IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 January 2023, accepted 3 March 2023, date of publication 10 March 2023, date of current version 22 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3256277

RESEARCH ARTICLE

Hybrid Chain: Blockchain Enabled Framework for Bi-Level Intrusion Detection and Graph-Based Mitigation for Security Provisioning in Edge Assisted IoT Environment

AHMED A. M. SHARADQH[1], HAZEM ABDEL MAJID HATAMLEH[2], AS'AD MAHMOUD AS'AD ALNASER[2], SAID S. SALOUM[3], AND TAREQ A. ALAWNEH[1]

[1]Electrical Engineering Department, Al-Balqa Applied University, Amman 11134, Jordan
[2]Applied Science Department, Al-Balqa Applied University, Ajloun 26824, Jordan
[3]Computer Engineering and Networks Department, Jouf University, Sakaka 42421, Saudi Arabia

Corresponding author: Ahmed A. M. Sharadqh (dr.ahmed.sharadqah@bau.edu.jo)

ABSTRACT Internet of Things (IoT) is an emerging technology and its applications are flattering amidst many users, as it makes everything easier. As a consequence of its massive growth, security and privacy are becoming crucial issues where the IoT devices are perpetually vulnerable to cyber-attacks. To overcome this issue, intrusion detection and mitigation is accomplished which enhances the security in IoT networks. In this paper, we proposed Blockchain entrenched Bi-level intrusion detection and graph based mitigation framework named as HybridChain-IDS. The proposed work embrace four sequential processes includes time-based authentication, user scheduling and access control, bi-level intrusion detection and attack graph generation. Initially, we perform time-based authentication to authenticate the legitimate users using NIK-512 hashing algorithm, password and registered time are stored in Hybridchain which is an assimilation of blockchain and Trusted Execution Environment (TEE) which enhances data privacy and security. After that, we perform user scheduling using Cheetah Optimization Algorithm (COA) which reduces the complexity and then the access control is provided to authorized users by smart contract by considering their trust and permission level. Then, we accomplish bi-level intrusion detection using ResCapsNet which extracts sufficient features and classified effectively. Finally, risk of the attack is evaluated, and then the attacks graphs are generated by employing Enhanced k-nearest neighbor (KNN) algorithm to identify the attack path. Furthermore, the countermeasures are taken based on the attack risk level and the attack graph is stored in Hybridchain for eventual attack prediction. The implementation of this proposed work is directed by network simulator of NS-3.26 and the performance of the proposed HybridChain-IDS is enumerated based on various performance metrics.

INDEX TERMS IoT network security, hybrid chain, access control, intrusion detection system (IDS), attack graph generation, deep learning method.

I. INTRODUCTION

In recent years, the proliferation of Internet of Things (IoT) has gone massive day by day. The real-time systems which are built by adopting IoT are known as Cyber-Physical Systems (CPS) [1]. The IoT is defined as the distributed internet connections over smart devices for various real-time applications in which lot of smart sensors are playing a major part. It is estimated that, the usage of IoT devices increases to 500 billion by 2025 and beyond. Many of the applications associated with adoption of IoT are civilian purposes, smart agriculture, detection and tracking of an object, etc.., [2], [3], [4] Even though, IoT technology is likely to be applied in various fields however, since its development security is considered

The associate editor coordinating the review of this manuscript and approving it for publication was Hang Shen.

as a major problem. The security threats in the IoT devices would drastically affect the QoS and also theft the users' valuable private information [5]. To overcome the security issues, many of the existing works have undergone research based on IoT security.

Former security measures taken by the existing works includes authentication, access control methods, deployment of firewall, and trust computation of the users etc., [6]. Although the above prior methods provides security but lack of considerable intelligence leads to poor detection. In addition to that, the existing works were limited with any one of the prior methods which also increases the security threats. To provide strong detection and defense mechanism which are suitable for large-scale dynamic environment, Intrusion Detection System (IDS) for IoT are performed [7], [8]. The IDS is considered the effective tool for IoT which can handle large number of real-time flows which can detect and mitigate the nature of flows (i.e. normal or malicious). The IDS can be classified into two types such as [9],

- Signature-based Intrusion Detection System
- Anomaly-based Intrusion Detection System

Many of the existing works adopted signature IDS for IoT by training the tool with many real-time datasets such as UNSW-NB15, DAS-CIDS, NSL-KDD, CIDDS-001, etc.., [10], [11]. However, the signature-based IDSs are limited with detecting only known attack patterns while leveraging the unknown attack patterns leads to poor security. Besides, anomaly-based IDS is also adopted by many of the existing works for detecting unknown attack patterns [12]. However, the features taken for anomaly traffic detection by the existing works were not so effective also leads to poor security. The Artificial Intelligence (AI) algorithms such as Machine learning (ML) and Deep Learning (DL) algorithms play a major part in IDS [13], [14]. However, the adoption of conventional algorithms lacks with high complexity and speed. In some works, blockchain technologies along with AI algorithms are also adopted whereas the conventional blockchain structure limits the scalability [15]. To leverage the existing issues, the proposed work adopts robust IDS mitigation mechanism for IoT using advanced DL and blockchain technology.

A. AIM AND OBJECTIVES

The main aim of this research is to provide security in the IoT environment by performing bi-level intrusion detection and alert generation. In addition, the research also identifies the problems of considering security scarcity, false alarm rate and poor countermeasure. The main objectives of this research are to provide security by implementing an intrusion detection system with low computational time and high accuracy. The remaining objectives of this research are described as follow,

- To enhance the security in IoT, authentication was performed to authenticate the legitimate users and access control was provide to preserve data privacy by considering the generated access policy based on their attributes and role.
- To improve the intrusion detection accuracy, bi-level intrusion detection are performed to detect the malicious traffic in the network by extracting the significant features.
- For timely detection and mitigation, the risk assessment is evaluated and attack graphs are generated to detect the attack in advance and attack paths are evaluated from the attack graph for alert generation.

B. RESEARCH MOTIVATION

To provide security in IoT, many existing works perform intrusion detection system and alarm generation to reduce the malicious traffic in the network and severity level of attack which lack of detection accuracy, network security scarcity, poor mitigation and risk assessment. We are motivated to solve the existing problems are described as follows,

i. **Lack of Security:** Most of the existing work, wouldn't provide any access control to the legitimate users and some existing work issues all kinds of access to the legitimate users which leads to security breaches and data privacy scarcity. In addition, the user's data were stored in cloud server where the attackers can easily access that leads to security breaches.
ii. **High False Positive Rate:** Several existing works categorize the type of attack by only considering the limited features (i.e. statistical features) which leads to high false positives. In addition to that, all the existing work classifies the user's traffic as normal and abnormal however, the suspicious packets are either fall into normal or malicious also leads to high false positive rate.
iii. **Poor Mitigation & risk assessment:** After attack detection, in many of the existing works the countermeasure (i.e. alert generation) was not taken and then the most of work alert generation was performed randomly which leads to poor mitigation. In addition, the risk assessment was not evaluated to determine the attack severity level and the network was unaware of the severity level.

C. RESEARCH CONTRIBUTION

In this paper, we proposed HybridChain-IDS framework for enhancing security in IoT network by performing intrusion detection. The contribution of this research are illustrated as follows,

- In first, we perform Time-based authentication by utilizing NIK-512 to authenticate the legitimate users using Trust Authority (TA) which display the user registered time, then it provides security key to users through acquiring password and registered. Then, the user account will block more than three failure attempts. Moreover, the registered time and password are stored in hash format in HybridChain which is an integration of TEE and blockchain which improves network scalability and data privacy.
- In second, before accommodates the access control, the authenticated users are scheduled based on numerous

parameters delay, throughput, resource energy and priority using Cheetah Optimization Algorithm (COA) which reduces the complexity. Then the access control is provided by Smart Contract to the authorized users based on their trust and permission level.

• In third, we implement effective bi-level intrusion detection using sufficient features where the suspicious packets are examined again to identify whether the packets are normal or malicious that helps to enhance the network security.

Finally, risk assessment is evaluated to analyse the impact of the attacks and the attack graph is constructed to identify the attack path. After that, the risk-based countermeasures are taken to strengthen the network security and the attack graph is stored in Hybridchain for eventual attack prediction. Furthermore, the network are refreshed and reconstructed to prevent from packet loss.

D. PAPER ORGANIZATION

This paper is farther organized into several sections which are defined as follows, Section II represents the existing research works and its limitations. Section III demonstrates the major problems which are faced on intrusion detection in IoT. Section IV illustrates the Proposed HybridChain-IDS research methodology which encompasses of mathematical equations, pseudocode and algorithm workflows. Section V describes the simulation setup and comparison results of the simulation results and research summary of proposed work. Section VI terminates the HybridChain-IDS framework.

II. LITERATURE REVIEW

In this section, the literatures of existing works are summarized, which are associated with the proposed HybridChain-IDS framework. In addition to that, the sections encountered several research gaps and its limitations. This section further categorized into three subdivisions which are represented as follows,

A. INTRUSION DETECTION SYSTEM USING MACHINE LEARNING

In work [16], the author proposed a method for anomaly detection using traffic features in IoT network. Initially, The IOT gateway centric security monitoring system collect the IOT device traffic in centralized location. The TCPdump and Wireshark packet analyzer was used to capture the traffic data then the information entropy was extracted from the traffic data. The Naive Bayes (NB) algorithm was used for extracting the statistical features and classification it classifies into two classes such as benign and malicious. Finally, the module cut off the communication between the infected node and an alert notification was sent to other users. However, the authentication was not performed where all the users are considered as legitimate users which affect the network security. Two-level anomaly detection based on flow features in IoT network was proposed [17]. Initially, the TCPdump and Wireshark are used for analyzing the network flow and then the flow-based features are extracted from the network flow. Based on the flow features the Decision tree (DT) algorithm was used for binary classification which classifies into normal and malicious. Finally, the types of attack such as brute force, heartbleed, botnet, DOS and DDOS are classified by Random Forest (RF) from the occupied malicious network flow. However, the Random forest was used for categorizing the attack types which generate the large number of tree while training the attack detection model which leads to high complexity. In work [18], the author proposed a security model for smart monitoring and attack detection. Initially, every device in IoT network are registered with an edge device then based on the query, time and location the authentication mechanism was proceeded. After that, the JnetPcap has utilized for capture the network packet and decode it, subsequently, the feature extraction was performed by PcapWT. The classification was performed by Support vector machine (SVM), Artificial neural network (ANN) and Decision tree (DT) thus classifies into malignant and benign traffic. Finally, the malignant traffic was categorized into DOS, DDOS and Botnet then the alert was generated. However, the edge device was authenticated by only considering the query, time and location which are insufficient to estimate their legitimacy that affects the network security level.

