DOI: 10.15849/IJASCA.220720.02

Int. J. Advance Soft Compu. Appl, Vol. 14, No. 2, July 2022
Print ISSN: 2710-1274, Online ISSN: 2074-8523

PDIS: A Service Layer for Privacy and Detecting Intrusions in Cloud Computing

Rusul Mumtaz[1], Venus Samawi[2]*, Aysh Alhroob[1], Wael Alzyadat[3], Ikhlas Almukahel[1]

[1,2] Faculty of Information technology, Isra University, Jordan
e-mail: Engroro93@yahoo.com, e-mail: Vensu.Samawi@iu.edu.jo,
e-mail: Ikhlas.almukahel@iu.edu.jo, e-mail: Aysh@iu.edu.jo
[3]Faculty of Science and IT, Al-Zaytoonah University of Jordan, Jordan
e-mail: Wael.alzyadat@zuj.edu.jo

Abstract

Cloud computing faces numerous challenges in many areas including security and privacy issues. In this work, a developed approach is suggested to tackle three security and privacy issues: network intrusion detection (NID), privacy, and internal attacks. A decision tree (J48) has been used to generate a set of rules based on the CICIDS2017 dataset to solve the NID problem. The accuracy of the generated rules approaches 99.8%. A set of policies are attached to the data file on the bases of a sticky policy to preserve privacy. A new approach is suggested based on blockchain to detect internal attacks in real-time, in which a set of trustees-chain are identified by the data owner. Any data modification conducted by a trusted member will be reported to all members of the trust group including the owner. The developed approach suggests adding a Privacy and Detecting Intrusions Service (PDIS) layer as part of the cloud computing main service model. PDIS includes the three suggested approaches above (NID, sticky policy, and trustees-chain). Finally, a web-based application is implemented to act as casework to validate PDIS and evaluate its reliability.

Keywords: *Cloud computing, Privacy, Machine learning, Internal Intrusion Detection, Network Intrusion Detection, Real Time Auditing*

Received 11 February 2022; Accepted 4 May 2022

1 Introduction

The scalability of information, documents, multimedia resources, and web and mobile applications motivated people, universities, and organizations to use clouds. Cloud and fog computing could be considered a shared pool of resources that can easily be accessed by users from any computer or mobile device. Therefore, cloud computing is the key point in the ubiquitous computing vision [1]. The high demand for cloud utilization, specifically in educational and business fields, results in setting up many cloud data centers to provide resource sharing based on a "pay-per-use" model [2-4]. Cloud computing provides considerable advantages from shared resources and software applications to provide a substantial amount of storage space [2, 5, 6]. Considering cloud architecture, modern cloud data centers construct the cloud computing services (CCS), which is a set of layers placed on top of the hardware. Each of these layers acts as one service model. SaaS, PaaS, and IaaS are the main services in cloud computing [5, 6]. Cloud data storage enables users to store their data online instead of using their own local storage devices (local hard drives, flash drives, etc.). Some advantages of using clouds, among others, are the ability to access data using ubiquity computing and sharing resources. Recently, individuals and firms greatly use clouds due to the substantial storage and high-performance computing capabilities offered by cloud-centers. However, storing files (private data, business information, applications, etc.) in clouds is highly vulnerable to various attacks and privacy violations. Moreover, almost all ubiquity-computing systems suffer from security and privacy weaknesses, which are technically difficult to maintain. Consequently, security and privacy are a challenge in general, specifically in cloud and fog computing. Although cloud centers provide outsourcing and multi-tenancy, sharing platforms and resources may cause considerable problems concerning client privacy violations, security, and intrusions caused by either intruders or legal clients. The major concern in cloud computing to date is how to protect user privacy, substantial data, and software from unauthorized access and how to inhibit intrusion attacks. Therefore, new strategies are needed to handle privacy and security problems to protect client data and computation [4, 7]. Cloud attacks could be classified based on the following: threat-type (internal threat by cloud client or external attack threat), attack-type (violation of client privacy, enormous data, and intense computation, data integrity, and confidentiality), and the victim service-layer (IaaS, PaaS, or SaaS) [8]. Different approaches are used but are not efficient to manage all attacks. The other point is that no general solution could be used to protect all cloud service layers from various attacks [9]. Furthermore, cloud centers must provide a privacy technique to guarantee a certain

level of privacy for the client's data. Therefore, a new method is suggested to provide a certain privacy level for clients utilizing an intelligent technique, which can be achieved by adding an intelligent privacy and detecting intrusions service (PDIS) layer. The PDIS layer acts as one of the cloud computing layers, placed on top of the other service layers, to ensure a highly reliable privacy control and protect user data and information when using cloud computing services.

The contribution of this study is to suggest an intelligent approach that detects internal attacks in real-time, detects network intrusions and enhances privacy, in the cloud computing SaaS layer. The suggested intelligent approach utilizes the data mining method and sticky policy and blockchain to develop intelligent privacy, and confidentiality service layer (PDIS), which can detect internal intrusions (made by trust members) immediately and detect and prevent external intrusions (NID). The PDIS layer is laid between the client and CCS layers. The proposed PDIS layer consists mainly of two phases:

Intrusion detection (ID) phase: a set of rules is generated by utilizing the data mining approach (Decision tree J48). These rules are generated based on a set of attributes (audit trail, type of access, type of service, client type, IP address, etc.) provided by the datasets (CICIDS2017). The decision-tree algorithm (J48) is applied to the dataset, and the results are analyzed using the R statistical computing environment to generate the most proper set of rules to detect and prevent intrusions. The generated rules will be a part of the decision-making phase. The DAMP duty is to either reject or accept user requests based on the results of the ID rules.

Privacy and protection phase: sticky policies (to provide access control restrictions) and trustees-chain technique based on blockchain philosophy (a new approach suggested for tracking data access and immediate detection of an internal attack, which could be made by trust members) have been used to construct a layer that guarantees a certain level of protection and privacy to clients without the need for the third party. Sticky policies are carried out based on client information and access control rules that are specified by the data owner. On the other hand, the new technique based on blockchain technology is applied to inform all trustworthy parties (chain members specified by the data owner) about any updating and/or accessing attempt made by the trusted parties (i.e., any accessing or updating attempts are reported to all chain members to notify them with the latest access and updates made on a data file. This process will preserve real-time auditing of data changing (by who, when, in addition to last data updates), which enables the trust group to detect any suspicious data access.

