EURASIP Journal on Wireless Communications and Networking

RESEARCH

Open Access

Hybrid intrusion detection system using blockchain framework

S. R. Khonde[1,2*] and V. Ulagamuthalvi[1]

*Correspondence: khondeshraddha21@gmail.com
[1] Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India
Full list of author information is available at the end of the article

Abstract

Data security and confidentiality are major goals now days due to the extensive use of the internet for data sharing. In modern era, most of the networks are compromised by intruders to grab access to private, confidential, and highly secured data. An intrusion detection system (IDS) is widely used to secure the network from getting compromised by intruders. Most of the IDS share the signatures of the novel attacks detected by anomaly approach for improving the detection rate and processing time. Security of signature shared by nodes is becoming a considerable problem. This paper presents a novel framework blockchain based hybrid intrusion detection system (BC-HyIDS), which uses the blockchain framework for exchanging signatures from one node to the other in distributed IDS. BC-HyIDS works in three phases where it uses both detection methods and blockchain in the third phase to provide security to data transferred through the network. This system makes use of a cryptosystem to encrypt the data stored in blocks to improve security one level higher. Hyperledger fabric v2.0 and Hyperledger sawtooth is used to implement system. Blockchain framework is created as a prototype using distributed ledger technology which helps in securing signature exchange. Performance of BC-HyIDS is evaluated in terms of accuracy, detection rate, and false alarm rate. From results, it is observed that a 2.8% increase in accuracy, 4.3% increase in detection rate, and a reduction of 2.6% in FAR is achieved. Blockchain performance is evaluated using Hyperledger fabric v2.0 and Hyperledger sawtooth on throughput, processing time, and average latency. BC-HyIDS shows improved performance when used with blockchain.

Keywords: Blockchain, Intrusion detection system, Secured communication, XGBoost, Isolation random forest, Artificial neural network, Ensemble approach

1 Introduction

In today's modern era, huge use of internet provides an ease for data sharing and exchange. This makes security a big issue to deal with. Data exchanged in network is mostly private and confidential. To provide security to this data a secured mechanism is needed. Intruders use attacks to break security of network to steal information exchanged. Security to the network is provided by the intrusion detection system (IDS). An IDS is used to provide security to network against attacks. It helps in monitoring normal as well as abnormal activities of network. IDS generate alarm when any malicious activity is observed in the network. Malicious activities are detected using two methods as Signature based detection (SIDS) and Anomaly based detection (AIDS). SIDS method use standard dataset of intrusion detection to identify malicious activities in the network. Standard dataset consist of signatures, which define the pattern of attack. Packet entering in the system is analyzed using this dataset. If the pattern is matched then it is consider as malicious activity and alarm is generated for administrator. Otherwise, the packet is considered as normal and passed in the network. Limitation of SIDS is unidentified attack cannot be handled by this system. If old or outdated dataset is use it does not contain signature of unidentified attacks. This leads to degrade in performance of SIDS. AIDS method use behavior based analyses to detect malicious activities. Normal behavior of all type of traffic is stored in the form of rule. These rules are use as pattern to analyze and detect packet entering in the network [1]. Each packet behavior is match with the rules, if match is found it is consider as normal behavior. Otherwise, any deviation in rule is considered as malicious activity. Limitation of ADIS is this method generates more false alarms. Slight deviation in the rule is considered as malicious behavior. In most of the network it is difficult to find the normal behavior of packets. In current internet scenario there is a need of IDS, which can offer collaborative, distributed, cross-platform and various protection services. This will help in improving network security and all new attacks can be identified [2].

Due to emerging of new attacks in the modern era, a need for intelligent and innovative IDS arises for data exchange and collaboration. Nodes connected in network can collaborate with each other to share data such as signature dataset, network resources, attack signature, data alerts and many more. Data exchange between nodes can be at risk if an intruder becomes a part of the network and can observe all activities and data passing. Intruder can capture, modify or delete data passing within the network. More secured mechanism is required to provide security to data exchanged or transfer within the network. Tampering of data in exchange between nodes can lead to network harm. Intruders can easily modify the signatures, datasets, files, logs and many more. Control of data in the hand of intruder takes network at high risk.

A standard platform use to provide security to data exchanged between nodes of distributed network is blockchain [3, 4]. It provides distributed and shared data structure to exchange information in peer-to-peer network [5]. Another feature of blockchain is it allows replication of data on number of nodes. Replication of data increase security and single node cannot be a bottleneck for network [6, 7]. In distributed IDS network blockchain is used for improving security because of its immutability and consensus protocols followed among nodes [8]. Blockchain is use by most of the security applications for multimedia and confidential data sharing [9, 10]. In current era, most of the system applications make use of blockchain framework due of its various advantages [11, 12]. To improve performance various security application use blockchain [13, 14]. Recent areas like internet of things [15], intrusion detection system [16], financial services and many other applications have adopted blockchain [17]. In web application development [18] and cloud computing [19] blockchain is an emerging area for research.

In this paper, a blockchain based framework is proposed, to exchange signatures of new attacks in the distributed network. Proposed work use signature as well as anomaly attack detection methods in a hybrid approach. Packet entering in the system is analyzed by signature detection. If any malicious activity is detected then packet is discarded, otherwise it is passed to the anomaly detection. Anomaly detection method analyze packet and check for normal activities. If packet is normal then it is passed in network otherwise packet is handed over to blockchain framework. This framework is responsible for signature creation and distribution in the network. Blockchain provides security to the signatures while exchange in between nodes of distributed network. Intruders cannot tamper signatures are it is replicated on number of nodes in network. Newly created signature is used by all nodes to update dataset. Updated dataset helps in improving performance of signature detection. The proposed framework is the first IDS which makes use of blockchain for signature exchange. Features of this architecture are it's fully distributed and fault tolerant. If any node fails in this architecture, then also data can be made available due to replication of data on number of nodes. Intruders cannot tamper data as it is available on number of nodes and it is impossible to change data at number of locations. This architecture basically makes used of authorized nodes for data exchange. Due to various features, this architecture is the unique IDS for distributed networks.

Organization of the paper is as: Sect. 2 provides the basics of blockchain and related work. In Sect. 3, the detailed architecture for the proposed BC-HyIDS system is elaborated along with the smart contracts used in the proposed system. Section 4 gives the experimental results of blockchain and IDS system using various performance parameters. Section 5 gives a discussion and system evaluation of BC-HyIDS in terms of network characteristics. Section 6 describes the conclusion and future enhancements possible for the proposed work.

2 Related work

Blockchain is emerging as a solution for each application where the security of data matters and needs to improve upon in terms of security in data exchange [20, 21]. Security against signatures is provided using the blockchain framework. Only authorized nodes can read, update, delete and modify signatures in blockchain [22]. Types of blockchains available are Public [23–25], Private [26] and Consortium blockchain [27, 28]. Public blockchain are most widely used blockchain where each user can participate. Private blockchain is restricted to authorized nodes and these nodes can only participate in blockchain. Consortium blockchain is used by multiple organizations together to create a private network. Most of the organizations are using IDS for securing their network from the unwanted attacks happening in the network. Most of the IDS available in the market are having hybrid architecture which makes use of various classifiers for detection of attacks and maintaining the security of data transferred in the network. In survey, it is observed that only 3% of IDS available in the market have used blockchain for malware detection. A new scheme based on hybrid and ensemble model for new generation networks is frequently used by researchers now days. Most of the work is done on IDS depending on the methodology for attack detection based on signature and anomaly detection. Most of them use many supervised, unsupervised, and semi supervised algorithms for detection of attacks and share or create a signature using various approaches such as deep learning, data mining, and cloud [29–31]. Most of IDS use the ensemble techniques to improve performance over individual classifiers [32–35]. Feature selection techniques are also used along with ensembles in most of the existing IDS systems

[36–38]. IDS system makes use of a hybrid approach to improve its performance [39–41] based on the detection approaches available. A new term is coined in this area to use convolutional neural networks for getting IDS work efficiently [42]. All hybrid architectures are using various classifiers, which are used for signature as well as anomaly detection. Signature based detection shows the limitation of detection rate. Modern attack signatures are not available in the standard datasets, which tends towards decreasing the detection rate of signature-based detection. On the other hand, anomaly-based detection shows more false alarm rate compared to signature-based detection. This is possible because anomaly-based detection creates an alarm even if a slight deviation from the normal traffic is observed. In the proposed approach, we combine both the detection approach so that limitations can be overcut and the performance of IDS can be improved. From the literature survey, it is observed that most of the IDS do not share the signatures detected by the anomaly approach. Some hybrid approaches are sharing a signature in the network from one node to the other but without considering any security aspect. Thus, a need for intelligent IDS is raised, which makes use of secured mechanism [43]. In proposed architecture, hybrid IDS use both detection methods and provides a security mechanism to exchange the signature obtained after anomaly detection within the network. As per the requirement of proposed work private blockchain is used for developing blockchain prototype. As per our knowledge, the proposed architecture is the first one of its nature, to provide a solution for transferring signatures from one node to the other in the distributed network using blockchain technology.

2.1 Contribution

Based on the observation of various researches carried out in blockchain area, there is a gap clearly identified in malware detection and blockchain. Some of the limitations of the existing IDS are listed below:

- Existing IDS does not use both detection methodologies as signature based and anomaly based in a single system.
- In the existing scenario, none of the cooperative intrusion detection systems are creating a signature for distribution.
- These IDS are not having any secure mechanism for sharing signatures within the network.
- Less number of IDS works in a fully distributed manner using blockchain technology.

To provide a solution for all the above identified gaps of existing IDS, we propose a new system based on blockchain technology, which is a Blockchain-based hybrid Intrusion Detection System (BC-HyIDS). This novel system is the first system providing all advantages and features required for networks to keep it secure as per our knowledge. Some of the highlights of BC-HyIDS are given below:

- BC-HyIDS uses signature and anomaly based detection in a hybrid manner for attack detection.
- BC-HyIDS use benchmark dataset CIC-IDS 2017 for creating the signature.
- BC-HyIDS provides mechanism to exchange signatures using blockchain.