B. INTRUSION DETECTION SYSTEM USING MACHINE LEARNING

An adversarial attack detection model with only black-box access in the IoT network was proposed [19]. Initially, the raw packets are collected and the relevant information (i.e. IP address, MAC address, port and packet size and packet timestamp) are extracted using NFQueue and Tshark. The temporal statistics are calculated and the statistical features are extracted and clustered to form a feature map by Monte-Carlo method. Finally, the three-layer auto-encoders are integrated to learn the behaviours of each cluster and classify into normal and malicious. However, all the users are considered as legitimate users where the malicious traffic will increase in network due to the presence of illegitimate users which leads to security breaches. In work [20], the author proposed a lightweight method for intrusion detection system in IoT environment. Initially, all type of access control was given to every legitimate users to access the data. Then the B-events collection component monitors and records the user current activity to train the anomaly detection model. Based on the network traffic threshold the auto-encoders classify the network traffic as normal or intrusion. Finally, the D-alarm component blocks the intruder user and then notification was sent to the system administrator to take necessary measures. However, without authenticating the legitimate users the access control was provided to all the users where the attackers can easily access the data which affects the network security. Collaborative intrusion detection model using deep learning architecture in IoT network was proposed [21].

Initially, the CICFlowMeter tool was deployed to process the data and to extract the features such as destination port, protocol, flow duration, the total number of packets in the forward direction, the number of packets per second of traffic flows, and the average size of the packet. Then the ensemble-based multi-feature selection was used to select the important features based on specific threshold. Based on the feature extraction the Generative Adversarial Network (GAN) was utilized to classify as normal or malicious traffic. However, the Generative Adversarial Network (GAN) was used for classification where the traditional problem of this algorithm is it will unstable during the training which become harder to train that leads to high false positive rate.

In work [22], the author proposed a low-complexity cyber-attack detection in IoT edge computing (LocKedge) for multi-attack detection. Initially, the raw traffic data are normalized by min-max normalization method to convert into numerical and categorical. The feature extraction was performed by principal component analysis (PCA) which extracts the features and reduces the dimension. Finally, based on the features extraction, the LocKedge utilize the traditional neural network algorithm to classify the multiple attacks such as DOS, DDOS, OS, fingerprint etc.., However, the feature extraction was performed using principal component analysis (PCA) which is not able to find optimal principal components and is sensitive to outliers where the feature extractions are not efficient which affects the detection accuracy. In work [23], deep learning integrated with optimization algorithm to perform intrusion detection in IoT network. Initially, the work consists of three-phase data collection, pre-processing and intrusion detection. The data are collected and pre-processed, where the data are standardized and into standard normal distribution to reduce data redundancy. Finally, the based on the statistical features, intrusion detection was performed by Adaptive particle swarm optimization algorithm with convolutional neural network (APSO-CNN) to reduce the training complexity and increase intrusion detection accuracy. However, the intrusion detection was performed using APSO-CNN which preform effectively, the statistical features are only considered for intrusion detection where the inadequate features leads to high false positive rate. Author in [24], proposed a novel deep learning enable intrusion detection mechanism. Initially, the framework consists of four modules including database module, intrusion detection system module, controller module and synthesizer module. The raw data packets are captured by data collector and packets are processed by label coding, feature scaling and feature extraction. The generative adversarial network (GAN) was employed, which generates synthetic samples to overcome the data imbalance issue. Then the controller module performs two task where sending synthetic request to the IDS module and evaluating the pending request. Finally, the intruder was detected using GAN. However, this work considers all the users as legitimate users and allows the users to access the network which leads to security breaches due to presence of illegitimate users.

C. INTRUSION DETECTION SYSTEM USING MACHINE LEARNING

Author in [25], proposed a distributed consensus based trust model (DCONST) to detect the multiple-mix attack. Initially, the Trust authority (TA) distributed the asymmetric key to IoT nodes and during the data communication between two nodes the trust authority provide symmetric key to both sender and receiver to encrypt the transferred packets. The trust evaluation was performed to measure the node reputation by sharing their cognition then the DCONST model detect the malicious node and begin node. Then the DCONST detect the concrete attack behaviours and cluster them by K-Means clustering subsequently the malicious node was categorized into DCONST-light, DCONST-normal and DCONST-proactive. However, after attack detection the malicious node was categorized into light, normal or proactive, where there is no countermeasure taken to alert the users and block the malicious node which leads to security breaches.Blockchain enable framework for intrusion detection in IoT Fog-Cloud architecture was proposed in [26]. The privacy-preserving blockchain was employed for secure data transmission. Initially, all the entities in the network are authenticated by Trust Authority (TA) and security was provided. Then the feature extraction was performed by principal component analysis (PCA) thereby reducing dimension. Finally, the intruders are classified using Gradient boosting algorithm. However, the PCA was utilized for feature extraction where this algorithm consumes huge time while working with outliers and missing values which increase high latency. Author in [27], proposed a distributed intrusion detection framework using fog computing to improve network security. Initially, the raw traffic are captured and pre-processed by standard Scaler normalization method. The blockchain was implemented for security purpose and mining pool was integrated with intrusion detection system to detect suspicious attack. Finally, the statistical features and packet features are extracted then XGBoost and Random forest algorithm was utilized separately for intrusion detection where XGBoost achieve better accuracy. However, the blockchain was accomplished for privacy preserving which increase network security, but this traditional blockchain suffers with non-scalability.

III. PROBLEM STATEMENT

DNN based network intrusion detection model for IoT gateways was proposed [28]. The network traffic is captured and statistical features are extracted by Damped incremental statistical algorithm to detect the intrusion and countermeasure was taken. The main problems of this research are listed as follows,

- Here all the IoT users are considered as legitimate users where the number of malicious traffic increases due to the presence of malicious users which affects the security level in the network.
- In this work, the Damped incremental statistical algorithm was utilized for feature extraction, where the

limited feature (statistical feature) was extracted this is not enough to analyse the attack category which leads to low detection accuracy.
- The proposed attack detection model utilizes the deep neural network, which requires massive data to train the model which leads to high computational power and increase high complexity.

Anomaly-based intrusion detection framework to protect the IoT device was introduced in [29]. Then the statistical features are extracted and intrusion was classified by one-class algorithms and precautions were taken by action manager. The major problems of this research are explained below,

- The proposed Passban signal an anomaly even if the incoming traffic contains a pattern that is not an attack, but somehow it diverges from the routine traffic which leads to false positive rate.
- Here, the isolation forest algorithm was utilized for classification where the model generate large number of tress while leads to high computational time.
- In this work, the alert notification was sent to the users randomly without determine the attack path which leads to high latency.

A deep blockchain framework to execute security-based collaborative intrusion detection system (CIDS) was proposed in [30]. Here, privacy based encrypted data transmission was accomplished using blockchain and (Bi-LSTM) was utilized for CIDS at cloud network. The problems of this research are defined as follow,

- The deep blockchain framework was utilized for data preserving and privacy data transmission even though it performs well, it is traditional blockchain that suffers from lack of confidentiality due to its non-scalability.
- Here the Bidirectional LSTM was proposed to perform CIDS, which will hinder its applicability on large data and high energy consumption due to its high computational time.
- In this work, the network traffic was classified as normal or malicious, whereas the suspicious traffic will be taken as normal traffic which leads to a high false positive rate.

Collaborative intrusion detection system (CoLL-IoT) to detect the malicious activities in IoT device was introduced in [31]. The raw packets are captured by chi square algorithm for feature extraction and intrusion detection was classified using XGBoost. This major problems of this research includes are narrated below,

- Here, the XGBoost algorithm was utilized for intrusion detection where the algorithm does not perform well on sparse and unstructured data and the algorithm is very sensitive to the outliers which leads to hardly scalable.
- In this work, the feature extraction was performed by chi-square method where it is difficult to interpretation and it need large sample size which leads to high energy consumption.

Research Solutions: Initially, time-based authentication is performed by Nik-512 hashing algorithm which will hash the users password and registered time and store it in blockchain that leads to increase the security level, then the access control is provided to the user based on their role by priority entrench user scheduling to achieve better QoS thereby reduce in complexity. The Bi-level intrusion detection is implemented by extracting the significant features to categorize the attack types which will improve the detection accuracy. The feature extraction and intrusions detection is performed by ResCapNet which combines of Residual network and capsule network where the capsule network extracts the features significantly with small sample size which reduces the high energy consumption. Furthermore, bi-level intrusion detection are executed where the first level IDS classifies normal, suspicious and malicious then the suspicious traffic are classified as normal or malicious in second level which will reduce the false positive rate. Then the attack graph is generated to detect the attack path which will utilize to notify the users optimally which helps to reduce the high latency. Finally, hybrid chain is proposed by combining blockchain and trusted execution environment (TEE) which minimizes the computational burden and increases the blockchain and privacy scalability.

IV. HYBRIDCHAIN-IDS SYSTEM MODEL

In this research, we concentrate on providing security in the IoT environment through effective bi-level intrusion detection. This proposed methodology consists of several layers including physical layer consists of IoT users (i.e. IoT devices), edge layer consists of edge nodes and cloud layers consists of cloud storage. Figure 1 represents the architecture of the proposed HybridChain-IDS framework. In this work, we proposed Hybrid chain, which is combined of blockchain and trusted execution environment that helps to reduce the network computation burden and provide high security. The blockchain-based authentication and access control is proposed to achieve high security and privacy preservation.

A. PHYSICAL LAYER

This is a fundamental layer of IoT network which is responsible for gathering data from all IoT users for data transmission and storing their data in cloud server from various sensors in secure manner. The IoT devices can access in any location through mobile phones, laptops, computers, etc..,

B. TRUSTED AUTHORITY (TA)

The Trust Authority is deployed in the physical layer by blockchain for providing authenticity to IoT users by achieving their credentials and affording them with security keys.

C. EDGE LAYER

The edge layer is comprised of several edge nodes which are responsible for collecting the network traffic. Furthermore, the bi-level of IDS is accomplished in edge layer to strengthen the security and privacy of IoT users.