Although several studies utilized a decision tree (J48), to our knowledge (based on a literature review), none of them analyzes the decision tree results and generates ID rules. In this work, the R statistical computing environment is used to generate ID rules

based on analyzing the output of the decision tree. These rules could be used to detect and prevent intrusion. This work leverages the blockchain philosophy to preserve data owner privacy and detect internal attacks. To our knowledge and based on literature reviews, no attempts have been made to use the blockchain philosophy to detect real-time insider attacks.

The rest of this paper is organized as follows. Section 2 illustrates the theoretical background concerning the cloud environment, deployment model, and security threats in the cloud. Several approaches concerning privacy, ID, and data security are also explained. Section 3 explains the PDIS layer, along with the structure of the suggested system. Section 4 presents the implementation details of the system, provides additional information about the used techniques and applying them to the case study, and illustrates which system is put into use. Moreover, this section tests and evaluates the functionality and efficiency, showing the behavior of the system under multiple circumstances. Then, the section runs scenarios showing its strong points in preserving privacy, detecting intrusions, and marinating the authenticity of the data within the cloud computing environment. Section 5 concludes the study.

2 Cloud Security and Privacy Issues

In the last decade, most applications leverage resources and services provided by cloud infrastructure. Therefore‹ cloud computing becomes a significant paradigm that is utilized by vast consumers to make use of upon-request storage and high-performance computing with reduced cost. Nevertheless, utilizing cloud infrastructure may cause serious security and privacy problems, exposing user data and computations to various breaches and attacks [6, 10, 11]. As mentioned in Section 1, this work focuses on cloud security and privacy. This section reviews cloud computing security and privacy, which has significant implications for our proposed framework. Cloud deployment models and Cloud computing threats (security, privacy, and data integrity) will be illustrated. Intrusion detection will also be explained. The existing intrusion detection systems and methodologies relating to sticky policy and blockchain that are found in related work are discussed. Kinds of literature relating to privacy in the cloud are reviewed and illustrated. Finally, the main contribution of this work, which distinguishes it from other related works, is clarified.

2.1 Cloud Computing: Deployment Models

Thousands of services are available in the cloud environment to preserve the rapid utilization of cloud services. The server that is close to the client area is more likely to be utilized. Deployment models are classified based on the location (concerning client area), and the service provider and manager. Majorly, cloud deployment models are classified into: Private, public, community, and hybrid cloud infrastructure [10, 12, 13].

2.2 Cloud Environment: Security and Privacy Threats

Using the cloud environment has several benefits, ranging from resource sharing to utilizing high-performance computing and mass storage. Despite these benefits, users of the cloud environment experience serious security and privacy threats. These threats affect the various technologies and systems applied in cloud environments including network connections, information and databases, virtualization, managing transaction, resource sharing, and load balancing, and long and short storage management. Therefore, security and privacy should be maintained in cloud computing. Cloud security threats may result from internal attackers (service provider's employees, clients with authority to access cloud services, or any third party), or external attackers (unauthorized users). Based on the attacker skill, cloud environment attackers are classified to [13]:

- An attacker who utilizes uncomplicated techniques and tools (Random attackers).
- Attackers with limited skills attacking particular cloud providers or servers using advanced attacking methods, by publicly accessible tools (Weak attackers).
- Organized and financed group of skilled attackers (Strong attackers).
- Highly skilled professional attackers (Substantial attackers). They are not easily being discovered even by professionals in eCrime.

Security and privacy in the cloud could be classified into four main categories [6,13,14]:

1) *Intrusions and Malware Attacks*: Harmful Software (*Malware*), which is used to access and collect sensitive data, and breach computer processes. Intrusion detection systems, firewalls, and Anti-Malwares are needed to suspend the harm of Malware and enhance cloud client confidence in the Physical-Machines (PMs) that are used to store their data in the datacenter
2) *Authorization and Data Integrity:* The major problems in cloud computing are how to preserve that data is only accessed and modified by authorized users and how to maintain and assure the accuracy and consistency of data through its life cycle. The integrity of data within SaaS, which is considered a complex hosting in a cloud environment, is a serious problem that needs to be resolved. Therefore, obligates users to access control rules, and developing client notification systems is mandatory to preserve data integrity.
3) *Availability:* how to preserve data availability despite system failures and DoS (denial of service) attacks. Inhibiting DoS and ensuring executing applications correctly (i.e., ensuring the validity and integrity of computations) is essential to clients.
4) *Security of Data in the Cloud:* to resolve this problem, ciphering algorithms are used to encrypt data, and certain policies are enforced to control data sharing. In

this work, we are interested in solving security issues concerning *Intrusions and Malware Attacks, privacy utilizing sticky policy and blockchain*.

2.2.1 Intrusion Detection (ID)

ID refers to the detection of suspicious activity through monitoring and analyzing events occurring on computers or networks. ID has become the main technology used to monitor traffic and detect intrusions across the network [15, 16]. The Intrusion Detection System (IDS) is used as a defense layer to protect networks from ever-growing attacks before causing considerable harm [17, 18]. The IDS utilizes certain methods to detect intrusive behaviors in a computer system, then issues alerts and reports these behaviors, or even blocks them [19]. IDSs are mainly divided, according to their scope, into two main categories: namely, (NIDS) and host-based intrusion detection systems (HIDSs). The NIDS is placed at a strategic point(s) within the network to keep the traffic among all connected devices on the network analyzed and monitored to detect any illegal/abnormal activity, whereas the HIDSs work on individual hosts or individual devices on the network. A HIDS monitors the packets (inbound and outbound) from a certain device only and notifies the client or administrator when abnormal or illegal activity is detected [15, 20]. Many ID techniques are utilized to tackle intrusion attacks in a cloud environment (such as Virtual Machine Introspection (VMI) and Hypervisor Introspection (HVI) based approaches) [21]. To develop IDSs, a benchmark dataset is needed to generate rules to protect networks from malware and intrusive attacks. Many datasets are utilized by researchers to assess the accuracy of their suggested ID system (IDS). In [18] the set of datasets (DARPA98, KDD99, ISC2012, and ADFA13) are explored. It was found that most of these datasets contain old intrusion signatures. Therefore, a new dataset (CICIDS2017) is built, which includes benign (normal flows) network flows in addition to eleven network attacks (anomaly flows), which reflect real network flows (normal and anomaly). The new dataset is utilized to specify the best features set that detect network attacks are specified using the Random-Forest Repressor algorithm. Seven classification algorithms (K-nearest neighbor, Random Forest, decision tree, Adaptive Boosting, Naive-Bayes, Multi-layer perceptron, and Qualitative Data Analysis) are used to assess the performance of the new dataset. Several studies focused on addressing issues of ID within cloud computing environments, which can be classified into intrusions against the cloud itself, and intrusions that target individual machines inside the cloud. Deep learning approaches such as Recurrent Neural Networks are used to detect malware [22]. IDS for a cloud environment was revised to point out the drawbacks and features of each IDS system based on cloud characteristics [17]. In [23], a multilayer perceptron along with feature selection is used to develop an anomaly-based intrusion detection system. The developed IDS is based on an enhanced correlation-based feature selection (CFS), which identifies anomalies at early stages, and improves detection accuracy. Attacks on a networked computer system may be classified into two main categories: goal-based attacks and used protocol-based attacks.