- BC-HyIDS supports distributed architecture of the network.

Table 1 shows comparison of existing IDS with proposed distributed IDS with some parameters.

Next section elaborates the detail illustration of all phases and methodologies used in the implementation of BC-HyIDS.

3 Proposed methodology

The proposed architecture is an innovative approach for sharing signatures in the distributed environment of IDS. This is a novel architecture as none of the IDS works on a hybrid approach combining both detection techniques. Most of the hybrid approaches are based on combining various types of classifiers as supervised, unsupervised, and semi-supervised algorithms. Proposed architecture uses both types of detection methods as signature-based as well as anomaly based for attack detection and improving the security of network. Blockchain is used for signature transfer and it would be of its first kind to be used in distributed IDS as per our knowledge. This model is proposed for a distributed environment where each node is connected to the other in a distributed fashion. Each node will be able to detect attacks by analyzing packets entering through the network. Whenever the packet reaches at node, it gets captured and detected for a malicious pattern using signature-based detection phase. In this phase, all classifiers are trained using the modern CIC-IDS 2017 dataset. Classifiers which are used in this phase are artificial neural network (ANN), Isolation Random Forest (IRF), and XG boost. These all classifiers will ensemble together to get the final prediction of the analysis using majority voting algorithm. If packets are analyzed as attacks, then it gets directly rejected by the node, otherwise it would be forwarded to the second phase as anomaly detection. This phase improves security of the network since IDS validates each packet twice by both detection techniques. If the dataset is not up to date with the signatures of modern attacks, the malicious pattern is also analyzed as a normal packet. Node checks it twice using anomaly-based detection for assurance about a normal packet. Anomaly detection is based on the behaviour of each network. Packet behaviours are matched with the rules specified by genetic algorithms.

Signature creation and transfer using blockchain framework phase is used once the attack is detected by the anomaly detection phase. This phase is added in this architecture to securely transfer signatures from one node to other. In this phase, node is

Table 1 Comparison of existing and proposed IDS

Parameters	Existing IDS	Proposed distributed IDS
Detection methodologies	Any one—Signature detection or Anomaly detection	Both—Signature and Anomaly detection in hybrid approach
Signature creation	Very often	For all new attack detected
Mechanism for signature sharing	Not secure	Secured
Distributed	No	Fully distributed
Fault tolerant	No	Yes
Blockchain technology	No	Yes

responsible for creating, and encrypting signature of the novel attack. Further it is incorporated as a block in the blockchain, which will be used for transferring signatures in the network. All nodes receive the signature and update their dataset to ensure that next time the same attach would be taken care. This will reduce the detection time as well as the processing time of the node. Figure 1 shows the general architecture of the proposed distributed IDS which shows the top view of the architecture. Detail of each phase is explained in the next section.

3.1 Phase 1: signature based detection

Signature based detection is mainly performed using a basic approach whereby the signatures of various attacks are stored in standard datasets. Many standard datasets for intrusion detection are available and can be leveraged for training various classifiers for identification of attacks. Each dataset has its own set of attack signatures. Number and type of attacks are different in various datasets. In the proposed system, CIC-IDS 2017 dataset is used for training and testing classifiers. Comparison of classifier performance is observed and presented in the results section. With this approach, packets entering in the system are analyzed based on the signatures available in the datasets. Once the packet enters in this phase, classifiers try to match it with various available signatures. If it matches, then the packets are discarded, otherwise it is passed to second phase for further analysis. To evaluate the performance of system, all classifiers are used in an ensemble manner to avoid biased prediction. Classifiers used in this phase are ANN, IRF, and XG-boost to test real-time traffic. Ensemble is used to improve the performance of IDS in this phase. Ensemble approach provides better accuracy as a compare to the individual classifiers in terms of detection rate and accuracy. Detailed system model used in Phase 1 for signature detection is shown in Fig. 2.

All three classifiers are used in an ensemble approach to improve performance of the signature-based detection phase. Ensemble of all classifiers is done using majority voting algorithm. Majority voting algorithm is used to boost the performance of individual

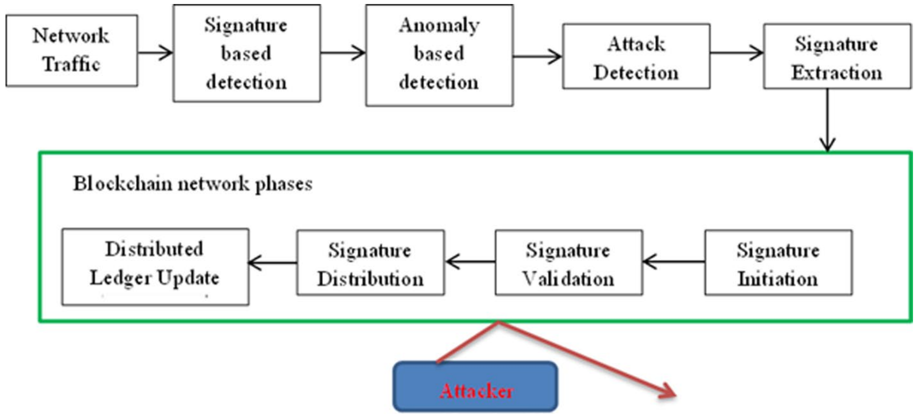


Fig. 1 System architecture of proposed system BC-HyIDS using blockchain. The top view of the architecture which elaborates all the phases of prototype. Each step of the architecture is explained including blockchain framework. All the steps included in blockchain framework are explained in block. This framework will provide security to the system such that attackers will not able to penetrate attacks in network. This framework provides immutable data exchange

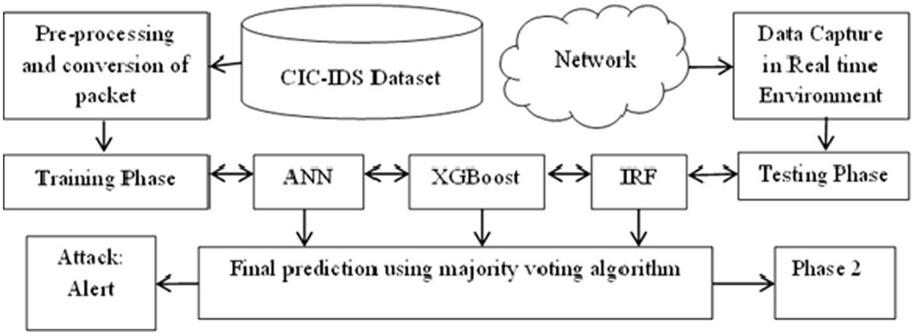

Pre-processing and conversion of packet
CIC-IDS Dataset
Network
Data Capture in Real time Environment
Training Phase
ANN
XGBoost
IRF
Testing Phase
Attack: Alert
Final prediction using majority voting algorithm
Phase 2

Fig. 2 BC-HyIDS phase 1: signature based detection architecture. Phase 1 architecture from proposed system is elaborated in figure. It shows input, output and processing steps for signature based attack detection. Classifiers used where artificial neural network, XGBoost and isolation random forest. All classifier are trained using CIC-IDS 2017 dataset. This is a standard intrusion detection dataset used for modern attack detection. Ensemble approach using majority voting algorithm is used to get final prediction as attack or normal packet

classifier [44]. To avoid biased output given by individual classifier majority algorithm is used. This algorithm works on odd number of classifiers. To find the output of this algorithm inputs are collected as a vote from odd number of classifiers [45]. Majority voting algorithm takes input as the prediction made by each classifier after analyzing packets. Votes collected from all classifiers are analyzed and according to the majority predictions, the final prediction or classification of the packet is decided. Final output will be malicious, that is attack or normal. If it is malicious, then an alert is generated by the administrator and the packet is discarded from network. If the output is normal, then packets are passed to Phase 2: Anomaly based detection for further behavior analysis for packets. Detailed system architecture for Phase 1 is shown in Fig. 2.

3.2 Phase 2: anomaly based detection

Anomaly based detection is mostly based on behavior-based approach for intrusion detection. In behavior-based detection the classifiers are trained using rules that show the behavior of the packet and network. Normal behavior is used for training various classifiers and testing is done in real-time environment.

All classifiers are trained using rules which show the normal behavior of the data entering in system. Various machine learning and deep learning algorithms are used to do behavior analysis now day. The reason behind doing behavior analysis for anomaly detection is these are novel types of attacks whose signatures are not available in any standard dataset. For anomaly detection, as the signature is not available, the behavior or pattern of such attacks can help to recognize malicious activities in the network. To find the normal behavior of the data entered in the system, a genetic algorithm is used. This algorithm is used to find the normal behavior patterns for all normal activities. Any behavior against these normal activities is considered as malicious behavior.

3.2.1 Genetic algorithm

Genetic algorithm is mainly used for intrusion detection based on behavior analysis. The behavior of the system is defined in terms of rules. These rules are defined using genetic algorithms and works in four-step process as initial population generating, chromosome designing, fitness value calculate, and genetic operator designing, to find the behaviour patterns in terms of rules. With the help of this process, genetic algorithm finds the rule-based dataset as an output. CIC-IDS 2017 dataset is used as an input. Rule based databases will be used to train classifiers so that they can perform behaviour analysis of each data entering in the system. These rules are used to represent the normal behaviour of data. If any packet data is not satisfying any rule, then it is considered as a malicious behaviour. If packet data match with at least one of the rule, then it is considered as normal behavior. Using CIC-IDS dataset, optimal features can be found to get the final rules. Fitness is also known as goodness of population. While evaluating, operations like crossover and mutation is used to make it more specific.