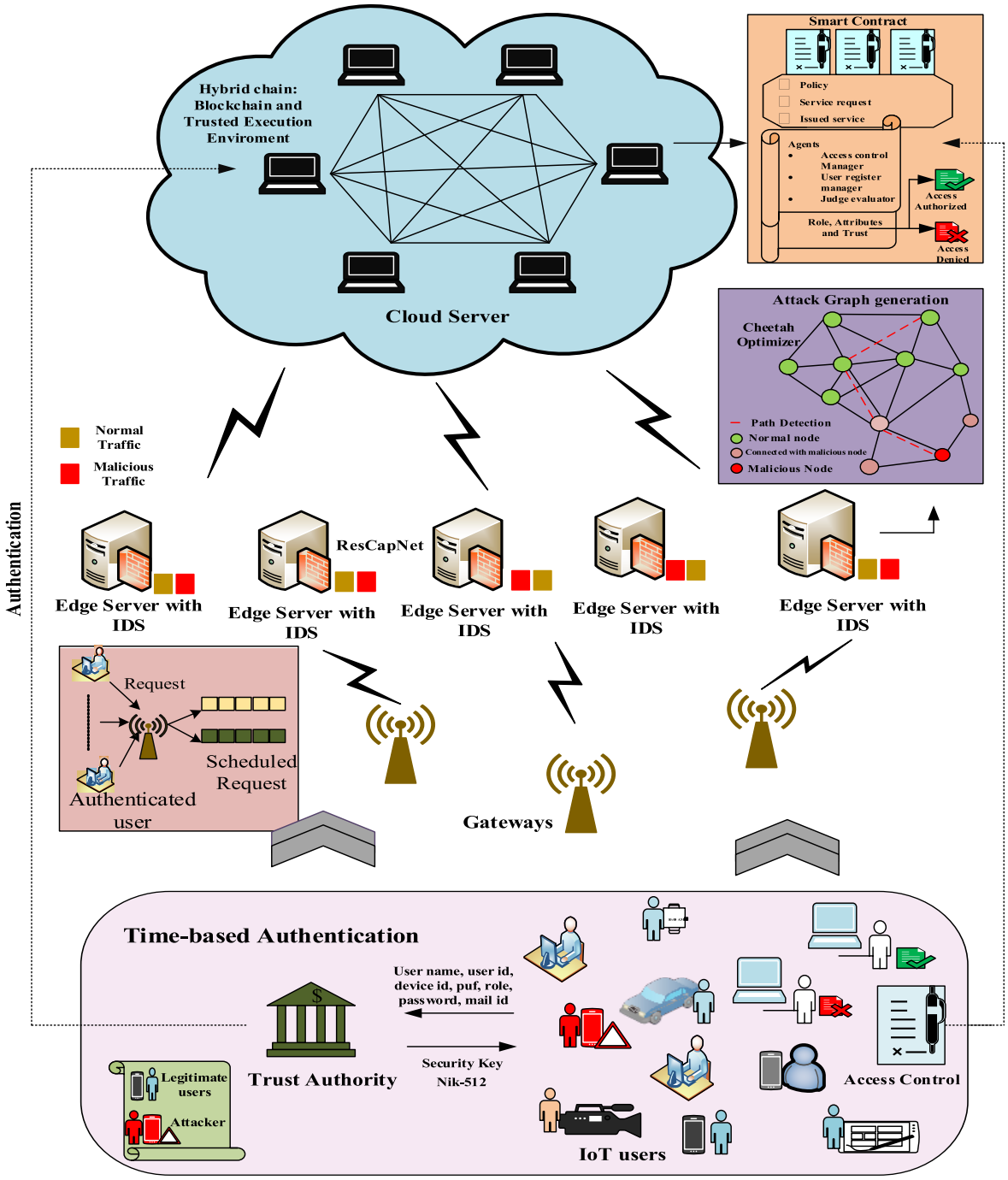

FIGURE 1. Architecture of the proposed HybridChain framework.

D. CLOUD LAYER

The cloud layer is composed of blockchain to increase network security and reduce computational burden. Moreover, it is responsible for performing countermeasures to enhance network security.

E. HYBRID BLOCKCHAIN

The hybrid chain incorporates blockchain with Trusted Execution Environment (TEE) which adopts hierarchical network for minimizing computational burden and allows storing transactions securely.

1) TIME-BASED AUTHENTICATION

Initially, we perform authenticating the IoT users ($\ni$) to ensure legitimacy. For that, the $\ni$ are register with their details such as user name ($\mathbb{N}$), user ID (α), device ID (Δ), PUF (β), role ($\nexists$), password (ρ) and mail ID (ϑ) to the trusted authority (TA) which sends the details into the blockchain to enhance security. After registering, the trust authority displays the

user registered time in hour (δ), minute (γ) and seconds (ε) then TA provides security key, based on the user credentials including password (ρ) and their registered time hour (δ), minute (γ) and seconds (ε). The steps involved in registration and authentication are defined below.

• Step 1: Initially, the ($\ni$) is registered to Trust Authority by providing the credentials ($\daleth$), (α), (Δ), (β), ($\nexists$), (ρ) and (ϑ) which can be composed,

$$TA \leftarrow Reg\left\{(\daleth), (\alpha), (\Delta), (\beta), (\nexists), (\rho), (\vartheta)\right\} \tag{1}$$

where, $Reg\left\{(\daleth), (\alpha), (\Delta), (\beta), (\nexists), (\rho), (\vartheta)\right\}$ denotes the registration of ($\ni$) with parameters ($\daleth$), (α), (Δ), (β), ($\nexists$), (ρ), *and* (ϑ) respectively.

• Step 2: Once ($\ni$) is registered, the Trust Authority displayed the user registered time which is used for user login.

$$TA \leftarrow dis\left\{(\delta), (\gamma), (\varepsilon)\right\} \tag{2}$$

where, $dis\left\{(\delta), (\gamma)\ (\varepsilon)\right\}$ denotes the hour (δ), minute (γ) and seconds (ε) which display ($\ni$) registered time.

• Step 3: After ($\ni$) registered, the TA generates the 512-bit secret key to the registered user for authentication which is illustrated as,

$$TA \leftarrow SK_{512}[(\rho), (\delta), (\gamma), (\varepsilon)] \tag{3}$$

where, (ρ), (δ), (γ), (ε) denotes the password with user registered hour, minute and seconds. After authentication, the blockchain stores the password and user registered time in hash format which cannot compromise by the attackers this improves the security level. For that purpose, we proposed NiK-512 hashing algorithm which resistant all cryptographic attacks, including quantum collision attacks. The cryptographic hash function is developed for hash output length of 512-bits which utilizes Miyaguchi-Preneel Structure for it generates 512-bits long values and X is stored as an array of 16 32-bits elements. The hashing of SK_{512} are divided into 512-bit blocks and the padding of last block with zero to the proportion of 512 bits.

In the beginning, the S is computed as 0, the function works whether keyless mode or while key value assuming that key mode is utilized. The SK_{512} computed with first block of password and registered time block being processed. Furthermore, the compression function is employed which taken from internal state of current values then password and registered time block that have to be processed. The input obtains the arrays of current values X and Y. In addition, the classical memory required are estimated as:

$$m = 2^{l/5} \tag{4}$$

For $\bar{\mathrm{A}}_i$, according to the formula (4) the transformation is executed (modulo 16 are taken as indexes of elements).

$$\mathrm{X}_i := (\mathrm{X}_i \gg 1) \oplus (\neg Y_i) \oplus (X_{i+6} \wedge \neg Y_{i+3}) \tag{5}$$

The array is revolved, the element of $(i)-$ th becomes $(i-1)-$ th element, the zero element as last element. For X_i with indices, $2 \leq i \leq 16$ transformation is executed as:

$$\mathrm{X}_i := (\neg \mathrm{X}_i \gg 1) \oplus Y_i \oplus \left(\neg X_{i-6} \bigwedge Y_{i-3}\right) \tag{6}$$

For all X_i and Y_i, the transformation are executed according to following formula:

$$\begin{cases} \mathrm{X}_i := \mathrm{X}_i mod 2^{32} \\ \mathrm{Y}_i := \left(\mathrm{Y}_i + \mathrm{X}_i x \mathrm{X}_{(i+d^3) mod 16}\right) mod 2^{32} \end{cases} \tag{7}$$

where d denoted as number of current round (i.e. for first round $d = 0$ and $d = 31$ as last round). Furthermore, then the secret key hashed with 512-bits (SK_{512}) was generated and provided to the $\ni$ by TA. The user should remember the time displayed after registration, during every login the user intends to enter their username, password and displayed register time. If the user forgot their password or registered time, then by choosing forgot password the trust authority will send a security code to the user registered mail id that allows the user can view their password and registered time which is also limited to only three times. The threshold T is calculated as:

$$T\ (r, s) = -\sum_{n \epsilon \chi} r(n) \log s(n) \tag{8}$$

where, r *and* s are discrete probability distribution and n denotes the threshold range limited for the user which is set as three threshold ranges.

$$\mathfrak{Q} = \begin{cases} 0 & if\ 0.3 \geq n\ Mail\ Generated \\ 1 & if\ 0.3 < n\ User\ Blocked \end{cases} \tag{9}$$

Through this authentication, the security level is increased and unauthorized users are eliminated which reduces the computational complexity and malicious traffic in the network. Furthermore, by storing the (SK_{512}) in hash format at hybrid permission blockchain, the confidentiality is enhanced. The hybrid chain combines of blockchain with Trusted Execution Environment (TEE) it comprises of four layers includes data layer, verification layer, estimation layer and application layer. The computational burden are minimized by employing hierarchical network by reducing latency of on-chain by executing major heavy weight computation in off-chain. The hybrid chain is advantaged by enabling each participant to share their data through secure communication protocol. Moreover, the hybrid chain elongates the reservation memory, which permits the blockchain application to execute in TEE that enhances the storing of transactions securely and documentation of whole storage of key-value codes situated in TEE outside. The data layer consists of data storage and techniques of encryption in blockchain embraces of chain structure, data blocks, hash function and digital signature. The verification layer comprises of transmission protocol where the verification of execution result is performed by utilizing Practical Byzantine Fault Tolerance (PBFT) consensus algorithm. The estimation layer is conducted for verification of transactions and execution of smart contract in Virtual machine (VM) and key

Pseudocode Time-Based Authentication

Input: User Credentials
$\{(\mathbb{N}), (\alpha), (\Delta), (\beta), (\nexists), (\rho), (\vartheta)\}$
Output: Authenticated or Not
Begin
For all User in registration $\{(\mathbb{N}, \alpha, \Delta, \beta, \nexists, \rho, \vartheta)\}$ ***do***
Perform registration using Eq. (1)
Display $\ni$ registered time using Eq. (2)
NiK-512 generate SK_{512} using Eq. (3) and Store $\ni$ credentials and SK_{512} in Blockchain
End For
For User in Login Phase **do**
If $(\acute{E}_{pr} == \cup_{pl})$ **then** // $\acute{E}_{pr}$ user current password and time, $\cup_{pl}$ registered password and time.
Authentication success
Else
Execute threshold range for $\ni$ password recovery using Eq. (8)
If $(0.3 \geq n)$ **then** // n threshold range
obtain password
Else
Account Blocked
End if
End if
End For
End

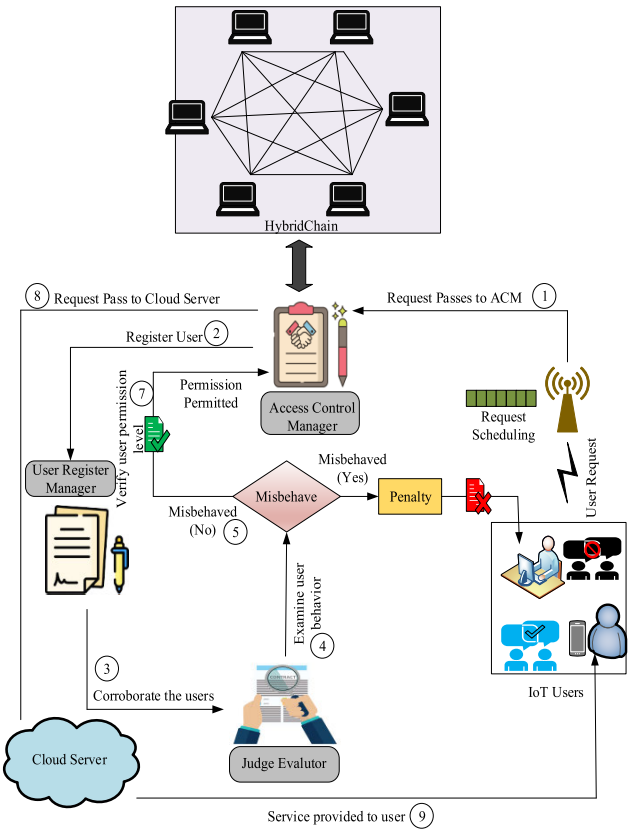

FIGURE 2. User scheduling and access control.

management. Furthermore, the access control is provided by confidentiality preserving smart contract with high performance. The application layer is configurable implementation of blockchain, smart contracts and algorithms.