2.2.2 Sticky Policy and Blockchain

Clients have no control over their data when storing it in a remote server. The full control is assigned to a service provider or a third party. Therefore, clients need to be able to control their data when it is stored in a cloud. Sticky policy is a technique used to enhance privacy and maintain data protection of the client's files (computation-jobs and data) in a cloud environment. To preserve privacy in a cloud environment, a certain policy (specified by the owner) is attached to the significant client's file (users need to preserve their profiles and other confidential information from internal or external intruders). Sticky policies indicate the terms and limitations attached to a client's files that control the user's access to that file (i.e. file owners could specify who can access his file, and how to be handled through clarifying file-access-policy) [11, 24]. [25] protects individual components of data entity through embedding sticky policies into OOXML (open office XML) without access control to the content. The sticky policies attached to the data are secured by using the modern public-key Identity-Based Encryption (IBE) scheme. Authors claim that a sticky policy can be used to facilitate sensitive data protection utilizing standard OOXML and XACML built-in features. To enhance the privacy degree provided by the sticky policy technique, client's files could also be ciphered to be used by limited the domains or by a specific group of users, such as utilizing client's data in research fields, or in a set of platforms to preserve the certain extent of security features [26]. Using a sticky policy by clients means the cloud provider allows clients to manage their files. However, some cloud providers prohibit clients from managing their files. In this case, the third party (trusted one) could manage the client's files on his behalf and allow clients to monitor [27]. A Blacklisted users authorized and trusted list of users, intrusion detection notifications could be utilized to enhance privacy [14, 28].

Blockchain refers to the technique that allows members to keep a "ledger" that contains all transaction data also allows them to update their ledgers to maintain their integrity in the case of a new transaction. Since advancements in internet encryption technology have enabled members to verify the reliability of a transaction, the only failing point that has arisen in the transaction that comes from the reliance on a third authorized party had been resolved [29]. Recently, Blockchain might be the solution for maintaining data integrity as it inherently resists modifying data. Blockchain is not only a data structure, but it is used as a history for data modification. Therefore, it could be used to improve data integrity and audit the data accessing processes by users to help in detecting abnormal behaviors chain [29, 30]. In]31[, suggests using two different methods to share data in medical records based on their sensitivity. A consortium blockchain is used to share privacy-sensitive parts, and a public blockchain to share the non-sensitive parts. Blockchain is a promising approach, which is recently been utilized by researchers to preserve privacy and data integrity. The threats handled by blockchain resulted from malicious data modification and updating without informing all trusted group members.

3 PDIS Layer: The Proposed Model

Nowadays, the security of exchanging and saving data over a CCS as well as to maintaining privacy and data integrity are significant problems for both users and cloud service providers [11]. To enhance the trustworthiness of the cloud clients concerning the service providers and preserve resource confidentiality, an intelligent layer needs to be developed by the respective platforms. In this study, the PDIS layer is proposed to address security, privacy, and internal attack detection problems in CCS. PDIS layer mainly consists of two phases: intrusion detection module (used to detect and prevent intrusions), and privacy and protection module (employing sticky policy and access control to inset the terms of use and access policies of the data and maintain detection of internal attacks utilizing Blockchain). The suggested approach is applied to a simulated system to secure teacher-student-learning web-based applications. Therefore, the suggested approach fits private cloud infrastructure as a deployment model, which could be managed by the organizations. The technique used in this study relies on granting reading and posting privileges of encrypted data (using AES cryptosystem) by instructors at the university, and authorized students (or other users) after fulfilling the policy's rules and conditions specified by the data owner (rules of access control). The suggested PDIS layer is meant to be placed on top of the main services, it is integrated with the SaaS layer (see **Error! Reference source not found.**1).

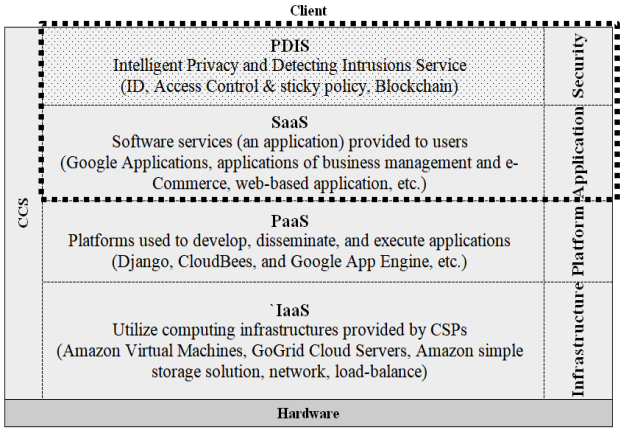


Fig. 1**:** The Proposed Model of cloud computing main service

The designed service layer is mainly consisting of two modules, namely: Intrusion detection module, and Privacy and protection module.

A) Intrusion Detection Module:

The IDS is known as a rule-based pattern matching system, which could be constructed and verified through the system usage. Any significant deviation from the normal usage is flagged as abnormal. The IDS mainly aims to detect any unusual activity, which breaches the authentication and security policy of computer networks. The process of detection is performed by the system administrator manually by analyzing the user's log information. A NIDS is used to monitor the environment's behavior of the network (message flow, access requests, etc.) to protect the network's data from attacks and threats, mainly unauthorized access, and DoS. **Error! Reference source not found.** 2 shows the main module of the developed NIDS and the steps followed to report an intrusion.