Fitness value is calculated for each feature set and the feature having strong fitness value will be considered for rule-based dataset generation. This dataset is then used to train classifiers. Equation (1) is used to calculate the fitness value. Threshold considered for the fitness value is 0.90. All chromosomes with a value nearer to the threshold are considered for dataset generation. Classifiers used are ANN, IRF, and XG-boost.

$$\text{Fitness value} = (a/A) - (b/B) \tag{1}$$

where A: total number of attacks; a: number of attack connections correctly classified by the individual classifier; B: normal connections in the population; b: number of normal connections correctly classified by classifiers.

Figure 3 shows the process of genetic algorithm which starts from the formulation of population and chromosome and then evaluates each of them with crossover and mutation operator to get the most important and specific chromosome. The population size, crossover folds, and mutation size can vary in the case of benchmark datasets. The motto behind this process is to get specified and accurate rules defining the normal behavior of the data transferred in the network. This individual chromosome is then converted into rules to create a dataset. This rule-based dataset is used further to train classifiers for malicious activity detection as Phase 2 of the proposed framework. Any deviation other than the rules mentioned would be considered as a malicious behaviour and that data is declared as attack. Once the attack is identified, an alert is generated by the system administrator and the packet is passed to the third phase signature generation and distribution. This phase will convert the packet into the signature for distribution among

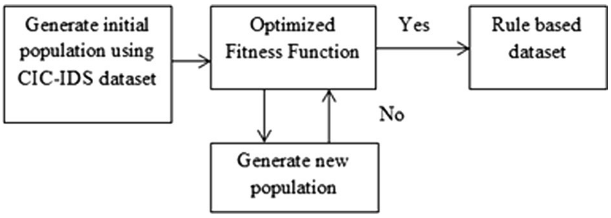

Generate initial population using CIC-IDS dataset
Optimized Fitness Function
Yes
Rule based dataset
No
Generate new population

Fig. 3 Genetic algorithm process for rule based dataset generation. Genetic algorithm is used to generate rule based dataset. Process works in four steps as initial population generation for features selected from CIC-IDS dataset. This dataset consist of more than 80 features out of that only 22 features are used to describe normal behavior of the packet in the network. Optimized set of features is finalized using fitness function. All the features with good fitness value are considered for rule dataset generation. Process continues until optimized set is found

nodes in the distributed network. Otherwise, normal packet data is allowed to transfer in the network. To avoid biased prediction and reduce false alarm generation, ensemble approach is used, which makes use of majority voting algorithm to give the final prediction. System Architecture for phase 2: Anomaly based detection is shown in Fig. 4.

3.3 Phase 3: blockchain framework for signature extraction and distribution

In BC-HyIDS, phase 3 is used to securely distribute signatures over the distributed network. Input received in this phase is packets which are predicted as attacks by the anomaly detection phase [46, 47]. This phase works in three steps, first from the received packet; a signature is created and in the second step this signature is uploaded as a block and verification is done. Third step will distribute the signature over all nodes connected in the distributed network. Permissioned private blockchain is used to transfer signatures in this phase. Broad architecture of the blockchain framework along with the individual node structure is shown in Fig. 5.

As shown in Fig. 5, the blockchain is distributed over a distributed network. All nodes in the network are attached in a distributed manner which follows one of the consensus protocols such as proof of work (PoW), proof of stack (PoS), proof of authority (PoA). In BC-HyIDS, phase 3 makes use of the blockchain platform for signature extraction, upload, and distribution. Blockchains can be of various types such as public, permissioned, non-permissioned, and consortium blockchains. These blockchains can be implemented on various platforms such as Ethereum [48] and Hyperledger. As per the requirement of BC-HyIDS, permissioned blockchain is developed with the help of Hyperledger which uses of the consensus protocol as a proof of stack (PoS) [49]. Hyperledger is a platform which is used to build customized applications on the permissioned blockchain. As the need of distributed IDS, we have implemented permissioned blockchain along with some features of public blockchain. Permissioned blockchains consists of nodes which have the authority to be a part of this blockchain. Each node attached into this network is an authorized node which

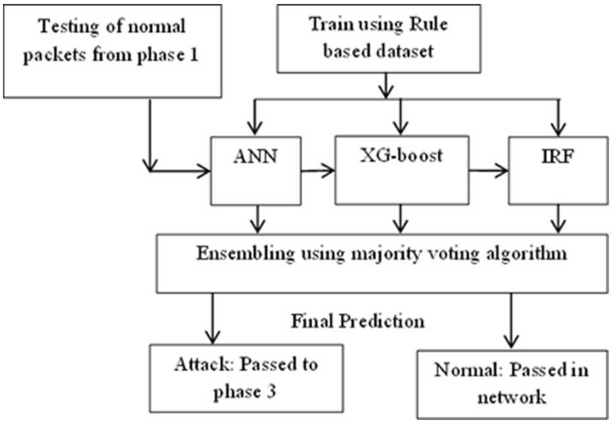

Testing of normal packets from phase 1
Train using Rule based dataset
ANN
XG-boost
IRF
Ensembling using majority voting algorithm
Final Prediction
Attack: Passed to phase 3
Normal: Passed in network

Fig. 4 BC-HyIDS phase 2: anomaly based detection architecture. Detailed architecture of phase 2 that is anomaly based detection for system. Input will be taken from phase 1, the packets which are considered as normal. Rule dataset is used to train classifiers. Rule based dataset is generated using genetic algorithm. It consists of rules describing normal behavior of packets entering in the network. Ensemble approach with majority voting algorithm is used to find final prediction as attack or normal packet

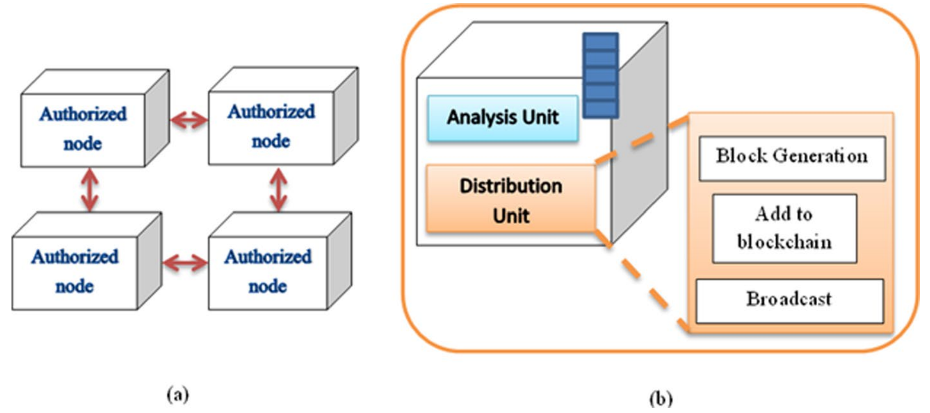


Fig. 5 Blockchain framework: **a** blockchain network of BC-HyIDS and **b** individual node structure. BC-HyIDS system is a private blockchain network implemented using Hyperledger platform. As it is a private and permissioned network it consists of only authorized nodes. All nodes are connected to each other in a chain. Each node is able to do all functionality using analysis and distribution unit

is responsible for extracting the signature, create a block, and distribute among all remaining authorized nodes of the network. BC-HyIDS consist of two types of nodes as initiator node and the validator node. Initiator nodes are the authorized nodes which are responsible for signature creation. Validator node performs the validation of the signature and converts it into a block for distribution. In BC-HyIDS some are initiator nodes and some are validator nodes. Validator nodes can also be work as an initiator node if required. The objective behind adding this phase to the BC-HyIDS system is to provide security to signatures distributed in the network from attackers.

Each node consists of an analysis unit (AU) and a distribution unit (DU). Analysis units mainly analyzes all packets entering in the node through network. This unit makes use of both the phase of BC-HyIDS, that is, signature-based detection and anomaly-based detection. This also helped to update the dataset with new signatures which consider an attack by anomaly detection phase. Along with the analysis and distribution unit, each node will consist of the complete ledger of the blockchain, a structure used to carry the data called as block, and a transaction, a basic unit of blockchain. Transactions are nothing but signatures which were extracted by nodes from the packet. Signature extraction block upload and distribution are explained in the next section.

3.3.1 Extraction of signature

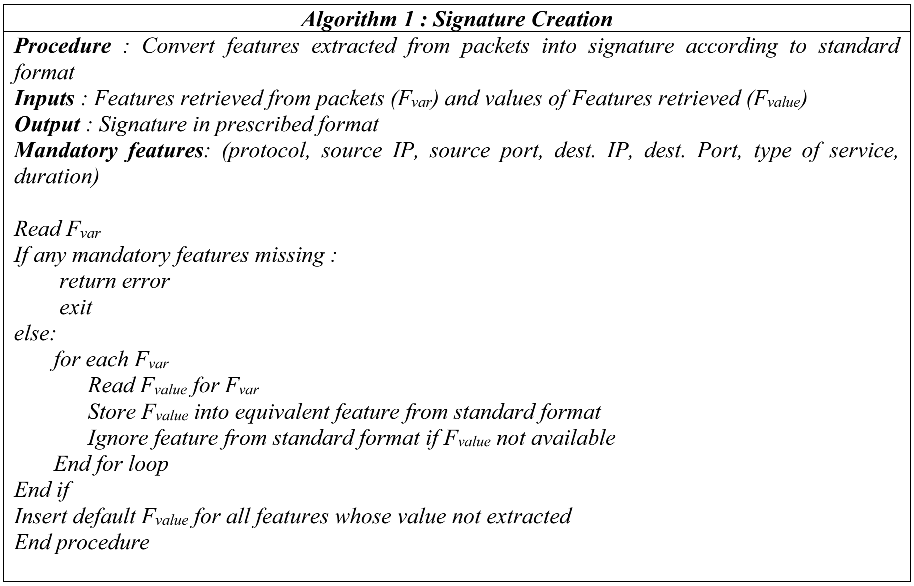
Signature extraction is performed from the packets received as input from the anomaly detection phase to create the signature. Features are selected as per the format of CIC-IDS 2017 dataset as phase 1 of BC-HyIDS uses this dataset for attack detection. CIC-IDS 2017 dataset uses 22 features for the detection of various types of attacks. For creating signature, a script is returned which will take the input as a packet and generates its signature equivalent to the features of CIC-IDS 2017 dataset.

