USER-BEHAVIOR TRUST MODELING IN CLOUD SECURITY

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Maryam Malaa Alruwaythi

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Computer Science

October 2019

Fargo, North Dakota

North Dakota State University
Graduate School

Title

USER-BEHAVIOR TRUST MODELING IN CLOUD SECURITY

By

Maryam Malaa Alruwaythi

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Kendall E. Nygard
Chair

Brain Slator

Oksana Myronovych

Benjamin Balas

Approved:

11/6/19 — Date

Kendall E. Nygard — Department Chair

ABSTRACT

With the cloud computing increasing in popularity by providing a massive number of services such as recourses and data center, the number of attacks is increasing**.** Security is a basic concern in cloud computing, and threats can occur both internally and externally. Users can access the cloud infrastructure for software, operating systems, and network infrastructure provided by the cloud service providers (CSPs). Evaluating users' behavior in the cloud-computing infrastructure is becoming more important for both cloud users (CSs) and the CSPs that must ensure safety for users accessing the cloud. Because user authentication alone is not enough to ensure the users' safety and due to the rise of insider threats, the users' behavior must be monitored. User-behavior trust plays a critical role in ensuring the users' authenticity as well as safety.

To address the research problem, we proposed two models to monitor the users' behavior in the cloud and then to calculate the users' trust value. The proposed models improve the current trust models. Our proposed models address the issue of trust fraud with the concept of "slow increase." The proposed models deal with malicious conduct by constantly aggravating the penalty approach (principle of "fast decline"). The proposed models reflect the users' latest credibility through excluding the expired trust policy in the trust calculation. The proposed models evaluate users based on a large amount of evidence which ensures that the users' trust value is stable. We generate a dataset to simulate audit logs containing the designed user-behavior patterns. Thus, we use the dataset to evaluate our proposed models.

ACKNOWLEDGEMENTS

First, I am deeply thankful to my lovely family, parents, sisters, and brothers, for supporting me while I pursued this degree. Thanks go to my great husband, Abulmajeed Alruwaythi, who encouraged me to achieve this degree and motived me when I need it. Thank you to my lovely sons, Yusuf and Hossam, who have been my motivation, inspiration, and drive.

I would also like to thank my supervisor, Dr. Kendall E. Nygard, for his support and advice while completing my research. Without Dr. Nygard's guidance and constant feedback, this dissertation would not have been completed. Finally, I would like to thank my dissertation committee's members: Dr. Oksana Myronovych who always motived and encouraged me when I needed it, along with her valuable guidance, and Dr. Brian Slater and Dr. Benjamin Balas for agreeing to serve on my committee and for their valuable expertise and precious time.

DEDICATION

This doctoral disquisition is dedicated to my parents, husband, sons, and siblings. This dissertation would never have been published without them. I'm pleased to have them in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

NIST	National Institute of Standards and Technology
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
SLAs	Service-Level Agreements
CPU	Central Processing Unit
IP	Internet Protocol
CSA	Cloud Security Alliance
ENISA	European Network and Information Security Agency
CSPs	Cloud Service Providers
CSCs	Cloud Service Clients
IBAC	Identity-Based Access Control
TBAC	Trust-Based Access Control
MAC	Mandatory Access Control
DAC	Discretionary Access Control
ABAC	Attribute-Based Access Control
RBAC	Role-Based Access Control
QoS	Quality of Services
TBUB	Trust Based on User Behavior
IDS	Intrusion Detection System
SNMP	Simple Network Management
RMON	Remote Monitoring

AHP	Analytic Hierarchy Process
FANP	Fuzzy Analytic Network Process
MTBAC	Mutual Trust-Based Access Control Model
FCE	Fuzzy Comprehensive Evaluation
FART	Fuzzy Adoptive Resonance Theory
RPTM	Reward and Punishment Trust Model
FRBAC	Flexible Role-Based Access Control
AIMD	Additive-Increase, Multiple-Decrease
UBADAC	User Behavior Assessment-Based Dynamic Access Control
DTBAC	Dynamic Trust-Based Access Control
FMUBCT	Framework for Monitoring User Behavior and Compute User Trust
FUBT	Fuzzy Logic Approach Based on User Behavior Trust
AWS	Amazon Web Services

CHAPTER 1. INTRODUCTION

1.1. Introduction

Trust can play a key role in addressing cyber-security concerns. Cybercrimes cost the world nearly $3 trillion in 2015. This figure is expected to increase to $6 trillion by 2021 [1]. In recent years, the demand for computing has gradually increased. With a variety of technological advances, people can access wide-ranging information repositories and multiple resources. These interactions become cheaper and more powerful than before. One such technology is cloud computing which has been defined by the National Institute of Standards and Technology (NIST) [2] as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." There are three types of service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Cloud computing has worthy features which are attractive to governments and business owners. By 2023, the U.S. government will exceed $10 billion with a growth rate of 16% from 2018-2023 of spending on cloud computing [3]. One cloud-computing feature is the pay-as-you-go pricing model, meaning that the user pays for computing resources as they are needed. Another advantage is reducing the operating cost. Because the cloud environment can rapidly allocate and de-allocate resources on demand, the service provider does not need to provision resources for the peak load. The cloud service is web hosted, so cloud users can easily access it with an internet connection. Thus, one cloud-computing advantage is that cloud users can access the cloud wherever and whenever they want. Because of outsourcing the service infrastructure to the cloud, users shift the maintenance expenses to the cloud service providers (CSPs).

With cloud computing's increased popularity for providing a massive number of services, such as resources and data centers, the number of attacks is increasing [4]. The cloud platform's security is an important factor for cloud development. Storing important business data and confidential information in a cloud environment requires that a high-security mechanism be applied in the cloud platform. Basic security protections, such as firewalls and traditional access control, are not able to satisfy security requirements with the expansion of cloud computing. A lot of significant resources are stored in the cloud. The number of cloud users may be hacked because of limitations with basic security protection. To strongly implement a secure, reliable, and safe cloud-computing environment, we need to consider the trust issue. With cloud computing, there is need for mutual trust from the cloud users and the cloud service provider; neither one is unessential. Because of the user lacks controllability of data and resource, there is mistrust about cloud computing. The mistrust could happen based on the following reasons: leaking sensitive data, data-loss risk, data-release risk, storage-location risk, service distraction, cloud-provider breakdown risk, etc. [5,6]. These problems will damage the information security, data availability, and integrity, leading to economic and finance losses [7].

The trust should be bi-directional; cloud users trust the cloud service provider, and the cloud service provider trusts the user. When users trust the service provider, they wish to store their data in the cloud and to use the cloud for their daily work environment. On other hand, provider trust allows the users to deploy their applications and to execute tasks on cloud, and to directly access data and the cloud infrastructure for software, operating systems, and network infrastructure which are provided by the cloud service providers (CSPs). Thus, the malicious user may deploy a malicious code. This code may cause a huge disaster by occupying central processing unit (CPU) time, memory space, and other resources; taking control of the virtual

machine and possibly attacking other users; and potentially attacking the underlying platform which provides the operational environment [5,6]. The malicious users could be competitors, hackers, opposition, etc. Therefore, it is a critical time for the cloud service provider to monitor the users' behavior in order to detect and to prevent malicious users, which is the objective of this dissertation. This research contributes by building models to monitor the users' behavior and to calculate the user's trust value by utilizing fuzzy logic as the artificial intelligent technique as well as sliding windows technique. The dissertation's output, extracted common behavior patterns, could be used to detect insiders by comparing them with the users' runtime activities. Then, the trust value is calculated and used as another factor for user identification.

1.2. Motivation

Currently, most cloud service providers use a common security mechanism to identify users and access management (IAM) through authentication and authorization. However, IAM has disadvantages which make it unable to satisfy the security requirements within the expansion of cloud computing. The first issue is that, before the user starts an interaction, the IAM assigns authority to the user without monitoring the user's actual behavior throughout the interaction. Therefore, it is impossible to detect and to prevent malicious action, disoperation, and risk behavior effectively. The second concern is that the IAM only verifies the user's trust with the identity, test username, password, and internet protocol (IP) address in order to identify the user's identity, without verifying the user's behavioral trust based on the historical evidence of interactions; IAM cannot detect and prevent potential malicious users [8]. Authorized users still carelessly utilize risky behaviors which might lead to data damage or leaks with the secure resources. There is an increasing risk of an insider threat to data security, and nearly two-thirds of the recent attacks or data leaks have been caused by insiders [9, 10]. Security consciousness

moves from the traditional access-control defense to holistic behavioral solutions in order to detect and to prevent ongoing insider threats before they actually occur [11, 12, 13, 14]. Finally, a user's identity could be stolen by hackers.

Because insiders already have privileges to access the organization's information and assets, it is even more difficult to defend compared to defend attacks from outsiders. The authorized users' activities must be monitored and controlled on an ongoing basis. Because of the large number of people using cloud services, including employees, contractors, customers, and partners; the complexity and changing nature of user-behavior patterns; and the lack of fully featured signature-based detection capabilities, system administrators for cloud-computing systems have had major challenges to identify user-behavior patterns. With these disadvantages, the IAM is unable to solve the cloud-computing security issues effectively. Expanding and updating traditional access-control technology is urgently needed in order to address trust issues and to improve the cloud-computing security issues.

The concept of "trust" was first introduced in computer science by M. Blaze in 1996 [15]. The basic idea is to admit the imperfection of security information in an open system; the system's security decision requires a reliable third party to supply additional security information. One branch of trust management theory is the trust based on user behavior (TBUB). TBUB is a comprehensive evaluation of the user-interaction behavior by using quantized evaluation results to represent the user's creditability on the cloud and to identify risky and malicious users [16]. Therefore, TBUB provides a valid decision-making basis for an access-control system and improves the reliability for the allocation of authority. TBUB is a trending technology that addresses security issues in the cloud. However, there are remaining problems to

solve; identifying the problems for design models and then improving the TBUB performance is the research problem for this dissertation.

1.3. Dissertation Statement

The aim of my research is to improve and to develop a user-behavior trust model for cloud computing. This study includes improving the current user-behavior evaluation methods and calculating the user's creditability based on user behavior during the interaction with the cloud as well as the user's behavior history. In addition, the model should be able to prevent a malicious user's access to the CSPs whenever a user behaves abnormally. Moreover, the model should be able to update the user's trust value and the CSPs changes for a user's authority rule in a timely manner.

It is significant to secure the cloud from the insider user. I propose two models to improve security by calculating the user's trust value. I use multiple steps in each model: find the common behaviors, which is defined as a frequently discovered pattern, from the dataset. Then, use fuzzy logic and the sliding windows technique to compute the user's trust value. The dissertation provides valuable knowledge for anomaly detection and trust-computation research by extracting a small set of categorized, representative patterns and then using the techniques to compute the trust value.

1.4. Dissertation Contribution

This dissertation presents models to calculate the trust value based on the user's behavior in the cloud. Based on the trust value, the user can be allowed to or denied access and use of the cloud. This research's major contributions are as follows:

- Design ,and implement the user-behavior trust model based on proposed equations, sliding windows ,and fuzzy logic. The proposed model can be applied on different cloud

deployment types. The model considers all the evaluation principles which were not considered with the previous models. In addition, we designed user-behavior patterns based on the user's activity history, which allows the discovery of normal user behavior. Our proposed model considers three evidence types: security, login, and operation. Moreover, the model reflects three types of trust: direct, history, and comprehensive.

- Design ,and implement the user-behavior trust model based on fuzzy logic. The proposed model is useful for all cloud deployment types. The model utilizes all the evaluating principles by using different equations than model one. Our proposed model considers four evidence types: security, login, operation, and performance.The model considers five types of trust: direct, history, indirect trust (from other users in same domain , and different domains in the same CSP), and comprehensive.
- Based on the lack of having a user-behavior dataset, we developed a behavior-simulator algorithm that allows the generation of dataset. This algorithm provides a testing benchmark to evaluate the user-behavior trust models.
- Present case studies to demonstrate the functionality and performance of the proposed models. The experiment has been organized as follows:
    - Verification: we run multiple experiment cases to demonstrate that the proposed models can produce the user's history pattern and efficiently prevent a malicious user from accessing the cloud service provider whenever the user behaves abnormally. Moreover, the proposed models are able to update the user's trust value in a timely manner; then, the cloud service provider changes the user's authority.

- There is a comparative analysis with the existing models; we run different experiment cases to validate that the proposed models are more efficient than the existing models.

1.5. Publication and Peer Review

Five papers were published for this dissertation. The first and second papers were specifically about the proposed models which are described in Chapters 3 and 4. The third paper was a literature review which is described in Chapter 2. With these three papers, Maryam Alruwaythi was responsible for the innovations described, but the dissertation's academic adviser was a co-author; Krishna was a co-author who edited (grammar and paper layout). Maryam Alruwaythi was the peer reviewer for the fourth and fifth papers.

1. **Maryam Alruwaythi**, Krishna Kambhampaty, and Kendall E. Nygard, "User Behavior Trust Modeling in Cloud Security," The 5th Annual Conference on Computational Science & Computational Intelligence, Las Vegas, NV,2018.
2. **Maryam Alruwaythi** and Kendall E. Nygard, "Fuzzy Logic Approach Based on User Behavior Trust in Cloud Security," Proceedings of the 19th Annual IEEE International Conference on Electro Information Technology (EIT 2019), Brooking, SD, 2019.
3. **Maryam Alruwaythi**, Krishna Kambhampaty, and Kendall E. Nygard, "User Behavior and Trust Evaluation in Could Computing," Proceedings of the 34th International Conference on Computers and Their Applications, Honolulu, HI,2019.
4. Kendall E. Nygard, Ahmed Bugalwi, **Maryam Alruwaythi**, Aakanksha Rastogi, Krishna Kambhampaty, and Pratap Kotala, "Elevating Beneficence in Cyberspace with Situational Trust," Proceedings of the 32nd International Conference on Computer Applications in Industry and Engineering (CAINE 2019), San Diego, CA, 2019.

5. Krishna Kambhampaty, **Maryam Alruwaythi**, and Kendall E. Nygard, "Trust and Its Influence on Technology," Midwest Instruction and Computing Symposium, Fargo, ND, 2019.

1.6. Dissertation Structure

This dissertation is organized as follows:

- Chapter 2. Background and Related Work.
- Chapter 3. Model 1: Framework for Monitoring the User's Behavior and Computing the User's Trust. This chapter presents the first model which improves the existing models. In this model, we apply two techniques, sliding windows and fuzzy logic, to compute the cloud user's creditability.
- Chapter 4. Model 2: Fuzzy Logic Approach Based on User-Behavior Trust in Cloud Security. This chapter presents the second model which improves the first model by adding performance evidence and indirect trust to compute the comprehensive trust. With this model, we only apply fuzzy logic to compute the cloud user's creditability.
- Chapter 5. Experiments and Simulation Results.
- Chapter 6. Conclusion and Future Work.

CHAPTER 2. BACKGROUND AND RELATED WORK

2.1. Cloud Computing

The cloud-computing model allows for convenient and on-demand access to shared resources, such as servers, storage, applications, software, and services, that can be dynamically delivered as needed. The National Institute of Standards and Technology (NIST) [2] defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." The essential features of NIST's cloud computing [2] are as follows:

- **On-demand self-service:** Cloud computing provides on-demand services where users can access computing resources as needed.
- **Resource pooling:** The resources include CPUs, networks, and storage resources. These resources are pooled to provide multiple users with simultaneous access.
- **Broad access to the network:** Cloud-computing resources can be accessed over a network through a wide range of devices, such as personal computers, tablets, and smartphones.
- **Measured service:** Cloud resources, such as memory, network bandwidth, and CPU usage, can be monitored and measured.
- **Rapid elasticity:** Resources for cloud computing can be allocated elastically or distributed according to the consumers' needs. These computing resources can be scaled depending on workload variations.

2.1.1. Cloud Service Delivery Models

Based upon the NIST definition, cloud computing has three distinct service models [2]:

- **Software as a Service (SaaS):** This software service is provided via the internet. Clients can use software applications without installing, maintaining, or updating the software; and without managing the cloud infrastructure. Examples of SaaS are Google Docs, and Game.
- **Platform as a Service (PaaS):** This type of cloud computing provides clients with a platform to deploy their created applications using provider-supported programming languages, libraries, services, and tools. Without the need to manage the underlying cloud infrastructure, clients can use this cloud platform. Examples of PaaS are Google App Engine, and Microsoft Azure.
- **Infrastructure as a Service (IaaS):** This type of cloud computing provides clients a cloud frame for computing resources, such as networks, CPU, memory, and storage. Clients can allocate or deallocate resources, in a dynamic manner and as needed, without managing the underlying cloud infrastructure. Instead of buying additional hardware, paying for the system's software maintenance, or purchasing relevant system software, users can directly develop their own platforms and applications in the cloud infrastructure (service layer).

2.1.2. Cloud Deployment Models

Cloud deployment models are primarily divided into the following categories:

- **Public**: A general cloud provider owns the infrastructure behind a public cloud. A public cloud houses many services for various customers, so multiple tenants access them from multiple locations. Web interfaces are used to access services on a common basis. This

model is based on a pay-per-use business approach and is typically low-cost, providing highly scalable services. The cloud's resources are located at an off-site location, making this model's deployment less secure and riskier than others because malicious activities can occur with the service delivery models. In this case, customer-to-provider service-level agreements (SLAs) must be well detailed and analyzed [17,18].

- **Private:** is a new term used by some providers to describe offers that emulate private-network cloud computing. This cloud is located in an organization's internal enterprise data center. With a private cloud, the provider's scalable resources and virtual applications are pooled and available for sharing among the cloud users. A private cloud differs from a public cloud in that the organization manages all cloud resources and applications, similar to the functionality of the intranet. Due to specified internal exposure, private-cloud usage can be much more secure than a public cloud. Only the organization and designated stakeholders can access a private cloud [17, 18].
- **Hybrid:** is a combination of two or more other models of cloud deployment that a secure network centrally manages. A hybrid cloud is traditionally considered a mix of private and public clouds, bringing together the advantages of each type and overcoming their barriers. This model is managed by both the organization and a third-party entity and is located both on- and off-site [17,18].
- **Community:** The model of community cloud deployment is one that multiple organizations control and share. The cloud is usually set up to foster a common interest among multiple owners. The cloud can be managed by an owners' committee or a third-party organization and can be located on- or off-site. Community members are free to access the cloud data [17, 18].

2.1.3. Cloud-Computing Security

With cloud computing increasing in popularity, as a result of providing massive numbers of services, such as resources and data centers, the number of attacks is increasing [4]. A cloud's platform security is becoming an important factor with cloud development. Storing important business data and confidential information in the cloud environment requires that high-security mechanisms be applied to the cloud platform. The cloud-computing security issues have been widely affected by both academics and the IT industry. According to International Data Corporation (IDC) a market research firm statistic which rate cloud-computing development challenges or issues, among other challenges, the security issue is highly concerning at 87.5% [19]. The Cloud Security Alliance (CSA), NIST, and the European Network and Information Security Agency (ENISA) enumerate the following data threats for the cloud-computing environment [20]:

- **Data breaches:** Due to side-channel timing attacks on virtual machines, an organization's sensitive internal data may fall into the hands of its competitors. This type of attack can be designed to extract private cryptographic keys which are used on the same physical server with other virtual machines.
- **Corruption of stored data:** Because the cloud provider has root access to physical machines, this access enables the provider to modify or delete the client's data. The provider may interfere with the data, making them unusable, or alter the data in such a way that the system cannot detect the alteration. The interfering with the data constitutes a serious threat to the application data's integrity.
- **Data loss:** The stored data may be lost due to accidental deletion; a loss of encryption keys; or physical disasters, such as flooding, earthquake, and fire.

- **Denial of service:** Attackers can generate huge numbers of fake requests for a certain cloud server, forcing the server to consume processor power, memory, disk space, and network bandwidth. Denial of service attack, ultimately, causes an intolerable system slowdown and makes the service unavailable to other customers.
- **Malicious internal users:** These people can include current employees, administrators, other third-party service providers, or contractors who have access to and may misuse this access. The misuse causes intentional harm, affecting the confidentiality, integrity, and availability of an organization's sensitive data.

In conclusion, the study of cloud-computing security research is evolving gradually. S. Yu et al. [21] believe that this research can be divided in three ways:

- Cloud-computing system issues and network-security issues.
- Cloud-computing issues in data protection.
- Trust issues in the cloud-computing environment among entities, such as cloud service providers (CSPs) and cloud users. Currently, solving the cloud computing's security issues involves setting up a feasible cloud-security framework for specific security risks and using this framework to study key security technology. The trust-based access-control policy provides a possible solution for cloud computing's s trust issues.

2.2. Access-Control Technology

Access control is one of the most important elements in the field of information security. Access control is a fundamental and traditional mechanism for data security which is used to control unauthorized access for computing resources and data [22, 23]. Access control is a mechanism where services know whether to honor or deny requests to access computing resources and data. The system for the access-control model should include the subject, object,

and policy. The subject is a user who has permission to access the resources, which are the objects under certain policies. In cloud computing, the access-control model takes different actions, such as identification, authentication, and authorization, before accessing the resources. The access-control model has two types which can be applied to the traditional IT environment and the cloud environment:

- Identity-based access-control (IBAC) model.
- Trust-based access-control (TBAC) model.

2.2.1. Identity-Based Access-Control Model

The IBAC is a user identity-based access-control mechanism where access authorizations for specific objects or resources are assigned based on the user's identity. There are four types of IBACs: the mandatory access-control (MAC) model, the discretionary access-control (DAC) model, the attribute-based access-control (ABAC) model, and the role-based access-control (RBAC) model.

2.2.1.1. Mandatory Access-Control Model

The MAC [24] is an access-control policy where an initiator for a subject or request may perform some kind of operation on a particular object or resource. When a subject tries to access an object or the information in an object, an authorization rule is enforced, by reviewing security attributes, to determine if the access can occur. The MAC assigns different security levels to each subject and object in order to establish secure access to objects or the flow of information within objects. The subject's security attributes reflect the level of authority that can be obtained by that subject, and the object's security attributes reflect the object's sensitivity [25]. Although the MAC model protects information flow or leaks within an object, it does not guarantee the information's complete secrecy within an object [26].

2.2.1.2. Discretionary Access-Control Model

The DAC model is an access-control policy that provides service access to the owner of an object or resources. The basic idea of DAC is that the subject's owner determines the authority for other accessing subjects, and the subject that obtained the accessing authorization may further grant privileges to other subjects [26]. Most operating systems, including Windows, Linux, and Macintosh, as well as most Unix types are based on DAC models. When a file is created in these operating systems, admin decide which access privileges to give other users. When people want to access a file, the operating system makes a decision based on the privileges assigned to the file. From a security point of view, the MAC model is more secure than the DAC model.

2.2.1.3. Attribute-Based Access-Control Model

The user attribute is considered in the access-control model attribute in order to make access-control decisions. The user's attributes can be location, age, birth date, or role [27]. Each attribute has unique and unobtrusive values. To allow or deny access, this model checks the user's attribute against the predefined policy for a particular system or organization. Because there are a large number of users in cloud computing, deciding on a large number of attributes is a very complex task.

2.2.1.4. Role-Based Access-Control Model

The RBAC [28] model is an access-control procedure or mechanism where the decision about access control is made based on the user's predefined roles. This model's primary objective is to allow users based on their roles and permissions. A user's role and permission must be authorized before accessing any cloud resources. A specific user can be assigned more than one role, and more than one user can have a specific role. New permissions can be granted

for roles because the application may change according to user's requirements. Figure 1 shows the RBAC model's core, consisting of four elements: user, object, role, and permission, where the permissions are the type of operations applied to a resource or object. A role is a job function within a specific organization regarding the user's authorization and responsibility.