2) USER REQUEST SCHEDULING AND ACCESS CONTROL

After successful authentication, based on the user role and attribute the policy is generated. The user sends a service request to the blockchain where the access control manager in smart contract collects the user request that consists of mixed types of service requests. Hence we need to schedule the user request to reduce the latency and waiting time which helps to use available user resources. Figure 2 illustrates the flowchart of user scheduling and access control. For that, we proposed Cheetah Optimization Algorithm (COA), the steps involved are represented below as follows,

In order to perform user scheduling, the cheetah (cloud server) search the hunting prey (user) in two modes includes scanning mode and active mode, depending upon the fitness value the cheetah might select the optimal mode to hunt the prey. The fineness value where estimated by considering delay, throughput, resource energy and priority. Moreover, the cheetah optimization algorithm consists of three strategy includes searching strategy, sitting & waiting strategy and attack strategy (rushing and capturing). In mathematical modelling of searching strategy of cheetahs, assume $C_{i,j}^{k}$ represent the current position of cheetah $i = (1, 2, \cdots, m)$ at arrangement of $j = (1, 2, \cdots, L)$) where m denotes number of cheetah population and L is optimization problem in dimension. Furthermore, the new position of cheetah i is updated by utilizing random search function, arbitrary step size and current position of each arrangement are follows,

$$C_{i,j}^{k+1} C_{i,j}^{k} + \hat{z}_{i,j}^{-1} . \tau_{i,j}^{k} \tag{10}$$

where $C_{i,j}^{k+1}$ and $C_{i,j}^{k}$ are the forthcoming and current position of cheetah i at arrangement j, k denotes time of current hunting, $\tau_{i,j}^{k}$ and $\hat{z}_{i,j}^{-1}$ are step length and randomized parameter of cheetah i at arrangement j. Let, K denotes length of maximum hunting time, and the randomization parameter is the second term where the random numbers are normally distributed $\hat{z}_{i,j}$ from the standard distribution. In most slow walking search, the step length $\tau_{i,j}^{k} > 0$ is set at $0.001x\ k/K$ where in some case $\tau_{i,j}^{k}$ can be regulated between the distance of cheetah i and its leader or neighbour. The updation of every arrangement of cheetah is perform by assuming $\tau_{i,j}^{k}$ equal to $0.001\ x\ k/K$ which is multiplied by maximum of step size. For others, $\tau_{i,j}^{k}$ in every cheetah's arrangement is estimated through the multiplying distance between cheetah position i and cheetah selected randomly. Depending on the distance between the prey and leader, the leader position is chosen entrenched some variables of prey position are changed to obtain best solution. Furthermore the optimization problem can be effectively solved by employing any randomised parameter with random step size (i.e. $\hat{z}_{i,j}^{-1}$ and $\tau_{i,j}^{k}$).

The mathematical modelling of sit and wait strategy, in most of the cases the prey attempt to escape from cheetah, to avoid this the cheetah decide to trap (lying on ground or hiding among shrubs) and hunt the prey by getting closure. Hence, here the cheetah residue at its position and allow the prey to come nearer by waiting, this attempt can be illustrate as follows:

$$C_{i,j}^{k+1} = C_{i,j}^{k} \tag{11}$$

where $C_{i,j}^{k+1}$ is updated cheetah position and $C_{i,j}^{k}$ denotes current position of cheetah i at arrangement j. This strategy acquires best solution without change every cheetahs continuously, so it evade premature convergence. Besides the mathematical modelling of attack strategy, the cheetah utilizes flexibility and speed strategy to hunt the prey. The attacking approach of cheetah can be defined as follows:

$$C_{i,j}^{k+1} = C_{E,j}^{k} + \hat{z}_{i,j} \cdot v_{i,j}^{k} \tag{12}$$

where $C_{E,j}^{k}$ represent the prey current position in arrangement j (best position of current population), $\hat{z}_{i,j}$ *and* $v_{i,j}^{k}$ are turning factor and the interaction factor of cheetah arrangement. $C_{E,j}^{k}$ is the cheetah's rushing tactics by employing maximum speed to get closure to prey in short time. Hence, it evaluates the $i-$ th new position of cheetah based on current prey's position. Furthermore, the $v_{i,j}^{k}$ deliberate interaction during capturing phase between the leader and cheetah or between cheetahs. This factor, mathematically defined the difference in neighborhood cheetah's position, $C_{E,j}^{f} = (f \neq i)$ and cheetah's position $i-$ th, $C_{i,j}^{k}$. The random number of $\hat{z}_{i,j}$ turning factor is equal to

$$\left|z_{i,j}\right|^{exp(z_{i,j}/2)} \sin\left(2\pi z_{i,j}\right) \tag{13}$$

where normally distributed $z_{i,j}$ is standard normal distribution of random numbers which deliberate the cheetahs sharp turns of capturing phase. By utilizing these strategies, the cheetah optimizer performs effective user scheduling to reduce the complexity.

After user scheduling, access control is performed, in our process, the access control is provided by smart contract in blockchain where the smart contract generates by multiple agents to manage data and service sharing among network users. The multiple agents are the Access control manager (ACM), user register manager (URM) and judge evaluator (JE). The ACM is the main smart contract that administers the access control among IoT device. Whenever, ∋ generate the request ACM is executed and it forwards the request of ∋ by checking correlated permission level. The URM creates the registration table to store user credentials acquired while authentication and also it stores the information of user-accessed service (data) with time. Moreover, the JE judge the user behavior and evaluate the trust value based on user behavior to provide access control. The misbehavior includes, when the ∋ send numerous of request simultaneously for service and the ∋ who cancelled their generated request. Once if the ∋ have been misbehaved,

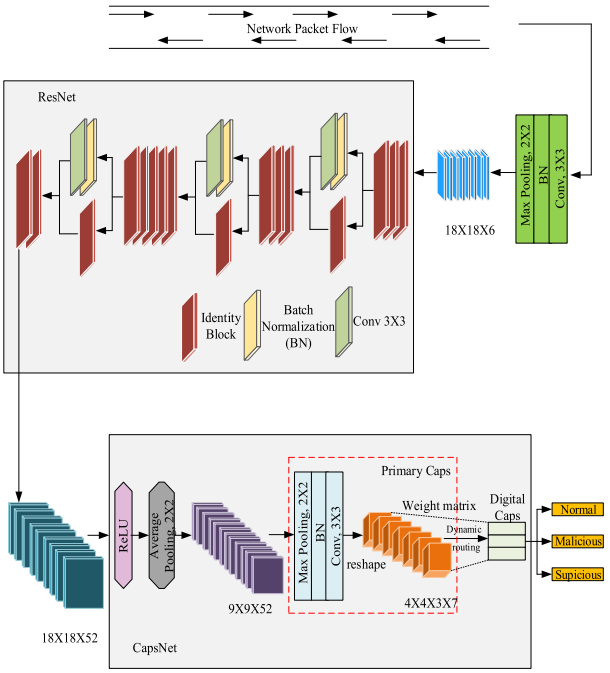

FIGURE 3. Workflow of ResCapsNet.

then the corresponding penalty is regulated for that specify person by turned off their state for particular time to reduce complexity. The trust computation embraces of input vector $\ni = \ni_1, \ni_2, \ldots$ denotes each user, weight vector (depends on user behaviour) $= G_1, G_2, \ldots$. The output of demanding the weights $G_{i(i=[1\ldots h])}$ to inputs $\ni_{j(j=[1\ldots q])}$ is the trust value which is generated based on,

$$Trust = \sum_{i=1}^{q} G_i \ni_i \tag{14}$$

Once the trust value is estimated, the trust level is assumed as low (misbehaved) and high (not misbehaved). Further, the permission levels are examined based on their role and the requests are permitted or repudiated, corresponding to their permission level and if the trust level is high. Otherwise, the access is denied and the alert message was generated for each user (i.e. Access Granted !, Requests are Concealed !, Static Check Stoped !, Misbehavior Detected !,Static Check failed & Misbehavior Triggered !). By performing user scheduling and access control, the complexity is reduced thereby enhancing security level.

3) BI-LEVEL INTRUSION DETECTION

After providing access control, bi-level intrusion detection in the network is carried out effectively to enhance network security. The bi-level intrusion detection is implemented by deep network ResCapNet algorithm which is combined of capsule network (Capsnet) and residual network (ResNet). Figure 3 demonstrates the workflow of ResCapsNet.Initially, the first level of IDS is performed in the edge layer, where the filtration of incoming network packets based on the packet

flow are captured by gateways. Then, based on the network packet flow the packet features are extracted by ResNet. The ResNet is employed to optimize the network layer and then to achieve the identity of mapping and assure that the layer of input and output identity are same. In ResNet, the identity layer are regulated automatically by performing training and several layer of this original network are changed into residual block. The residual operation is illustrated below as follows:

$$H = V_2\mu\left(V_2 a\right) \tag{15}$$

$$b = H\left(a, \{V_i\}\right) + a \tag{16}$$

$$b = H\left(a, \{V_i\}\right) + V_w a \tag{17}$$

where μ in equation (15) denotes non-linear ReLU function, b is the shortcut common output of second ReLU. The input and output dimension of equation (17) required to change, includes changing of linear transformation V_w can be execute on a using shortcut operation and number of channels.