Standard format for signature creation is as follows

$$\{\text{MAC address},\ \text{IP address},\ \text{Public Key},\ \text{Private Key},\ \text{Type},\ \text{Port},\ \text{Features}\}$$

where MAC address—MAC address of the node that is responsible for signature extraction; IP address—IP address of the node that is responsible for signature extraction; Public Key—Public key of node responsible for signature extraction; Private Key—Private key of node responsible for signature extraction from a pair of public–private keys of node; Type—Type of attack whose signature is extracted by node. In this case, it will be considered as "Novel"; Port—Active communication port of the authorized node; Features—22 features of the CIC-IDS 2017 dataset extracted from packets.

Post extracting features from packets, the initiator node executes the signature creation algorithm as given below to convert the extracted features into the prescribed signature format. All Initiator nodes follow smart contracts to create the signature in the required format. Smart contracts are the set of rules prescribed stored in the system which will be used by the blockchain if a certain action is done. For example, if the initiator wants to create a signature from extracted features, then system will automatically follow rules and formats prescribed by the signature creation algorithm. The Algorithm 1 is used for signature creation is given below.



Algorithm 1 : Signature Creation

Procedure *: Convert features extracted from packets into signature according to standard format*
Inputs *: Features retrieved from packets (F_{var}) and values of Features retrieved (F_{value})*
Output *: Signature in prescribed format*
Mandatory features*: (protocol, source IP, source port, dest. IP, dest. Port, type of service, duration)*

```
Read $F_{var}$
If any mandatory features missing :
    return error
    exit
else:
    for each $F_{var}$
        Read $F_{value}$ for $F_{var}$
        Store $F_{value}$ into equivalent feature from standard format
        Ignore feature from standard format if $F_{value}$ not available
    End for loop
End if
Insert default $F_{value}$ for all features whose value not extracted
End procedure
```

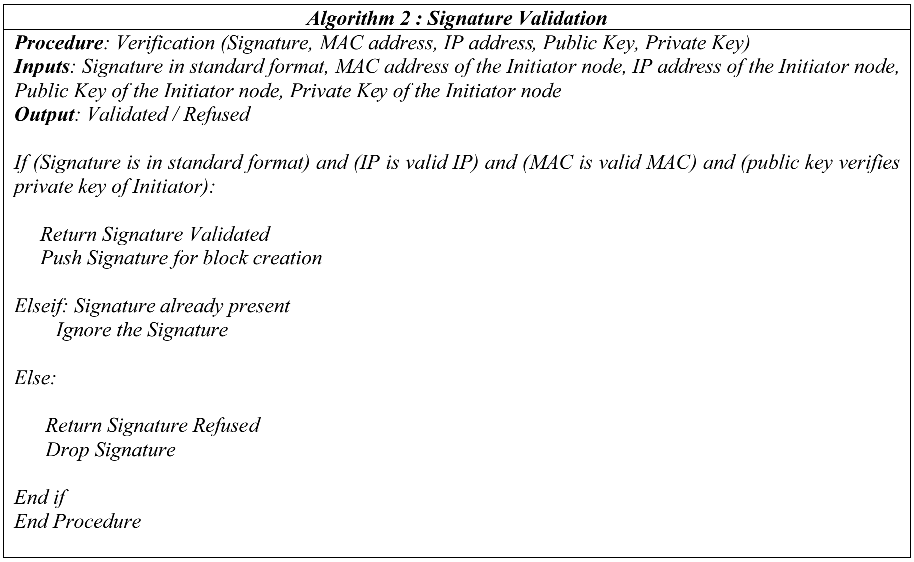
As shown in Algorithm 1, signatures are created in the standard format and saved in the format explained above. The created signature is encrypted by the private key of initiator node and sent to validator node for validation.

3.3.2 Validation of signature

Signature validation is carried out by the validator node. Signature validation is an important state in this phase as all initiator nodes has the authority to create a signature of the anomaly attacks detected by their analysis unit. The BC-HyIDS system has some of the nodes working as validator nodes. The responsibility of the validator node is to check the validity, authorization, and significance of the signature before incorporating it to the blockchain. Following is the checklist followed by the validator node.

- The signature submitted as a transaction is in the prescribe format and generated using the smart contract used for signature creation.
- It also verifies whether the same signature uploaded in the past by some other initiator. If so, then a new block is not created by validator, otherwise this signature will be considered for block creation.
- Is signature is valid and created by the anomaly detection phase of the initiator node.
- Is the initiator is an authorized node to create a signature and send it for validation.

Validator node completes the validation according to all parameters explained above. Validation process is followed as a smart contract and it is to be followed by each initiator and validator node before confirming the incorporation of signature to the blockchain. Once the signature is validated, the node initiates the process of block creation and incorporation in the blockchain. The Algorithm 2 used by the validator node for completing the validation process is given below.



```
Algorithm 2 : Signature Validation
Procedure: Verification (Signature, MAC address, IP address, Public Key, Private Key)
Inputs: Signature in standard format, MAC address of the Initiator node, IP address of the Initiator node,
Public Key of the Initiator node, Private Key of the Initiator node
Output: Validated / Refused

If (Signature is in standard format) and (IP is valid IP) and (MAC is valid MAC) and (public key verifies
private key of Initiator):

    Return Signature Validated
    Push Signature for block creation

Elseif: Signature already present
      Ignore the Signature

Else:

     Return Signature Refused
     Drop Signature

End if
End Procedure
```

As shown in the Algorithm 2 above, if all conditions are satisfied then signatures are verified by the validator, otherwise it is refused by the validator and the signature gets dropped. In case if the same signature is already created by the other initiator node, then it will be ignored by the validator. The validated signature is further passed for block creation to get added in the blockchain.

3.3.3 Signature block creation

The next step post validating the signature is block creation to add the same in the blockchain network. Hyperledger is used to implement the BC-HyIDS system. According to the system architecture, any new attack detected by the node is to be converted into a signature so that it can be used by other nodes for future use. Once the validator node validates the signature, new block is created using Hyperledger format. Block structure used in BC-HyIDS is as shown in Fig. 6.

As shown in Fig. 6, the block is divided into three sections as header, data, and metadata. Explanation of all sections is given below.

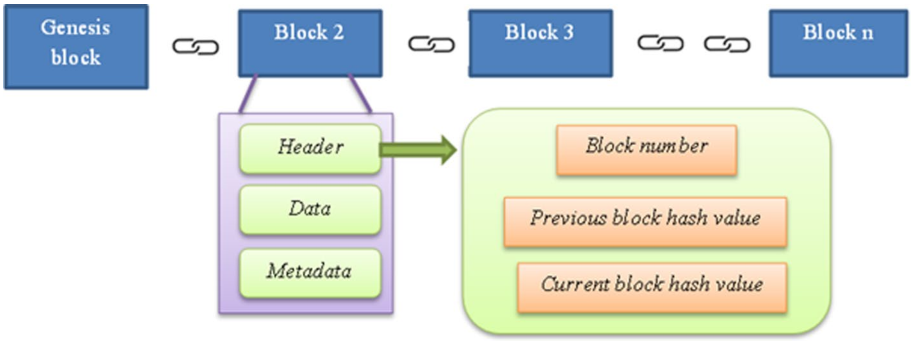


Fig. 6 Block structure for hyperledger blockchain used in BC-HyIDS. Hyperledger blockchain along with block structure is shown. First block of this blockchain is genesis block. Each block consists of header, data and metadata. Each block is connected with other using hash functions. Unique identification is given to each block called block number

a. ***Header***: It is a block header which gives information about the block. It consists of the following information.

- *Block number*: Block number is the identification assigned to each block newly created in the system by the validator node. This block number is used to access a block in the blockchain in future communication.
- *Previous block hash value*: 256 bits previous block hash value computed using SHA256.
- *Current block hash value*: 256 bits current block hash value in hexadecimal format.

In Hyperledger, to create a chain of blocks as per blockchain architecture, each block is connected to the next and previous block with the link that is called as block hash value. Default cryptographic hashing algorithm used by Hyperledger is Secure Hashing Algorithm (SHA256). It is a successor of SHA-1 and SHA-2 hash algorithms. It generates a unique 256 bit (32 bytes) hash value for each block. Hash values are mostly represented in hexadecimal format. SHA256 algorithm has not been compromised in any manner till date. This is the main reason of using SHA256 as a default hash algorithm in Hyperledger.

b. ***Data***: Block second section is Data section. This section consists of an actual signature which is created in the standard format.

c. ***Metadata***: Metadata consists of the data about the block like timestamp when the block is created, consensus protocols, private key of the initiator and validator, and signature details if any.

3.3.4 Distribution of signature

Next step after block creation is the distribution of the block to all nodes and add it to the blockchain. Once the block is created, validator nodes add the block to the existing blockchain. This information is broadcasted in the network to all nodes. Once the

Table 2 Execution time—query

No. of transactions	Hyperledger fabric v2.0	Hyperledger sawtooth
10	0.16	0.08
100	0.98	0.56
1000	10.25	3.12
10,000	57.28	15.26

Table 3 Average latency—query

No. of transactions	Hyperledger fabric v2.0	Hyperledger sawtooth
10	2.26	1.25
100	3.59	2.11
1000	17.23	13.66
10,000	72.45	32.12

validator block is received by all nodes, updating of the individual ledger occurs. All nodes update their ledger and can use the new signatures for further processing. Once all nodes finish with the operation, blockchain is committed and the block is permanently attached to the chain. In BC-HyIDS nodes save new blocks in the ledger and the read signature need to be updated in the CIC-IDS 2017 dataset. Updating the dataset helps nodes for further analysis of packets entering in the network.