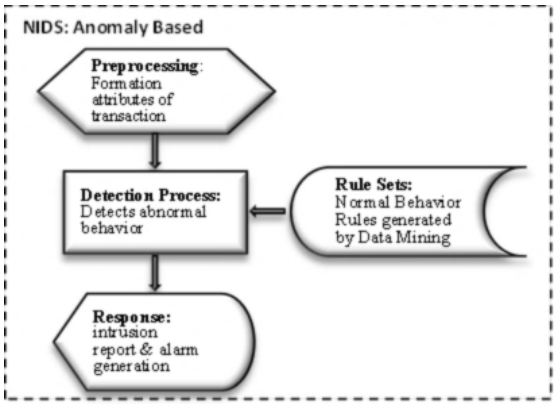


Fig. 2:The developed NIDS module within PDIS layer

In this work, anomaly-based IDS is developed, where rule sets of normal behavior are generated based on data mining approaches. Rule sets are generated through two stages: the *features-selection stage* and the *rules-generation stage*. In the feature selection stage, the decision tree (J48 algorithm) classifier is used to select and approve the best feature set. In the rules-generation stage, the ARules-Package in R-studio is used to generate a set of rules (NID rules) using the set of features resulting from the feature selection stage. The dataset CICIDS2017 is used to build the NID rule sets. CICIDS2017 is developed by the Canadian Institute of Cyber security. The dataset contains features along with the most recent attacks (threats) that are not addressed by previous ID datasets (e.g., the KDD99 dataset). Although CICIDS2017 contains 85 features, not all features are significant (some may cause degrading of detection accuracy). Based on [18], a certain 25 features are enough to detect intrusions. Therefore, we decide to test the ability of the recommended 25 features in constructing rules to detect intrusions compared with the 85 features of the dataset. Decision tree-J48 is used to test the ability of the dataset in detecting network intrusions. With the 25 features, the accuracy reaches ≈99.8% even with 10-fold cross-validation. Therefore, we decide to construct the NID rules (based on the 25 selected features) using ARules Package in R studio. Numerous rules are generated to cover all attacks and normal cases with various confidence. Therefore, we decide to count on normal rules with confidence = 1 to allow or ban user access, as illustrated in the detection process.

Detection Process

If the user-access- features match any of the normal rules and confidence=1, *then*
Allow access, and Call authentication step
Else Report an intrusion to the data owner & Ban user access

The NIDS collects data about the network's packet's traffic and registered abnormal behaviors (utilizing the normal rules with confidence=1) to discover possible threats. Based on the detected parameters, the developed IDS takes proper action when a threat is detected, mainly by reporting this intrusion threat to the data owner of the network, and/or blocking the user that initiated that suspicious data access behavior.

B) Privacy and Data Protection Module

This module is used to preserve privacy and data protection in the cloud, which mainly consists of three stages (see **Error! Reference source not found.** 3):

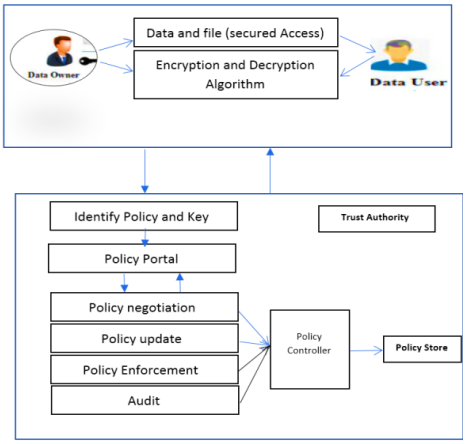


Fig. 3: Sticky Policy Components in CCS

• *Data Encryption Stage:* This stage is used to keep data confidential by encrypting it using the AES encryption algorithm. AES is a symmetric block cipher that can encrypt data blocks of 128 bits using symmetric keys 128, 192, or 256. NIST recommends using the AES128 and AES256 algorithms to encrypt long-term storage data. In addition, customer compliance requirements often include the AES algorithm. Therefore, we choose to use the AES128 algorithm to encrypt data. In this work, the AES128 algorithm is applied to data that the instructor (the data owner) wishes to encrypt. The shared key that is generated during the encryption process is held by the data owner's domain controller, unlit legitimate data users request access to encrypted data. Upon client request, an e-mail is sent to the data owner with the request, who can grant access according to the policy's rules that are set for that specific data file. All these actions are registered in the "data audit" file, along with any changes or alterations to the original data file in the chain.

• *Policy Rule Engine Stage*: sticky policy is used to insert the terms of use and access policies of the data. The policy engine is the core component of the trusted authority domain, where security is assuring through recording data requests and privilege granting behaviors over the network's traffic. When data are encrypted, access is only possible if granted by the data owner and the policy engine. When the policy's rules are met and constraints on the data are satisfied, the required data are decrypted and become accessible.

The work conducted in this research is developed based on [27] due to the importance of preserving security for big data [32], where several sub-components are used: the policy portal, policy controller, policy negotiation component, policy update component, enforcement component, and policy store. depicts these components. Each of these components is responsible for a portion of the policy application and rules settings:

1) Policy portal acts as the receiver of data access requests from any party (data owner or data user) and the sender of responses to these parties.

2) Policy controller is responsible for deciding on granting access to data or not, where this decision is forwarded to the corresponding component.

3) Policy negotiation component is responsible for negotiating access to data by requesting components.

4) Policy updating components when the data owner makes updates to the policy rules that are related to data, this component registers these updates and matches policy requirements.

5) Policy Enforcement component is used to ensure the fulfillment of policy rules on related data. When data requirements are met, these component actions on designated data are performed. Afterward, notifies the data owner to release the decryption key and identify the user that has access to related data. Enforcement of actions to other components like the policy negotiation and policy updating components is also assured through this component.

6) Policy store is responsible for keeping track of files and rules storage matters and keeps track of data and policy auditing actions.

Enforcement of the proposed sticky policy is the most critical component of the entire policy procedure. Encryption and decryption of data based on data users will be an efficient and effective policy enforcement procedure because the policy sticks to the data, which is only decrypted if the policy rules are applied. A sticky-policy-based approach for cloud security is used with different degrees of stickiness. This research mainly adopted the loose-couple binding policy, where data fragments and their sticky policies are stored separately and encrypted, as this procedure provides great infrastructure compliance.

• *Privacy and Data protection Stage*: This stage is used to ensure that data are not accessed or modified by unauthorized users or in an unauthorized manner and kept protected and correct. For that, a simple Blockchain algorithm is applied. Data protection is concerned with ensuring that the data files can only be modified by the file owner or group of trustees. In this work, the sticky policy is combined with a simple blockchain technique for the following:

1) Guarantee that the user who intended to modify or update the data file is authorized. This process is done by specifying a group chain (number of users are selected and given full authority to access and update data files).

2) Store audit files that record all access and modification attempts. Each update on the data file will be reported to

3) All chain members to notify them of the latest update made on the data file.

A sticky policy with the aid of a simple chain list, represented by the audited file and the notification notes passed to the trusted chain group helps in enforcing privacy and data integrity.