Once the significant features are extracted, intrusion detection is accomplished by CapsNet. The Capsule network can fetch spatial information and more important features by representing the features in vector and also it can provide high accuracy in less training data which helps to reduce the high energy consumption. The CapsNet is comprised of capsules where the neuron generates its output as scalar, and capsule output as vector. The extent of each vector describes the evaluated probability of object existence, and the aspect of each vector enrol the object posture parameters incorporates exact rotation, thickness, position, object size and tilt. The CapsNet functions as equation follows,

$$\hat{e}_{j|i} = W_{ij} e_i \tag{18}$$

$$D_j = \sum_i \mathcal{C}_{ij} \hat{e}_{j|i} \tag{19}$$

where vectors are the input and output of capsule, e_i and ω_j, the output e_i of previous capsule is multiplied with affine transformation matrix W_{ij} for turning e_i into $\hat{e}_{j|i}$. Then the weighted sum D_j is estimated corresponding to weight $\mathcal{C}_{ij}$ which is coupling coefficient enumerated by the iteration of dynamic process. $\mathcal{C}_{ij}$ is measure includes capsule as i and the activate capsule as j.

$$\omega_j = \frac{\|D_j\|^2}{1 + \|D_j\|^2} \frac{D_j}{\|D_j\|} \tag{20}$$

where the activation function of D_j is compressed rather of ReLU, hence the extent of vector final output ω_j is among 0 and 1. The output of activation function is achieved through compression function. The Capsule Network evaluates the output through estimating intermediate value $\mathfrak{P}_{ij}$ by iteration of dynamic routing. The prediction vector $\hat{e}_{j|i}$ in equation (18) and (19) is the prediction through capsule i and has effect of output capsule j. The two capsules are high correlated, if activation vector has huge similarity with prediction vector where the similarity is computed through prediction vector and activation vector of scalar product. Hence, the similarity score $\mathcal{C}_{ij}$ might appraise both possibility of feature attribute and feature existence, embrace neurons, that barely contemplate the feature existence possibility. Furthermore, if activation e_i of capsule i is notably low, therefore the e_i is proportional of $\hat{e}_{j|i}$ extent, $\mathcal{C}_{ij}$ might be still low; if the detail feature of capsule is not activated, where the overall feature and correlation among detail feature is notably low. The $\mathfrak{P}_{ij}$ coupling coefficient is quantified by softmax of $\mathcal{C}_{ij}$ in equation (22).

$$\mathcal{C}_{ij} \leftarrow \hat{e}_{j|i} X \omega_j \tag{21}$$

$$\mathfrak{P}_{ij} \leftarrow \frac{exp\left(\mathcal{C}_{ij}\right)}{\sum_c exp\left(\mathcal{C}_{ic}\right)} \tag{22}$$

Hence, ResCapsNet classified the network packets into normal, malicious and suspicious. The ResCapsNet defends the integrity of the information and performs effectively which helps to improve intrusion detection accuracy. In this work, the ResCapsNet is adopted and modified to be appropriate for intrusion detection. The Resnet-34 comprises of four partitions, where each partition has 3, 4, 6 and 3 of the identity blocks. Identity block in each partition contains 64, 128, 256 and 512 filters individually. Consecutively, to extract the significant features with low complexity, convolutional kernel size is minimized in the first convolutional layer from 7 to 3. Ever since, the number of filters reduced is utilized for every identity block in four partitions subsequently to 16, 28, 40 and 52, then there is no classification layer is acquired for generating output. The dynamic routing parameter as digit caps for data is set to 3. The network traffic flow is classified in three classes (normal, malicious and suspicious) hence the numbers of vector in primary and digit caps are set to 3 furthermore the number of capsules taking part in digit caps is set to 3.

Likewise, the second level IDS are performed where the suspicious network traffic are analyzed using ResCapsNet to ensure network security. Here the packet features are again examined by ResCapNet to classify the suspicious network traffic as normal or malicious. Finally, if the malicious traffic was detected then the attack type was categorized by considering the behavioral, spatial, temporal and content features. By performing, bi-level intrusion detection the network security level is amplified.

4) ATTACK GRAPH CONSTRUCTION AND MITIGATION

After the intrusions are detected, the risk assessment is evaluated and attack graph is generated to provide risk-based countermeasures by path detection based on attack graph which are illustrated in subdivisions.

a: RISK ASSESSMENT AND ATTACK GRAPH GENERATION

Once the bi-level IDS is completed, the risk assessment was performed to identify the severity level of the intrusion. For generating attack graph and detecting shortest attack path, we need to calculate the risk of detected attacks. The attack category and the attack mode are integrated to analysis the

severity level of attack (i.e. password-based attack is considered as low risk and vulnerability based are considered as high risk). The attack impact and feasibility are estimated to execute the risk assessment. The attack impact Ⱨ is evaluated as:

$$Ⱨ = Ð + Þ + Ø \tag{23}$$

where Ð denotes data loss, Þ is legislation or privacy and Ø *represents* the operation. According to the impact parameters, the sum is perhaps generated to acquire the attack impact level. Then the attack feasibility $\mathbb{F}$ is expressed as:

$$\mathbb{F} = \mathbb{W} + \mathfrak{Z} + \mathbb{K} \tag{24}$$

where the attack feasibility is generated by considering parameters (𝕎) window of opportunity, (𝔷) equipment and (𝕂) TOE knowledge. Furthermore, the risk value Y is described as:

$$\curlyvee = \sqrt{\mathfrak{m}_{\curlyvee}(Ⱨ)^2 + \mathfrak{n}_{\curlyvee}(\mathbb{F})^2} \tag{25}$$

where $\mathfrak{m}_{\curlyvee}$ and $\mathfrak{n}_{\curlyvee}$ are weight parameters of Ⱨ and $\mathbb{F}$. The risk contributions are concluded as same by both attack impact and feasibility where $\mathfrak{m}_{\curlyvee}$ and $\mathfrak{n}_{\curlyvee}$ are set at 0.5. After risk assessment, the risk level should be regulated based on the evaluated attack impact and feasibility level. The matrix perhaps by utilizing the calculated risk value in (25) and the risk level is represented as follows:

$$\mathfrak{t} = \mathfrak{h}(Ⱨ\ell + \mathbb{F}\ell) \tag{26}$$

where $\mathfrak{t}$ is the risk level computed by Ⱨℓ value of attack impact level and $\mathbb{F}\ell$ value of feasibility level and $\mathfrak{h}$ denotes the risk function of Ⱨℓ and $\mathbb{F}\ell$. Based on the quantified risk level, attack level is categorized into low and high.

Then the attack graph was generated to detect the attack path and take optimal countermeasure action. The attack graph generation is carried out by improved k-Nearest Neighbor, the kNN is enhanced by integrating Graph Neural Network (GNN) which will effectively learn the attack structure and provide a significant attack graph. Assume that the training attack training set $\Phi = \{(x_n, \mathcal{Y}_n)\}_{n=1}^{\mathfrak{q}}$ is given, where $x_n \in \mathbb{Q}^{\varsigma}$ is the $n-$ th input vector for input variables and $\mathcal{Y}_n$ is the label vector for output variables. This method reconstructs each input vector x_n into graph $\mathcal{G}_n = \forall(x_n; \Phi)$ where $\forall$ represents the transformation function. The $\mathfrak{k}$ and $\mathfrak{X}$ are the numbers of nearest neighbor and distance function which are the two hyper parameters required to be determined which are only employed to operate the transformation function $\forall$ for kNN search from Φ; but they are not exploited explicitly in learning procedure. For every x_n, its kNN occurrence are searched from $\Phi \setminus \{(x_n, \mathcal{Y}_n)\}$ entrenched on distance function $\mathfrak{X}$, illustrated by $\mathfrak{P}(x_n) = \left\{\left(x_n^{(i)}, \mathcal{Y}_n^{(i)}\right)\right\}_{i=1}^{\mathfrak{k}}$.

The proposed mechanism is an end-to-end method, it adapts graph neural network for graph construction by utilizing message passing neural network framework for enhancing general node and the edge node including isomorphic invariance. To determine kNN regulation from the attack training set Φ, it constructs GNN that conducts on graph representation $\mathcal{G} = \forall(x; \Phi)$ for input vector x provide the training Φ to identify the equivalent label vector $\mathcal{Y}$ as $\hat{\mathcal{Y}} = f(\mathcal{G}) = f(\forall(x; \Phi))$. It initially embedded each v^i into ς dimensional initial node denotes vector which utilizes embedded function ß as $\mathcal{M}^{(0),i} = ß(v^i), i = 0, \ldots, \mathfrak{k}$. For $\mathcal{G}$ graph construct, a message passing process is executed by using two major functions: message function ϖ and update function ϑ. The node representation of vectors $\mathcal{M}^{(w),i}$ are modified as:

$$\mathcal{S}^{(w),i} = \sum_{j|v^j \in \mathfrak{O}/v^i} \varpi\left(r^{i,j}\right) \mathcal{M}^{(w-1),j}, \forall i \tag{27}$$

$$\mathcal{M}^{(w),i} = \vartheta\left(\mathcal{M}^{(w-1),i}, \mathcal{S}^{(w),i}\right), \forall i \tag{28}$$

After N time message passing steps, set of node representation of vectors $\{\mathcal{M}^{(w),i}\}_{w=0}^{\mathbb{N}}$ is achieved per node. The 0-th node of set $\{\mathcal{M}^{(w),0}\}_{w=0}^{\mathbb{N}}$ is then progressed with the function of readout ξ to acquire final identification of label $\mathcal{Y}$ as follows:

$$\hat{\mathcal{Y}} = \xi\left(\left\{\mathcal{M}^{(w),0}\right\}_{w=0}^{\mathbb{N}}\right) \tag{29}$$

The fundamental functions ß, ϖ, ϑ and ξ are parameterized like neural networks, ß is the two-layer fully connected function in neural network along ς tanh units in each units. Furthermore, the specified attack training data $\Phi = \{(x_n, \mathcal{Y}_n)\}_{n=1}^{\mathfrak{q}}$, the method determine a given task kNN rule from Φ in the form of $\hat{\mathcal{Y}} = f(\forall(x; \Phi))$. The prediction method f is trained entrenched on representation of graph $\mathcal{G}$ utilizing the objective function $\cup$ as follows:

$$\begin{aligned} \cup &= \frac{1}{\mathfrak{q}} \sum_{(x_n, \mathcal{Y}_n) \in \Phi} \mathfrak{L}(\mathcal{Y}n, \hat{\mathcal{Y}}n) \\ &= \frac{1}{\mathfrak{q}} \sum_{(x_n, \mathcal{Y}_n) \in \Phi} \mathfrak{L}\left(\mathcal{Y}n, f(\forall(x; \Phi))\right) \end{aligned} \tag{30}$$

where $\mathfrak{L}$ denotes the loss function, which depends upon the target task. Moreover, the generated attack graph was stored in blockchain where the attackers cannot access are modify it, will improves the network security.

b: ATTACK PATH DETECTION AND MITIGATION

From the generated attack graph, the attack path is detected by considering attack root privilege, source attack node, target node and stage weight information. Initially, the attack node path is detection to discover the shortest attack path. The attack node path disclosure method represents security state and relationship among its states and connection matrix of all the hosts is acquired in network. The attack node path detection is defined as follows:

Assume that node $\mathfrak{Q}_i$ denotes network states where $\hat{\mathrm{S}} = \{\mathfrak{Q}_0, \mathfrak{Q}_1, \ldots, \mathfrak{Q}_n\}$ the collection of all network states is, and

then the attack graph is taken as:

$$\mathcal{G} = \frac{\left\{\hat{\mathrm{S}}, \mathfrak{E} | \mathfrak{Q}_0, \mathfrak{Q}_m, \mathcal{T}\right\}}{\mathfrak{Q}_i X \mathfrak{e}_i} \tag{31}$$

where, $\mathfrak{e}_i$ edge is utilized to represent intrusion attack mode, which alleviates the condition of $\mathfrak{E} = \{\mathfrak{e}_0, \mathfrak{e}_1, \ldots, \mathfrak{e}_n\}$, $\mathfrak{E}$ describes the all possible methods of attack in network, $\mathcal{T}$ describes the security attributes of network. Suppose that $\eth$ denotes the vulnerabilities set in network, $\hat{\mathrm{R}}$ describes attack rule, and S denotes the connection relationship. $\mathcal{T}$ is estimated by:

$$\mathcal{T} = \frac{\left\{\eth, \hat{\mathrm{R}}, \mathrm{S}\right\}}{\mathfrak{Q}_m} \otimes \mathfrak{Q}_0 \tag{32}$$

where $\mathfrak{Q}_m$ is attack state achieved through intrusion map, $\mathfrak{Q}_0$ defines initial state and the paths utilized by attack maps are moderated by $\mathfrak{Q}_0$. Furthermore, the $\wp$ denotes the authority state of real-time intruder, where $\lambda\left(\wp_j\right)$ predicted attack effect of intruder is calculated as follows:

$$\lambda\left(\wp_j\right) = \frac{\kappa\left(\mathfrak{Q}\right)}{\wp \otimes \mathrm{H}\left(\varepsilon\right)} X \mathfrak{b}\left(\mathfrak{Q}\right) \tag{33}$$

where $\mathrm{H}(\varepsilon)$ describes host attribute, $\kappa\left(\mathfrak{Q}\right)$ describes any behavior of attack in network, and $\mathfrak{b}\left(\mathfrak{Q}\right)$ describes every attacks on all possible paths of attack graph. Moreover, optimal attack path is discovered to predict the attack intention of IoT environment. Assume that $\eth$ denotes the vulnerabilities set, $\mathcal{w}_{cve}$, $\mathcal{w}_{pre}$ and $\mathcal{w}_{post}$ denotes CVE number according to vulnerability, $\mathcal{I}$ denotes the intention set that attack might reach in the network, $\check{\mathrm{I}}_{name}$ and $\check{\mathrm{I}}_{gap}$ denotes the graph name and intention action point respectively, then Υ_{tab} transfer correlated of attack behaviour among nodes are calculated as:

$$\Upsilon_{tab} = \frac{\mathcal{J} * \left\{\check{\mathrm{I}}_{gap} X \check{\mathrm{I}}_{name}\right\}}{\eth \pm \left\{\mathcal{w}_{cve}, \mathcal{w}_{pre}, \mathcal{w}_{post}\right\}} \tag{34}$$

Assume that $\mathcal{G}(\nabla, \mathfrak{E})$ describes graph of attack path which is explicated as an itemized graph, ∇ describes node set in distribution state at different level such as protection domain, vulnerability and host and $\eth_{\mathfrak{y}}$ and $\mathfrak{y}$ describes host vulnerability set, then

$$\mathcal{G}(\nabla, \mathfrak{E}) = \frac{\eth_{\mathfrak{y}}}{\nabla, \mathfrak{E}} X \frac{\left\{\mathcal{w}_j, \mathcal{w}_{j+1}\right\} X \left\{\mathfrak{y}_0 X \mathfrak{y}_1\right\}}{\Upsilon_{tab}} \tag{35}$$

where $\mathcal{w}_j$ and $\mathcal{w}_{j+1}$ describes vertex sets, $\mathfrak{y}_0$ and $\mathfrak{y}_1$ describes the edges of newly directed. Suppose that $\mathfrak{G}_{rl}$ is assigned as real-time attacker location, $\mathfrak{R}$ denotes key condition that the $\check{\mathrm{I}}$ intention of attacker can be recognized. If there is the path denoted by $\mathfrak{G}_{rl} \rightarrow \mathfrak{R}$ in $\mathcal{G}$, it can be determined that the reachable intention $\check{\mathrm{I}}$ will be reconstructed into the problem of path search among nodes. The attack intention reachability can be calculated as:

$$\grave{\mathrm{A}}_{\mathcal{G}_{(\mathcal{T},n)}} = \frac{\check{t}_f \mp n_{\mathfrak{y}j}}{\mathfrak{G}_{rl} X \check{\mathrm{I}}} X \mathcal{G} \frac{\mathrm{U}_o \mp \acute{\mathrm{P}}_{fj}}{\mathfrak{R} \mp \mathfrak{G}_{rl} \rightarrow \mathfrak{R}} \tag{36}$$

where t'_f illustrates connection matrix of $\mathcal{G}$, $n_{\mathfrak{y}j}$ denotes the number of nodes in $\mathcal{G}$, U_o denotes transition of attackers from one node to another node, and $\acute{\mathrm{P}}_{fj}$ denotes each vulnerable point attribute. Moreover, the shortest path P^* of attack intention recognition is evaluated as follows:

$$P^* = \frac{\kappa_d}{\mathbb{Z}_f} X \frac{\grave{\mathrm{A}}_{\mathcal{G}_{(\mathcal{T},n)}} X \mathcal{G}(\nabla, \mathfrak{E})}{\grave{\mathfrak{v}}_{\mathfrak{y}_u}} \tag{37}$$

where κ_d describes the vulnerable point in difficulty degree and $\mathbb{Z}_f$ describes hiding degree of vulnerable point. The attack map is explained corresponding to the correlation connection between hosts, the attack intention reachability is evaluated, the attack intention recognition in probability is obtained, the attack intention and the shortest path is achieved which anticipate the abnormal information through attack intention.

Once the intrusion is detected, then countermeasures are taken based on the risk assessment and attack path detected. If the attack risk level is low, then the alert message is generated to the specific user, and if high risk level is detected, then the administrator blocks the communication between the malicious node and the alert message is generated to that specific users and correlated nodes which are connected with malicious node and also the correlated nodes are examined to identify if any other nodes are attacked. Then the network was refreshed and reconstructed to avoid packet loss. The network reconstruction is performed by obtaining attack chains from attack graph to analysis about attack scenario. For instance, if $A - B - C - D - E$ is the generated attack path where the D is malicious node which tends to be high risk, then the D malicious node is blocked and the alert message is sent for other nodes A, B, C *and* E. Furthermore, these nodes are examined and the E node is reconnected with its one-hop relation of node C (i.e. $A-B-C-E$) which will be prohibited from packet loss thereby enhancing reliability.

V. EXPERIMENTAL RESULTS

In this section, we represent the proposed HybridChain-IDS framework in an IoT environment. This experimental research comprises of three subsections specifically simulation setup, comparison analysis and research summary. The result section illustrates that the proposed work achieve superior performance with compared to previous work.

A. SIMULATION SETUP

The simulation result of this proposed work is implemented by NS-3.26 network simulator which improves the performance of this research. The proposed framework is compared with several performance metrics and proven that our work achieves superior performance. Table 1 describes the system configuration and Table 2 describes the network parameters configuration.

B. COMPARATIVE ANALYSIS

In this section, we represented the comparison analysis between the proposed HybridChain-IDS framework and existing works where we consider two existing works such

TABLE 1. System configuration.

Hardware Configuration	Hard disk	60 GB
	RAM	2 GB
	Processor	Pentium dual core and above
Software Configuration	Operating system	Ubuntu 14.04 LTS
	Network Simulator	NS-3.26

as Lit-IDS [28] and Passban-IDS [29]. The main objective of this research is to provide security in IoT network by accomplish intrusion detection. The proposed work achieved better performance in terms of accuracy, detection rate, false alarm rate, precision, recall and F1-score concerning number of IoT users.

1) IMPACT OF ACCURACY

This metric is utilized to estimate the accuracy of proposed HybridChain-IDS framework. The highest accuracy demonstrates the system detects the intrusion accurately. Generally, the accuracy is obtained as the summation of true negative and true positive are divided by total samples. The accuracy $\left(\hat{\mathrm{A}}\right)$ is mathematically represented as follows:

$$\hat{\mathrm{A}} = \frac{\mathfrak{t} + \mathfrak{p}}{\mathfrak{t} + \mathfrak{p} + \mathrm{d}' + \mathfrak{n}} \tag{38}$$

where $\mathfrak{t}$ represents the true positive, $\mathfrak{p}$ denotes the true negative, d' shows the false positive and $\mathfrak{n}$ denotes the false negative.

Figure 4 represents the comparison of accuracy with respect to number of IoT users. The comparison result describes that the proposed work achieves high accuracy when compared to other two previous woks such as Lit-IDS and Passban-IDS. In our work, the significant features are considered for performing intrusion detection which increases the detection accuracy. In addition to that, by performing effective bi-level intrusion detection where the suspicious packets are also classified as normal or malicious by contemplating significant features using ResCapsNet. The existing works are performed intrusion detection by considering limited features which tends to high false alarm rate and the suspicious packets are not examined. The proposed HybridChain-IDS framework achieves 14% better accuracy compared to existing works.

2) IMPACT OF DETECTION RATE

This metric is utilized to estimate the rate of attack detection in IoT network. Commonly, this is represented as ratio of number of detected attacks to the number of user increasing

TABLE 2. Network parameters for HybridChain-IDS.

Network Parameters	Value
No. of IoT users	100
No. of Gateways	4
No. of Edge Server	4
No. of Cloud Server	1
No. of Trust Authority	1
Hybrid Chain	1
Simulation Area	1000m×1000m
Simulation Time	300s
Modules	Wi-Fi, Ipv4, Internet
Initial Energy	100J
Node mobility	10 m/s
Transmission range	150m
Channel bandwidth	100 MHz
Mobility type	Random waypoint
Time interval of packets	1s
Number of packets	~1000
Packet data rate	100 Mbps
No. of retransmission	7
Traffic Type	TCP/IP, UDP
Size of packets	64, 128, 256, 512, 1024 bytes

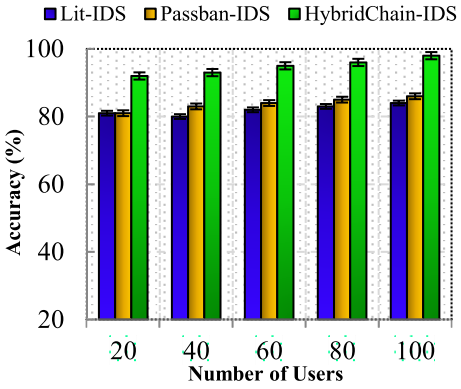

FIGURE 4. Accuracy vs. number of users.

which can be defined as follows:

$$\mathcal{D}_r = \frac{\grave{\mathrm{A}}_d}{\mathrm{O}'} \tag{39}$$

where, $\mathcal{D}_r$ describes the detection rate of attack, $\grave{\mathrm{A}}_d$ describes the attacks detected and O' represent the increasing users. The network with high detection rate can achieve a secure network. Figure 5 represent the comparison of detection rate with respect to number of IoT users of both proposed and existing works. The comparison results illustrate that the proposed work has attained better detection rate with compare to Lit-IDS and Passban-IDS existing works. In our work, we perform time-based authentication to exclude illegitimate users which reduces the malicious traffic in the network and avoids misclassification. Furthermore, bi-level

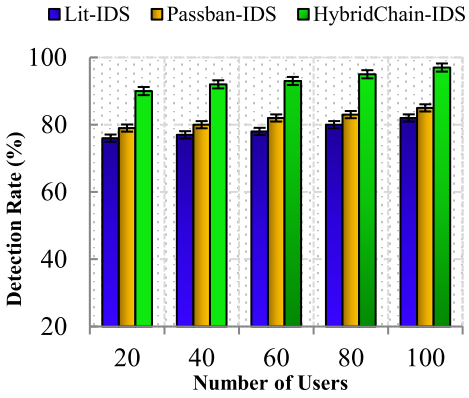

FIGURE 5. Detection rate vs. number of users.