4 Experiment results

In this section, the performance of the implemented permissioned private BC-HyIDS blockchain is elaborated. BC-HyIDS implemented using Hyperledger fabric v2.0 and Hyperledger sawtooth. Performance parameters used for evaluation are execution time, average latency, throughput, and transaction processing time. System performance is evaluated in two parts as the performance of blockchain and the performance of IDS with and without blockchain. Based on the conducted experiments, Hyperledger sawtooth provides better results compared to Hyperledger fabric v2.0. In terms of the accuracy of IDS, it improves if a blockchain is used (Tables 2, 3, 4, 5, 6).

4.1 Performance evaluation of blockchain

Many approaches are provided to check the performance of blockchain [50, 51]. Performance of the blockchain in BC-HyIDS is evaluated according to the number of nodes in the network and the execution time required. Types of nodes are Initiator and Validator nodes. Parameters used to evaluate the performance of both nodes are

Evaluation of execution time Evaluation of execution time for the two platforms is done by varying the frequency of transactions in the network [52]. Execution time is analyzed for different functions such as time required executing a simple query, the initiation process, and the validation process. In general, query processing execution time increases

Table 4 Execution time—initialization process

No. of transactions	Hyperledger fabric v2.0	Hyperledger sawtooth
10	1.32	0.65
100	2.32	1.25
1000	10.56	6.12
10,000	74.23	55.33

Table 5 Average latency—initialization process

No. of transactions	Hyperledger fabric v2.0	Hyperledger sawtooth
10	3.22	1.22
100	8.21	5.69
1000	26.11	23.12
10,000	56.21	48.35

Table 6 Execution time—validation process

No. of transactions	Hyperledger fabric v2.0	Hyperledger sawtooth
10	5.23	2.33
100	10.23	6.12
1000	25.21	21.33
10,000	72.33	58.21

as the number of transactions increased. Execution time taken by sawtooth is better compared to Fabric v2.0 for the simple query function, as shown in Fig. 7a and Table 2. As the dataset grows, large the execution difference between both platforms goes on increasing. Execution time for initiation function and validation function is better, if sawtooth is used compared to Fabric v2.0 when the number of transactions increased. However, when a small dataset is used, then v2.0 shows better execution time compared to sawtooth. Figures 8a and 9a along with Tables 4 and 6 demonstrates the execution time required for the initiation and validation process for both implementations as Fabric v.2.0 and sawtooth.

Evaluation of average latency Average latency is evaluated with different values of transactions for both implementations as Fabric v2.0 and sawtooth. Latency average is calculated for executing a simple query initialization process and validation process. It is observed that the latency time is more in v2.0 compared to sawtooth as transaction go on increasing. The comparison between both for the average latency values is shown in Figs. 7b, 8b and 9b and Tables 3, 5 and 7 for simple query, initialization process and validation process respectively.

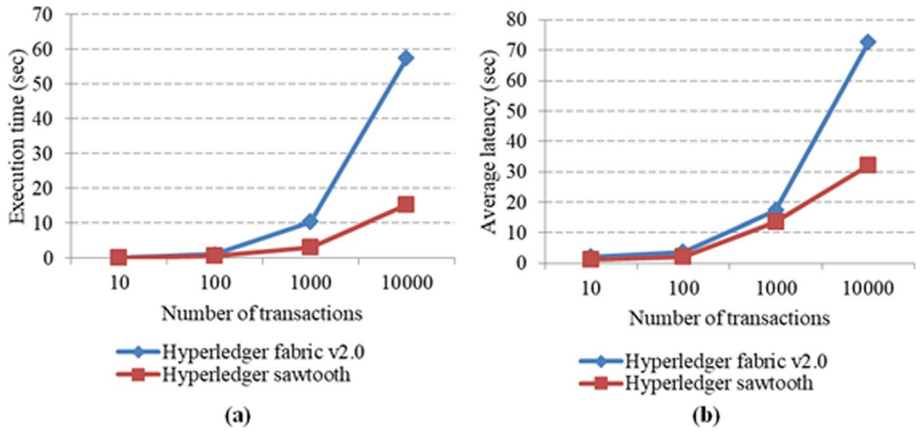


Fig. 7 Blockchain performance for simple query: **a** execution time and **b** average latency. Performance of blockchain that is third phase of the system is presented in graphical form for simple query execution. Simple query is a simple operation executed by node such as read a block. **a** Execution time required for simple query using Hyperledger fabric 2.0 and sawtooth is represented. **b** Average latency required for simple query execution on system when implemented using Hyperledger v2.0 and sawtooth is graphically represented

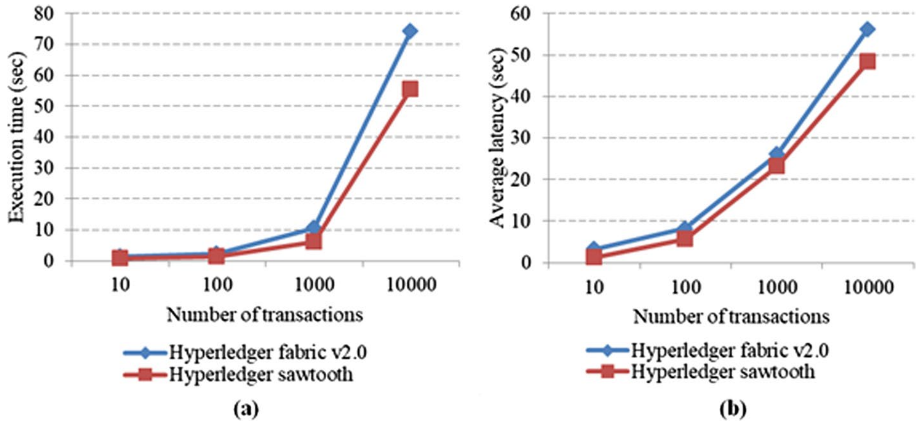


Fig. 8 Blockchain performance for initialization process: **a** execution time and **b** average latency. Performance of blockchain that is third phase of the system is presented in graphical form for initialization process execution. Initialization process is an operation executed by node when nodes needs to add new block created from the signature. **a** Execution time required for initialization process using Hyperledger fabric 2.0 and sawtooth is represented. **b** Average latency required for initialization process execution on system when implemented using Hyperledger v2.0 and sawtooth is graphically represented

Evaluation of throughput Throughput of the system is the number of transactions executed by the node per unit time. Average Throughput of a network depends on the number of nodes available in the network. As the number of node initiator and validator increased then the throughput of the system can also get increased. Throughput depends on the number of features such as the block size allowed in the blockchain or the number of nodes in the network. Figure 10a, b demonstrate the throughput of the system with different block size and number of nodes in the network, respectively.

Evaluation of transaction processing time Transaction processing time is the time taken from the point when a transaction has initialized on the node up to validation,

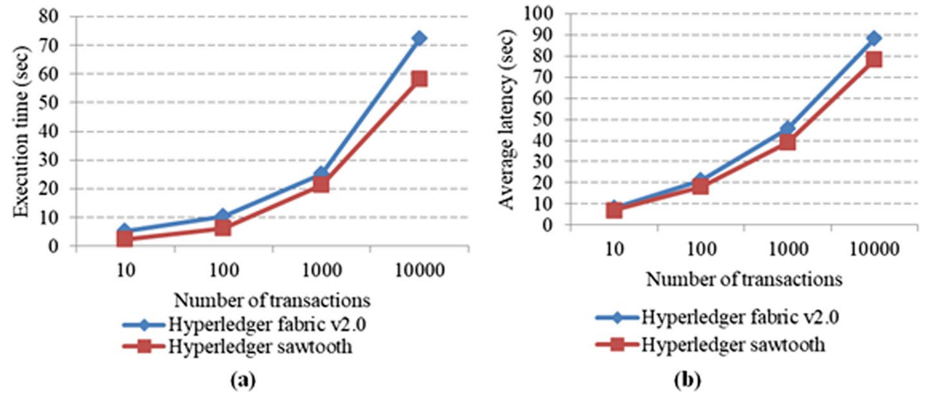


Fig. 9 Blockchain performance for validation process: **a** execution time and **b** average latency. Performance of blockchain that is third phase of the system is presented in graphical form for validation process execution. Validation process is an operation executed by node when nodes verify blocks to add in the blockchain. **a** Execution time required for validation process using Hyperledger fabric 2.0 and sawtooth is represented. **b** Average latency required for validation process execution on system when implemented using Hyperledger v2.0 and sawtooth is graphically represented

Table 7 Average latency—validation process

No. of transactions	Hyperledger fabric v2.0	Hyperledger sawtooth
10	8.12	6.88
100	21.23	17.98
1000	45.33	39.12
10,000	88.12	78.12

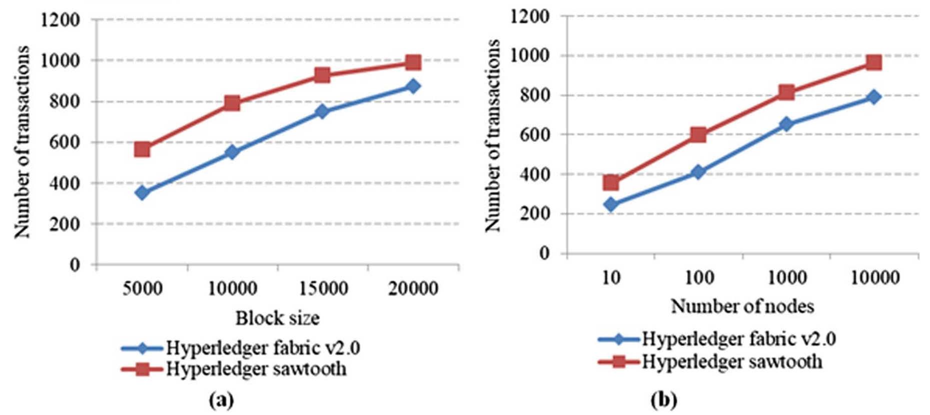


Fig. 10 Blockchain performance parameter throughput: **a** different block size and **b** different number of nodes. Performance parameter throughput of the blockchain is presented in graphical form. Throughput of the system is tested by variation in block size and number of nodes. **a** Throughput of system when different block size is used in blockchain using Hyperledger fabric 2.0 and sawtooth is represented. **b** Throughput of system when different number of nodes when same block size is used when implemented using Hyperledger v2.0 and sawtooth is graphically represented

completion, and addition of blocks in the blockchain. Transaction processing time is not the same as latency as used in most evaluation parameters [53, 54]. In this latency, assumed is the time taken by the network to add blocks in the blockchain. Whereas we assume that the latency and time required including a block in the blockchain is included in Transaction Processing time. By increasing the degree of parallel operations, the transaction processing time can be increased. Figure 11a, b show the transaction processing time in consideration of different block sizes and number of nodes in network, respectively. Average transaction processing time can be easily calculated by the execution time required for each transaction and the total transaction processing time divided by the number of transactions.