3.1 PDIS Structure

Error! Reference source not found. 4 shows the architecture of the proposed data security and privacy system. The PDIS layer is logically divided into three main domains, namely, the data owner domain, the third-party (organization) domain, and the data user's domain.

• The *Data owner domain*, which encrypts and stores the data using AES128, maintains its integrity and existence within a chain (defining a chain of trusted users with their authority to access and control data files). This domain specifies the accessing policy to various users and sticks them to the file based on [26]. The sticky policies are separately encrypted using the identity-based encryption (IBE) algorithm due to its simplicity and robustness [11]. IBE is an asymmetric cryptography algorithm. The policies are encrypted by the owner using his key, in this case, even if the trusted third party can decrypt the policies with his private key, he cannot change them for the client's benefit.

• The *Third-Party domain (organization)*, which is responsible for:

1) Intrusion detection and prevention
2) User authorization check (check user ID, password, access control, and policies attached to the requested data).
3) Decrypting data before granting the request.
4) Report access request to the trustees' (members of the chain group).
5) Record the access request in audit file (user ID, time and date of data access, type of data accessing, IP address of the log in the device).

• The *Data user's* domain, requests access to data from its owner, and receives a plain data file.

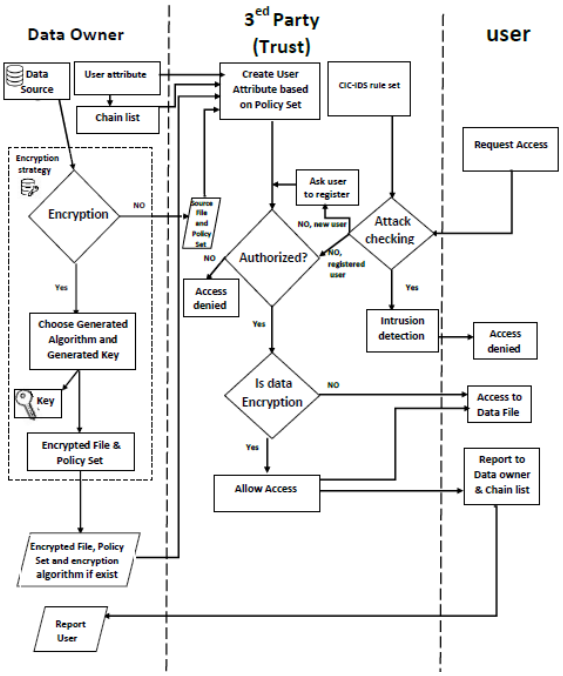


Fig. 4: General PDIS workflow

3.2 Flow Control of PDIS

Data on the cloud can be found in two forms, namely, the data created by its owner before being uploaded to the cloud system and the data that were created directly on the cloud environment using cloud services. The control flow of the suggested approach when a user sends a request to access a file is as follows:

- When users request a certain application or data file, the request will be first passed through the ID phase to guarantee that the request is normal (legitimate) and not an anomaly (attack).
- If the request is not detected as normal (based on the normal CIC-IDS rule sets), the request will be denied. Otherwise, the user request will be passed to the privacy phase to check if the user is authorized.
- In the case of an unauthorized user, the owner will need to decide either to add the user to the authorized list or to ban the request.
- In the case of authorized request, the access control of the target user will be specified based on the sticky policy determined by the owner. This step is to prevent illegal access to data or applications by legally unprivileged users (i.e., when users are privileged to read data but not modify it, allow to read and ban modify operation).

- The owner will determine the group of users who have full access authority, called a chain group (trust group), to maintain privacy and data protection. All types of access to the data or application will be audited and kept in an audit file. Furthermore, group members will be informed about the accessing operation and data updates (utilizing blockchain approach). This step will hinder internal intruders from breaching the data because any access (legitimate or illegitimate) will be reported to the trustees' chain and will also be recorded in the audit file. The owner can also activate the "protection option" on the classified data. To access data secured with a "protection option", it needs the consent of two members of the trust group (chosen randomly each time).

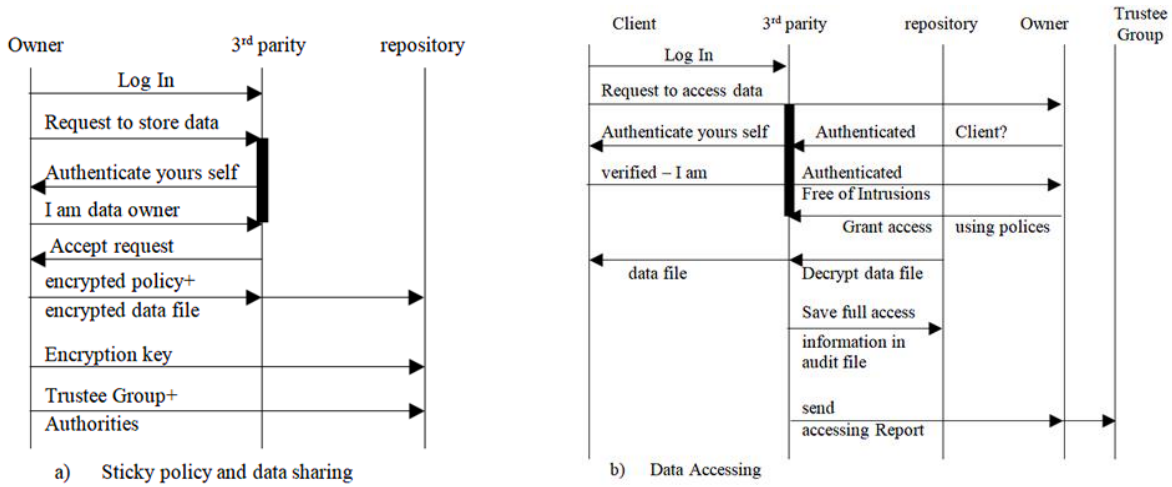
Secure data with protection Option

Request to access Data "D" by an authorized user
If protection-option. D is off, then
  Allow access, update the audit file
Else
  Randomly choose two members of trust group (SG)
  Ask for access permission from (SG)
  If both Agree, Allow access
  Else Ban user access
Report to the trust group