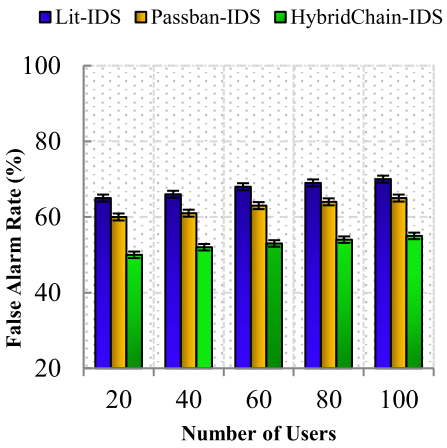

FIGURE 6. False alarm rate vs. number of users.

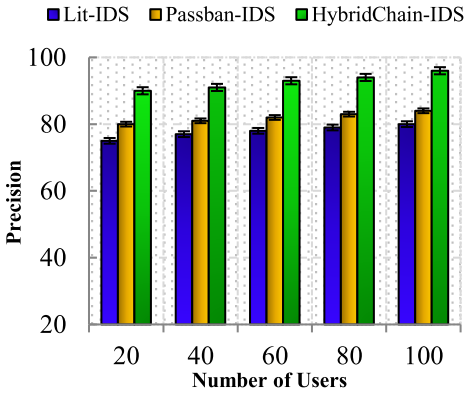

FIGURE 7. Precision vs. number of users.

intrusion detection is performed by considering effective features, where bi-level intrusion detection enhances the attack detection rate. In existing works, the authentication is not accomplished where illegitimate users can also access the network, this increases the numerous of malicious traffic in the network and leads to misclassification and high complexity that might result in ineffective detection thereby low detection rate. The proposed work reached 15% high detection rate compared to existing works.

3) IMPACT OF FALSE ALARM RATE

This metric is used to evaluate the rate of false alarm in IoT environment. Generally, the false alarm rate is defined as the ratio of false alarm to summation of true negative and the false positive. The false alarm rate $(\mathrm{d'}_r)$ can be formulated as follows:

$$\mathrm{d'}_r = \frac{\mathrm{d'}}{\mathfrak{p} + \mathrm{d'}} \tag{40}$$

where $\mathrm{d'}_r$ denotes the false alarm rate, $\mathrm{d'}$ describes false positive and $\mathfrak{p}$ denotes the true negative. A network with low false alarm rate can improve the accurate intrusion detection in the network. Figure 6 illustrates the comparison result of false alarm rate in both proposed and existing works. The comparison result shows that proposed work HybridChain-IDS achieve low false alarm rate with compare to existing works. In our research, we perform bi-level intrusion detection by employing ResCapsNet where in first level of intrusion detection is performed which classifies into three categories namely normal, malicious and suspicious. Moreover, in second-level intrusion detection the suspicious traffic flow are identified whether it is normal or malicious which reduces the false alarm rate. Furthermore, the existing works classified the network traffic as normal or malicious where the malicious traffic are taken as either normal or malicious which affects the network security and increase the high false alarm rate. The proposed work achieves 15% low false alarm rate when compared to existing works.

4) IMPACT OF PRECISION

This metric is used to calculate the value of positive predictive based on specificity; furthermore it also defines the detection performance. The precision is measured as ratio of true positive to the summation of false positive and true positive. The mathematical representation of precision is illustrated as follows:

$$\mathfrak{p} = \frac{\mathfrak{t}}{\mathfrak{t} + \mathrm{d'}} \tag{41}$$

where $(\mathfrak{p})$ represent the precision. Figure 7 describes the comparison result of proposed HybridChain framework and existing works with respect to number of IoT users. The comparison result shows that the proposed work achieves better performance compared to existing works. In our work, we perform time-based authentication to authenticate the legitimate users and authorized user are only entrance to the network this will exclude the illegitimate users and reduce the innumerable malicious traffic. The ResCapsNet is proposed to perform bi-level intrusion detection where this algorithm extracts the significant features and spatial information is fetched effectively which helps to perform accurate detection. The existing works are limited with network

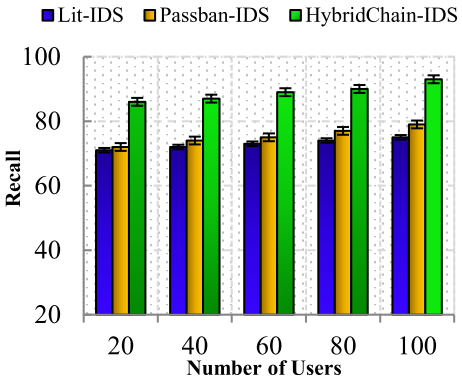

FIGURE 8. Recall vs. number of users.

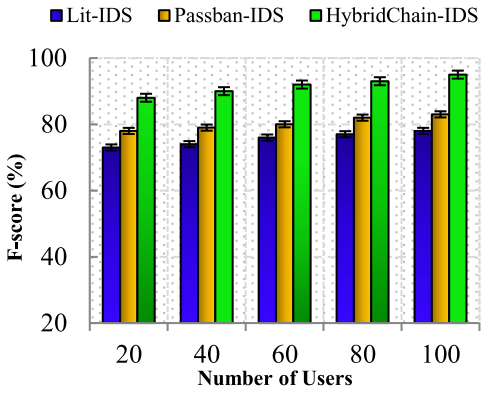

FIGURE 9. F-score vs. number of users.

security level and true positive rate. The unauthorized users are allowed in the network where the intruder can act as legitimate users and easily compromise the legitimate users which increases the malicious packets in the network and leads to less positive predictive rate. The proposed work attains 16% better positive predictive value compared to existing works.

5) IMPACT OF RECALL

This metric is measured the value of negative predictive based on sensitivity of the intrusion detection in IoT network. The recall is evaluated the proportion of true positive to the additive of false negative and true positive which is represented as follows:

$$\text{Recall} = \frac{t}{t + n} \tag{42}$$

where (Recall) represent the recall. Figure 8 represents the proposed and existing recall value with respect to number of IoT users. In our research, we perform user scheduling and access control to increase network security.

The user scheduling is executed to reduce the complexity and the users are scheduled on priority based which improves the access control service. The trust-aware access control support increasing the data integrity level and reducing negative prediction through permission-based access control. Moreover, the suspicious packets are examined again to reduce the negative prediction. The existing works are neither provides access control in considerable manner where the users can approach the data without any restrictions and condition that affect the data privacy. The intrusion detection is performed with insufficient features which tend to high negative predictive value. Moreover, the suspicious packets are not examined which increases the high negative predictive value. The proposed work accomplished 18% better negative predictive compared to existing works.

6) IMPACT OF F-SCORE

The F-Score is described as the harmonic mean of precision and recall. Generally, the F-score is evaluated as proportion of recall product to the precision and recall summation. The F-score is formulated as follows:

$$F = 2 \times \frac{p + \text{Recall}}{p \times \text{Recall}} \tag{43}$$

where (F) denotes the F-score. Figure 9 illustrates the comparison result of both proposed and existing works F-score values with respect to number of IoT users. In, the proposed HybridChain-IDS, the F-score are enhanced with the number of IoT users. Furthermore, the increasing of F-score also denotes the accuracy of this work that is if the F-score is obtained high then the accuracy of intrusion detection is increased. The HybridChain-IDS performs time-based authentication to enhance network security. The user scheduling and access control is provided to authorize users based on their trust and permission level which improves data privacy thereby reducing in complexity. The Hybrid blockchain enhances the security in IoT environment. Moreover, bi-level intrusion detection is executed by considering effective features to amplify the detection accuracy. The proposed work achieves 17% high F-score compared to existing works.

C. RESEARCH SUMMARY

The research summary divides into two subdivisions, where the security analysis and research highlights are illustrated.

1) SECURITY ANALYSIS

In this section, we define security analysis of the proposed HybridChain-IDS framework. We examine the security dispensed by HybridChain-IDS framework in IoT network. We enumerate our research with 5% of malicious node in the network. The comparison results prove that our work achieves high security. In our work, we concentrate on identifying the brute force attack, SYN flood attack and phishing attack which are explained below,

a: BRUTE FORCE ATTACK

Our proposed framework mitigates brute force attacks. The characteristic of brute force attacks are falls into weak password-related targets where the attackers seek to guess the password through numerous attempts. In our work, time-based authentication is executed to avoid the brute-force attacks. Once, the users is registered using their credentials

TABLE 3. Numerical analysis of proposed and existing work.

	Lit-IDS	Passban-IDS	Hybrid Chain-IDS
Accuracy (%)	82	83.8	94.8
Detection rate (%)	78.6	81.8	93.4
False Alarm rate (%)	67.6	62.6	52.8
Precision	77.8	82	92.8
Recall	73	75.4	89
F-score	75.6	80.4	91.6

then the TA displayed the user registered time and provides security key based on the password and registered time which is stored in hash format using NIK-512 in blockchain.

b: SYN FLOOD ATTACK

Our proposed work performs against the SYN flood attacks which are type of DDoS attack focused to make server unavailable by sending numerous of packets. In our work, the authenticated users are scheduled based on priority and then the access control is provided based on their trust and permission level. In case, if the user transmits numerous of packets then that user's state will turn off.

c: PHISHING ATTACK

This is a type of attack where the attacker transmits fraudulent message outlines to trick the legitimate user for revealing personal information. In our work, we execute bi-level intrusion detection where suspicious packets are examined again. Furthermore, the blockchain ensures every transaction which enhances network security.