4.2 Performance evaluation of BC-HYIDS

BC-HYIDS performance is evaluated with the performance parameters used to evaluate IDS systems. Most of the IDS systems use various machine learning and neural network techniques for attack detection. Each IDS system can use one of the detection methods as signature based or anomaly-based detection. The proposed BC-HyIDS system makes use of both detection methods as signature and anomaly based. To improve the accuracy and detection rate with reduced false alarm rate BC-HyIDS makes use of ensembling technique and genetic algorithm in two phases. To provide two-layer security to the network BC-HyIDS use two phases and each packet goes through analysis twice. Performance of the system is the check of accuracy, detection rate, and false alarm rate parameters. Performance of BC-HyIDS is evaluated with and without blockchain. It is observed that signature based detection phase shows improvement in detection rate and accuracy when used along with blockchain compared to signature distribution. Table 8 shows the performance of each phase of BC-HyIDS with and without blockchain.

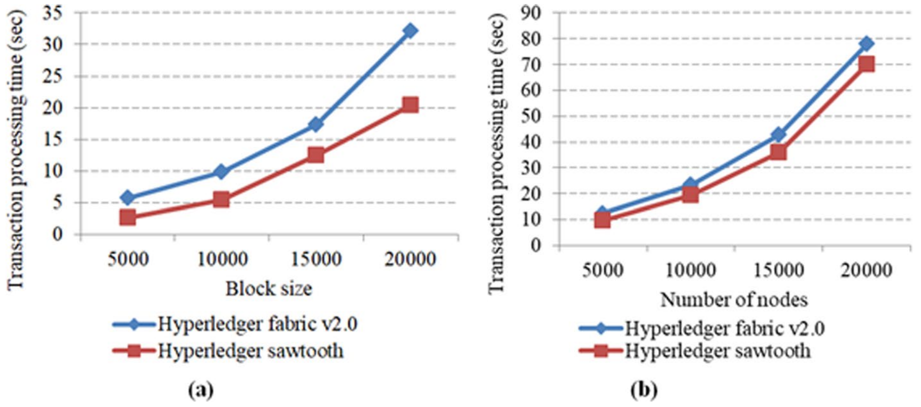


Fig. 11 Blockchain performance parameter transaction processing time: **a** different block size and **b** different number of nodes. Performance parameter transaction processing time of the blockchain is presented in graphical form. Transaction processing time required is tested by variation in block size and number of nodes. Transaction is nothing but operations executed by nodes for initialization and validation of blocks. **a** Transaction processing time of system when different block size is used in blockchain using Hyperledger fabric 2.0 and sawtooth is represented. **b** Transaction processing time of system when different number of nodes when same block size is used when implemented using Hyperledger v2.0 and sawtooth is graphically represented

From Table 8, it is observed that Phase 1 performance increased due to signature update in the dataset using blockchain. BC-HyIDS shows the improvised accuracy and detection rate. Accuracy increased by 2.8% approximately. Detection rate increased by 4.3% approximately and the false alarm rate reduced by 2.6%. From the comparison of BC-HyIDS with and without blockchain, it is observed that BC-HyIDS shows better performance in all parameters when used with blockchain framework. Figure 12a–d shows the graphical representation of the performance parameters accuracy, detection rate, and false alarm rate along with comparison.

4.3 Comparison of BC-HyIDS with existing IDS

Figure 13 shows comparison of existing hybrid IDS with BC-HyIDS. Most of the researches used ensemble and hybrid approach to improve overall performance of IDS. BC-HyIDS system used ensemble hybrid approach to improve performance of system. To improve the performance of IDS in distributed network blockchain phase is used for exchange of signatures. This phase improves performance of the signature based detection method due to update in dataset with novel signatures. Table 9 shows the comparison of BC-HyIDS with existing hybrid approaches. From Table 9 it is observed that the proposed BC-HyIDS system provides better accuracy and detection rate as compared to existing IDS. A reduction in false alarm rate is also observed. Ensemble methods used supervised, unsupervised, and semi-supervised classifiers. Al-Yaseen et al. [55] used hybrid model of a multi-level SVM and ELM model to classify normal behaviour and known attacks and an adaptive SVM model to learn and classify unknown attacks. Accuracy obtained by hybrid model is 95.86% with all features of KDD99 dataset. Three-tier architecture was used [56] to clean and pre-process the data along with support vector machine. Accuracy obtained in this hybrid architecture is 94.71% and false alarm rate is 3.8%. A hybrid approach [57] by combining *K* Nearest Neighbour with combined strangeness isolation algorithm was used to detect dos, probe, U2R and L2R attack on KDD dataset. Accuracy of 95.1% with false alarm rate 3% is obtained. Hybrid approach was developed [41] to detect anomaly-based and misuse-based attack with KNN algorithm and obtained 93.29% accuracy and 1.78% false alarm rate. Use of various techniques in distributed intrusion detection system improves accuracy. By using chi square feature reduction technique, the accuracy improved to 97% and 1.13% false alarm rate [58].

From studies it is observed that BC-HyIDS provides better accuracy and reduced false alarm rate due to secure transfer of signature from one node to other in distributed

Table 8 Performance of BC-HyIDS with and without blockchain

Performance parameters	Phase 1: SD of BC-HyIDS (%)	Phase 2: AD of BC-HYIDS (%)	HYIDS (without blockchain) (%)	Phase 1: SD of BC-HyIDS (%)	Phase 2: AD of BC-HYIDS (%)	BC-HYIDS (with blockchain) (%)
Accuracy	95.6	91.2	**95.7**	98.2	93.2	**98.5**
Detection rate	96.5	93.4	**94.5**	98.2	96.1	**98.8**
False alarm rate	2.4	4.5	**3.8**	1.8	2.3	**1.2**

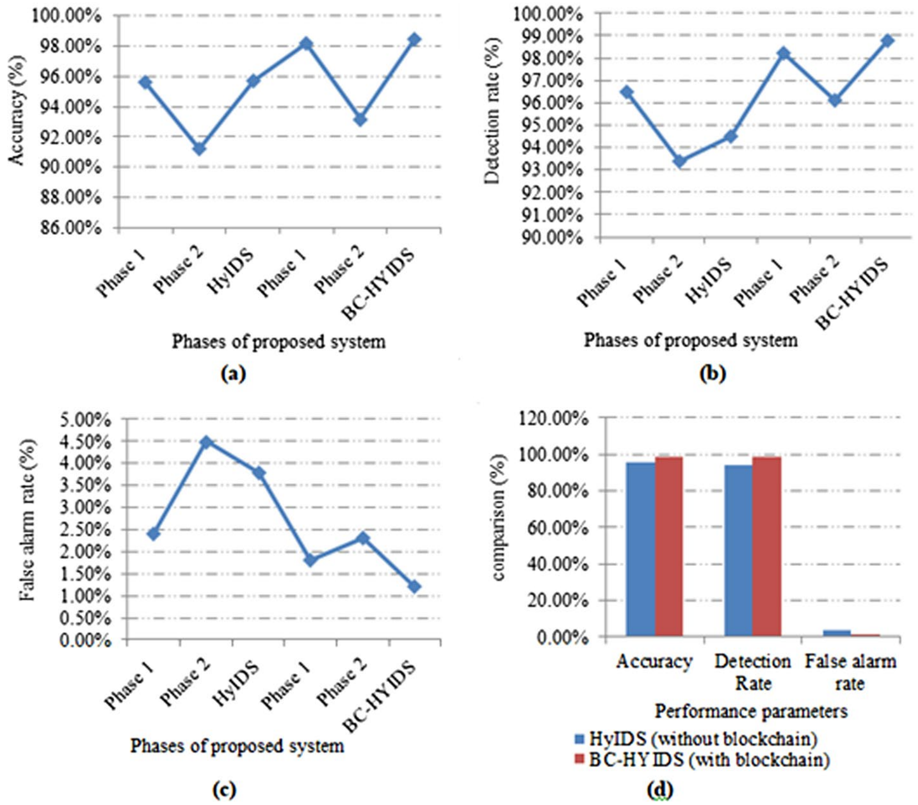


Fig. 12 BC-HyIDS performance measures: **a** accuracy for all phases of BC-HyIDS; **b** detection rate for all phases of C-HyIDS; **c** false alarm rate for all phases of BC-HyIDS and **d** comparison of BC-HyIDS with and without blockchain. Performance of BC-HyIDS as a single system is tested using various parameters. These all parameters are tested with and without blockchain. **a** Accuracy for all phase of BC-HyIDS when used with and without blockchain is represented. **b** Detection rate for all phase of BC-HyIDS when used with and without blockchain is represented. **c** False alarm rate for all phase of BC-HyIDS when used with and without blockchain is represented. **d** Comparison of whole system when used with and without blockchain is graphically represented in terms of accuracy, detection rate and false alarm rate

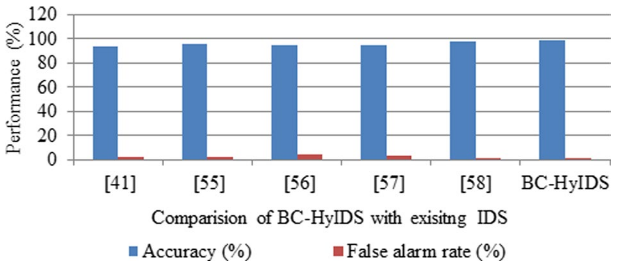


Fig. 13 Comparison of BC-HyIDS with existing IDS. Proposed BC-HyIDS with blockchain as a whole system is compared with existing hybrid IDS. Graphical representation shows better performance of BC-HyIDS as compared to existing IDS for distributed environment

environment. Blockchain used in BC-HyIDS helped to improve performance of signature detection in hybrid approach.