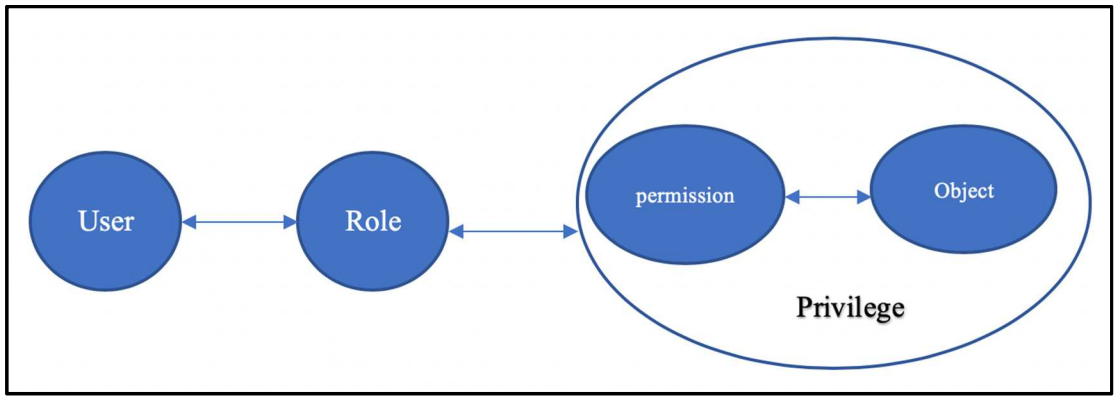

Figure 1. The relationship between the RBAC's elements.

The RBAC model has many advantages compared to the DAC and MAC models. The RBAC model's primary drawbacks are that it can only be applied within a closed network. Because this model is based on identity, it only checks the user's identity information to authorize the person. If the user performs any malicious activity or operation on the cloud's resources, the RBAC cannot control the user's behavior during the interaction or malicious operation.

2.2.2. Trust-Based Access Control Model

Because cloud computing is a very popular form of internet application, a large number of users exist, and user behavior is always uncertain and dynamic. Therefore, different risks affect cloud resources. The identity-based control models cannot be applied to cloud computing. The following section explains the trust concept, trust in cloud computing, and characteristics of trust.

2.2.2.1. Concept of Trust

The concept of trust has been studied in several domains, such as sociology, economics, and psychology. Sociologist Sztompka stated, "Trust is a bet about future contingent action of others" [29]. In psychology, trust "is believing that the person who is trusted will do what is expected" [30]. There are different definitions of trust, but "common to these definitions are the notions of confidence, belief, faith, hope, expectation, dependence, and reliance on the goodness, strength, reliability, integrity, ability, or character of a person or thing" [31]. The trust concept originally came from sociological theory, and the concept of trust has been introduced into the domain of computer science because the connection between people can easily refer to the interactions among computers, machines, and internet entities. The concept of trust in the computer science domain was proposed by Anderson [32]. Blaze et al. [33] were the first team to handle the internet security problem by using the trust-management concept to build a trust model for distributed systems. Simultaneously, Abdul-Rahman and Hailes [34] proposed the trust-metric mathematical model to handle internet security issues.

2.2.2.2. Trust in Cloud Computing

In the area of information security, trust is one of the most important issues. To strongly implement secure, reliable, and safe cloud-computing environments, we must consider the trust issue. Yew [35] defines computational trust in cloud systems as a "specific level of subjective assessment of whether a trustee (cloud provider) exhibits characteristics that are consistent with the trustee's role." In cloud computing, there is need for mutual trust from the cloud users and the CSPs; neither one is unessential. The user's lack of control for data and resources leads to the mistrust of cloud computing based on the following reasons: leaks of sensitive data, data-loss risk, data-release risk, storage-location risk, service distraction, and cloud-provider-breakdown

risk [5, 6]. These problems damage the information security, data availability, and integrity, leading to economic and finance losses [7].

Several studies have been done about trust in cloud computing. In practice, Patil and Shyamasundar [36] presented a comparative analysis of different identity-based trust-management approaches that integrate technology with other factors. Urquhart [37] described the biggest issue in cloud computing as trust, stating that customers and service providers need more trust because of the dynamic nature of cloud computing. Abdul-Rahman and Hailes [38] suggested reputation as the expectation of agent behavior based on information or observations about the agent's past behavior. Shyamsundar and Patil [39] explained the design and implementation of the delegation system and role-based authorization.

Paul [40] proposed the trust model based on quality of service (QoS). This model evaluates the CSPs based on QoS (availability, reliability, turnaround, efficiency, and data integrity). Gholami and Arani [41] presented the proposed turnaround trust model, which is the development of the trust model based on QoS [40] to assist cloud-service clients (CSCs) to select the trusted CSPs. This model is based on the QoS (availability, reliability, data integrity, and response time) and the implementation speed. Tan et al. [42] proposed a trust model based on the SLA and behavioral evaluation. They divided the proposed model into three parts: trust selection, implicit factors, and dynamic trust. Trust selection was based on six parameters (availability, reliability, integrity, average degree of fulfillment, similarity, and integrated value [Intevalue]).

Jin et al. [43] proposed the Stadam model; it is the SLA trust model based on anomaly detection and multi-cloud. In Stadam, the cloud consumer uses multiple providers simultaneously. By utilizing anomaly detection algorithms to check whether the providers fulfill

the SLA, the providers' trust value will update, and the number of requests that are sent to the provider will change.

2.2.2.3. Characteristics of Trust

Despite the diversity among the existing trust definitions and despite the lack of a precise definition in the literature, a large convergence exists about which properties the trust concept satisfies. The most important trust characteristics are reported in the following section; these guidelines are important for modeling trust [31,35]:

- **Trust is multidimensional:** Here, trust is an oriented relationship between the trustor and the trustee. In addition, a trustor could trust a trustee for a certain purpose but not for another purpose.
- **Trust is subjective:** Trust is a personal opinion, wherein a trustor's evaluation of the trustee can be entirely different from some other trustor's evaluation. According to Grandison and Sloman [44], trust is reflected as a subjective and personal phenomenon based on a variety of factors or evidence, some of which may be more powerful than others.
- **Trust can be measured:** Trust values can be used to represent the different degrees of confidence that a trustor may have in trustee. Trust is measurable also provides the basis for modeling trust and evaluating computations.
- **Trust depends on history:** Which means that past experience can affect the current trust level.
- **Trust is dynamic:** With time, confidence usually changes non-monotonically. It can be periodically refreshed or revoked, and must be able to adapt to the changing circumstances in which the trust decision is made. Confidence is sensitive to many

factors, events, or context changes. Solutions should consider the notion of learning and reasoning to handle this dynamic property of trust. A sophisticated trust-management approach is required for the dynamic adaptation of the trust relationship between two entities.

- **Trust is conditionally transferable:** Trust information may be transmitted or received through a chain of recommendations.
- **Trust is a composition of several attributes:** "Trust is really a composition of many different attributes: reliability, dependability, honesty, truthfulness, security, competence, and timeliness, which may have to be considered depending on the environment in which trust is being specified" [44]. Compositionality is an important feature to make trust calculations.

2.3. Trust Based on User Behavior

To a certain extent, the behavioral trust-based access-control mechanism can avoid the above-mentioned vulnerabilities. It can detect and control malicious users in a timely manner through real-time user-behavior monitoring and analysis; for users with correct identities who have admitted to malicious behavior, the system will either reduce their degree of authority or deny access. In other words, trust based on user behavior is a comprehensive evaluation of the user's interaction behavior by utilizing quantified evaluation results to represent the user's trust degree to the cloud as well as to identify risky and malicious users [45]. Therefore, trust based on user behavior (TBUB) provides a valid decision-making basis for the access-control system and improves security in the cloud.

2.3.1. Obtaining Evidence of User-Behavior Trust

Different evidence types should be considered when modeling user behavior in cloud computing and evaluating trust values. To obtain effective evidence, we primarily consider that the gathered evidence is comprehensive, true, and reliable. Trust evidence can be obtained by using software or hardware.

- To determine the number of access times, the times for scanning important ports, and the number of times for operation failures, an intrusion detection system (IDS), such as Tcpdump, is used as long as the network card is set to licentious mode [45].
- There are different types of log and audit trails, such as application log, system log, network management, and audit recording [46]. Audit trails can record the IP source as well as the destination address's (user packet) operation time, duration, and type.
- Bandwidth [47] is a network-flow detection tool which is used to gather performance evidence by getting Transmission control protocol (TCP), Internet Control Message Protocol(ICMP), Hypertext Transfer Protocol(HTTP), User Datagram Protocol(UDP), Virtual private network (VPN), and Peer-to-peer(P2P) data flow based on the IP.
- Cisco's NetFlow Monitor [48] tool is used to collect security and performance evidence, such as real-time monitoring of the data flow, the number of accesses using illegal connections, scanning sensitivity, and important ports.
- Network-management software is based on a standardized protocol, such as simple network management (SNMP), remote monitoring (RMON), or Cisco Works Software [49].
- Security products such as firewalls and access-control systems can capture various evidence.

- Different hardware, such as genius hard probes and NetScout, can gather the evidence directly [50].

2.3.2. Principles for Evaluating User Behavior

In this section, we present the principles that should be considered while modeling user behavior for cloud computing [6, 51]:

- **Principle 1: Expired user behavior should not be considered;** when the user stopped accessing the cloud or has not accessed it recently then the behavior records are out of date. Thus, the user should then be evaluated as a strange user.
- **Principle 2: Recent user behaviors affect the trust value;** new behavior must be more important and affect the trust evaluation more than long-term behavior because, with trust calculations, we consider the most recent behavior.
- **Principle 3:** Abnormal behavior plays an important role in trust evaluation than traditional behavior.
- **Principle 4: Trust evaluation is based on a large amount of user-behavior data; t**he creditability of the trust value is based on a large amount of user-behavior evidence. The evidence in the cloud should be large enough to ensure that the result is stable. If the amount is small, then the results are not representative and are unstable.
- **Principle 5: Slow-rise strategy is to prevent fraud risk in the trust evaluation;** this strategy is based on a large number interaction with cloud to achieve accurate trust values. This principle prevents users from gaining a high trust value when they have a small number of interactions.

- **Principle 6: Punish non-trusted user is based on rapid-decline strategy;** this strategy punishes users when abnormal behavior is detected. Punishment quickly decreases the trust value.
- **Principle 7:** The trust value will decrease whenever repeated malicious behaviors have occurred; repeated malicious behavior decreases the trust value more rapidly than the first occurrence.
- **Principle 8: Trust evaluation should consider avoid cheating;** because of the trust degree is a collaboration of different trust types, each type of trust must have weight in order to avoid too much influence being received from indirect trust .

2.4. Related Work

Evaluating user behavior in cloud computing has been investigated with several research papers to monitor user behavior and to improve the security for cloud computing. Different evidence, such as security, reliability, performance, and operation, exists. Each model has used several evidence types to evaluate user behavior; some of them use security evidence while others use reliability evidence. In addition, several principles have been followed for most of the research papers. This section presents the existing user-behavior models, divided into three sections: evidence of user-behavior trust, the principles covered by existing user-behavior models, and summary tables.

2.4.1. User-Behavior Evidence

To calculate trust values for the users, different types of user-behavior evidence must be collected to investigate the users' behavior in the cloud. The following sections present different evidence that has been used with the existing models.

2.4.1.1. Security Evidence

Security evidence presents the cloud-service user's security characteristics. This evidence, such as users carrying viruses or scanning important ports during their access sessions, is recorded in the user's log files. Security evidence is important when a system monitors user behavior to prevent damage from occurring to cloud services and resources. Table 1 presents several types of evidence that have been used for different research papers.

Table 1. Examples of security evidence.

ID	Evidence Item	Explanation	Resource
SI.1	Scan important resource ports	Does the user scan important ports on the cloud?	[6,52, 53]
SI.2	Carry virus	Does the user carry viruses during the access session?	[6, 53, 54]
SI.3	Illegal connection	Does the user gain access from an illegal connection? The definition of an illegal connection is based on the system requirements.	[6, 52, 55, 56, 57]
SI.4	Input security-sensitive keywords	Does the user input security-sensitive keywords? Sensitive keywords are based on the system requirements.	[6, 52, 53,55]
SI.5	Use proxy	Does the user utilize a proxy?	[52,53]
SI.6	Access other user accounts	Does the user access other users' accounts?	[5]
SI.7	Delete other user folders	Does the user delete other users' folders?	[5]
SI.8	Create a file/folder in other users' accounts	Does the user create files/folders in other users' accounts?	[5]
SI.9	Modify other users' data files	Does the user modify other users' data files?	[5]

2.4.1.2. Performance Evidence

Performance evidence presents cloud-service users' performance characteristics. This evidence, such as CPU occupancy rate and memory occupancy rate, is recorded in a user's log files. Performance evidence is important when a system tracks user behavior to prevent any

damage to the cloud's services and resources. Users with low performance metrics can throttle resources, preventing usage by other people. Table 2 presents several types of evidence that have been used for various research papers.

Table 2. Examples of performance evidence.

ID	Evidence Item	Explanation	Resource
PI.1	CPU occupancy rate	How much does the user typically utilize the CPU?	[6,55, 58]
PI.2	Memory occupancy rate	How much does the user typically utilize the memory?	[58]
PI.3	User's IP transmission delay	How many delay times that the user typically spends sending the packet?	[6, 52, 53, 59, 60]
PI.4	User's bandwidth occupancy rate	What is the user's bandwidth occupancy rate?	[53]
PI.5	User's storage-resource occupancy rate	How much does the user typically utilize the cloud's storage?	[53, 60]
PI.6	User's throughput capacity	How much does the user typically utilize the throughput capacity?	[53, 61]

2.4.1.3. Login Evidence

Login evidence presents the cloud-service users' login characteristics. This evidence, such as login time, login path, or the IP address utilized to access the cloud, is recorded in the users' log files. Login evidence is important when a system tracks user behavior in order to prevent damage to cloud services and resources. Table 3 presents several types of evidence that have been used for various research papers.

Table 3. Examples of login evidence.

ID	Evidence Item	Explanation	Resource
LI.1	Login certification	The username and password are correct.	[56, 57, 62, 63, 64]
LI.2	Login path	What is the source of request address (User-agent, IP)?	[53, 56, 57, 62]
LI.3	IP address	Does the user access the cloud from an unusual IP address?	[56, 57, 59, 62, 63, 64]
LI.4	Login-time preference	Does the user login to the cloud at his/her time preference (usual time)?	[55, 62, 63, 64]
LI.5	Exceed authority attempt	Does the user exceed the number of times login authority?	[26, 52, 61, 65, 66]

2.4.1.4. Reliability Evidence

Reliability evidence presents the cloud-service users' reliability characteristics through interactions with the CSPs. This evidence, such as user data-error rate or user IP packet loss, is recorded in the user's log files. This evidence determines whether users can access the CSPs on a secure network. Then, users could bring viruses to the CSPs and other users of the same CSPs. Table 4 presents several types of evidence that have been used for various research papers.

Table 4. Examples of reliability evidence.

ID	Evidence Item	Explanation	Resource
RI.1	User's data-error rate	How high is the user's data-error rate?	[26,53, 65]
RI.2	User's IP packet loss rate	Does the user typically lose a packet?	[26,55, 59, 60, 61, 62]
RI.3	Connection-establishment failure rate	How is the user's connection? Is there any failure while the user connects to the cloud?	[65,65,66]

2.4.1.5. Operation Evidence

Operation evidence presents the cloud-service users' operation characteristics. This evidence, such as the common functions with which users typically work in the cloud, operation

times, or operation duration, is recorded in the user's log files. Operation evidence is important when a system tracks users' behavior to prevent damage to cloud services and resources. By tracking users' behavior in the cloud, we know when people typically work and the common functions that users typically utilize in the cloud; then, we can compare new behavior with the historical behavior to find a user's trust value. Table 5 presents several types of evidence that have been used for various research papers.

Table 5. Examples of operation evidence.

ID	Evidence Item	Explanation	Resource
OI.1	Operation duration	How long does the user typically work in the cloud?	[62, 63, 64]
OI.2	Common function	What is the common function that the user typically does in the cloud?	[54, 56, 59, 62, 67]
OI.3	Operation time	What is the operation time that the user typically works in the cloud?	[54, 59, 62]
OI.4	Operation frequency	How many times does the user typically work in the cloud?	[54, 62, 63, 64, 68]
OI.5	Operation action (copy and paste, save delete, and print)	What is the operation type that the user typically does in the cloud?	[67]
OI.6	Operation action (upload, retrieve, and download)	What type of document does the user typically upload, retrieve, or download?	[64]

2.4.2. Existing User-Behavior Evaluation Models

Based the literature review, we acknowledge that there are two types of access-control models which are applied in the research papers: the authority-based access model and the role-based access model. We have classified the research papers into two sections based on the type of access-control model.

2.4.2.1. Authority-Based Access Control

The authority-based-trust access-control model assigns users with operating constraints, such as software deployments, data uploads, and different software service levels, across the entire service system. Bendale and Shah [5] developed a SaaS application to monitor user behavior in the private cloud; they proposed a variety of policies to evaluate the users' behavior. In the trust equation, if the user has violated a certain number of policies, the user is malicious. This model can evaluate users and can detect abnormal user behavior and malicious users when the principles are violated. However, this model does not reflect all of the principles and, thus, cannot reflect a user's actual behavior patterns or behavior trust. In addition, this model does not consider the fraud-risk problem and cannot prevent malicious users from receiving high trust values for short-term good behavior.

Tian et al. [6] proposed a new method based on the fuzzy AHP model to calculate the weight of behavior evidence as well as the users' trust values. In addition to improving security defenses, the authors have used multiple detection engines. These engines are used to conduct a comprehensive inspection of suspicious files. This model reflects the access time principle but fails to reflect the remaining principles. Ma and Zhang [52] proposed a new method based on improvements to the AHP method. This model considers the expiration trust record principles by creating three interaction ranges: positive, negative, and uncertain. Behavior in the negative range means that it is far from the current time and should not be included in the trust calculations. Behavior in the uncertain range means that the record is uncertain with the weight for trust calculations. Behavior in the positive range means that it is a new behavior and has a high weight for trust calculations. In addition, this model applies the principles of recent

behavior and trust fraud risk through the slow-rise and punishment strategies. However, this model fails to consider the repeat abnormal-behavior principle.

Yang et al. [53] proposed a trust evaluation model for nodes in wireless sensor networks (WSNs) based on a fuzzy analytic network process (FANP) method. This model simply and clearly calculates the users' trust values and is more accurate than the ANP method. However, this model does not reflect any of the evaluation principles. Jun-Jian and Li-Qin [55] proposed a Dynamic-trust evaluation model to evaluate users' behavior in cloud computing by combining two methods: entropy and the analytic hierarchy process (AHP) model. The entropy method acquires an objective weight. However, the AHP method acquires a subjective weight. The advantage with Dynamic-trust evaluation model is that it can balance between objective and subjective weights to calculate the users' trust values. The model also calculates which users have consumed the largest amount of resources. This model considers a large number of evidence principles, and recent behavior has a large influence on trust values. However, this model has some drawbacks. It does not consider the expiration of trust records, repeated abnormal behavior, or recent behavioral changes. In addition, this model does not consider the fraud-risk problem. Malicious users who obtain a high trust value in a short-term period cannot be prevented from using this system.

Berrached and Korvin [56] proposed a fuzzy algorithm for reinforcing access control based on the history of a user's behavior. This model uses different evidence to evaluate user and to compute the amount of damage that the cloud can accept. However, this model does not reflect the rest of the evaluation principles. Jaiganesh et al. [58] proposed a system which used fuzzy adoptive resonance theory (FART) and neuro fuzzy techniques. With the fuzzy ART technique, they used memory, giga floating operation per second (GELOPS), and disk space for

each virtual client as the input factors. Then, they used unsupervised learning methods to train and to test virtual clients. In summary, this system has two steps: classifying the user's behavior and the user's learning methods, and then evaluating the user's behavior based on the neuro fuzzy systems. This model can classify users into four categories, secure, vulnerable, modified, and anomaly, based on the usage of resources (memory, GELOPS, and disk space), meaning that this system can distinguish between secure and anonymous users. This model's drawback is the failure to reflect the evaluation principles.

Junfeng and Xun [59] proposed a cloud-based user-behavior authentication model which utilizes multi-partite graphs. This model has three layers: the user-behavior evidence layer, building behavior multi-partite graphs, and the behavior-authentication layer. The advantage with this model is the combination of AHP and graph theory. Moreover, the authors identity re-certification and risk game to identify malicious users' cloud services more accurately and efficiently as well as to improve security. In addition, this model can distinguish between malicious and risk users. The malicious users' behavior is abnormal most of the time while the risk users' behavior is only abnormal some of the time. Finally, this model reflects a number of evidence principles. This model's drawback is similar to the model proposed by Jun-Jian and Li-Qin [55].

Liqin et al. [60] proposed an expression to compute trust values by considering self-adaptive algorithms in order to determine the number of interactions between cloud providers and users and to exclude the expiration records. In addition, the authors apply the slow-rise method to prevent attackers from attaining high trust values. On the other hand, this model does not reflect the principles of recent user behavior or repeat abnormal behavior. Yang and Yu [62] proposed a model based on multi-level, fuzzy comprehensive evaluation that combines

quantitative and qualitative evaluation models. The authors used AHP methods and fuzzy comprehensive evaluation (FCE). This model considers the number of evidence principles. This model does not consider the rest of the principles. Reena et al. [63] proposed a system with two technologies. First, user-behavior profiling computes the users' trust values. The user-profiling technique is based on how, when, and how much users access information. The second technique is decoy technology, which is used to download decoy files, instead of genuine files, to an untrusted user. This model can detect abnormal user access and can create decoy files by scrambling the contents of genuine files. This model only reflects the number of access times. Chen et al. [64] proposed a trust evaluation model based on the users' behavior data. This model outlines a set of cloud users' trusted behaviors from the data and sets a weight for each behavior category; that weight is then used to calculate direct trust. In addition, the authors calculate trust recommendations based on the interactions between one user and other cloud users. Next, by giving the historical trust value, the authors calculate the comprehensive trust which is based on direct trust, recommendation trust, and historical trust. This model reflects the following principles: expiration trust record, the number of access times, punishment, and synergy cheating. However, this model fails to consider the repeat abnormal-behavior principle or the slow-rise model.

Lin et al. [65] proposed a mutual trust-based access-control (MTBAC) model. This model has two parts. The first one evaluates the users' behavior using AHP. The second one evaluates the CSPs. According to the user-behavior trust value and the CSPs creditability, MTBAC assigns various users to multiple available CSPs. This model used the AHP method and recommendation trust to solve trust uncertainty problems. This model reflects a number of evidence principles. However, this model fails to consider the rest of the principles.

Mohsenzadeh et al. [66] proposed a model based on fuzzy mathematics theory in cloud computing. By using fuzzy mathematics theory, the trust evaluation's subjectivity is reduced. This model combines direct and indirect trust to calculate the users' trust values. The direct trust comes from the local domain and recommendations from the same cloud provider, but from a different domain. Indirect trust comes from other CSPs. In order to prevent a high influence for indirect trust, the authors assigned different weights for trust values from direct and indirect trust. The model is capable of preventing synergies by using the notion of a trust domain. According to the degree of trust for CSPs organizations, distinct weights are assigned to recommended trust and indirect trust in order to avoid excessive external effects. This model's drawback is the failure to reflect the evaluation principles.

Alguliev and Abdullaeva [67] proposed a system to detect masqueraders in the cloud-computing environment. This system has two phases: creating user profiles and detecting abnormal behavior. The creating phase consists of two components. In the first phase, the user's event log is recorded, and feature extraction occurs. In the profile-creating phase, three values are used (expectation, Ex; entropy, En; and excess entropy). In the detection phase, the cosine similarity method is used to compare the normal behavior with new behavior. The collaborative filtering method evaluates any deviation from normal behavior. This model is simple and can detect masquerader users. This model's drawbacks include that it does not consider the evaluation principles to compute trust. Kalaskar et al. [68] proposed a system that combines two technology user-profiling technologies to monitor users' behavior and then to distinguish between real and fake users. User-profile technology is based on multiple evidence mentioned in the evidence section. In addition, the authors used decoy technology to send bogus data to fake

users. This model considers the number of access times but fails to apply the rest of the principles.