4 Implementation and Analysis of Findings

In this section, the proposed PDIS is presented through a case study. A web-based learning application is developed to secure teacher–student data accessing to analyze the behavior of suggested PDIS. The web-*based learning application* has been implemented using Oracle with its Java package JDevelope. The system flow is tested through a set of user interfaces that are built on a web server. The developed application is used by instructors and students at the university. The instructor is a one-node connected with other instructors to control the university file-sharing system (instructors represent the trustee group in the blockchain module, while the course-coordinator is the owner). Moreover, the repository of files is shared among the users (students and instructors) based on authorization and authentication policy. One of the most critical aspects of such a system is how to distribute the authorization and authentication rules among the instructors and students while maintaining the privacy of each. In the case study, privacy and data protection are preserved through access policies defined by the data owner (course-coordinator). The policies are encrypted using IBE. Each encrypted policy file is linked to the corresponding data file using a relational database. The database is stored in the third-party domain (university). The database fields are updated (or add a new data file with its policies) only by the data owner after the authentication process (third parity sends a verification code to the data

owner to authenticate him). Security is enhanced by encrypting the data using AES128 before saving it in the third-party domain. Error! Reference source not found.5 illustrates the sequence diagram of the developed case study. The instructor (the data owner) shares some data and files on the system. Students who look to access those files (data users) must be authorized by the data owner to be able to use or download the data files. The data owner might encrypt the data (or parts of it) and enforce the sticky policy to be applied to the data. When the data user tries to access this encrypted data, they will have to be granted access by the data owner, after the third party (university) authenticates the user and approves the access (free of intrusions). The students (clients) who need to access the data files must be approved by the IDS to ensure that the access is legitimate, before sending a decryption request to the instructor. This case is also similar when data are not encrypted, but without asking for decryption,



only asking for reading access.

Fig. 5:Sequence diagram of web-based learning application

A) Data Owner Module

Two types of data are stored in the cloud. The first one is the data that are generated by the owner before being loaded into the cloud. The other is the data that are generated on the same cloud platform as part of the service. In this module, the data are stored in a distributed database at the organization (university) and to be accessed through the corresponding access policies. The data are encrypted by using the encryption algorithm AES128, and then, a set of trusted parties (chain members) are identified. Finally, the user access policy is specified, and encrypted using the IBE algorithm, which is then attached to the data file (using the sticky policy approach in Section 3). The policies are concerned with the following:

- Registration of user information.
- Determine the user's authority (read, update, etc.).
- Identify restrictions on the date, time, and place to access data.

B) ID Module

The user will perform the register step, at which user information is stored as user attributes to be used in policy matching. After, the user will send a request either to upload or read data. The ID module (at the third-parity domain) will check the access request against the set of normal rules (CIC-IDS rule sets), generated based on the CICIDS2017 dataset. The ID module is mainly consisting of two parts:

- *Part one (user verification):* Another level of security is to authenticate the user by sending a verification code to the email (or phone no.) he entered during the registration process. The user has to use the verification code as a password in the first login. The user will be given the authority to change his password later.
- *Part Two (rule matching):* this part is responsible for detecting if the user request is normal or an attack. Access attributes are extracted and matched against normal rules (with confidence=1), that are generated based on CICIDS2017. If user access attributes match one of the normal rules, then access is allowed (or else access is denied). In this work, the normal rules are generated, but we need to have access to log files to get user attributes, which is not allowed for security reasons. Therefore, this operation is accomplished by a third-party domain (the university).

C) Chain Module: Real-Time Internal Intrusion Detection

A group of trustees, who have been identified by the owner as chain members, will have the authority to add, delete, and modify the data. When the data are uploaded by one of these people, the data seem available for each person within this chain. Whenever the shared data are modified, these modifications appear on a page called audit project. The blockchain shows the change in the data once the user makes an update. All users in the chain (based on level) are notified about the access and can identify who made it and when (date and time). Moreover, the blockchain may increase the negative effect of bad use of authorization if not controlled by sticky policy.

4.1 IDS: Rule Extraction & Evaluation

The dataset preprocess stage is needed to overcome the problems concerning record duplication, data dimensionality, missing attributes, and others, which may affect the accuracy of the extracted rules (wrong rule-extraction means a gap in the IDS) to generate the ID rule sets. The dataset (CICIDS2017) considers the network traffic monitoring during a group of days. First, we integrate all these days into one dataset with more than 2 million records; each record consists of 85 attributes. The dataset covers 12 types of attacks. The main problem is the existence of noise data as part of the dataset. This noise concerns duplicate records and missing attributes (missing values). Therefore, noise removal is needed (as preprocessing step) to improve system accuracy and enhance the confidence of the generated rules. These rules will be used to detect intrusions accurately. Intrusions may cause serious harm to sensitive data. The

preprocessing step focuses on removing duplicate records and remove missing attribute.

Another problem is many attributes (85 attributes) in the dataset. Many attributes may sometimes degrade the system accuracy as some attributes have a negative effect on the detection systems. In addition, considerable rules will be generated as detection rules. We must select the most proper subset of attributes to reduce the number of attributes without affecting system accuracy. In [18], 25 features are specified as the best feature subset that could be used to detect intrusions without affecting system accuracy. In this work, we trained J48 decision tree using the 25 features with 10-fold cross-validation. The accuracy of the testing results reached 99.8% (on average), which is comparable to utilizing full features accuracy. Reducing the number of features not only improves system accuracy but also helps in reducing the number of rule sets (i.e., rules could be generated using the 25 features with preserving high detection accuracy). R studio is used to generate IDS rules due to its analytic tool for the Statistical data. Many packages of R tools are used to generate the IDS rules. Several rules are generated from the R tool when the 12 types of attacks are considered. Therefore, we decided to rely on rules that detect normal access with confidante = 1. Any access that does not match any of the normal rules will be denied regardless of the attack type. The high accuracy of the proposed IDS raises confidence in the IDS rules. The trust in the ID technique improves the security of the system, thereby enhancing privacy in the later phase.

4.2 Sticky Policy: Flow Evaluation

As mentioned earlier, the system is composed of three main domains: the data owner, the data users, and the trusted authority (third-party authentication). The first two domains were explained earlier, whereas, in this section, the trusted authority component will be thoroughly investigated because the evaluation of the sticky policy takes place in it. The third-party authenticator (university) will decide which part of the policy suits the received data and access requests from the other domains and produce a proper response. This response can be in three outcomes: true (positive), false (negative), or unknown. The evaluation of the policy rules starts by categorizing requests to match the “common deny” rules. If the evaluations returned a “true” outcome, then this request is denied, and the issuing component is flagged as malicious. Otherwise, if the request fits into the “common permit” category, and the evaluation produced “true”, then this request is granted access to the data. Now, if a request is categorized as “domain deny,” and the evaluation outcome was “true,” then this request is denied, and the requesting party is blocked. Moreover, in the “domain permit” category, when evaluated as “true,” access is granted to domain components as stated in the policy. In case the evaluation returns “false” for any of the mentioned categories, then access will be denied. These actions and requests will be kept in log file records, which are accessed from the “audit” interface. The log file is used for network behavior analysis by the trusted authority component (third party), to assess malicious data usage