2) RESEARCH HIGHLIGHTS

In this section, we elucidate the experimental results in summary which also proven that the proposed HybridChain-IDS framework achieves superior performance through comparison results. The performance of proposed work is enumerated in terms of accuracy, detection rate, false alarm rate, precision, recall and F-score which are described in Figure 4 to Figure 9. Table 3 demonstrates the performance metrics in numerical analysis of proposed and existing works. The highlights of this research are described as follows,

- For enhancing the security in IoT, time-based authentication is performed to authenticate legitimate users using Nik-512 hashing algorithm.
- For increasing the detection accuracy, the significant features are extracted and bi-level intrusion detection is implemented by utilizing ResCapNet algorithm which improves the accuracy.
- For timely detection and mitigation, the attack graph was generated using improved KNN algorithm for timely attack detection in the future and attack path was evaluated to alert generation which reduce the attack severity level then the attack graph was stored in the blockchain.

VI. CONCLUSION

IoT environment, lack of security and privacy are the major issues. In this research, the HybridChain-IDS framework is proposed to execute effective bi-level intrusion detection. Initially, time-based authentication is achieved to authenticate legitimate users by providing 512-bit security key using NIK-512 hashing algorithm. The password and registered time are stored in Hybrid chain (Blockchain and TEE) and this blockchain improves the scalability of blockchain and data privacy. Then the authenticated users are scheduled to reduce the complexity and then access control is provided entrenched on user permission level and trust level. After that, we perform effective bi-level intrusion detection employing ResCapsNet which enhances network security. Furthermore, the risk assessment is executed to enumerate the attack impact level and attack graph is generated for attack path identification. Then the risk-based countermeasures are taken and the attack graph is stored in blockchain. Finally, the network is refreshed and reconstructed to hinder packet loss. The proposed HybridChain-IDS is achieved better performance in terms of accuracy, detection rate, false alarm rate, precision, recall and F-score.

REFERENCES

[1] A. N. Jahromi, H. Karimipour, A. Dehghantanha, and K.-K. R. Choo, "Toward detection and attribution of cyber-attacks in IoT-enabled cyber–physical systems," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13712–13722, Sep. 2021.

[2] S. Aheleroff, X. Xu, Y. Lu, M. Aristizabal, J. Pablo Velásquez, B. Joa, and Y. Valencia, "IoT-enabled smart appliances under industry 4.0: A case study," *Adv. Eng. Informat.*, vol. 43, Jan. 2020, Art. no. 101043.

[3] Y. B. Zikria, R. Ali, M. K. Afzal, and S. W. Kim, "Next-generation Internet of Things (IoT): Opportunities, challenges, and solutions," *Sensors*, vol. 21, no. 4, p. 1174, Feb. 2021.

[4] Q. V. Khanh, N. V. Hoai, L. D. Manh, A. N. Le, and G. Jeon, "Wireless communication technologies for IoT in 5G: Vision, applications, and challenges," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–12, Feb. 2022.

[5] T. Mohammed, A. Albeshri, I. Katib, and R. Mehmood, "UbiPriSEQ—Deep reinforcement learning to manage privacy, security, energy, and QoS in 5G IoT HetNets," *Appl. Sci.*, vol. 10, no. 20, p. 7120, 2022.

[6] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for IoT security," *Digit. Commun. Netw.*, vol. 6, no. 2, pp. 195–202, 2020.

[7] D. J. Atul, R. Kamalraj, G. Ramesh, K. Sakthidasan Sankaran, S. Sharma, and S. Khasim, "A machine learning based IoT for providing an intrusion detection system for security," *Microprocessors Microsyst.*, vol. 82, Apr. 2021, Art. no. 103741.

[8] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using LSTM based autoencoder," in *Proc. 16th ACM Symp. QoS Secur. Wireless Mobile Netw.*, Alicante, Spain, Nov. 2020, pp. 37–45.

[9] P. Maniriho, E. Niyigaba, Z. Bizimana, V. Twiringiyimana, L. J. Mahoro, and T. Ahmad, "Anomaly-based intrusion detection approach for IoT networks using machine learning," in *Proc. Int. Conf. Comput. Eng., Netw., Intell. Multimedia (CENIM)*, Surabaya, Indonesia, Nov. 2020, pp. 303–308.

[10] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT," *Proc. Comput. Sci.*, vol. 167, pp. 1561–1573, Jan. 2020.

[11] Z. A. El Houda, B. Brik, and L. Khoukhi, "'Why should i trust your IDS?': An explainable deep learning framework for intrusion detection systems in Internet of Things networks," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1164–1176, 2022.

[12] M. A. Alsoufi, S. Razak, M. M. Siraj, I. Nafea, F. A. Ghaleb, F. Saeed, and M. Nasser, "Anomaly-based intrusion detection systems in IoT using deep learning: A systematic literature review," *Appl. Sci.*, vol. 11, no. 18, p. 8383, Sep. 2021.
[13] M. H. Faruk et al., "Malware detection and prevention using artificial intelligence techniques," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 5369–5377.
[14] K. S. Kiran, R. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi, "Building a intrusion detection system for IoT environment using machine learning techniques," *Proc. Comput. Sci.*, vol. 171, pp. 2372–2379, Jan. 2020.
[15] T. M. Hewa, A. Kalla, A. Nag, M. E. Ylianttila, and M. Liyanage, "Blockchain for 5G and IoT: Opportunities and challenges," in *Proc. IEEE 8th Int. Conf. Commun. Netw. (ComNet)*, Hammamet, Tunisia, Oct. 2020, pp. 1–8.
[16] Y. Sun, J. Yu, J. Tian, Z. Chen, W. Wang, and S. Zhang, "IoT-IE: An information-entropy-based approach to traffic anomaly detection in Internet of Things," *Secur. Commun. Netw.*, vol. 2021, pp. 1–13, Dec. 2021.
[17] I. Ullah and Q. H. Mahmoud, "A two-level flow-based anomalous activity detection system for IoT networks," *Electronics*, vol. 9, no. 3, p. 530, Mar. 2020.
[18] R. W. Anwar, K. N. Qureshi, W. Nagmeldin, A. Abdelmaboud, K. Z. Ghafoor, I. T. Javed, and N. Crespi, "Data analytics, self-organization, and security provisioning for smart monitoring systems," *Sensors*, vol. 22, no. 19, p. 7201, Sep. 2022.
[19] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, and M. Qiu, "Adversarial attacks against network intrusion detection in IoT systems," *IEEE Int. Things J.*, vol. 8, no. 13, pp. 10327–10335, Jul. 2021.
[20] S. Fenanir, F. Semchedine, S. Harous, and A. Baadache, "A semi-supervised deep auto-encoder based intrusion detection for IoT," *Ingénierie des Systèmes d Inf.*, vol. 25, no. 5, pp. 569–577, Nov. 2020.
[21] L. Nie, Y. Wu, X. Wang, L. Guo, G. Wang, X. Gao, and S. Li, "Intrusion detection for secure social Internet of Things based on collaborative edge computing: A generative adversarial network-based approach," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 134–145, Feb. 2022.
[22] T. T. Huong, T. P. Bac, D. M. Long, B. D. Thang, N. T. Binh, T. D. Luong, and T. K. Phuc, "LocKedge: Low-complexity cyberattack detection in IoT edge computing," *IEEE Access*, vol. 9, pp. 29696–29710, 2021.
[23] X. Kan, Y. Fan, Z. Fang, L. Cao, N. N. Xiong, D. Yang, and X. Li, "A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network," *Inf. Sci.*, vol. 568, pp. 147–162, Aug. 2021.
[24] M. H. Shahriar, N. I. Haque, M. A. Rahman, and M. Alonso, "G-IDS: Generative adversarial networks assisted intrusion detection system," in *Proc. IEEE 44th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jul. 2020, pp. 376–385.
[25] Z. Ma, L. Liu, and W. Meng, "Towards multiple-mix-attack detection via consensus-based trust management in IoT networks," *Comput. Secur.*, vol. 96, Sep. 2020, Art. no. 101898.
[26] P. Kumar, R. Kumar, G. Srivastava, G. P. Gupta, R. Tripathi, T. R. Gadekallu, and N. N. Xiong, "PPSF: A privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2326–2341, Jul. 2021.
[27] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, "A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network," *J. Parallel Distrib. Comput.*, vol. 164, pp. 55–68, Jun. 2022.
[28] X.-H. Nguyen, X.-D. Nguyen, H.-H. Huynh, and K.-H. Le, "Realguard: A lightweight network intrusion detection system for IoT gateways," *Sensors*, vol. 22, no. 2, p. 432, Jan. 2022.
[29] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6882–6897, Aug. 2020.
[30] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9463–9472, Jun. 2021.
[31] H. M. Alshahrani, "CoLL-IoT: A collaborative intruder detection system for Internet of Things devices," *Electronics*, vol. 10, no. 7, p. 848, Apr. 2021.

AHMED A. M. SHARADQH received the Ph.D. degree in computer science and computing systems and networks from the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Ukraine, in 2007. Since 2009, he has been an Associate Professor with the Computer Engineering Department, Faculty of Engineering Technology, Al-Balqa Applied University. His research interests include the performance of networks, quality services, security networks, the IoT, image processing, digital systems design, operating systems, and microprocessors.

HAZEM (MOH'D SAID) ABDEL MAJID HATAMLEH was born in Irbid, Jordan, in 1973. He received the M.Sc. and Ph.D. degrees from the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," in 2007. He is currently an Associate Professor with the Applied Science Department, Ajloun University College, Al-Balqa Applied University. His current research interests include computer networks, wireless networks, the IoT, image processing, and computer graphics.

AS'AD MAHMOUD AS'AD ALNASER received the Ph.D. degree in computer engineering from the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute." He is currently an Associate Professor with the Department of Applied Science, Ajloun University College, Al-Balqa Applied University. His research interests include wireless and mobile networks, internet protocols, image processing, and graph theory and its applications.

SAID S. SALOUM was born in Irbid, Jordan. He received the Higher Diploma degree in radio-physics and electronics from Kaluga State University, Russia, in 1995, and the Ph.D. degree in computer engineering from Izhevsk State Technical University, Russia, in 2004. He is currently an Assistant Professor with the Computer Engineering and Networks Department, Jouf University, Saudi Arabia. His research interests include image processing, machine learning, and deep learning.

TAREQ A. ALAWNEH was born in Irbid, Jordan, in 1984. He received the B.S. and M.S. degrees in computer engineering from the Jordan University of Science and Technology (JUST), Irbid, in 2006 and 2009, respectively, and the Ph.D. degree in computer engineering from the University of Hertfordshire, U.K., in 2021.

From 2010 to 2013, he was a full-time Lecturer with the Electrical and Computer Engineering Department, Tafila Technical University (TTU), Al-Tafila, Jordan. He was an Assistant Professor with Fahad Bin Sultan University (FBSU), Saudi Arabia, in 2021. He is currently an Assistant Professor with the Electrical Department, Al-Balqa Applied University. His research interests include cache partitioning algorithms, low-power designs, cache coherence protocols, high-performance dynamic random access memory (DRAM) for multimedia applications, multi-core systems, tiled-chip multiprocessors (tiled-CMPs) systems, and the IoT.