Xiaoxue et al. [69] proposed the reward and punishment trust model (RPTM) to calculate the users' trust values. This model is based on recommendations from other users as well as the user's historical transactions. The RPTM applies recent behavior principles and trust fraud risks through the slow-rise and punishment strategy. This model effectively differentiates between legal and malicious users. However, this model fails to consider the expiration principle and repeat abnormal behavior.

2.4.2.2. Role-Based Dynamic-Access Control

The role-based trust access-control model assigns trust-checked users with predefined roles, rather than service degrees. Thus, this model assigns users with operating constraints for a particular object, such as reading, writing, revising documents or data, or configuring virtual machines. Banyal et al. [26] proposed the dynamic trust-based access-control (DTBAC) model to prevent malicious users from accessing the cloud-computing environment. This model can identify malicious users and quickly prevent them from accessing the cloud server. In addition, the DTBAC has succeeded in considering the principles for the number of times accessed and the fraud-risk problem. However, the DTBAC does not reflect the principles for the expiration trust record, recent behavior, and repeat abnormal behavior.

Jing et al. [54] proposed the user-behavior assessment-based dynamic access-control (UBADAC) model. This model has three parts: calculating user-behavior risk values, based on threat behavior; calculating user trust values, based on the user behavior's risk value; and mapping user trust values, with permission. This value determines the access rights for cloud resources. This model can calculate the risk value for user behavior based on the asset value,

vulnerability degree, and the threat for each resource in the cloud. The model then calculates the user trust values based on the risk values. This model considers some evaluation principles, such as time influence and repeated abnormal behavior principles. However, the model does not consider the expiration trust record, recent behavior, or the fraud-risk problem (slow-rise and punishment).

Yang et al. [57] proposed a model that incorporates a role-based access-control model with user-behavior trust. They proposed multiple contexts to evaluate the users' behavior. This model can provide scalable and flexible authorization strategies, and it defends multiple contexts for trust evaluation as well as different trust-evaluation methods. The drawback is that the model does not reflect any of the evaluation principles and is too complex to practice in the cloud-computing environment.

Deng and Zhou [61] proposed the flexible role-based access control (FRBAC) model. In this model, they use direct trust between the cloud service client (CSC) and the CSPs, based on user behavior. In addition, they use the recommendation trust from other CSPs nodes. By combining direct and recommendation trust, the model produces a user trust value. The FRBAC model uses the additive-increase, multiple-decrease (AIMD) algorithm to punish malicious users. However, the model does not reflect the principles of expiration trust record, recent behavior, repeat abnormal behavior, or slow rise. In addition, this model does not prevent synergy cheating for recommendation trust.

2.4.3. Summary of Related Work

In summary, according to this review, we have defined tables 6, 7, and 8to evaluate each model based on the applicable trust type, access-control model, principle, and evidence.

Table 6. Summary of trust types and the access-control model.

Reference Number	Trust Type				Access-Control Model	
	Direct	History	Indirect Trust /Users	Indirect Trust/ CSPs	Authority	Dynamic Role
5	•				•	
6		•			•	
26	•			•		•
52	•	•		•	•	
53		•			•	
54	•	•				•
55	•	•			•	
56	•	•			•	
57	•					•
58	•				•	
59	•				•	
60	•			•	•	
61	•		•	•		•
62	•				•	
63	•	•			•	
64	•	•	•		•	
65	•				•	
66	•			•	•	
67	•				•	
68	•				•	
69	•	•	•		•	

Table 7. Summary of principles.

Reference Number	Expired Behavior	Evidence	Recent User Behavior	Repeated Abnormal Behavior	Fraud Risk: Slow-Rise	Punishment	Synergy Cheating
5		•					
6		•					
26		•			•	•	
52	•	•	•		•	•	
53							
54				•			
55		•	•				
56		•					
57							
58							
59		•					
60	•				•	•	
61		•				•	
62		•					
63							
64		•				•	•
65		•					
66							•
67							
68		•					
69		•	•		•	•	•

Table 8. Summary of evidence.

Reference Number	Security	Performance	Login	Operation
5				•
6	•	•		
26			•	
52	•	•	•	
53	•	•	•	
54	•			•
55	•	•		
56	•		•	•
57	•		•	
58		•		
59		•	•	•
60			•	
61		•		
62			•	•
63			•	•
64			•	•
65			•	
66			•	
67				•
68				•
69				

According to Table 6, no existing model considers all trust types. To build an efficient model, we must consider all trust types instead of only focusing on the direct and historical interactions between the user and the cloud. The drawbacks of existing models lead us to the proposed model which considers direct trust, history trust, and indirect trust to calculate a user's trust value. According to Table 7, no single model considers all the principles, leading us to propose a model that considers all the principles to evaluate the users and to calculate their trust values. According to Table 8, no single model considers both the login location and time to determine if a user is malicious. For example, we can examine the login location and time between the past login and the current login: if a user logs in at 4 pm in India and then at 7 pm in the United States, that user is malicious. In addition, no model applies all evidence types to evaluate and to monitor users on the cloud, leading us to propose a model that considers all evidence types.

CHAPTER 3. MODEL 1: FRAMEWORK FOR MONITORING THE USER's BEHAVIOR AND COMPUTING THE USER'S TRUST

3.1. Introduction

Traditional access control, simple methods for virus detection, and intrusion detection are unable to manage variety of malicious and network attacks [70]. The number of users might get hacked because of limitation in basic security protection. To implement a secure, reliable, and safe cloud-computing environment, we need to consider the trust issue. A trusted cloud is guaranteed to be safe from user terminals; combined with the concept of a trusted network [71], it evaluates, forecasts, monitors, and manages the user's behavior to eliminate malicious data-center attacks which are performed by unwanted cloud users and hackers; as a result, there is improved cloud security.

The enhancement with the FMUBCT is detecting abnormal user behavior by creating a user-behavior history pattern. In addition, no model considers all the user-behavior evaluation principles that were described in Section 2.4.1. Thus, the FMUBCT includes all the evaluation principles to calculate a user's trust value. Some principles have been performed via a sliding-windows algorithm. Moreover, the FMUBCT successfully considers three types of trust—direct, history ,and comprehensive —to calculate a user's credibility. This part is achieved by implementing a fuzzy logic approach to provide more intelligent access control. The evaluation is performed based on monitoring the user's actions and behavior in order to determine the user's genuineness and trustworthiness. Furthermore, the FMUBCT combines two techniques: fuzzy logic and sliding windows.

The remainder of this chapter is organized as follows. Section 3.2 describes the Background while Section 3.3 presents the user-behavior trust model based on fuzzy logic.

Section 3.3.1 details the model's Logic Structure, and Section 3.3.2 explains the FMUBCT's phases and the algorithms used in each phase. Furthermore, Section 3.3.3 demonstrates how the user-behavior trust model based on fuzzy logic reflects the trust evaluation principles. Finally, Section 3.4 concludes the chapter by summarizing the proposed model.

3.2. Background

3.2.1. Fuzzy Logic Approach

In 1965, Lotfi Zadeh, a professor at the University of California at Berkeley, invented fuzzy logic. Fuzzy logic is an extension of Boolean logic and serves as a form of logic or probabilistic logic that is highly valued; it utilizes reasoning that is approximate, rather than fixed and exact. Fuzzy logic is extended to handle the concept of partial truth, where the truth value can range from completely true to false. In addition, the range degree could be between 0 and 1. In an environment with imprecise, uncertain, and incomplete information, fuzzy logic can reason and make rational decisions. Because fuzzy logic allows vague human assessments to be included with the computer problems, it has been widely used for developing pattern recognition, identification, control systems, optimization, and intelligent decision-making [72,73, 74, 75].

For decades, fuzzy logic has been successfully applied in many fields, such as industrial manufacturing, automobile production, automatic control, hospitals, banks, libraries, and academic education. In 1976, Blue Circle Cement in Denmark, developed an industrial application for cement kiln control [76]. Dattathreya et al. [77] proposed an intelligent system based on fuzzy logic to detect and eliminate potential fires in hybrid electric-vehicle engines and battery compartments. Moreover, Cao and Liu [78] used fuzzy logic to recognize human action, and Nguyen et al. [79] proposed a weight-estimation method using fuzzy logic to improve security for co-authentication in the Android platform. Because fuzzy logic has been employed

in several fields, we see its adoption in much of the current scientific research for several disciplines.

Implementing fuzzy logic requires three steps. The first step is fuzzification which converts crisp data into fuzzy data and membership function. The second task is the fuzzy-inference process which combines the membership function with the fuzzy rule to define the output fuzzy value. The final step is defuzzification which transforms an output fuzzy value to a crisp value.

3.2. Proposed Model

3.3.1. Logic Structure

We proposed the FMUBCT to improve the security for cloud computing by enhancing traditional access control. This improvement is achieved by checking the user's trust value before the authorized user can access the cloud. In addition, the model can monitor the user's behavior while the user interacts with the cloud in order to avoid malicious attacks from an abnormal user. The FMUBCT consists of eight primary components: the authentication, authorization, behavior-monitoring, user-profiling, comparison, trust computation, trust management, and fuzzy logic modules. Figure 2 illustrates the model's flow chart.

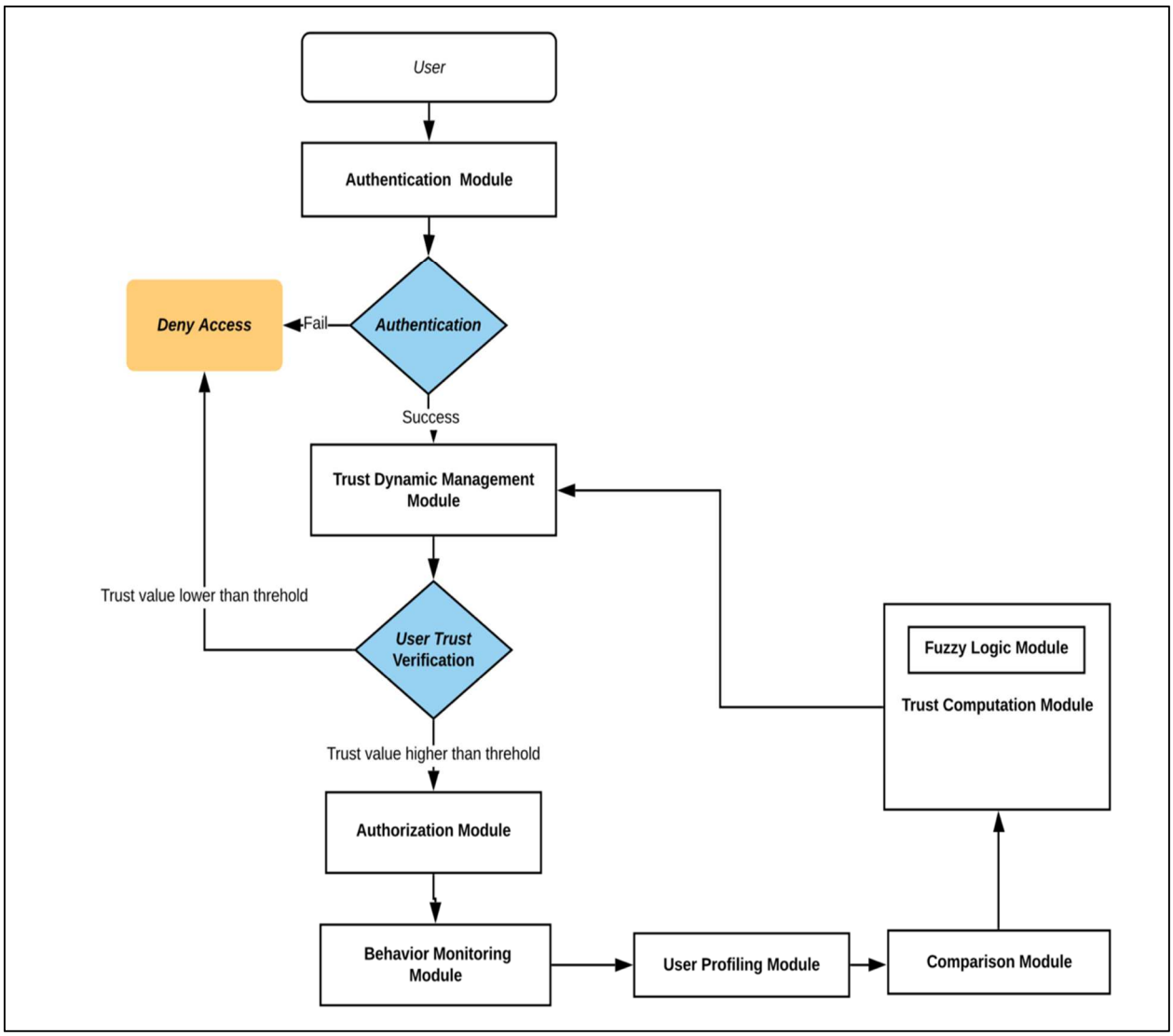

Figure 2. Flow chart for the model.

The function for each FMUBCT module is as follows:

- **Authentication Module:** verifying the user's identity when the user requests access to the cloud.
- **Authorization Module:** granting the user access to the operation and resources based on the user's trust value.
-

- **Behavior-Monitoring Module:** monitoring of the user's behavior, in real time, throughout the interaction process. Collect the required behavior evidence; standardize the collected evidence with data pre-processing; and store the database of the user's behavior evidence.
- **User-Profiling Module:** creating a user-behavior pattern based on the user's history.
- **Comparison Module:** comparing the current user's behavior with user's history pattern. This comparison to obtained deviation cases, between current behavior and history pattern, are considering as malicious attacks.
- **Trust Computation Module:** applying the trust evaluation equation to the user-behavior evidence in order to calculate direct trust and history trust. Afterward, send the values to the fuzzy logic module.
- **Fuzzy Logic Module:** applying the fuzzy logic approach to compare two types of trust; then, update the user's trust value in the trust database.
- **Trust Dynamic Management Module**: querying the user's trust value from the trust database when a user requests access to the cloud. In addition, verifying the updated user-trust value dynamically modifies the user's degree of service and operating authority.

The following steps explain the FMUBCT's working procedure:

- A user requests access to the cloud by entering his/her identity; then, the cloud service provider verifies the user's identity. If the user fails the authentication, the CSPs will deny access.
- If the user passes the authentication modules, the trust dynamic management will query the user's trust value from the trust database. If the user's trust value is less than the minimum for the threshold user, the CSPs will deny access.

- If the user's trust value exceeds the threshold, the authorization module will grant the user access to the operation and resources. Figure 3 illustrates the flow chart for step 1 through 3.

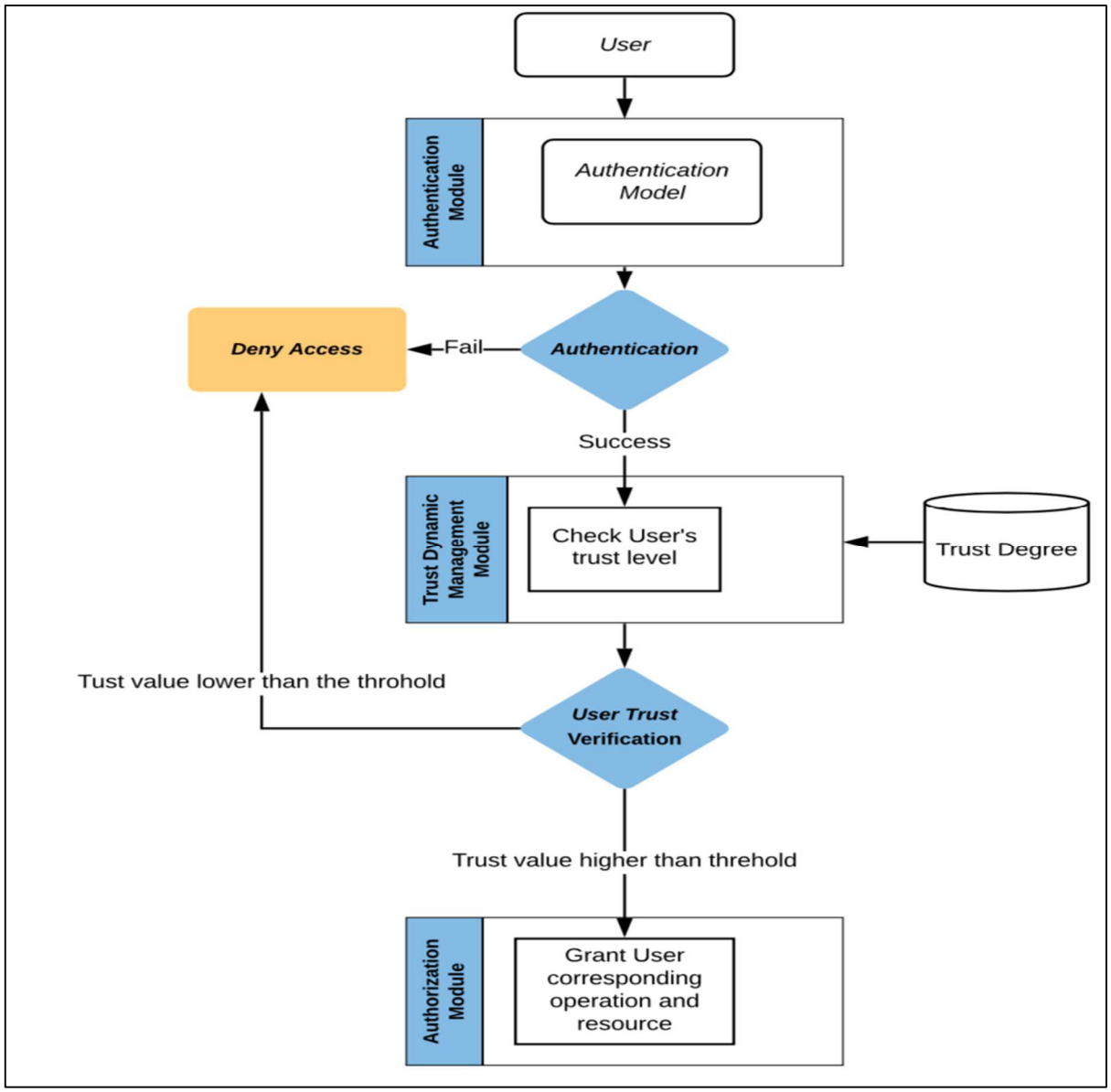

Figure 3. Flow chart for steps 1 through 3.

- The behavior-monitoring module checks the user's behavior during the access process. The module collects evidence and stores it in the user-behavior database. To reduce the computation, the user-profile module works every two weeks to produce a user's

behavior pattern. Figure 4 illustrates the flow chart for the behavior- monitoring and user-profiling modules.

- Meanwhile, the current behavior is recorded and sent to the comparison module, which begins work by comparing the user's new behavior with the user's history pattern. Figure 5 illustrates the flow chart for the comparison module.
- The comparison result is sent to trust computation module to calculate the direct trust. Meanwhile, the trust computation module calculates the user's history trust value. Figure 6 illustrates the flow chart for the trust computation module.
- Based on knowledge about the direct trust and history trust, the fuzzy logic module works to infer the user's trust value and to store that value in the trust database. Figure 7 illustrates the flow chart for the fuzzy logic module.
- The trust dynamic management module notices the updated user-trust value and, then, dynamically modifies the user's degree of service and operating authority.
- The update for the user's trust value is fed back to the user.

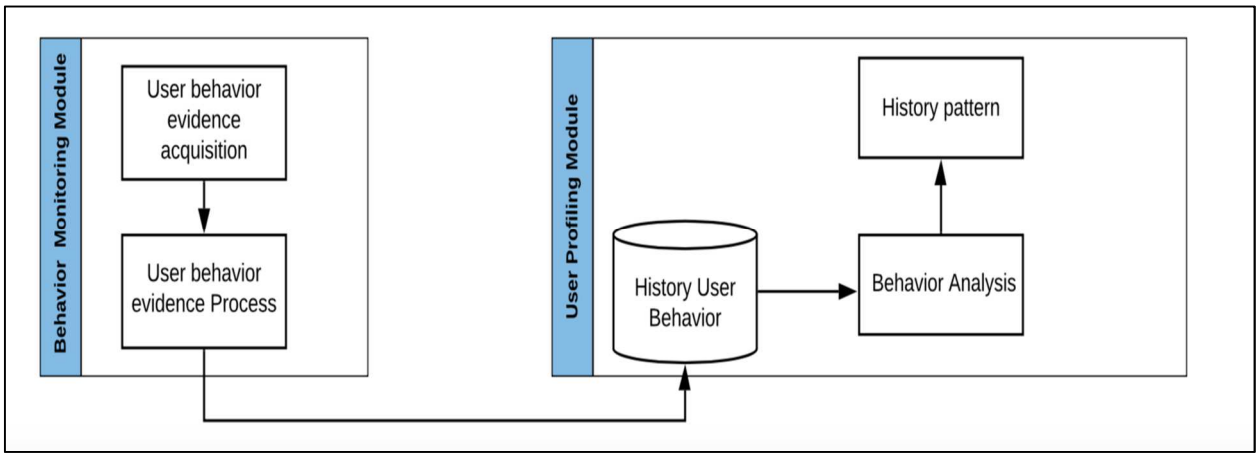

Figure 4. Flow chart for the behavior-monitoring and user-profiling modules.

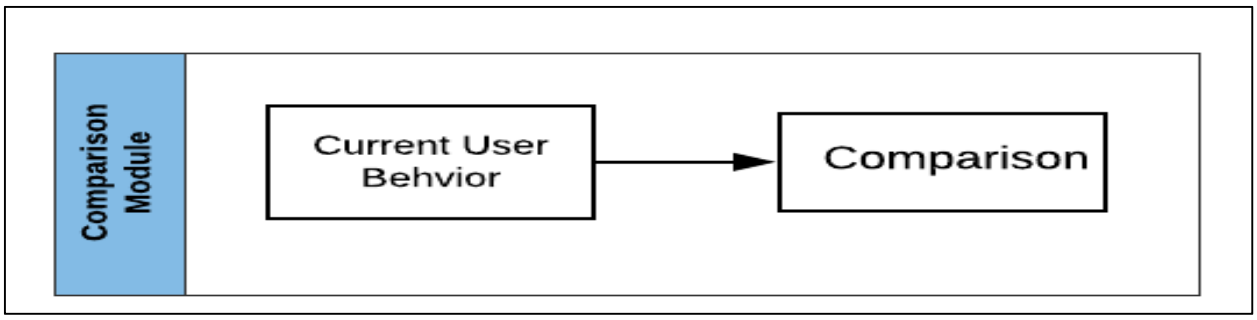

Figure 5. Flow chart for the comparison module.