Table 9 Comparison of BC-HyIDS with existing IDS

Hybrid classifiers	Accuracy (%)	False alarm rate (%)
Hybrid KNN [41]	93.29	1.78
SVM-ELM [55]	95.86	2.13
Three tier IDS [56]	94.71	3.8
CSI-KNN [57]	95.1	3.0
Fusion chi-square and SVM [58]	97	1.31
BC-HyIDS	98.5	1.2

5 Discussion

The proposed system BC-HyIDS is an Intrusion Detection System which makes use of the blockchain framework to distribute signatures in the network. In BC-HyIDS blockchain framework is implemented using Hyperledger Fabric v2.0 and Hyperledger sawtooth. Blockchain used in this system is a permissioned based private blockchain with some features of public blockchain. Main reason to choose private blockchain along with public is this system is meant for an organization, so mostly it will be used by authorized users. Hyperledger provides flexibility in implementing blockchains over private blockchains. The reason using two versions of Hyperledger is to improve the performance of system. BC-HyIDS works in three phases such as Signature based detection, Anomaly based detection, and Signature creation and distribution. While using blockchain in phase 3 for the distribution of signatures of novel attacks, the performance of signature based detection increases approximately by 2.8%. BC-HyIDS is the novel framework which makes use of both detection techniques along with an immerging blockchain platform. This is the first IDS system which is implemented on a blockchain platform with both detection techniques as per our knowledge. This system works in a distributed fashion as no central controller is used for controlling operations in network. All nodes are authorized nodes and have the authority to create and validate signatures of a novel attacks. During implementation, care was taken to reduce the processing time in phase 3 such that the throughput of the system can be increased. To summarize BC-HyIDS is a unique IDS implemented on a blockchain platform to improvise the performance of general IDS systems.

System evaluation: BC-HyIDS were evaluated based on network characteristics. As the network plays a vital role in the implementation of IDS, most of the network characteristics are considered. System evaluation parameters are elaborated below.

1. Data sharing: Data sharing in the network is done using blockchain in BC-HyIDS. Data shared in blockchain is secured and immutable as it is replicated at each node in blockchain.
2. Computation cost: Computation in phase 3 is complex if the number of nodes is more in the network, but if the numbers of nodes are less, computation cost is less. Once the signatures are validated, then computation cost goes down in terms of the time required to read it. With the help of Hyperledger sawtooth used for private

as well as public blockchain, we succeed in reducing the computation cost for BC-HyIDS.

3. Bandwidth overhead: As the number of nodes increases in the network bandwidth overhead is happened in the network. As per observation, bandwidth usage is more as the number of nodes increased in the network.
4. Trust management: BC-HyIDS provides enough trust management as all nodes connected to the system are part of the same organization and follows all agreed smart contracts and consensus protocols.
5. Scalability: BC-HyIDS supports horizontal scalability as the number of nodes can be increased in the system. Performance of the system up to 10,000 nodes is checked in the real-time environment. More number of nodes can be added to the network.
6. Security aspects—BC-HyIDS is a more trusted and more secure system which provides security to the data or information passing through network. All nodes coordinate the operation in the distributed fashion. Making use of hash algorithms in blockchain makes the network more secure for communication.

6 Conclusion and future directions

This paper presents a novel system BC-HyIDS which is one of the first Intrusion Detection System, implemented along with blockchain features as per our knowledge. In today's modern world where most of the data is insecure because of novel attack methods used by intruders, this system is a solution for it. To make BC-HyIDS unique, this system is implemented and works in three phases such as Signature based detection, Anomaly based detection, and Signature creation and distribution. All phases work together to form a single system and is installed on every node of the network. The dataset used is CIC-IDS 2017 for testing and training various classifiers in both detection phases. Blockchain framework allows BC-HyIDS to securely exchange signature in between various nodes of distributed network. System performance is evaluated for only blockchain and for whole BC-HyIDS system. Blockchain performance is evaluated in terms of execution time, average latency, throughput and transaction processing time. Hyperledger sawtooth comes up with better performance in terms of computation cost as compared to v2.0.

BC-HyIDS and HyIDS system is also evaluated with parameters such as accuracy, detection rate, and false alarm rate. BC-HyIDS provides better performance if used along with blockchain compared to blockchain. As blockchain is used for the performance of signature based detection increased drastically. BC-HyIDS provides an increased accuracy by 2.8%, detection rate by 4.3% and reduction in false alarm rate by 2.6%. Overall BC-HyIDS provides improved performance when used with blockchain. In future work, we would like to implement these IDS on full public blockchains. Internal malicious node identification can be a big challenge when the public blockchain is used. Additionally, we are interested in checking the performance of this system using various platforms like Ethereum, Corda, Azure, Blockcypher, or Factom to check the system performance in all types of blockchains.

Abbreviations

IDS: Intrusion detection system; BC-HyIDS: Blockchain based hybrid intrusion detection system; SIDS: Signature based intrusion detection system; AIDS: Anomaly based intrusion detection system; ANN: Artificial neural network; IRF: Isolation random forest; AU: Analysis unit; DU: Distribution unit; SHA: Secure hashing algorithm.

Acknowledgements

The authors are thankful to Dr. S.H. Gawande of M.E.S. College of Engineering, Pune, India for continuous support and encouragement.

Authors' contributions

SK carried out the architecture design, related study, prototype design, implementation for blockchain framework and manuscript drafting. UV carried out statistical analysis for prototype and helped to finalize and draft the manuscript. All authors read and approved the final manuscript.

Funding

This work is not supported fully or partially by any funding organization or agency.

Availability of data and materials

The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Ethical approval

The experiments are numerical simulations on a computer platform. They do not require and ethical valuation. There is no research carried out on Humans, animals, plants, or other living tissues.

Author details

[1]Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.
[2]Department of Computer Engineering, M.E.S. College of Engineering, S. P. Pune University, Pune, Maharashtra, India.

Received: 26 December 2020 Accepted: 17 January 2022
Published online: 27 June 2022

References

1. M. Ahmed, A.N. Mahmood, H. Jiankun, A survey of network anomaly detection techniques. J. Netw. Comput. Appl. **60**, 19–31 (2016). https://doi.org/10.1016/j.jnca.2015.11.016
2. J. Manan, A. Ahmed, I. Ullah, L.M. Boulahia, D. Gaiti, Distributed intrusion detection scheme for next generation networks. J. Netw. Comput. Appl. **147**, 102422 (2019). https://doi.org/10.1016/j.jnca.2019.102422
3. P.J. Taylor, T. Dargahi, A. Dehghantanha, R.M. Parizi, K.R. Choo, A systematic literature review of blockchain cyber security. Digit. Commun. Netw. **6**, 147–156 (2020)
4. D. Berdik, S. Otoum, N. Schmidt, D. Porter, Y. Jararweh, A survey on blockchain for information systems management and security. Inf. Process. Manag. **58**(1), 102397 (2021). https://doi.org/10.1016/j.ipm.2020.102397
5. Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, L. Sun, A blockchain based truthful incentive mechanism for distributed P2P applications. IEEE Access **6**, 27324–27335 (2018). https://doi.org/10.1109/ACCESS.2018.2821705
6. T. Dinh, R. Liu, M. Zhang, G. Chen, B. Ooi, J. Wang, Untangling blockchain: a data processing view of blockchain systems. IEEE Trans. Knowl. Data Eng. **30**(7), 1366–1385 (2018). https://doi.org/10.1109/TKDE.2017.2781227
7. M. Miraz, M. Ali, Applications of blockchain technology beyond cryptocurrency. Ann. Emerg. Technol. Comput. **2**(1), 1–6 (2018)
8. W. Meng, E. Tischhauser, Q. Wang, Y. Wang, J. Han, When intrusion detection meets blockchain technology: a review. IEEE Access **6**(1), 10179–10188 (2018). https://doi.org/10.1109/ACCESS.2018.2799854
9. N. Kumar, S. Aggarwal, Core components of blockchain, in *Advances in Computers*. ed. by S. Aggarwal, N. Kumar, P. Raj (Academic Press, London, 2020). https://doi.org/10.1016/bs.adcom.2020.08.010
10. M. Jan, J. Cai, X. Gao, F. Khan, S. Mastorakis, M. Usman, M. Alazab, P. Watters, Security and blockchain convergence with internet of multimedia things: current trends, research challenges and future directions. J. Netw. Comput. Appl. **175**, 102918 (2021). https://doi.org/10.1016/j.jnca.2020.102918
11. A. Ramachandran, M. Kantarcioglu, Using blockchain and smart contracts for secure data provenance management. arXiv:1709.10000 (2017)
12. K. Toyoda, P. Mathiopoulos, I. Sasase, T. Ohtsuki, A novel blockchain-based product ownership management system (POMS) for anti counterfeits in the post supply chain. IEEE Access **5**, 17465–17477 (2017). https://doi.org/10.1109/ACCESS.2017.2720760
13. B.T. Rao, V.L. Narayana, V. Pavani, P. Anusha, Use of blockchain in malicious activity detection for improving security. Int. J. Adv. Sci. Technol. **29**(3), 9135–9146 (2020)
14. A.R. Mathew, Cyber security through blockchain technology. Int. J. Eng. Adv. Technol. **9**(1), 3821–3824 (2019)
15. J. Sengupta, S. Ruj, S. Bit, A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. J. Netw. Comput. Appl. **149**, 102481 (2020). https://doi.org/10.1016/j.jnca.2019.102481