and access and tune policy's rules accordingly. Upon which, the decision about releasing or holding the decryption key is made. In the data user's component, when an access request is initiated, the authority component replies with a permit or deny to that request according to the policy's rules. If the response is to permit access, then data are decrypted using the decryption key shared by the data owner with the authority domain, and the download link is activated. By contrast, if the response was to "deny" access, then the download link stays disabled, and the user cannot download the file. Regardless of the response, the request is registered, along with all transactions and message requests in the log file's audit database. The trusted authority component can perform analytics on this database to enhance the ID process and tune the policy's rules according to recorded behavior. Finally, the response along with all transactions and message requests and accessing information are reported to the trustee group including the data owner. These two steps will overcome the internal attack problem given that any access type to data will be saved in the audit database and reported to the trustee group. Moreover, using sticky policies and encrypting data and policies will preserve data security and reduce external attack problems.

5 Conclusion

In cloud computing, security and privacy issues have become a significant problem that needs a smart solution. Data in the cloud must be protected against malware, internal attacks, and external attacks, including preserving privacy and data integrity. In this study, we tackled the security and privacy problem by developing a PDIS layer, which is placed above the SaaS layer in the cloud service model. The PDIS layer mainly consists of three modules, namely, IDS, privacy, and access control (owner module), and the data integrity model. Anomaly-based IDS has been developed by using the data mining approach and the CICIDS2017 dataset. Rule sets that detect normal access with confidante = 1 are generated and used to detect anomaly access. Any access that does not match any of the normal rules will be denied regardless of the attack type. System accuracy reaches 99.8%. The second module is the owner module, which is developed to maintain privacy. Accessing policy is specified by the data owner and attached to the data file. The data file is encrypted with the strong encryption algorithm AES128 and then saved in third-parity storage, along with its policy. The policy is encrypted using the IBE algorithm before attaching it to the data file. The policies are encrypted by the owner using his key, and the trusted third party can decrypt the policies with his private key, but he cannot change them for the client's benefit. The sticky policy ensures that the data owner is the only legitimate user who can grant access to encrypted data, and any access to the data should be only through or by that owner. The policy is enforced over three areas, namely, the data owner, the policy enforcement engine, and the user's data. Each of these areas is composed of several components that integrate to help in making this security procedure solid. The developed system in this research adopts the sticky policy procedure described above,

on an application that helps a university instructor publish teaching material and other files over a cloud computing system, and grant access to students only after approving this access. The final module is the chain (blockchain-based), which is used to preserve data integrity and hinders internal attack. The data owner will define the trustee group along with the authorization list. Any access to data will be reported to all trustee groups and the owner. Furthermore, the history of data accessing will be kept in the audit file. The developed PDIS proved to be effective and easily adaptable, which can rapidly process requests and obtain replies from the data owner. The sticky policy applied provides further security enforcement, where an IDS is a vital part of it. The ID and owner models preserve privacy and reduce external attacks, whereas the blockchain module maintains data integrity and hinders internal attacks. The prototype of PDIS is a private deployment model which is considered a limitation. In future work, PDIS could be changed to a hybrid cloud deployment model by connecting it with a third-party public-cloud service provider for trustee-list applications and integrating them by using common cloud management and automation platform.

References

[1] Al-Dulaimy, A., Taheri, J., Kassler, A., Hoseiny Farahabady, M. R., Deng, S., & Zomaya, A.: MULTISCALER: A Multi-Loop Auto-Scaling Approach for Cloud-Based Applications, IEEE *Transactions on Cloud Computing*, doi: 10.1109/TCC.2020.3031676.

[2] Xiao, Z., & Xiao, Y.: Security and privacy in cloud computing. IEEE communications surveys & tutorials, 15(2), 843-859 (2013).

[3] Pearson, S.: Privacy, Security and trust in cloud computing. In: P. S., & Y. G. (Eds.), Privacy and Security for Cloud Computing, pp. 3-42. London: Springer (2013).

[4] Alhroob, A., & Samawi, V. W.: Privacy in Cloud Computing: Intelligent Approach. The 16th Annual Meeting, International Conference on High-Performance Computing & Simulation (HPCS2018), pp.1063–1065, Orléans, France: IEEE (2018).

[5] Cook, A., Robinson, M., & Ferrag, M. A.: Internet of Cloud: Security and Privacy Issues. In: M. B., D. H., D. S., & J. A. (Eds.), Cloud Computing for Optimization: Foundations, Applications, and Challenges. Studies in Big Data, 39, 271-201, Springer (2018).

[6] Al-Dulaimy, A., Itani, W., Shamseddine, M., & Taheri, J.: Privacy-Aware Job Submission in the Cloud. IEEE Middle East and North Africa Communications Conference (MENACOMM) pp. 1-6. Manama, Bahrain: IEEE (2019).

[7] Zhang, Q., Wang, G., & Liu, Q.: Enabling Cooperative Privacy-preserving Personalized search in cloud environments. Information Sciences, 480, 1–13(2019).

[8] Popli, M., & Gagandeep.: A Survey on Cloud Security Issues and Challenges. 6th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 230-235. New Delhi, India: IEEE (2019).

[9] Li, P., Li, J., Huang, Z., Gao, C.-Z., Chen, W.-B., K.: Privacy-preserving outsourced classification in cloud computing. Cluster Computing, 21, 277-286 (2018).

[10] Tabrizchi, H., & Rafsanjani, M. K.: A survey on security challenges in cloud computing: issues, threats, and solutions. J. Supercomput (2020).

[11] Miorandi, D., Rizzardi, A., Sicari, S., & Coen-Porisini, A.: Sticky Policies: A Survey. IEEE TKDE, 20 (2019).

[12] Rao, V. U., & Neelima, D. (n.d.): Axially symmetric space-time with strange Quark matter attached to string cloud in self-creation theory and general relativity. Int J Theor Phys 52, pp. 354–361 (2013).

[13] Sen, J.: Security and privacy issues in cloud computing. In: Cloud Technology: Concepts, Methodologies, Tools, and Applications, pp.1585-1630. IGI Global. (2015).

[14] Prathima, S., & Priya, C.: Privacy Preserving and Security Management in Cloud-Based Electronic Health Records-A Survey, In: P. SL., S. L., S. G., & B. D. (Eds.), Lecture Notes in Networks and System, Intelligent Computing and Innovation on Data Science Proceedings of ICTID, 118, pp. 21-29. Singapore: Springer (2020).