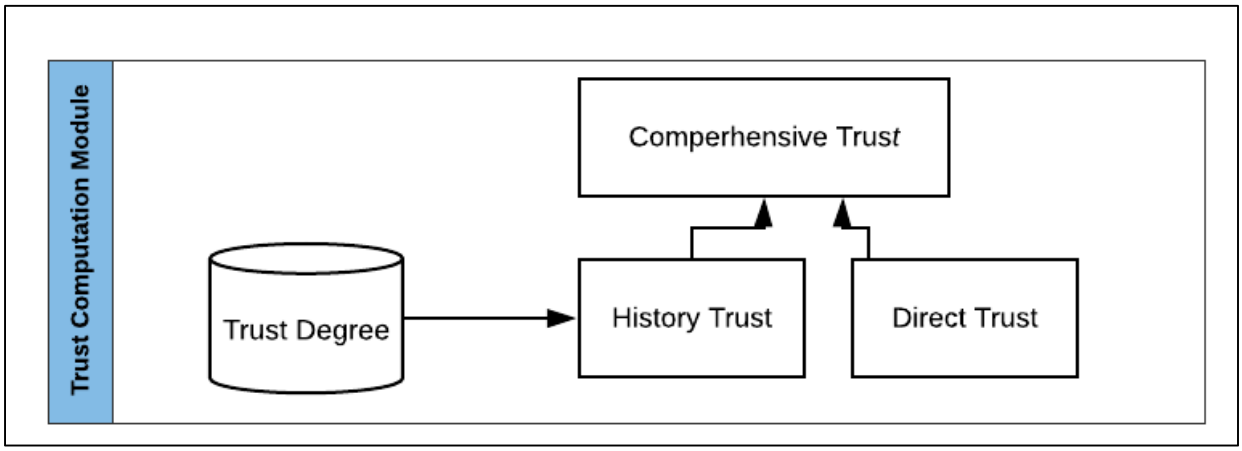

Figure 6. Flow chart for the trust computation module.

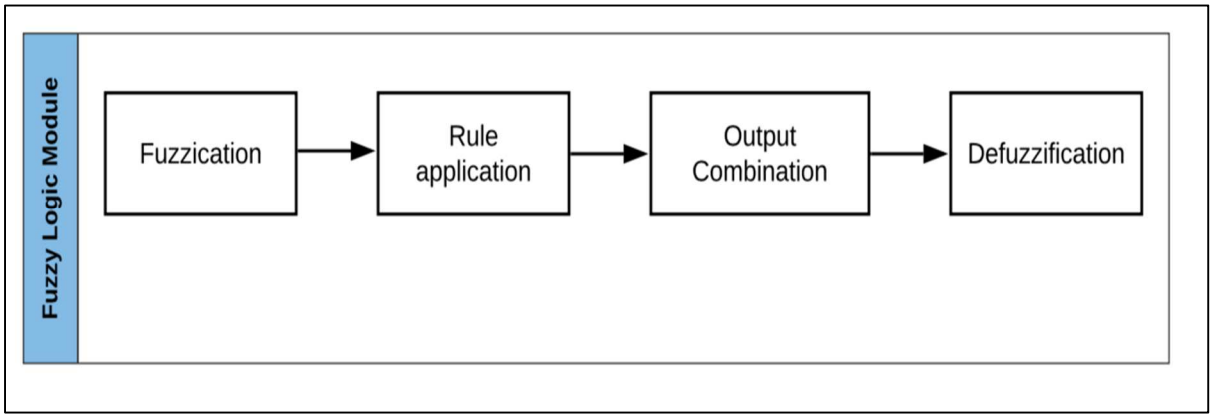

Figure 7. Flow chart for the fuzzy logic module.

3.3.2. The FMUBCT Module's Algorithms

In this section, we present the algorithms and equations used for each module. In Section 3.1, we mentioned that the FMUBCT contains eight modules. We now explain the major ones: the user-profiling, comparison, trust computation, and fuzzy logic modules.

3.3.2.1. User-Profiling Module

To recognize and to detect a user's anomalous behavior, a user-profile pattern must be created to distinguish a user's normal behavior from abnormal behavior. In this section, we describe the algorithm underlying our user-profiling modules and discuss how the profiles are designed based on the user's interest data. In the FMUBCT, we consider three evidence types: security, login, and operation. The user's behavioral evidence can be directly obtained from the detection tools, which were discussed in Section 2.3.2. Thus, we extract the evidence from the software's or hardware's log file.

The user can log in to the cloud via different devices and any time. Because the username and password can be stolen, we should consider the history habits, such as the usual IP address and the preferred login time. The history habits are helped to discover abnormal login behavior that could result in multiple security issues which can affect the cloud. For the login evidence, we considered two factors:

- The usual IP addresses.
- The user's preferred login time.

Regarding security evidence, the scanning port is one of the most frequent attacks that could occur in the cloud. No natural harm occurs in the cloud; however, the attacker will know the ports' status. Afterward, a distributed denial-of-service attack occurs, which attacks the services available through port scanning [80, 81, 82]. Attackers can identify and interfere with

potential system vulnerabilities. Port-scan attacks occur frequently and are becoming increasingly complex [82].

Carrying a virus while logging into the cloud is one harmful issue that could occur when storing data and can present different problems with the data. These issues could result in the data's reduced integrity and availability [55, 59]. Connecting to the cloud with an illegal connection is a security issue that we should consider in the security evidence. Using an illegal or unsecure network to connect might affect and inject the cloud through the user's identity, utilizing an unsecure connection could lead to the user's identity being stolen by an attacker. Due to the previously stated reasons, we consider the most important factors regarding security evidence:

- The number of times that a user scans an important cloud port.
- The number of times that a user installs a virus in the cloud.
- The number of times that a user connects to the cloud through illegal connections.
- The number of times that a user types any sensitive keywords.

When users utilize the system, they do not operate with the same behavior. Each user exhibits a habit behavior that can help to distinguish between the user's current behavior and history behavior. Regarding the operating evidence, we consider three factors:

- The common access services.
- Common action that the user typically performs on the cloud, such as downloading or uploading documents or user applications.
- The user's maximum duration of service usage.

3.3.2.2. The Analysis of Cloud Users' Behavior Data

Multiple users access the cloud. In this model, we set the total number of cloud users to be N.

$$Cu = \{u_1, u_2, .., u_i\} \quad ,$$

where u_i represents the cloud user;

$$i \in N, N \text{ is a natural number. } 0 < i \leq N \tag{1}$$

In equation 2, each user exhibits a set of behaviors that can be recorded in the cloud.

$$Cu_i = \{Ub_1, \ldots\ldots.. Ub_j\} \quad ,$$

where Ub_j represents the user's behavior;

$$j \in N, N \text{ is a natural number. } 0 < j \leq N \tag{2}$$

Each user's behavior features a set of evidence. This model only focuses on three evidence types.

$$Ub_{ij} = \{E_L, E_S, E_O\} \quad ,$$

where E_L: login evidence,

E_S: security evidence,

E_O: operation evidence (3)

Each evidence has multiples factors; evidence set E is defined as follows:

$$E_{ij} = \{F_1, F_2, .. \ , F_k\} \quad ,$$

$$\text{where } k \in N, N \text{ is a natural number } 0 < k \leq N \tag{4}$$

To simplify the evaluation process, the cloud-user behavior-analysis process is performed using a simple statistical method. We use mathematical statistics to obtain the frequency sets based on each cloud user's data. The ith cloud user, therefore, exhibits frequent behaviors.

The user interaction would continue to increase with the realization of time interactions. These interactions form the basis of creating the user-behavior pattern. Because the evidence's

importance would decrease over time, we decided to use the sliding-window technique to describe the evidence timelines as well as to decrease complexity and to increase the speed for producing the common user behavior. Figure 8 illustrate the timelines for the user interaction.

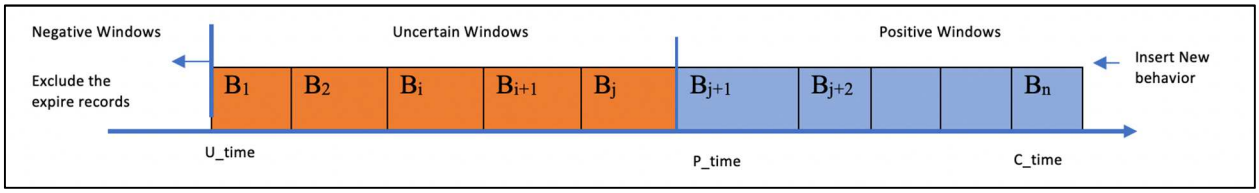

Figure 8. Sliding-window technique for user interaction.

We divided the user's interaction in the cloud into three categories: positive, uncertain, and negative interactions. Therefore, we established three windows: positive, uncertain, and negative windows. Each window size is based on time. For the positive windows, we filled out all interactions from two weeks prior to the current time to the current time. The size of the uncertain windows is based on the two weeks of positive time. The size for the negative windows starts from the time prior to the previous time in the uncertain windows. When the positive windows are full, then interaction shifts to the uncertain windows. Similarly, when the uncertain windows are full, the interaction is squeezed to the negative windows.

In the pattern, we use the positive windows to reflect the principle that recent behaviors play a more important role in trust evaluation. However, to consider the creditability and slow-rise principles, we use uncertain windows to ensure that we have sufficient historical information and to prevent trust deception. Having sufficient interactions means that the interactions exceed the uncertain windows to the positive windows. However, we use negative windows to exclude the expiration records. Because we use binary search, the complexity of the generating history pattern algorithm is O (log n) where is n number of records in positive and uncertain windows.

3.3.2.2. Comparison Module

To calculate the direct trust, we first compare the current behavior with the history pattern that was completed in the user-profiling module. Through equation 5, the module compares the current behavior with the user's history pattern.

$$E_i = \{F_1, F_2, \ldots, F_j\}$$

$$F_{ij} = \begin{cases} 1, & F_{ij} \in F_{ih} \\ 0, & F_{ij} \notin F_{ih} \end{cases}, \quad (5)$$

where E_i is the evidence type and the F_{ih} factor under the evidence represents the user's history pattern. Equation 5 is a crisp representation that inflicts a sharp boundary on a set where 0 indicates abnormal behavior and is not in the historical pattern. However, 1 indicates normal behavior that typically occurs. This module sends the comparison result to the direct trust in the trust computation module. The complexity of the comparison algorithm is O (n) where is n number of user's history patterns.

3.3.2.3. Trust Computation Module

3.3.2.3.1. Direct Trust

Direct trust reflects the current interaction between the user and the cloud. After the direct trust computation, the results are compared with the history trust based on fuzzy logic. The module uses Equations 6-8 to calculate the direct trust.

3.3.2.3.1.1. Computation of Login Evidence

$$\mathbf{T_L = W1 * CVLIP + W2 * CVLT}\text{ , where} \quad (6)$$

- CVLIP is the comparison value of the current login IP address with the history IPs.
- CVLT is the comparison value of the current login time with the history login times.
- Weight is W, for which the total should be 1.

Algorithm 1 accepts the cloud's log file and history pattern as input to calculate login trust.

Algorithm 1. Computation Login Evidence

```
Input: Current Login behavior Ei =IP, LT, History Behavior Pattern HBP
Output: Login Trust Value Ts
while UID==HUID do
    if LIP == HIP then
        CVLIP=1;
    else
        CVLIP=0;
    end
    if LT == HLT then
        CVLT=1;
    else
        CVLT=0;
    end
end
SetWeight();
TL=W1*CVLIP+W2*CVLT;
return TL;
```

3.3.2.3.1.2. Computation of Security Evidence

$$\mathbf{T_S = W1 * SIP + W2 * CV + W3 * IC + W4 * SK}\text{, where} \quad (7)$$

- SIP is the scanning important port.
- CV is the carrying virus.
- IC is the illegal connection.
- SK is the sensitive keyword.
- Weight is W, for which the total should be 1.

Algorithm 2 accepts the cloud's log file to calculate security trust.

Algorithm 2. Computation Security Evidence

```
Input: Current Security behavior Ei =SIP, CV,IC
Output: Security Trust Value Ts
Ts ← 0;
Wi ← 0 − 1;
Wj ← 0 − 1;
while UID==HUID do
    if user scans important port then
        SIP=1;
    else
        SIP=0;
    if user carries virus then
        CV=1;
    else
        CV=0;
    if user uses illegal connection then
        IC=1;
    else
        IC=0;
    if user enters sensitive keyword then
        SK=1;
    else
        SK=0;
SetWeight();
Ts=W1*SIP+W2*CV+W3*IC+W4*SK; return Ts;
```

3.3.2.3.1.3. Computation of Operation Evidence

$$\mathbf{T_O = W1 * CVOS + W2 * CVOA + W3 * CVOD}\text{ , where} \quad (8)$$

- CVOS is the comparison value for the current service with the service history.
- CVOA is the comparison value for the current action (such as download or upload) with the action history.
- CVOD is the compassion value for the current operation's duration with the duration history.
- The weight, W, has a total of 1. The W_i in equations 6, 7, and 8 is based on the more important factor obtaining the highest value. An example of the important factor is in equation 6; when the user is scanning an important port, then SP has more weight than the other factors.

Algorithm 3 accepts the cloud's log file and history pattern as input to calculate operation trust.

Algorithm 3. Computation Operation Evidence

```
Input: Current Operation behavior Ei =OS, OA,OD, History Behavior Pattern HBP
Output: Operation Trust Value To
while UID==HUID do
    if OS == HOS then
        CVOS=1;
    else
        CVOS=0;
    end
    if OA == HOA then
        CVOA=1;
    else
        CVOA=0;
    end
    if OD == HOD then
        CVOD=1;
    else
        CVOD=0;
    end
end
SetWeight();
To=W1*CVOS+W2*CVOA+W3*CVOD ;
return To;
```

3.3.2.3.1.4. Computation of Direct Trust Level

$$\mathbf{T_D = \alpha * T_L + \beta * T_O + \gamma * T_S} \tag{9}$$

Coefficients $\alpha, \beta,$ and γ are applied as weighting for each evidence type. The more important the evidence is to the cloud, the higher weight that it receives. The total for the weights should equal 1. To calculate the weight from equations 6-9, we use Equation 10 as follows:

$$Wi = \begin{cases} Wi = Wj & \text{where } Fi == Fj \\ Wi > Wj & \text{where } Fi > Fj \\ Wi < Wj & \text{where } Fi < Fj \end{cases} \tag{10}$$

3.3.2.3.2. History Trust

To calculate the history trust, after a long period of interactions with the cloud, we calculate the mean of the previous trust values in the positive windows. When a user logs in to

the cloud for the first time with normal behavior, an initial history trust value of 0.5 is assigned. The new user's history trust value is updated using fuzzy logic.

$$T_H \begin{cases} = 0.5\ , & \text{first login to the cloud} \\ = T_C, & \text{based on the fuzzy logic module} \\ = T_{H/}pn, & \text{before the fuzzy module} \end{cases} \quad (11)$$

3.3.2.4. Fuzzy Logic Modules

We use the following methodology to calculate a user's comprehensive trust value:

1. Define the input and output

With the input, T_D, and the mean, T_H, the output of the fuzzy system is the comprehensive trust. The input's membership has four degrees: normal, general, suspicion, and abnormal. The membership degrees for the output are trusted, suspected, general trusted, and untrusted. Figure 9 presents the inputs and output for fuzzy logic module.

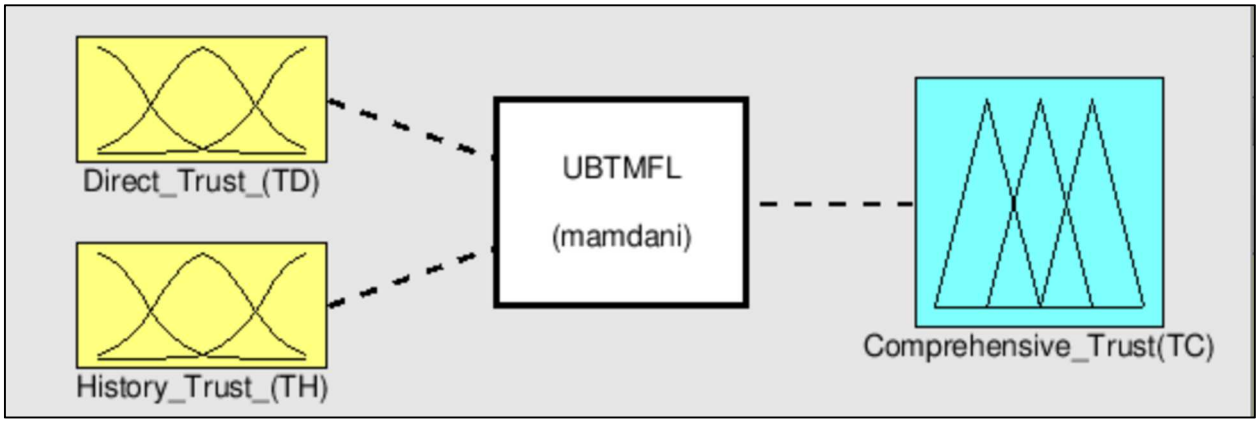

Figure 9. Input and output for the fuzzy logic module.

2. Construction of fuzzification function

We illustrate fuzzification by displaying the membership function for direct trust and history trust using a triangle view of variables, as depicted in Figures 10 and 11, respectively. In addition, Figure 12 illustrates the membership degree for the output. We define the direct and history trust values in Table 9, and the equivalent membership function is displayed in Figures

10 and 11. We define the comprehensive trust value in Table 10, and its equivalent membership function is depicted in Figure 12. Direct trust membership functions.

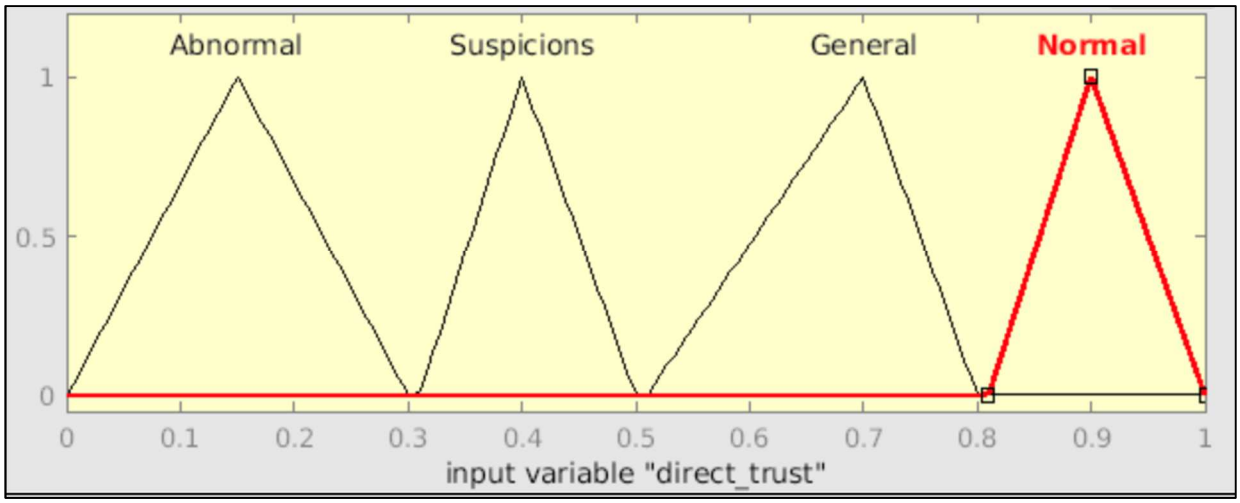

Figure 10. Direct trust membership functions.

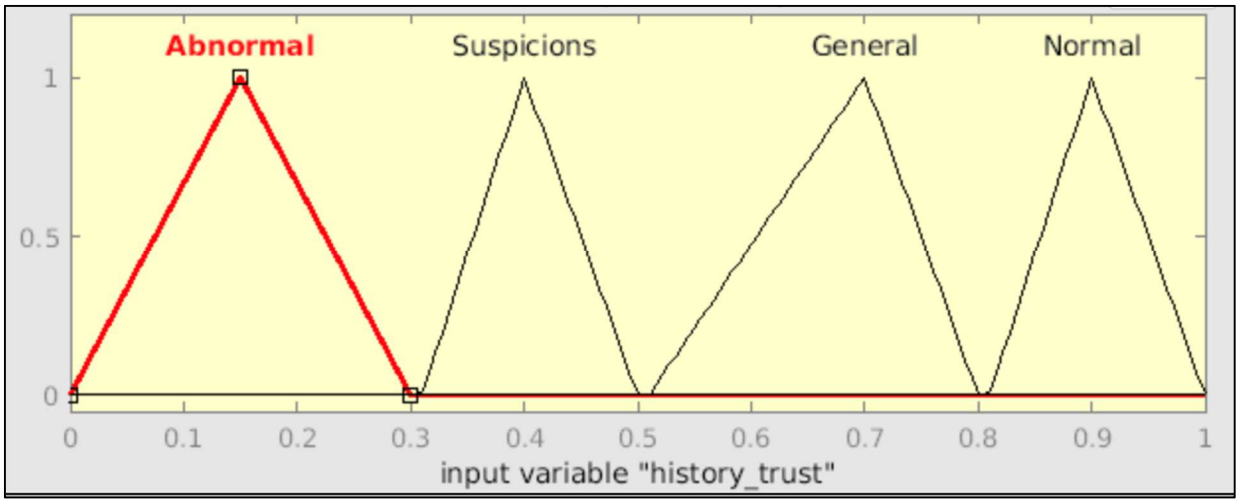

Figure 11. History trust membership functions.

Table 9. Fuzzy direct and history trust value.

Linguistic Direct and History Trust	Range	Fuzzy Number
Abnormal	0-0.30	(0 0.15 0.3)
Suspicions	0.31-0.5	(0.31 0.4 0.5)
General	0.51-0.80	(0.51 0.70 0.80)
Normal	0.81-1	(0.81 0.9 1)

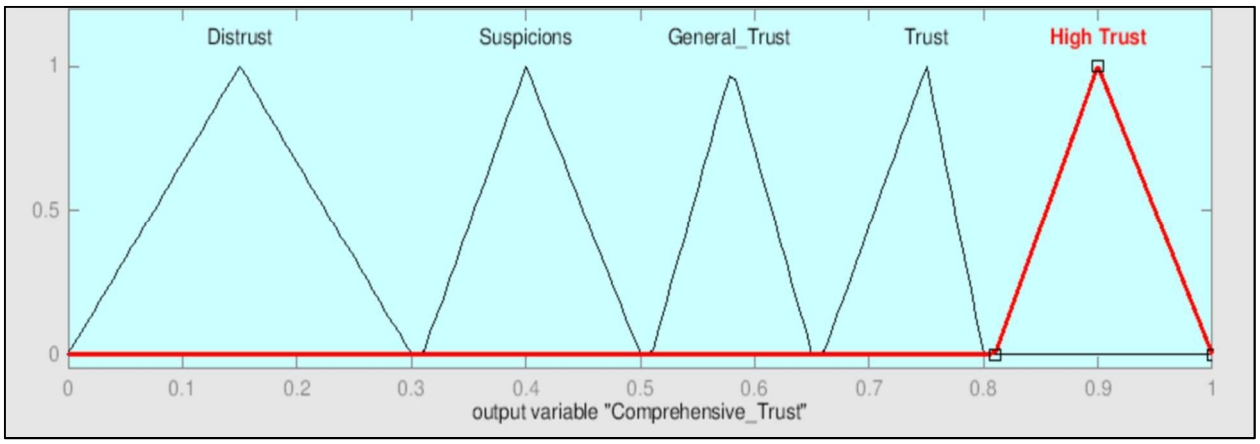

Figure 12. Comprehensive trust membership functions.

Table 10. Fuzzy comprehensive trust value.

Linguistic Comprehensive Trust	Range	Fuzzy Number
Distrust	0-0.3	(0 0.15 0.3)
Suspected	0.3-0.5	(0.31 0.4 0.5)
General Trust	0.51-0.65	(0.51 0.58 0.65)
Trust	0.66-0.8	(0.66 0.75 0.8)
High Trust	0.81-1	(0.81 0.9 1)

3. Defining fuzzy rules base

We employ the fuzzy inference system to generate rules and to control the output value. In Mamdani type, the fuzzy rule is simple and is represented by an IF-THEN relationship. The IF-THEN rule is written as follows:

IF X is a and Y is b and Z is c THEN w,

where X, Y, and Z are the crisp inputs; a, b, and c are the fuzzy clusters to which the inputs may correspond; and w is the output fuzzy cluster used for defuzzification. The possible number of generating rules for each input and output is 16. Table 11 presents the fuzzy rules.