16. C. Liang, B. Shanmugam, S. Azam, A. Karim, A. Islam, M. Zamani, S. Kavianpour, N. Idris, Intrusion detection system for the internet of things based on blockchain and multi-agent systems. Electronics **9**(7), 1120 (2020). https://doi.org/10.3390/electronics9071120
17. T. Golomb, Y. Mirsky, Y. Elovici, CIoTA: collaborative IoT anomaly detection via blockchain. arXiv:1803.03807v2, [cs.CY] (2018)
18. N. Agarwal, S. Hussain, A closer look at intrusion detection system for web applications. Secur. Commun. Netw. **2018**, 9601357 (2018). https://doi.org/10.1155/2018/9601357
19. H. Hamad, M. Al-Hoby, Managing intrusion detection as a service in cloud networks. Int. J. Comput. Appl. **41**(1), 35–40 (2012)
20. Y. Zhao, Y. Li, Q. Mu, B. Yang, Y. Yu, Secure pub-sub: blockchain-based fair payment with reputation for reliable cyber physical systems. IEEE Access **6**, 1229512303 (2018)
21. S. Cha, J. Chen, C. Su, K. Yeh, A blockchain connected gateway for BLE-based devices in the internet of things. IEEE Access **6**, 24639–24649 (2018)
22. K. Wüst, A. Gervais, Do you need a blockchain? IACR Cryptol. ePrint Arch. 375 (2017), Available http://eprint.iacr.org/2017/375
23. S. Nakamoto, BitCoin: a peer-to-peer electronic cash system (2008), Available http://bitcoin.org/bitcoin.pdf
24. E. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, M. Virza, Zerocash: decentralized anonymous payments from bitcoin, in *Proceedings of IEEE Symposium on Security and Privacy, Berkeley, CA, USA* (2014), pp. 459–474. https://doi.org/10.1109/SP.2014.36
25. G. Wood, Ethereum: a secure decentralised generalised transaction ledger. Document EIP-150 Revision (2016)
26. Linux Foundation. Hyperledger blockchain for business, Available https://www.hyperledger.org. Accessed 1 Oct 2017
27. Y. Chen, S. Chen, J. Liang, L. Feagan, W. Han, S. Huang, X. Wang, Decentralized data access control over consortium blockchains. Inf. Syst. **94**, 101590 (2020). https://doi.org/10.1016/j.is.2020.101590
28. J. Gu, B. Sun, X. Du, J. Wang, Y. Zhuang, Z. Wang, Consortium blockchain-based malware detection in mobile devices. IEEE Access **6**, 12118–12128 (2018). https://doi.org/10.1109/ACCESS.2018.2805783
29. G. Nadiammai, M. Hemalatha, Effective approach toward intrusion detection system using data mining techniques. Egypt. Inform. J. **15**, 37–50 (2014). https://doi.org/10.1016/j.eij.2013.10.003
30. R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-nemrat, S. Venkatraman, Deep learning approach for intelligent intrusion detection system. IEEE Access **7**, 14525–41550 (2019). https://doi.org/10.1109/ACCESS.2019.2895334
31. C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in cloud. J. Netw. Comput. Appl. **36**(1), 42–57 (2013). https://doi.org/10.1016/j.jnca.2012.05.003
32. A. Aburomman, M. Reaz, A survey of intrusion detection systems based on ensemble and hybrid classifiers. Comput. Secur. **65**, 35–152 (2017). https://doi.org/10.1016/j.cose.2016.11.004
33. Q. Qassim, A. Zin, M. Aziz, Anomalies classification approach for network—based intrusion detection system. Int. J. Netw. Secur. **18**(6), 1159–1172 (2016)
34. P. Tao, Z. Sun, Z. Sun, An improved intrusion detection algorithm based on GA and SVM. IEEE Access **6**, 13624–13631 (2018). https://doi.org/10.1109/ACCESS.2018.2810198
35. W. Feng, Q. Zhang, G. Hu, J.X. Huang, Mining network data for intrusion detection through combining SVMs with ant colony networks. Future Gener. Comput. Syst. **37**, 127–140 (2014). https://doi.org/10.1016/j.future.2013.06.027
36. Y. Xiao, C. Xing, T. Zhang, Z. Zhao, An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access **7**, 42210–42219 (2019). https://doi.org/10.1109/ACCESS.2019.2904620
37. X. Li, W. Chen, Q. Zhang, L. Wu, Building auto-encoder intrusion detection system based on random forest feature selection. Comput. Secur. **95**, 101851 (2020). https://doi.org/10.1016/j.cose.2020.1010851
38. Y. Zhou, G. Cheng, S. Jiang, M. Dai, Building an efficient intrusion detection system based on feature election and ensemble classifier. Comput. Netw. **174**, 107247 (2020). https://doi.org/10.1016/j.comnet.2020.107247
39. L. Li, Y. Yu, S. Bai, Y. Hou, X. Chen, An effective two-step intrusion detection approach based on binary classification and *k*-NN. IEEE Access **6**, 12060–12073 (2017). https://doi.org/10.1109/ACCESS.2017.2787719
40. S. Peddabachigiri, A. Abraham, C. Grosan, J. Thomas, Modelling of intrusion detection system using hybrid intelligent systems. J. Netw. Comput. Appl. **30**(1), 114–132 (2007). https://doi.org/10.1016/j.jnca.2005.06.003
41. C. Guo, Y. Ping, N. Liu et al., A two-level hybrid approach for intrusion detection. Neurocomputing **214**, 391–400 (2016)
42. K. Wu, Z. Chen, W. Li, A novel intrusion detection model for a massive network using convolutional neural networks. IEEE Access **6**, 50850–50859 (2018). https://doi.org/10.1109/ACCESS.2018.2868993
43. Z.A. Baig, S.M. Sait, A.R. Shaheen, GMDH-based networks for intelligent intrusion detection. Eng. Appl. Artif. Intell. **26**(7), 1731–1740 (2013). https://doi.org/10.1016/j.engappai.2013.03.008
44. M. AbdShehab, N. Kahraman, A weighted voting ensemble of efficient regularized extreme learning machine. Comput. Electr. Eng. **85**, 106639 (2020). https://doi.org/10.1016/j.compeleceng.2020.106639
45. S. Kumari, D. Kumar, M. Mittal, An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. Int. J. Cogn. Comput. Eng. **2**, 40–46 (2021)
46. M. Signorini, M. Pontecorvi, W. Kanoun, R. Pietro, BAD: a blockchain anomaly detection solution. IEEE Access **8**, 173481–173490 (2020). https://doi.org/10.1109/ACCESS.2020.3025622
47. M. Signorini, M. Pontecorvi, W. Kanoun, R.D. Pietro, Advise: anomaly detection tool for blockchain systems, in *2018 IEEE World Congress on Services (SERVICES)* (2018), pp. 65–66
48. X. Wang, J. He, Z. Xie, G. Zhao, S. Cheung, Contractguard: defend ethereum smart contracts with embedded intrusion detection. IEEE Trans. Serv. Comput. **13**(2), 314–328 (2020). https://doi.org/10.1109/TSC.2019.2949561
49. P. Gaži, A. Kiayias, D. Zindros, Proof-of-stake sidechains, in *2019 IEEE Symposium on Security and Privacy (SP)* (2019), pp. 139–156. https://doi.org/10.1109/SP.2019.00040
50. C. Fan, S. Ghaemi, H. Khazaei, P. Musilek, Performance evaluation of blockchain systems: a systematic survey. IEEE Access **8**, 126927–126950 (2020). https://doi.org/10.1109/ACCESS.2020.3006078

51. S. Smetanin, A. Ometov, M. Komarov, P. Masek, Y. Koucheryavy, Blockchain evaluation approaches: state-of-the-art and future perspective. Sensors **20**(12), 3358 (2020). https://doi.org/10.3390/s20123358
52. Q. Nasir, I. Qasse, M. Talib, A. Nassif, Performance analysis of hyperledger fabric platforms. Secur. Commun. Netw. **2018**, 3976093 (2018). https://doi.org/10.1155/2018/3976093
53. A. Baliga, I. Subhod, P. Kamat, S. Chatterjee, Performance evaluation of the quorum blockchain platform. arXiv:1809.03421 (2018)
54. S. Pongnumkul, C. Siripanpornchana, S. Thajchayapong, Performance analysis of private blockchain platforms in varying workloads, in *26th International Conference on Computer Communications and Networks, ICCCN* (2017). https://doi.org/10.1109/ICCCN.2017.8038517
55. W. Al-Yaseen, Z. Othman, Z. Nazri, Real-time multi-agent system for an adaptive intrusion detection system. Pattern Recognit. Lett. **65**, 56–64 (2016)
56. T. Hwang, T. Lee, Y. Lee, A three tier IDS via data mining approach, in *Proceedings of the 3rd Annual ACM Workshop on Mining Network Data, San Diego, CA* (2007), pp. 1–6
57. L. Kuang, M. Zulkernine, An anomaly intrusion detection method using the CSI-KNN algorithm, in *ACM Symposium on Applied Computing, Fortaleza (Ceara, Brazil)* (2008), pp. 921–926
58. G. Folino, P. Sabatino, Ensemble based collaborative and distributed intrusion detection systems: a survey. J. Netw. Comput. Appl. **66**, 1–16 (2016)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.