[15] Rai, K., & Devi, M. S.: Intrusion detection systems: A review. Journal of Network and Information Security, 1(2), 15-21 (2013).

[16] Kim, K., & Aminanto, M. E.: Deep Learning In Intrusion Detection Perspective: Overview and further challenges. 2017 International Workshop on Big Data and Information Security (IWBIS), pp. 5-10. Jakarta: IEEE (2017).

[17] Riaz1, A., Ahmad, H. F., Kian, A. K., Qadir, J., Rasool, R. U., & Younis, U.: Intrusion Detection Systems in Cloud Computing: A Contemporary Review of Techniques and Solutions. JISE, 33, 611-634 (2017).

[18] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A.: Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. InICISSP. The International Conference on Information Systems Security and Privacy (ICISSP 2018), pp. 108-116. SCITEPRESS -Science and Technology Publications (2018).

[19] Al Naqshbandi, S.M., Samawi, V. W., & Herik V., J.: Intrusion Detection By A Dynamic Length Rule, Global Journal on Technology, 3, 1460-1470 (2013).

[20] Ghosh, P., Biswas, S., Shakti, S., & Phadikar, S.: An Improved Intrusion Detection System to Preserve Security in Cloud Environment. IJISP, 14(1), 67-80 (2020).

[21] Mishraa, P., Pilli, E. S., Varadharajan, V., & Tupakula, U.: Intrusion detection techniques in cloud environment: A survey. Journal of Network and Computer Applications, 77, 18-47 (2017).

[22] M. A. Halim, Azizi Abdullah, & K. A. Z. Ariffin: Recurrent Neural Network for Malware Detection. International Journal of Advances in Soft Computing and its Application, 11(1) 46-63 (2019).

[23] Jabez, J., Gowri, S., Vigneshwari, S., Mayan, J. A., & Srinivasulu, S.: Anomaly Detection by Using CFS Subset and Neural Network with WEKA Tools. In S. C. Satapathy, & A. Joshi (Eds.), Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies, 107, pp. 675-682. Singapore: Springer (2019).

[24] Pearson, S., & Casassa-Mont, M. Sticky policies: An approach for managing privacy across multiple parties. Computer, 44(9), 60-68 (2011).

[25] Spyra, G., & Buchanan, W. J.: Protecting documents with sticky policies and identity-based encryption. 2016 Future Technologies Conference (FTC)، pp. 953-957. San Francisco, CA: IEEE (2016).

[26] Spyra, G., Buchanan, W. J., & Ekonomou, E.: Sticky policies approach within cloud computing. Computers & security, 70, 366-375 (2017).

[27] Li, S., Zhang, T., Gao, J., & Park, Y.: A sticky policy framework for big data security. 2015 IEEE 1st International Conference on Big Data Computing Service and Applications, pp. 130-137. Redwood City, CA: IEEE Xplore (2015).

[28] Samawi, V. W.: "SMCSIS: An IoT based Secure Multi-crop Irrigation System for Smart Farming", International Journal of Innovative Computing, Information and Control (IJICIC), 17(4), 1225–1241 (2021)DOI: 10.24507/ijicic.17.04.1225

[29] Yaga, D., Mell, P., Roby, N., & Scarfone, K.: Blockchain Technology Overview. National Institute of Standards and Technology (2019).

[30] Gaetani, E., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., & Sassone, V.: Blockchain-based database to ensure data integrity in cloud computing environments. the First Italian Conference on Cybersecurity (ITASEC17), pp. 146-155. Venice, Italy (2017).

[31] Cao, Y., Sun, Y., & Min, J.: Hybrid blockchain–based privacy-preserving electronic medical records sharing scheme across medical information control system. Measurement and Control. 1-14 (2020).

[32] Al_Zyadat, W. J. A., Alzyoud, F. Y., Alhroob, A. M., & Samawi, V. W.: Securitizing big data characteristics used tall array and mapreduce. International Journal of Engineering &Technology, 7(4), 5633-5639 (2018).Doi: 10.14419/ijet.v7i4.24404

Rusul Mumtaz earned her master's degree in software engineering from Isra University, Amman, Jordan in 2019. Her major research interest is in Engineering Sciences and Information Technology, cloud computing, and data mining.

Venus W. Samawi is full professor at Isra University, department of Computer science/Multimedia Systems. Venus Samawi became a Member of International Association of Engineers (IAENG). She received her BSc from University of Technology in 1987, the MSc and PhD degrees from Computer Science Department in Al-Nahrain University in 1992 and 1999 respectively. Dr. Samawi supervises many PhD and MSc theses concerning system programming, pattern recognition, network security, and text classification. She also, leads and teaches modules at both BSc and MSc Levels in computer science. She is a reviewer in number of conferences and Journals. Her special area of research is in pattern recognition, evolutionary computing, image processing, and natural language processing. Lately, Dr. Samawi main research interest is IoT.

Dr Aysh Alhroob earned his PhD in 2010 from the University of Bradford in the United Kingdom. He is a Full Professor of Software Engineering at Isra University. Since October 2021, he is acting as Dean of Scientific Research and Graduate Studies. Earlier, he was the Dean of the Information Technology Faculty. Furthermore, he is also a member and founding member of several national and international associations, such as the IAENG, Artificial Intelligence and Entrepreneurship Association, Association of Software Quality Assurance, and the Association of Arabic Content Enrichment. As for scientific research, he is interested in Big Data analysis, data science, and software testing using AI techniques. Aysh has published many articles on missing data prediction, Big Data classification, the accruing velocity of Big Data, and accruing the constraints using panel Big Data. Moreover, He is currently leading research that aims to improve decision-making in a range of critical scenarios by combining AI techniques and historical Big Data analysis. In terms of the quality of education and training, he is interested in the application of European quality standards in education and training and in coordination with Human Restart-European Board of Science and Development.

Wael Jumah Alzyadat is currently an Assistant Professor of software engineering. He also works with Al-Zaytoonah University, Jordan. The interesting research area encompasses the area of software analysis, intelligence systems, streaming data, and big data. Moreover, he established more than 30 published articles and achieved two copyrights.

Ikhlas Hasan Almukahel Teacher, Software Engineering Department, Faculty of Information Technology, Isra University, Amman, Jordan. Currently, is the coordinator of the quality department. Furthermore, Research interesting areas are Software Modelling, AI, Big Data, and Analytics Data. I'm committee steering in Big Data Coherences with KDD. Background Education is a master's degree in software engineering from Isra University, Amman, Jordan; Bachelor's Degree in Computer Software Engineering from Science and Technology University-Sana'a, Yemen.