Table 11. The FMUBCT's fuzzy rules.

Rule	TD	TH	Then
1	Normal	Normal	High Trust
2	Normal	Suspicions	General Trust
3	Normal	Abnormal	Suspected
4	Normal	General	Trust
5	General	Normal	General Trust
6	General	Suspicions	Suspected
7	General	Abnormal	Suspected
8	General	General	General Trust
9	Suspicions	Normal	Suspected
10	Suspicions	Suspicions	Suspected
11	Suspicions	Abnormal	Suspected
12	Suspicions	General	Suspected
13	Abnormal	Normal	Distrust
14	Abnormal	Suspicions	Distrust
15	Abnormal	Abnormal	Distrust
16	Abnormal	General	Distrust

4. Evaluating fuzzy rules

The system must use an inference engine to obtain the fuzzy value for the output. In our model, we consider the intersection (the fuzzy AND operation) to be given as follows:

$$\text{Min } \{\mu A(x), \mu B(x)\} \tag{12}$$

To obtaining crisp output, we use the center of gravity (COG) method of defuzzification because COG is one of the most popular methods. Table 12 presents the user's privilege based on the user's trust value.

Table 12. User's privilege.

User Type	Credibility	Privilege
Distrust	0-0.3	Access denied.
Suspected	0.31-0.5	Small quantity of basic services and low authority. High alert for this type of user.
General Trust	0.51-0.60	Basic services and general authority.
Trust	0.61-0.8	Large quantity of cloud services.
High Trust	0.81-1	Core services and superior authority.

3.3.3. Reflecting Trust Evaluation Principles

In this section, we explain our strategies to consider the evaluation principles.

3.3.3.1. Exclude Expired Trust Value Records

When the user stops logging into the cloud for an extended period, the time between the previous trust values and the current trust value is long. We use equation 13 to check for expiration records.

$$\mathrm{ExR} = \mathrm{CT} - \mathrm{LRT} > \mathrm{MT} \ , \tag{13}$$

where

- CT: current time
- LRT: last record time
- MT: maximum time for the model

Because the oldest records will not be squeezed from the positive windows, we designed a strategy to exclude the expiration records. When the model detects that the user has stopped accessing the cloud for an extended time period, the model replaces the record values with 0.5. By applying this strategy, the mean history trust is 0.5, which affects the comprehensive trust value in the fuzzy logic module. For a normal user, the trust value increases slowly by following the slow-rise strategy described in Section 3.3.3.3.

3.3.3.2. Recent User Behavior Affects the Trust Value

In the fuzzy logic module, we consider the recent-behavior principle by assigning the direct trust more weight than the history trust. This principle is important to ensure that the comprehensive trust value reflects the user's current state.

3.3.3.3. Slow-Rise Strategy

To prevent fraud for trust calculations when the user has fewer interactions with the cloud, we designed a slow-rise strategy. We used the sliding-window technique to check the number of interactions. We established three windows: positive, uncertain, and negative. Each window size is based on the number of interactions. To consider the creditability and slow-rise principles, we used uncertain windows to ensure that we have sufficient historical information and to prevent trust deception. the interactions exceed the uncertain windows to the positive windows mean we have sufficient historical information.

Our slow-rise strategy is that, first, we place the user in the general trust case if his/her interactions are in the uncertain windows. When the user first logs in to the cloud and behaves normally based on values from equations 6-9, we replace the value with 0.51, the smallest value of direct trust, because the general-trust fuzzy ranges from 0.51 to 0.65. When the user continues to behave normally in the cloud, we adjust the growth rate for the trust value by using equation 14.

$$T_D = T_D + \rho \quad , \quad \text{where } \rho = (0.01, .1) \tag{14}$$

However, we retain the value from equations 6-9 if the user behaves abnormally in the cloud, meaning that $T_D < 0.5$.

$$T_D = \begin{cases} T_D & , \ T_D < 0.50 \\ T_D + \rho & , \ T_D > 0.51 \end{cases} \tag{15}$$

We use equation 11 to calculate T_H. If the records are located in the uncertain winnows, T_C is calculated based on the fuzzy logic module. Figure 13 presents the timelines of user's interactions.

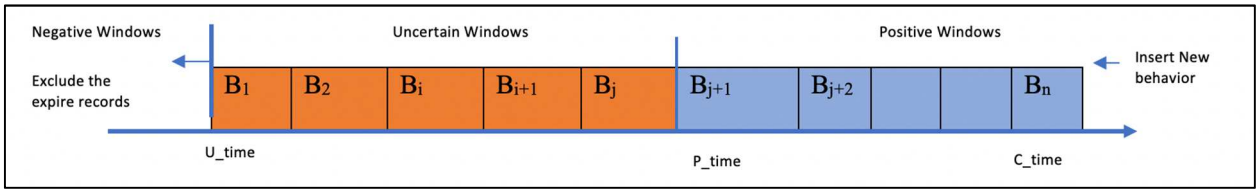

Figure 13. Sliding windows to calculate trust.

3.3.3.4. Punishment (Rapid-Decrease) Strategy and Repeating Malicious Behavior

If the user behaves maliciously ($T_D <= 0.5$), a portion of the recent trust value will be reduced to an untrusted value. Thus, the user's T_C is rapidly decreased to punish the user. Equation 16 is utilized to determine the number of windows that will be punished [51].

$$k = (\min\left[\alpha * \frac{T_{Old}}{T_{new}}\right], TR) \quad , \tag{16}$$

where

- T_{Old} = previous trust value
- T_{new} = newest trust value for the malicious behavior
- TR = number of trusted records

After changing the trust value, the module computes the average for the history trust value. Then, the direct trust and history trust is sent to fuzzy logic module to compute comprehensive trust.

For repeating malicious behavior, we use equation 17 to punish the user and to adjust the trust value to this new value.

$$\text{Punish Trust value} = \gamma * T_{new} \quad , \tag{17}$$

where γ is the penalty coefficient used for the intensity of the punishment: $\gamma = (0,1)$

If the user repeats malicious behavior, the γ will become lower, which decreases the final trust value by multiple penalties with the new trust value. We select the closest n records from the right window to punish the user and adjust them to the punish trust value. Then, we update the trust database with the new values.

3.4. Conclusion

In this chapter, we proposed a user-behavior trust model based on fuzzy logic and the sliding-window technique (FMUBCT) to evaluate user behavior and to detect abnormal user behavior in the cloud. This model used three types of user-behavior evidence: login, security, and operation. The model has three main modules: The first module is creating a user-history pattern based on common and frequent user behaviors. Because the importance of the evidence would decrease over time, we use the sliding-window technique to describe the timelines of the evidence; then, we only consider recent user behaviors to create a user-history pattern. The second module is a comparison to contrast the current user behavior with the history pattern. The third module is trust computation to calculate user's trust. This model considers three types of trust: direct, historical, and comprehensive. We used fuzzy logic to compute the comprehensive trust value. The FMUBCT is simulated, and the results indicate that it can prevent malicious users from accessing cloud service provider whenever the user behaves abnormally; this result is achieved by monitoring user behavior and then calculating the user's trust value. Moreover, this model is able to update user's trust value in a timely manner, leading the cloud service provider to change the user's authority. Finally, the FMUBCT reflects all the identified evaluation principles, ensuring that the user' trust value is accurate.

CHAPTER 4. MODEL 2: FUZZY LOGIC APPROACH BASED ON USER-BEHAVIOR TRUST IN CLOUD SECUIRTY

4.1. Introduction

With cloud computing gaining in popularity and by providing a massive number of services, such as resources and data centers, the number of attacks in the cloud is increasing. Cloud-platform security becomes an important factor for cloud development. Basic security protection, such as traditional access control (TAC), is unable to satisfy the security requirements with the expansion of cloud computing. For example, TAC is not able to prevent a malicious user from accessing the cloud. In this chapter, we propose the fuzzy logic approach based on user-behavior trust (FUBT).

The enhancement for the FUBT model is detecting abnormal user behavior by creating a user-behavior history pattern using the algorithm that was explained in Section 3.3.2.1. In addition, this model considers all the user-behavior evaluation principles described in Section 2.4.1. Thus, FUBT includes all of the evaluation principles to calculate a user's trust value, the same as FMUBCT from Chapter 3. The FUBT model improves the FMUBCT model with three types of performance evidence, login, security, and operation, to evaluate the cloud users. Moreover, we consider four types of trust: direct, history, indirect, and comprehensive. We present a fuzzy approach to calculate the direct and comprehensive trust values. FUBT is flexible and scalable because it considers more evidence to monitor and evaluate user behavior. Finally, the FUBT simulation shows that the model can effectively evaluate the users.

The remainder of this chapter is organized as follows: Section 4.2 presents the fuzzy logic approach based on user-behavior trust in cloud security. Section 4.2.1 details the model's Logic Structure, and Section 4.2.2 explains the FUBT's phases and the algorithms used for each

phase. Furthermore, Section 4.2.3 demonstrates how the FUBT model reflects the trust evaluation principles. Finally, Section 4.3 concludes the chapter by summarizing the proposed model.

4.2. Proposed Model

4.2.1. Logic Structure

We proposed the FUBT model to improve security for cloud computing by enhancing traditional access control. This improvement is achieved by checking the user's trust value before the authorized user can access the cloud. In addition, the model can monitor user behavior while the user interacts with the cloud in order to avoid malicious attacks from an abnormal user. The FUBT uses fuzzy logic to compute a user's trust value in order to provide more intelligent access control for the cloud. The FUBT model is the same as the FMUBCT model which consists of eight primary modules: the authentication, authorization, behavior-monitoring, user-profiling, comparison, trust computation, trust management, and fuzzy logic modules.

However, the difference between the FUBT and FMUBCT models is that FUBT only uses the fuzzy logic approach to calculate all trust types. In FMUBCT, we propose multiple equations to calculate direct trust. In FUBT, we consider indirect trust under the trust computation module. Figure 14 illustrates the flow chart for the trust computation module.

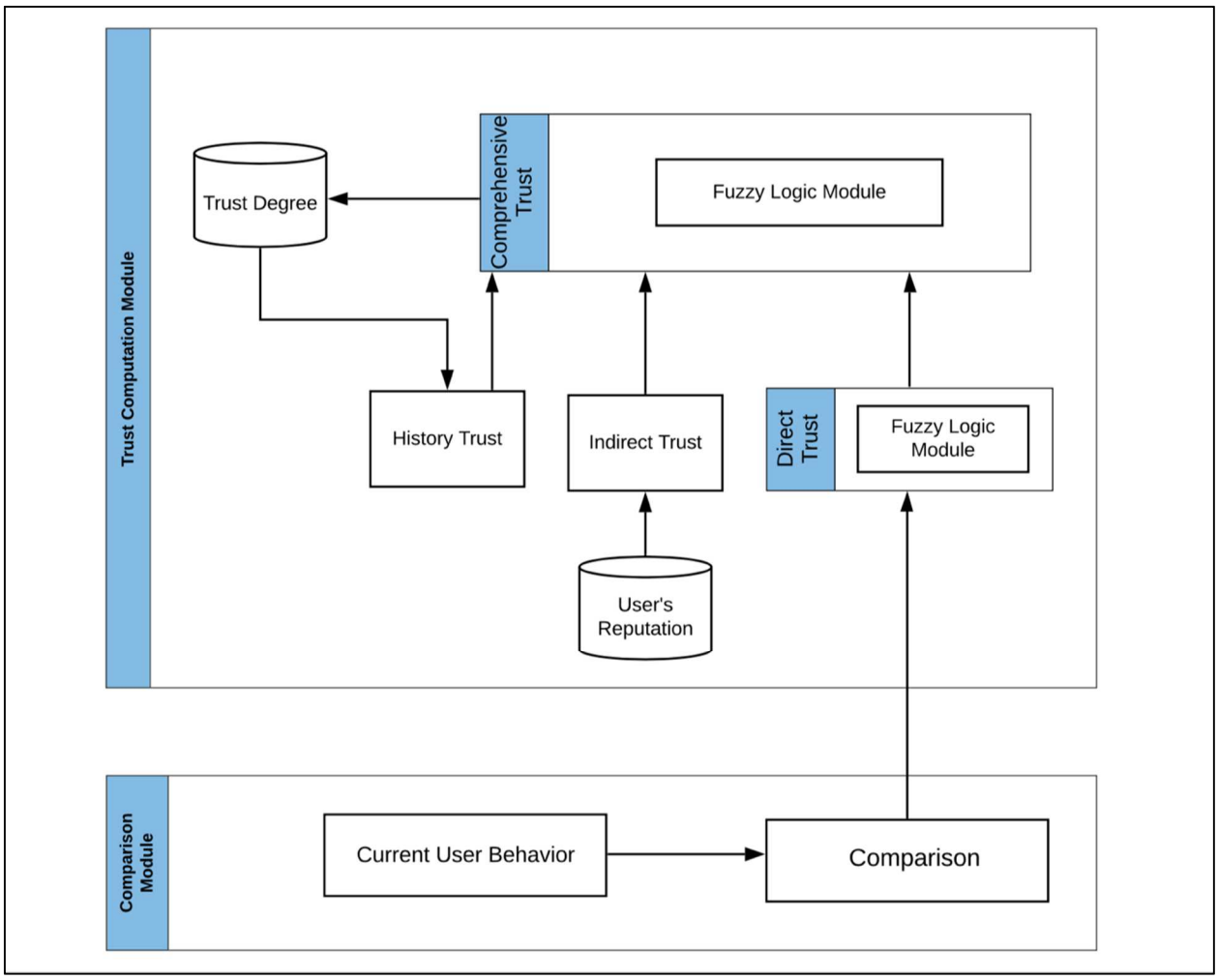

Figure 14. Flow chart for trust computation module.

4.2.2. FUBT Module's Algorithms

In this section, we present the algorithms and the equations used for each module. In Section 4.2, we mentioned that we have eight modules in the FUBT model. We will explain the modules that are improved from the FMUBCT: the user-profiling and trust computation modules.

4.2.2.1. User-Profiling Module

To recognize and to detect a user's abnormal behavior, a user-profile pattern must be created to distinguish a user's normal behavior from the abnormal behavior. The FUBT uses the algorithm that was explained in Chapter 3.

In FUBT, we consider four evidence types, security, login, operation, and performance, to monitor the user and to collect more evidence about the user. The FUBT considers performance evidence because utilizing resources such as memory, disk space, and CPU are important indicators to distinguish between normal and malicious users. For example, a malicious user utilizes the entire memory by creating an extreme number of mail messages. The FUBT considers the following three factors:

- User's memory occupancy rate.
- User's disk-space occupancy rate.
- User's CPU occupancy rate.

The user's behavioral evidence can be obtained directly from the detection tools which were mentioned in Section 2.3.2. Thus, we extract the considering evidence from the software's or hardware's log file.

4.2.2.2. Trust Computation Module

User-behavior characteristics vary. While historical behavior data can introduce a collection of trusted behavior, user behavior does not only rely on historical behavior. Thus, only using historical data to evaluate user behavior is too indiscriminate. Therefore, we should consider different trust types. In the FUBT, we consider four trust types to compute the user's trust value: direct, history, indirect, and comprehensive trust.

4.2.2.2.1. Direct Trust (DT)

It reflects the current interaction between the user and the cloud platform. FUBT uses the Mamdani type. We calculate direct trust in two phases, and each phase has five steps based on the fuzzy approach.

4.2.2.2.1.1. The Phase I

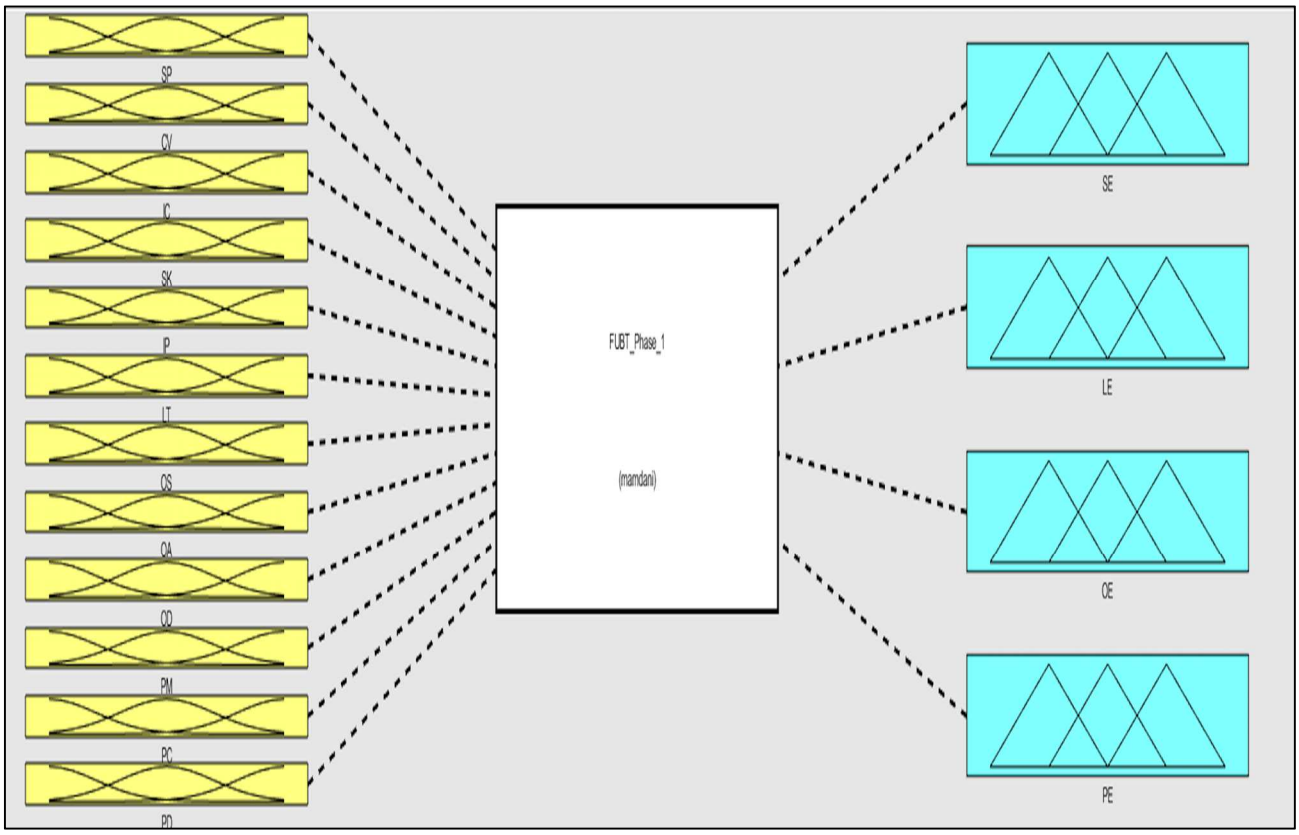

Figure 15. The input and output variables for the first phase.

The first step is a process to identify the input and output variables. Factors for each evidence type are input linguistic variables. The output variables are security, login, operation, and performance trust values. Figure 15 presents the input and output variables for phase 1. Table 13 illustrates the membership degree for each input. Tables 14 presents the membership degree for the outputs.

Table 13. Fuzzy input linguistic values and notations.

Linguistic Variables	**Factors**	**Notation**	**Membership**
Security	Scanning Port	SP	
	Carrying Virus	CV	
	Illegal Connection	IC	
	Sensitive Keyword	SK	Yes
			No
Login	IP	IP	Yes
	Time	LT	No
Operation	Service	OS	Yes
	Action	OA	No
	Duration	OD	
Performance	Memory	PM	Low
	CPU	PC	Medium
	Disk Space	PD	High

Table 14. Fuzzy output linguistic values.

Linguistic Variables	Notation	Membership
Security	SE	Distrust Trust
Login	LE	Distrust Suspect Trust
Operation	OE	Distrust Suspect General Trust
Performance	PE	Distrust Suspect General Trust High Trust

The second step is to construct the fuzzification function. Each input variable is simulated with the fuzzy set. In fuzzy logic, the membership of a given element in a set is determined as a fractional value between 0 and 1, known as a membership degree, which conveys an idea about how much of that element is contained within a set. However, in our model, some inputs only take two values: 0 and 1, where 0 means yes; the user carries a virus to the cloud, atypical time is spent in the cloud, or the user accesses unusual cloud services. However, 1 means no; the user behaves in the usual way: typical time spent in the cloud, does not carry a virus to the cloud, or does not scan an important port. To obtain the value for each factor, we use equation 18

$$x = random(range) \quad , \tag{18}$$

where range for yes is [0,0.5], and range for no is [0.51, 1].

We introduced a triangular shape as the membership function to define $\mu_{A_i}(x_0)$. As an example of input variables, we illustrate fuzzification by showing the membership function for SP, CV, SK, IC, IP, LT, OS, OA, and OD with a triangular view of the variables depicted in Figure 16. Figure 17 depicts the membership function degree for the performance factors, such as memory, CPU usage, and disk-space usage.

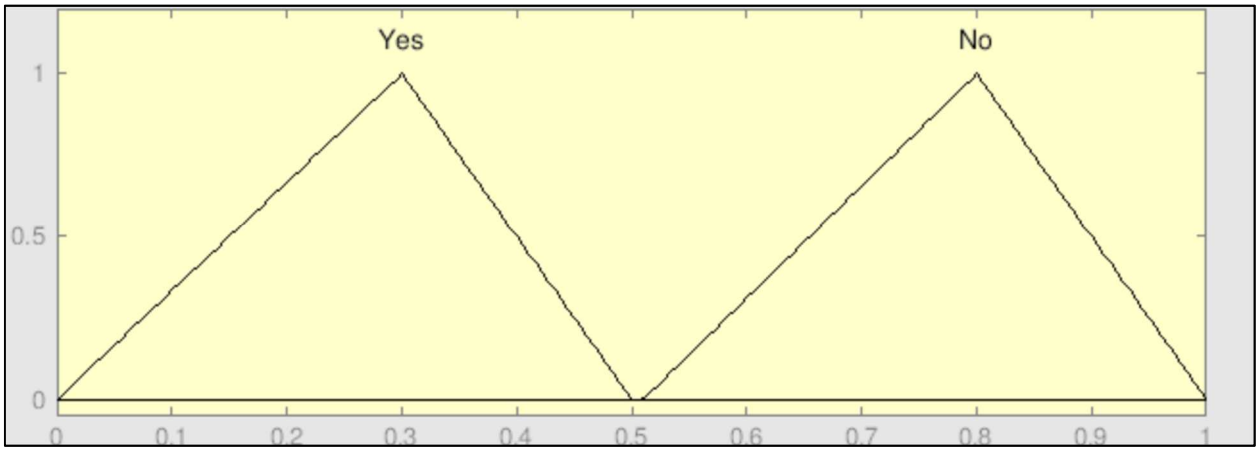

Figure 16. Security, login, and operation evidence inputs for the membership function.

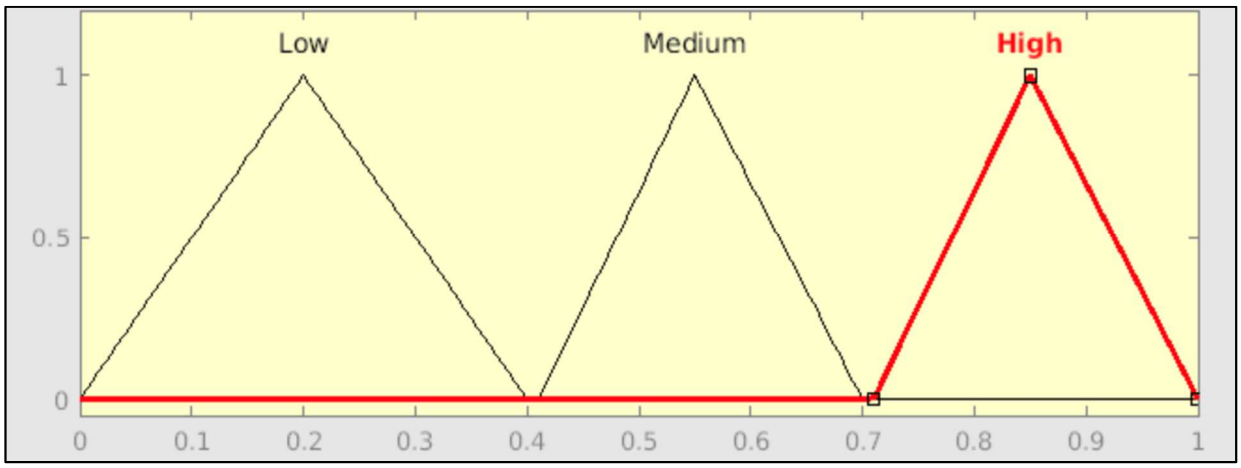

Figure 17. Performance-evidence inputs for the membership function.

The third step is to use the fuzzy-inference system to generate rules and then to control the output value. In the Mamdani type, the fuzzy rule is simple and is represented by an IF-THEN relationship. The IF-THEN rule is written as follows:

IF X is a and Y is b and Z is c THEN w ,

where X, Y, and Z are the crisp inputs; a, b, and c are the fuzzy clusters to which the inputs may correspond; and w is the output fuzzy cluster used for defuzzification. The possible number of generating rules for each input and output is 40. Tables 15-18 depict the fuzzy rule which we consider in phase one.

Table 15. Security-evidence fuzzy rules.

Rule	SP	CV	IC	SK	Then
1	Yes	Yes	Yes	Yes	Distrust
2	No	No	No	No	Trust

Table 16. Login-evidence fuzzy rules.

Rule	IP	LT	Then
3	Yes	Yes	Distrust
4	NO	Yes	Suspect
5	Yes	No	Suspect
6	No	No	Trust

Table 17. Operation-evidence fuzzy rules.

Rule	OS	OA	OD	Then
7	Yes	Yes	Yes	Distrust
8	Yes	No	Yes	Suspect
9	Yes	Yes	No	Suspect
10	No	Yes	Yes	Suspect
11	No	No	Yes	General
12	No	Yes	No	General
13	Yes	No	No	General
14	No	No	No	Trust

Table 18. Performance-evidence fuzzy rules.

Rule	PM	PC	PD	Then
15	H	H	H	Distrust
16	L	H	H	Distrust
17	H	L	H	Distrust
18	H	H	L	Distrust
19	H	M	M	Distrust
20	M	H	M	Distrust
21	M	H	H	Distrust
22	M	M	H	Distrust
23	H	M	H	Distrust
24	M	H	L	Suspect
25	M	L	H	Suspect
26	L	M	H	Suspect
27	H	M	L	Suspect
28	L	H	M	Suspect
29	H	L	M	Suspect
30	M	M	M	Suspect
31	M	M	L	General
32	L	M	M	General
33	M	L	M	General
34	H	L	L	General
35	L	H	L	General
36	L	L	H	General
37	M	L	L	Trust
38	L	M	L	Trust
39	L	L	M	Trust
40	L	L	L	High Trust

The fourth step is the process for evaluating fuzzy rules. The system has to use an inference engine to obtain the fuzzy value for the output. In our model, the intersection (the fuzzy AND operation) is given by

$$\text{Min } \{\mu A (x), \mu B (x)\} \tag{19}$$

For the first rule, we use the union (OR operation) which is given by

$$\text{Max } \{\mu A (x), \mu B (x)\} \tag{20}$$

The final step is defuzzification, the process to get the final, crisp output. There are different defuzzification methods. We use one of the most popular methods, the Center of Gravity (COG).

$$\text{COG(A)} = \frac{\int_{\min}^{\max} \mu_A(x)\, x dx}{\int_{\min}^{\max} \mu_A(x)\, dx} \tag{21}$$

The fuzzy output variable is also a linguistic variable, where the values are assigned grades of membership. We have two membership degrees for the security evidence: distrust and trusted. We have defined the security evidence value in Table 19, and its equivalent membership function is shown in Figure 18.

Table 19. Fuzzy security-evidence value.

Linguistic Security-Evidence Value	Range	Fuzzy Number
Distrust	0 - 0.50	(0 0.30 0.5)
Trusted	0.51 – 1	(0.51 0.7 1)

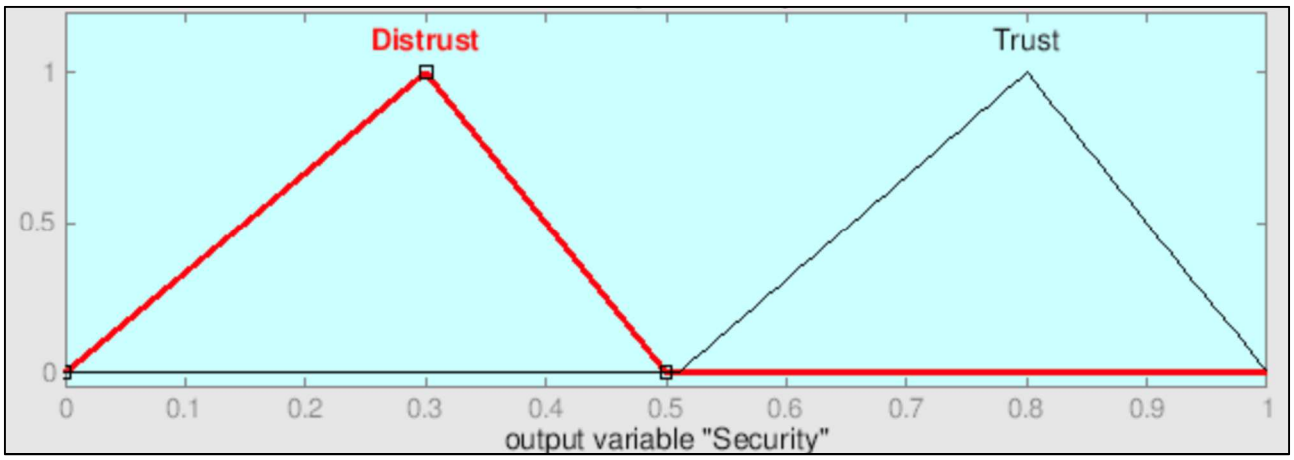

Figure 18. Security-evidence output membership function.

For the login evidence, we produce the three membership degrees which are presented in Table 20. The equivalent membership function is shown in Figure 19.

Table 20. Fuzzy login-evidence value.

Linguistic Login-Evidence Value	Range	Fuzzy Number
Distrust	0-0.30	(0 0.15 0.30)
Suspect	0.31-0.6	(0.31 0.45 0.6)
Trusted	0.61-1	(0.61 0.85 1)

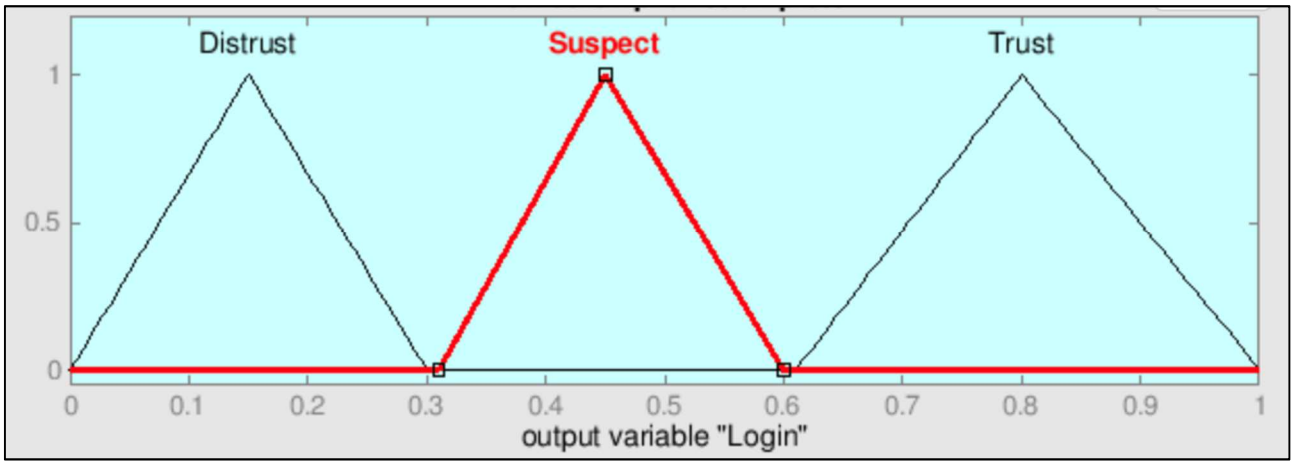

Figure 19. Login-evidence output membership function.

For the operation evidence, we produce four membership degrees which are presented in Table 21. The equivalent membership function is shown in Figure 20.

Table 21. Fuzzy operation-evidence value.

Linguistic Operation-Evidence Value	Range	Fuzzy Number
Distrust	0-0.30	(0 0.15 0.30)
Suspect	0.31-0.5	(0.31 0.4 0.5)
General	0.51-0.7	(0.51 0.6 0.7)
Trusted	0.71-1	(0.71 0.85 1)

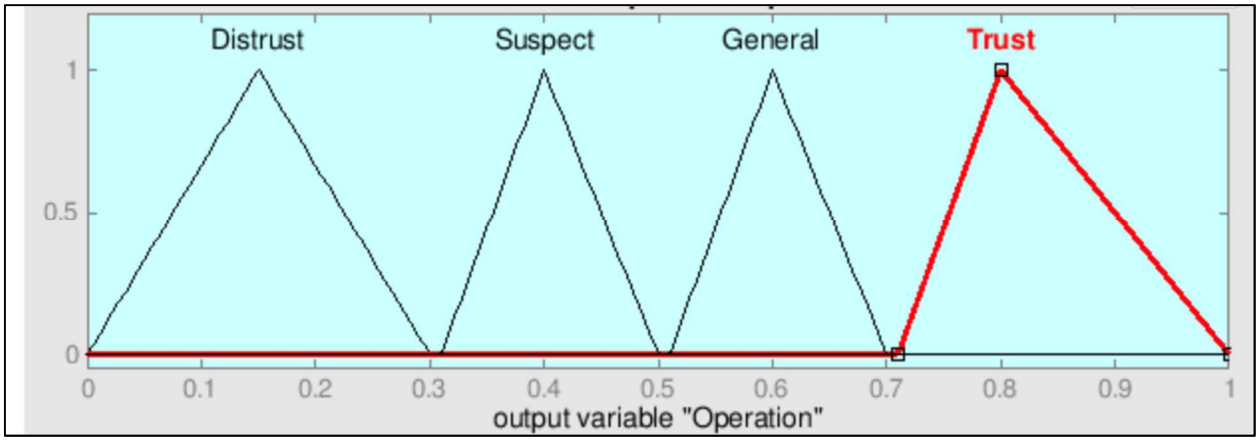

Figure 20. Operation-evidence output membership function.

For the performance evidence, we produce four membership degrees which are presented in Table 22. The equivalent membership function is shown in Figure 21.

Table 22. Fuzzy performance-evidence value.

Linguistic Performance-Evidence Value	Range	Fuzzy Number
Distrust	0-0.30	(0 0.15 0.30)
Suspect	0.31-0.5	(0.31 0.4 0.5)
General	0.51-0.65	(0.51 0.58 0.65)
Trust	0.66-0.8	(0.66 0.75 0.8)
High Trust	0.81-1	(0.81 0.9 1)

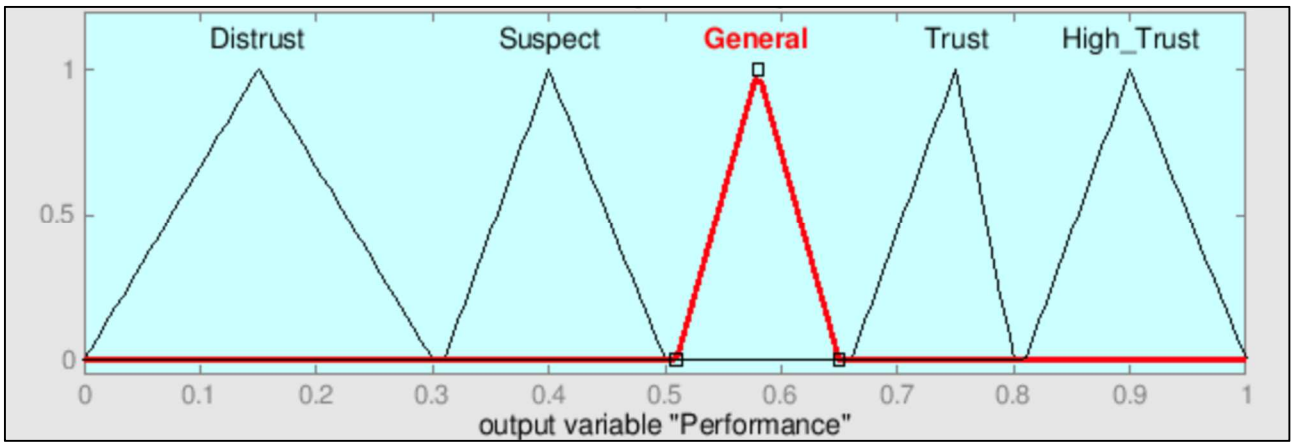

Figure 21. Performance-evidence output membership function.

4.2.2.2.1.2. The Phase II

For the first step, the inputs are outputs (SE, LE, OE, and PE) from phase I, and the direct trust value is the output. Figure 22 presents the input and output variables for phase II.

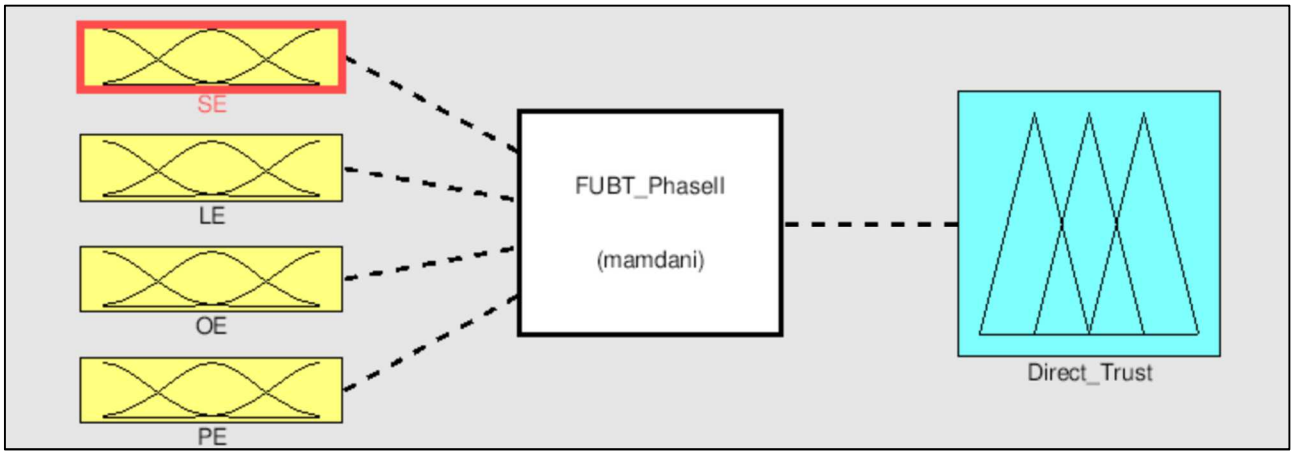

Figure 22. Input and output variables for the phase II.

The possible number of generating rules for each input and output is 11. Table 23 presents the fuzzy rules for the phase II.

Table 23.The phase II fuzzy rules.

Rule	SE	LE	OE	PE	Then
1	Distrust	Distrust	Distrust	Distrust	Distrust
2	Trust	Trust	Trust	High Trust	High Trust
3	Trust	Trust	Trust	Trust	Trust
4	Trust	Suspect	Trust	Trust	Trust
5	Trust	Suspect	Trust	High Trust	Trust
6	Trust	Trust	General	Trust	General
7	Trust	Trust	Suspect	Trust	Suspect
8	Trust	Trust	General	High Trust	General
9	Trust	Trust	Suspect	High Trust	Suspect
10	Trust	Trust	Trust	Suspect	Suspect
11	Trust	Trust	Trust	General	General
12	Distrust	Trust/Suspect	Trust	Trust/High Trust	Suspect
13	Distrust	Distrust	Trust	Trust/High Trust	Distrust
14	Distrust	Trust	General	Trust/High Trust	Suspect
15	Distrust	Trust	Suspect/Distrust	Trust/High Trust	Distrust
16	Distrust	Trust	Distrust	General/Suspect /Distrust	Distrust
17	Distrust	Trust	Trust	General	Suspect
18	Distrust	Trust	Trust	Distrust	Distrust

In this phase, we use intersection (the fuzzy AND operation). For the defuzzification, we use the center of gravity (COG). For direct trust, we produce four membership degrees which are presented in Table 24. The equivalent membership function is shown in Figure 23.

Table 24. Fuzzy direct-trust value.

Linguistic Direct-Trust Value	Range	Fuzzy Number
Distrust	0-0.20	(0 0.10 0.20)
Suspect	0.21-0.5	(0.21 0.3 0.5)
General	0.51-0.65	(0.51 0.58 0.65)
Trust	0.66-0.8	(0.66 0.75 0.8)
High Trust	0.81-1	(0.81 0.9 1)

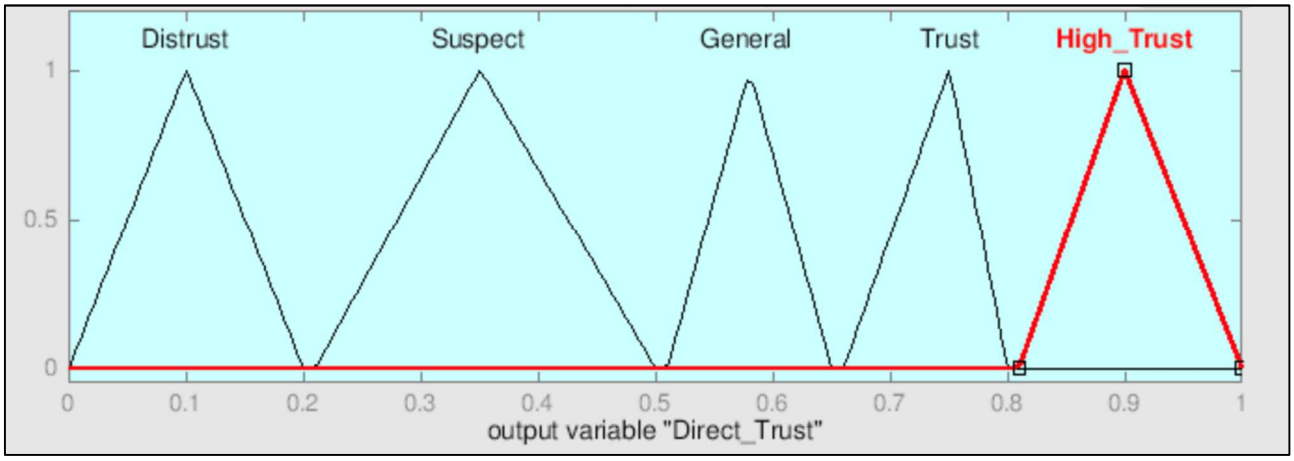

Figure 23. The phase II output membership function.

4.2.2.2.1.3. The Phase III

The last phase is to find the comprehensive trust value based on the three inputs: direct trust from phase II, indirect, and history. Figure 24 depicts the input and output variables for phase III.

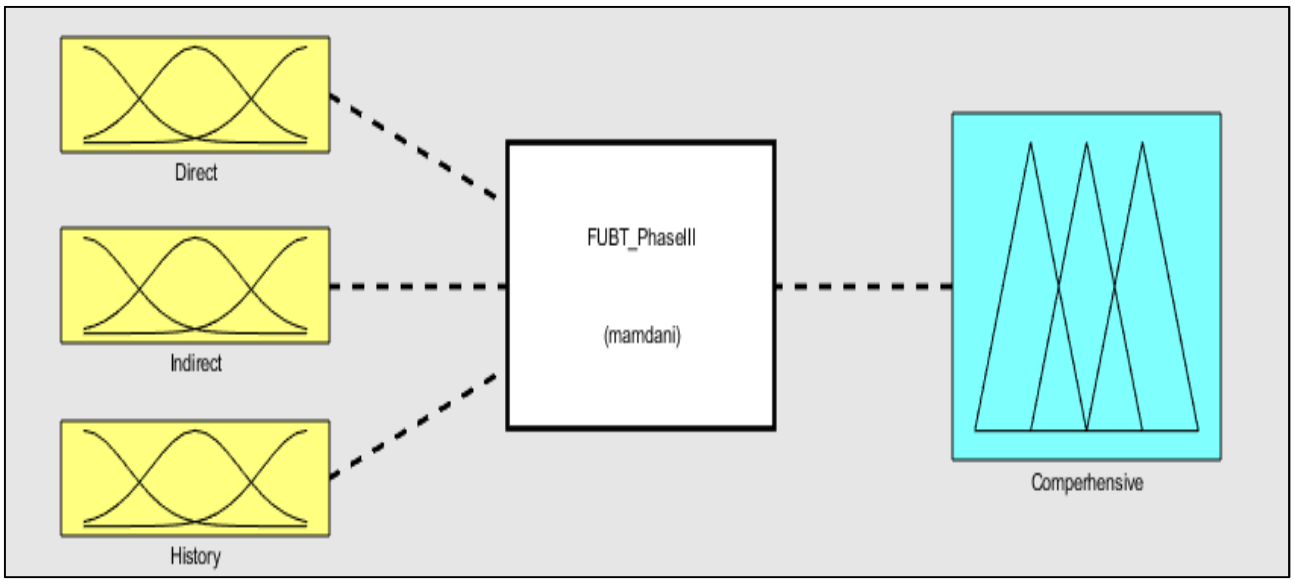

Figure 24. The input and output variables for the phase III.

4.2.2.2.2. History Trust (HT)

Reflects the last trust value which is affected by fuzzy logic for comprehensive trust. History trust has the same membership degree as direct trust from phase II.

4.2.2.2.3. Indirect Trust (IT)

IT is the score from another trusted user and same cloud provider, but from different domains; it is used to obtain the trust value for a new user in the domain when that user is an old user in another domain. In addition, if a malicious user has low trust values in other domains, then the user must be a malicious user in the new domain, too. The user's recommendation scores are stored in the database. The total (IT) is calculated using equation 22. Any trusted user can add or update a recommendation about a user based on the experience.

$$IT = \frac{\sum_1^i ITi}{i} \quad , \tag{22}$$

where IT is the score and i is the number of scores.

In this equation, we calculate the average for the recommendation scores to prevent synergies from cheating. The average protects the system from two user types: users with a

smaller number of recommendations from receiving a lower score and users with a higher number of recommendations from receiving a higher score. Figure 25 shows the membership degree for the indirect trust.

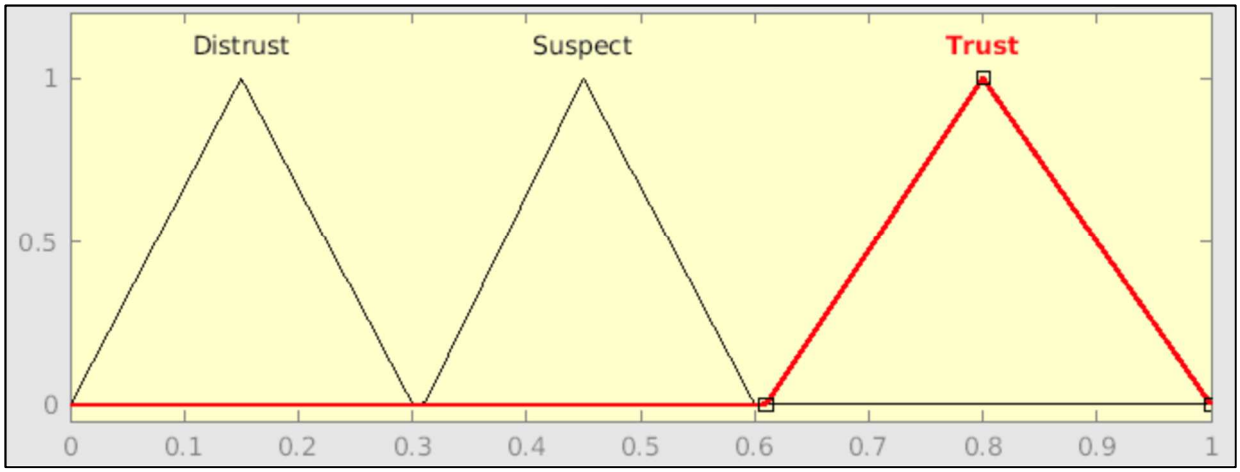

Figure 25. Membership function for indirect trust.

4.2.2.2.4. Comprehensive Trust (CT)

CT is the combination of direct, historical, and indirect trust. We use fuzzy logic to compute comprehensive trust. The inputs are DT, HT, and IT, and the output is CT. The rest of the fuzzy logic steps are the same as the steps in phase I and phase II. We illustrate fuzzification by showing the membership function for CT with a triangle view of variables (Figure 26). we produce four membership degrees which are presented in Table 25. The equivalent membership function is shown in Figure 26.

Table 25. Fuzzy comprehensive-trust value.

Linguistic Compressive-Trust Value	Range	Fuzzy Number
Distrust	0-0.20	(0 0.10 0.20)
Suspect	0.21-0.5	(0.21 0.3 0.5)
General	0.51-0.65	(0.51 0.58 0.65)
Trust	0.66-0.8	(0.66 0.75 0.8)
High Trust	0.81-1	(0.81 0.9 1)

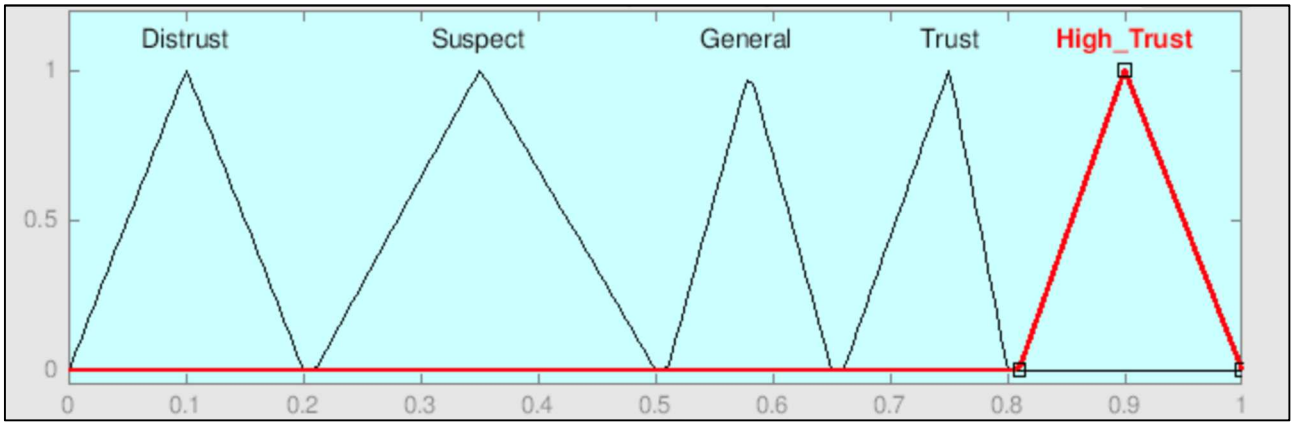

Figure 26. Comprehensive trust membership function in the FUBT.

Table 26 presents the fuzzy rules for the phase III. In this phase, we use intersection (the fuzzy AND operation). For the defuzzification, we use the center of gravity (COG).

Table 26. The phaseIII fuzzy rules.

Direct Trust	History Trust	Indirect Trust	Then
Distrust	-	-	Distrust
Suspect	Distrust	Distrust	Distrust
Suspect	Distrust/Suspect	Suspect/Trust	Suspect
Suspect	Suspect	Distrust/Suspect	Suspect
Suspect	General	Distrust/Suspect	Suspect
Suspect	General	Trust	Suspect
Suspect	Trust	Distrust/Suspect	Suspect
Suspect	Trust	Trust	Suspect
Suspect	High Trust	Distrust/Suspect	Suspect
Suspect	High Trust	Trust	Suspect
General	Distrust	Distrust	Distrust
General	Distrust	Suspect/Trust	Suspect
General	Suspect	Distrust/Suspect	Suspect
General	Suspect	Trust	General
General	General	Distrust	Suspect
General	General	Suspect/Trust	General
General	Trust	-	General
General	High Trust	Distrust/Suspect	General
General	High Trust	Trust	Trust
Trust	Distrust	Distrust/Suspect	Suspect
Trust	Distrust	Trust	General
Trust	Suspect	Distrust	Suspect
Trust	Suspect	Suspect	General
Trust	Suspect	Trust	Trust
Trust	General	Distrust	General
Trust	General	Suspect/Trust	Trust
Trust	Trust	Distrust	General
Trust	Trust	Suspect/Trust	Trust
Trust	High Trust	Distrust	General
Trust	High Trust	Suspect/Trust	Trust
High Trust	Distrust	Distrust/Suspect	Suspect
High Trust	Distrust	Trust	General
High Trust	Suspect	Distrust/Suspect	General
High Trust	Suspect	Trust	Trust
High Trust	General	Distrust	General
High Trust	General	Distrust/Suspect	Trust
High Trust	Trust	Distrust	General
High Trust	Trust	Suspect	Trust
High Trust	Trust	Trust	High Trust
High Trust	High Trust	Distrust/Suspect	Trust
High Trust	High Trust	Trust	High Trust

4.2.3. Reflecting Trust Evaluation Principles

The FUBT uses the same approach that was described in Sections 3.3.3.1 and 3.3.3.3.

4.2.3.1. Recent User Behavior Affects the Trust Value

With the fuzzy logic module, we consider the recent-behavior principle. In Phase II, for example, with rule 7, if the user accesses unauthorized service for a first time, then the user will be suspect. Thus, the operation-evidence trust value is suspect, which makes the direct trust suspect, too. For Phase III, with fuzzy rules, we give the direct trust more weight than the history and indirect trust. This principle is important to ensure that the comprehensive trust value reflects the user's current state.

4.2.3.2. Punishment (Rapid-Decrease) Strategy and Repeating Malicious Behavior

If the user has malicious behavior (direct trust <=0.5), the trust value is reduced based on fuzzy rules in the phase III. We use equation 17 (described in Section 3.3.3.4) when the user repeats malicious behavior. Thus, by repeating malicious behavior the user's trust value keeps decreasing until the user is denied access.

4.3. Conclusion

In this chapter, we proposed FUBT to evaluate the users' behavior and to detect abnormal user behavior in the cloud. This model used four types of user-behavior evidence: login, security, operation, and performance. In addition, this model considered four trust types: direct, historical, indirect, and comprehensive. We used fuzzy logic to compute the direct and comprehensive trust values. The FUBT model was simulated, and the results showed that it can effectively calculate the users' trust values. In addition, the FUBT model considered all the evaluation principles.

CHAPTER 5. EXPERIMENTS AND SIMULATION RESULTS

In this chapter, the simulation platform is defined. Simulation results and the analysis for the FMUBCT and FUBT models are presented.

5.1. Simulation Platform and Tools

The platform used for the simulation is MATLAB 9.5 with an Intel Core 5 Processor running at 2.3 GHz; there are 8 GB of RAM. Although it is difficult to find real-life user datasets, we generate a random dataset based on probability theory by using SAS 9.4.

5.1.1. Generating the Dataset

There is a lack of data from real system audit logs, especially in mission-critical and senior industries such as healthcare, banking, and the military. Consequently, we built an algorithm-based probability theory using SAS (Statistical Analysis System) 9.4 to generate a dataset; then, we used the dataset to validate the models in this dissertation.

Because of our models are based on evaluate user behavior in the cloud, we analyze the audio log from the AWS API to obtain information recorded in the cloud [83,84,85]. Figure 27 is an example of the AWS Cloud Trial user's identity and user's event; the example contains fields which define what action was requested, who requested this action, when, and where. We use the same fields for the AWS API in order to generate the dataset. Thus, we use information from AWS to generate our dataset.

```
{
       "Records": [{
              "eventVersion": "1.01",
              "userIdentity": {
                     "type": "IAMUser",
                     "principalId": "AIDAJDPLRKLG7UEXAMPLE",
                     "arn": "arn:aws:iam::123456789012:user/Alice",
                     "accountId": "123456789012",
                     "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                     "userName": "Alice",
                     "sessionContext": {
                            "attributes": {
                                   "mfaAuthenticated": "false",
                                   "creationDate": "2014-03-18T14:29:23Z"
                            }
                     }
              },
              "eventTime": "2014-03-18T14:30:07Z",
              "eventSource": "cloudtrail.amazonaws.com",
              "eventName": "StartLogging",
              "awsRegion": "us-west-2",
              "sourceIPAddress": "72.21.198.64",
              "userAgent": "signin.amazonaws.com",
              "requestParameters": {
                     "name": "Default"
              },
              "responseElements": null,
              "requestID": "cdc73f9d-aea9-11e3-9d5a-835b769c0d9c",
              "eventID": "3074414d-c626-42aa-984b-68ff152d6ab7"
       },
   ... additional entries ...
       ]
```


Figure 27. Example of AWS cloud trial.

5.1.2. Dataset Design

In this dissertation, we create an event dataset to simulate users' real data in the cloud, where each record has the following attributes: user ID, user packet (IP address, login date, login time, service, action, and duration), security factors (virus, illegal connection, scanning port, and inputting security-sensitive keywords), and usage (memory, CPU, and disk space). In our work, we used 14 attributes to evaluate users' behavior in the cloud. Because the data in an audit log are categorical data, we encode the categorical data to a number in order to reduce the complexity and to increase user-profiling algorithm's speed, producing common user-behavior patterns to compute the direct trust. For example, we substitute the cloud service's name with a

number, such as access storage service to 1 and access user's account service to 2; we continue using the same steps with actions and user ID. For security evidence, we have two values: 0 means yes, there is abnormal behavior, such as a user's illegal connection to access the cloud, and 1 means the user had a legal connection to access the cloud. For performance evidence, we have a range of 0 to 1 to present utility usage (CPU, memory, and disk space), where 0-0.4 means low usage, 0.41-0.7 means medium usage, and 0.71-1 means high usage.

There are various types of continuous probability distribution, such as normal distribution, exponential distribution, and generalized Bernoulli distribution. These distribution types can be used to indicate the demand distribution for the attribute values. With our proposed models, we use uniform and normal distribution to produce random data. In addition, to obtain more accurate random events, we have used the bootstrap resampling technique. Table 27 illustrates the rules that were used to generate data. By utilizing our proposed algorithm, we generate 7K records for 50 users for 43 days.

Table 27. The attributes' distribution definition.

Attribute	Representation	Domain	Value
User ID	U	50	1-50
IP Address	I	100	1-100
Date	D	43	12/13/2018-1/25/2019
Time	T	24	1-24 hour
Service	S	14	1-14
Action	A	5	1-5
Duration	DU	120	1-120 Minutes
Virus	CV	2	0 or 1
Illegal Connection	IC	2	0 or 1
Scanning Important Port	SIP	2	0 or 1
Sensitive Keyword	SK	2	0 or 1
CPU Usage	CU	3	0-1
Memory Usage	MU	3	0-1
Disk Space	TC	3	0-1

5.1.3. Dataset-Generation Algorithm

5.1.3.1. Replication data

In this approach, we divide the users into 2 types; 30 are normal, and 20 are malicious. We construct events by randomly selecting data from the domain for each factor using uniform distribution. Then, we duplicate each factor using uniform distribution in order to create records for each factor. Afterward, we merge factors to create events. For example, to generate a normal user event, the service is selected randomly from the services domain, which is from 1 to 14, and action, IP, time, and date have been selected randomly form their domains. We set conditions for the security factor and the performance factors. The security factor must be 1, meaning that the user does not misbehave in the cloud and that the performance factors should be in the low level. A malicious user should have more than abnormal behavior from the security or performance evidence. Table 28 gives an example of generating events using our algorithm. Finally, we combine the user-behavior events for all categories to create one dataset to validate our proposed models.

Table 28. Slice of generating events using our algorithm.

ID	Date	Time	IP	OS	OA	OD	SP	CV	IC	SW	PC	PM	PD
1	12/12/2018	6:56	1	2	5	13	1	1	1	1	0.14	0.03	0.03
1	12/12/2018	15:58	6	8	2	5	1	1	1	1	0.05	0.12	0.33
1	12/12/2018	8:52	6	15	3	14	1	1	1	1	0.01	0.16	0.34
1	12/13/2018	15:03	6	5	1	50	1	1	1	1	0.12	0.16	0.14
1	12/13/2018	0:41	1	14	4	61	1	1	1	1	0.33	0.17	0.36
……..													
……													
1	1/1/19	9:05	15	12	2	58	1	1	1	1	0.39	0.36	0.13
1	1/2/19	8:26	15	10	5	2	1	1	1	1	0.26	0.25	0.39
2	12/14/18	3:12	8	4	1	68	1	1	1	1	0.11	0.2	0.14
2	12/14/18	4:05	1	1	2	108	1	1	1	1	0.35	0.38	0.15
2	12/14/18	18:54	6	10	2	76	1	1	1	1	0.39	0.39	0.14
2	12/14/18	0:40	9	12	1	17	1	1	1	1	0.34	0.1	0.32
2	12/15/18	10:44	8	5	3	9	1	1	1	1	0.15	0.1	0.28
2	12/15/18	2:37	14	7	5	11	1	1	1	1	0.04	0.34	0.03
2	12/15/18	22:22	2	2	1	11	1	1	1	1	0.18	0.06	0.28
2	12/15/18	21:01	20	6	1	11	1	1	1	1	0.39	0.36	0.13
2	12/16/18	14:31	10	7	1	106	1	1	1	1	0.26	0.25	0.39
2	12/16/18	9:42	2	4	2	13	1	1	1	1	0.11	0.2	0.14
2	12/16/18	4:23	1	3	5	16	1	1	1	1	0.35	0.38	0.15
2	12/16/18	6:18	14	8	1	49	1	1	1	1	0.39	0.39	0.14
…….													
……..													
…….													
31	12/17/18	5:10	18	3	2	48	1	0	1	1	0.24	0.17	0.27
31	12/17/18	20:17	18	13	3	11	1	1	0	1	0.09	0.05	0.36
31	12/17/18	12:40	6	14	1	15	1	1	0	1	0.04	0.06	0.07
31	12/17/18	13:49	14	9	3	101	1	1	1	1	0.18	0.31	0.21
31	12/17/18	9:46	13	1	3	101	1	1	1	1	0.15	0.39	0.13
31	12/18/18	15:27	13	6	2	6	1	0	1	1	0.17	0.27	0.38
…….													
…….													
…….													
50	1/25/19	5:10	14	1	1	11	1	1	1	0	0.92	0.92	0.08
50	1/25/19	14:45	19	10	4	102	1	1	0	1	0.92	0.87	0.23
50	1/25/19	23:44	6	5	4	18	1	0	1	1	0.27	0.95	0.32
50	1/25/19	19:16	12	13	5	9	1	1	1	1	0.17	0.94	0.97

5.1.4. Statistical Analysis

Figure 28 illustrates that our proposed algorithm can generate events for users with the distribution. Figure 28 shows an example of the number of times accessing the cloud for a group of users (user1, user5, user10, user15, user20, user25, user30, user35, user40, user45, and user50) for 43 days. Thus, Figure 28 compares the number of accesses for 11 cloud users. User 10 accesses the cloud more than the other users presented in the figure.

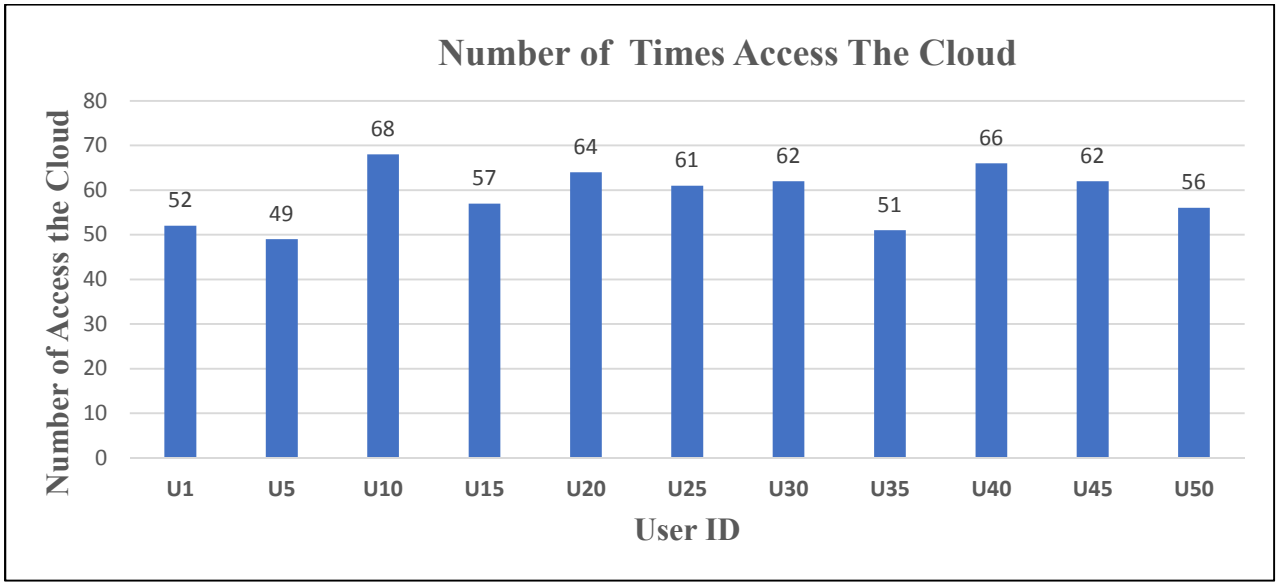

Figure 28. User-access distribution.

To present the distribution for accessing the cloud's services from Dec 12 to Dec 14, Figure 29 illustrates that the cloud users accessed service 4 on Dec 12 more than the other services. However, on Dec 13, service 6 was accessed more. Finally, occurrence for service 8 is above other services.

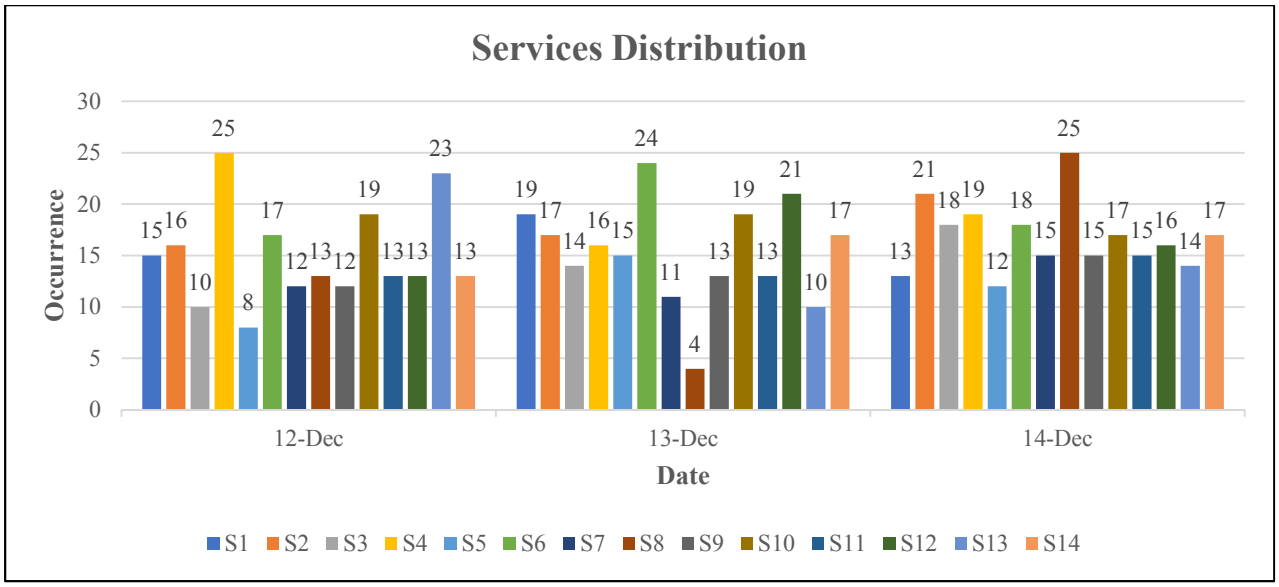

Figure 29. Service-access distribution.

To show the distribution for the cloud's action from Dec 12 to Dec 14, Figure 30 illustrates that the cloud users took action3 more than the other actions on Dec 12. On Dec 13, the users took action1 and action5 more than the others. Finally, the occurrence of action3 and action4 are above than the occurrence of other actions.

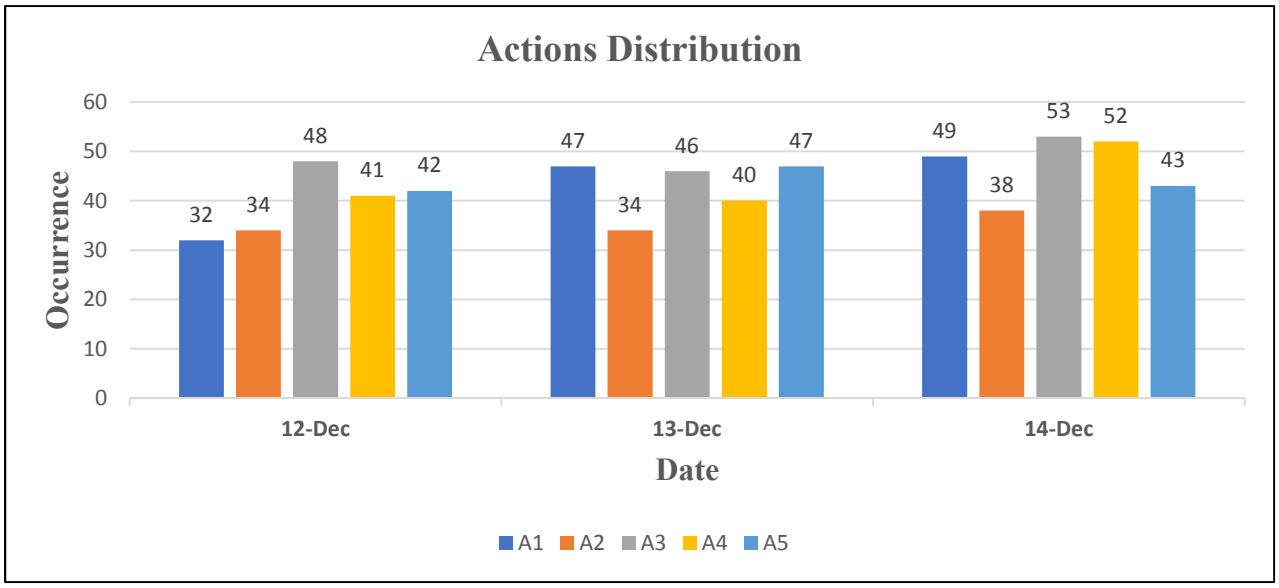

Figure 30. The user's action distribution.

Figure 31 shows that memory-usage distribution for the cloud users from Dec 12 to Dec 14. In Dec 12, there are 75 recodes indicate that usage of memory for some of cloud user in low level which is safe to cloud. However, there are 58 records in medium usage and 64 records in high usage which indicate there is heavy usage to the memory then cloud admin should consider. In Dec 13, the are 66 records on data show usage of memory in the cloud in low level, 71 records in medium level and 76 in high level. Finally, in Dec 14, there are 93 records in low level, 62 in medium level and 80 in high level.

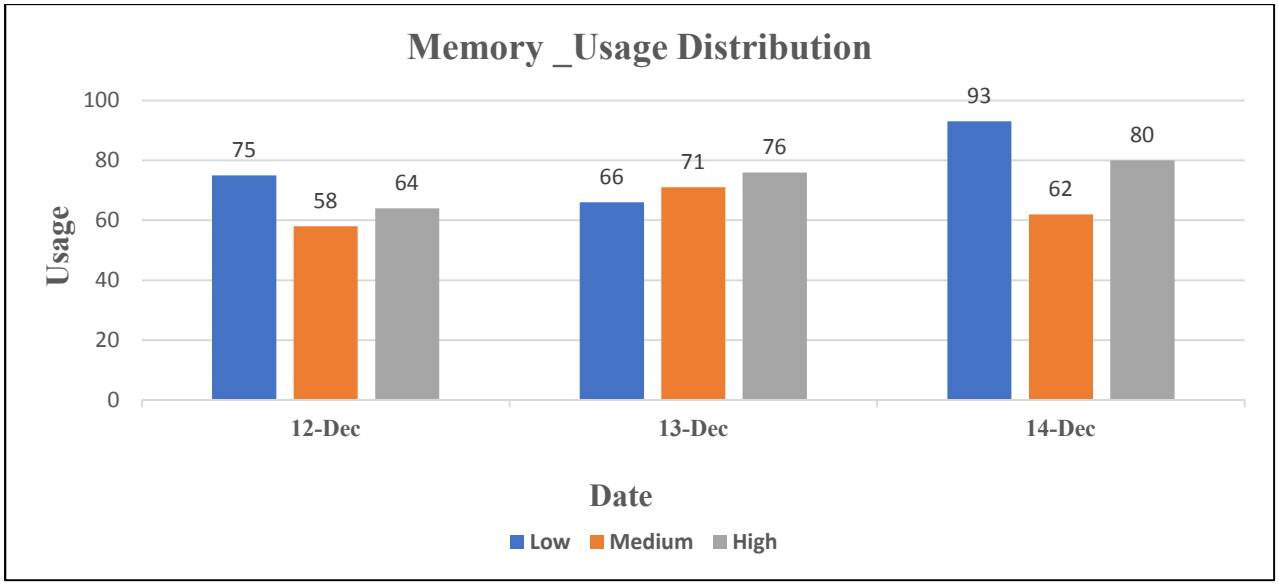

Figure 31. Memory usage.

Figure 32 shows Disk space usage distribution for the cloud from Dec 12 to Dec 14. Dec 12, there are 60 recodes indicate that usage of disk space for some of cloud user in low level which is safe to cloud. However, there are 72 records in medium usage and 81 records in high usage which indicate there is heavy usage to the disk space then cloud admin should consider. In Dec 13, the are 68 records on data show usage of disk space in the cloud in low level, 74 records in medium level and 68 in high level. Finally, in Dec 14, there are 69 records in low level, 67 in medium level and 86 in high level.

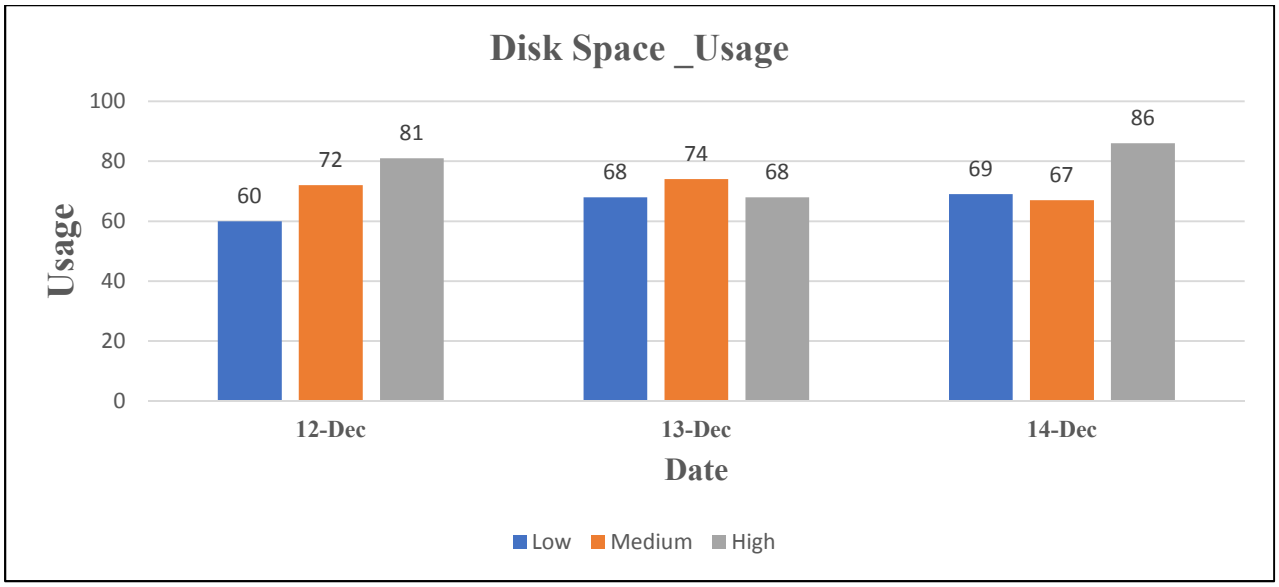

Figure 32. Disk-space usage.

5.2. Simulation Results and Analysis

This section presents multiple case studies to demonstrate the functionality, performance of the proposed models in the dissertation. The experiment in this section has been organized to:

- **Verification:** we run multiple experiment cases to demonstrate that the proposed models are able to produce common user behaviors. In addition, the proposed models efficiently prevent malicious user to access cloud service provider whenever the user behaves abnormally. Moreover, the proposed models are able to timely update user's trust value then the cloud service provider changes authority for the user.
- **Comparative analysis with existing models:** we run various **e**xperiment cases to validate that the proposed models are more efficient than the existing models.

5.2.1. Verification

In this section we developed different cases to evaluate effectiveness of our model. We divide the dataset to 5k as training data to produce history pattern and 2k as testing data to compute user's trust value.

5.2.1.1. The First Case to Evaluate producing User History Pattern

The FMUBCT and FUBT use the same algorithm to produce user history pattern. our proposed algorithm is able to find the user history pattern based on frequently user behaviors. Figure 33 illustrations that by running our user profile algorithm in different days for 50 users, our algorithm is able to find users' history patterns. For five days, the proposed algorithm discovers 147 patterns, ten days proposed algorithm discovers 498 patterns, fifteen days proposed algorithm discovers 931 patterns, and twenty days proposed algorithm discovers 1436 patterns for different users. We conclude from this figure, with more interactions in the cloud, more history patterns will occur.

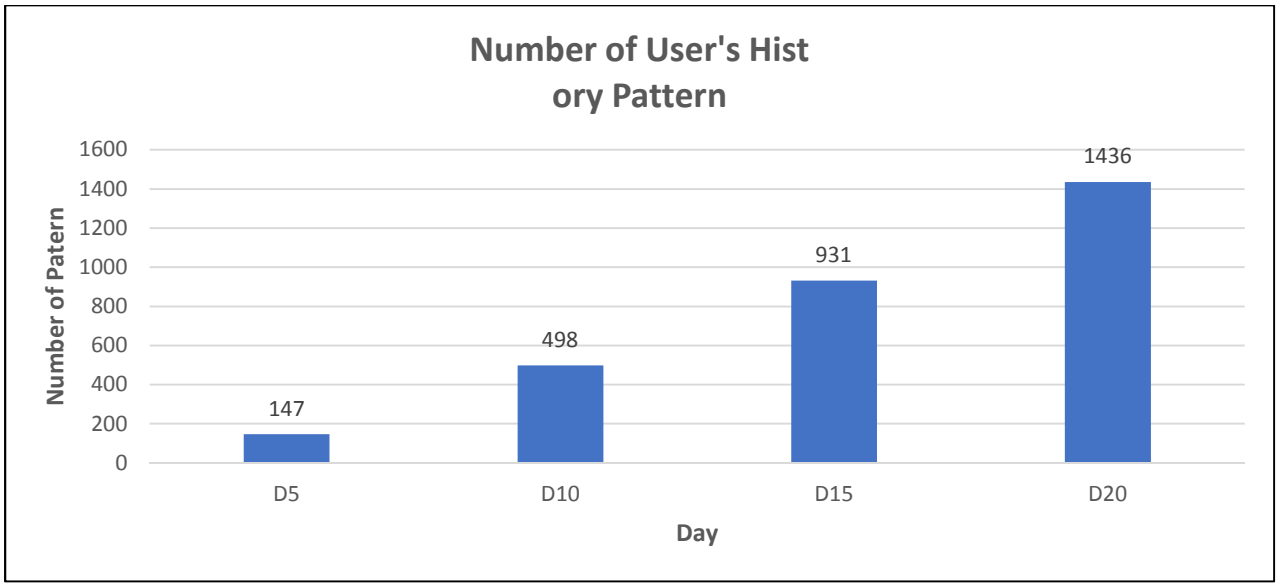

Figure 33. Users' history patterns from day5 to day20.

Figure 33 presents examples of how many patterns that our proposed algorithm finds for a group of users (user1, user5, user13, user22, user35, user38, and user40). Figure 34 shows that the more user interacts with cloud the more patterns we can get. For example, the number of extract patterns for group of users for twenty days accessing the cloud is greater than the number

of patterns for five days of accessing. In addition, Figure 34 presents user35 and user40 have more history pattern within twenty days than other users from this group of users.

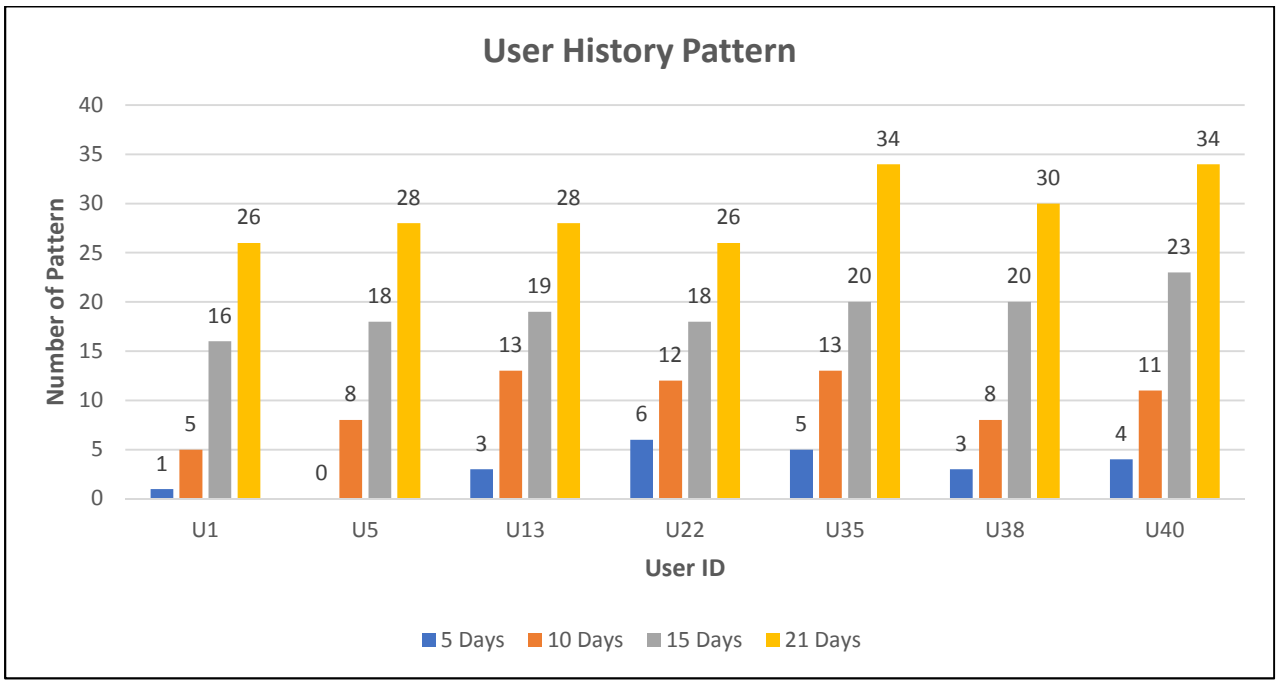

Figure 34. Increasing numbers for the users' history patterns.

In the following experiments, we provide our proposed model set for event records from the dataset. Each record has a set of factors (login, operation, security, and performance). Table 28 provides examples of the event records that were used for our experiment's cases.

5.2.1.2. The Second Case to Evaluate the Slow-Rise Strategy

The slow-rise strategy is a strategy to prevent trust fraud problem. Trust fraud problem is a user gains high trust value within a small number of interactions to the cloud. The FMUBCT and the FUBT have methods to consider the slow-rise strategy; they were described in Section 3.4.1. In this case, we run an experiment for user5 in order to evaluate our strategy to consider the slow-rise principle. In addition, we set the size for uncertain windows as 8 and the size for positive windows as 12. The adjusted growth is 0.01, and the initial trust value is 0.52.

Figure 35 and Table 29 indicate that, even though user5 behaves normally in the cloud from times 1 to 8, the trust value is in the general trust range (reaching a peak value of 0.58) because, based on our strategy of using the sliding-window technique, the number of interactions does not exceed the uncertain windows. Thus, the cloud cannot fully trust the user at this point. Beginning at time 9 while the user continues to behave normally in the cloud and the number of interactions is in the positive windows, the trust value slowly increases to reach a high degree of trust. Thus, the proposed models can prevent the trust-fraud issue where a user obtains a high trust value with a small number of interactions.

In addition, Figure 35 shows the difference between our proposed models at times 15 and 16; this variation occurs because the FMUBCT uses the average for the history trust while the FUBT uses the last comprehensive trust as the history trust. Moreover, based on the fuzzy rules for both proposed models, the comprehensive trust value is calculated. Tables 29 and 30 indicate how the proposed model calculates the user's trust value. The symbols in the tables present the membership functions: N is Normal; G is General; T is Trusted; and HT is High Trusted. Figures 36 and 37 show the sliding windows for our proposed models.

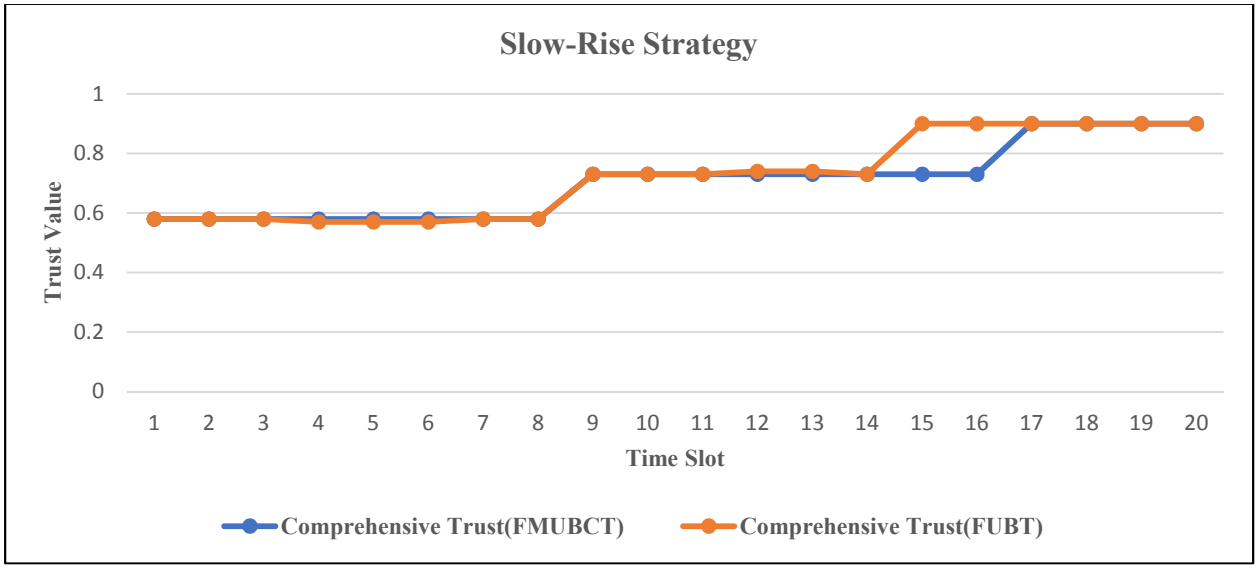

Figure 35. Evaluating the slow-rise strategy.

Table 29. Calculating the trust value based on the slow-rise strategy in FMUBCT.

Time Slot	Direct Trust	Average of History Trust	Comprehensive Trust (FMUBCT)
1	0.52 (G)	0.52 (G)	0.58 (G)
2	0.54 (G)	0.55 (G)	0.58 (G)
3	0.56 (G)	0.56 (G)	0.58 (G)
4	0.58 (G)	0.57 (G)	0.58 (G)
5	0.60 (G)	0.57 (G)	0.58 (G)
6	0.62 (G)	0.57 (G)	0.58 (G)
7	0.64 (G)	0.57 (G)	0.58 (G)
8	0.65 (G)	0.57 (G)	0.58 (G)
9	0.67 (N)	0.57 (G)	0.73 (T)
10	0.70 (N)	0.59 (G)	0.73 (T)
11	0.72 (N)	0.61 (G)	0.73 (T)
12	0.74 (N)	0.61 (G)	0.73 (T)
13	0.76 (N)	0.62 (G)	0.73 (T)
14	0.78 (N)	0.63 (G)	0.73 (T)
15	0.82 (N)	0.64 (G)	0.73 (T)
16	0.84 (N)	0.64 (G)	0.73 (T)
17	0.86 (N)	0.67 (N)	0.9 (HT)
18	0.88 (N)	0.67 (N)	0.9 (HT)
19	0.9 (N)	0.67 (N)	0.9 (HT)
20	0.92 (N)	0.67 (N)	0.9 (HT)

Table 30. Calculating the trust value based on the slow-rise strategy in FUBT.

Time Slot	Direct Trust	History Trust	Indirect Trust	Comprehensive Trust (FUBT)
1	0.52 ((G)	0.52 (G)	0.81 (T)	0.58 (G)
2	0.54 (G)	0.58 (G)	0.79 (T)	0.58 (G)
3	0.56 (G)	0.58 (G)	0.96 (T)	0.58 (G)
4	0.58 (G)	0.58 (G)	0.88 (T)	0.58 (G)
5	0.6 (G)	0.58 (G)	0.93 (T)	0.58 (G)
6	0.62 (G)	0.58 (G)	0.8 (T)	0.58 (G)
7	0.64 (G)	0.58 (G)	0.89 (T)	0.58 (G)
8	0.65(G)	0.58 (G)	0.92 (T)	0.58 (G)
9	0.67 (T)	0.58 (G)	0.92 (T)	0.73 (T)
10	0.7 (T)	0.73(T)	0.97 (T)	0.73 (T)
11	0.72 (T)	0.73(T)	0.9 (T)	0.73 (T)
12	0.74 (T)	0.73(T)	0.8 (T)	0.74 (T)
13	0.76 (T)	0.74 (T)	0.8 (T)	0.74 (T)
14	0.78 (T)	0.74 (T)	0.81 (T)	0.73 (T)
15	0.82(HT)	0.73 (T)	0.79 (T)	0.9 (HT)
16	0.84(HT)	0.9 (T)	0.91 (T)	0.9 (HT)
17	0.86 (HT)	0.9 (T)	0.91 (T)	0.9 (HT)
18	0.88 (HT)	0.9 (T)	0.9 (T)	0.9 (HT)
19	0.9 (HT)	0.9 (T)	0.85 (T)	0.9 (HT)
20	0.92 (HT)	0.9 (T)	0.9 (T)	0.9 (HT)

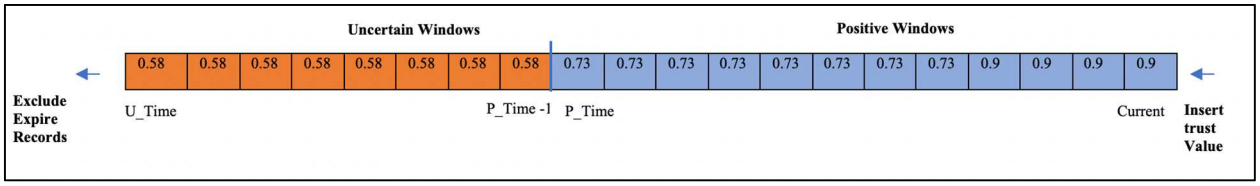

Figure 36. Sliding windows for FMUBCT.

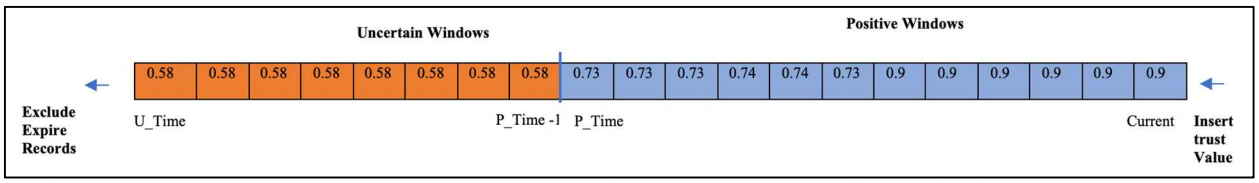

Figure 37. Sliding windows for FUBT.

5.2.1.3. The Third Case: Evaluate the Punishment (Rapid-Decrease) Strategy and Repeating Abnormal Behavior

We evaluate the FMUBCT and the FUBT when the user behaves abnormally. Figure 38 illustrates that our proposed models can detect the malicious behavior at time 10 when the user behaves abnormally by carrying a virus to the cloud, leading to a rapidly decreased user-trust value. At time 10, the FMUBCT uses equations 6-9 to calculate direct trust. The result of direct trust is 0.44, and the average for the history trust is 0.68. Then, values for direct trust and history trust are sent to the fuzzy logic module to compute the comprehensive-trust value. Based on fuzzy rule 8, if direct trust is suspicious and the history trust is general, then the comprehensive trust is suspect. Thus, the comprehensive-trust value is 0.4.

FUBT uses fuzzy logic in three phases; the computation of security evidence in phase I is 0.2, which is distrust. Then, the value is sent to phase two to compute the direct trust based on a comparison of the evidence values (login, security, operation, and performance); based on fuzzy rule 12, if SE is distrust while LE, OE, and PE are trusted, then DT is suspect; the result of direct trust is 0.33. Finally, compare direct trust, history trust, and indirect trust in phase III based on fuzzy rule 18, If DT is suspect while HT and RT are Trusted, then CT is suspect; therefore, the comprehensive trust value is 0.34.

From time 11 to 20, the user behaves normally in the cloud, so based on our slow-rise strategy, the user's trust value is slowly increasing. Figure 38 shows the difference between our proposed models at times 11 to 20; the variation occurs because the FMUBCT uses the average

for history trust while FUBT uses the last comprehensive trust as the history trust. Moreover, the difference of the values on proposed model are based on their fuzzy logic rules. In conclusion, the FMUBCT and FUBT models reduce the user's trust value by applying "rapid decrease" through their fuzzy rules. After the punishment, the user's trust value increases slowly when the user continues to behave normally.

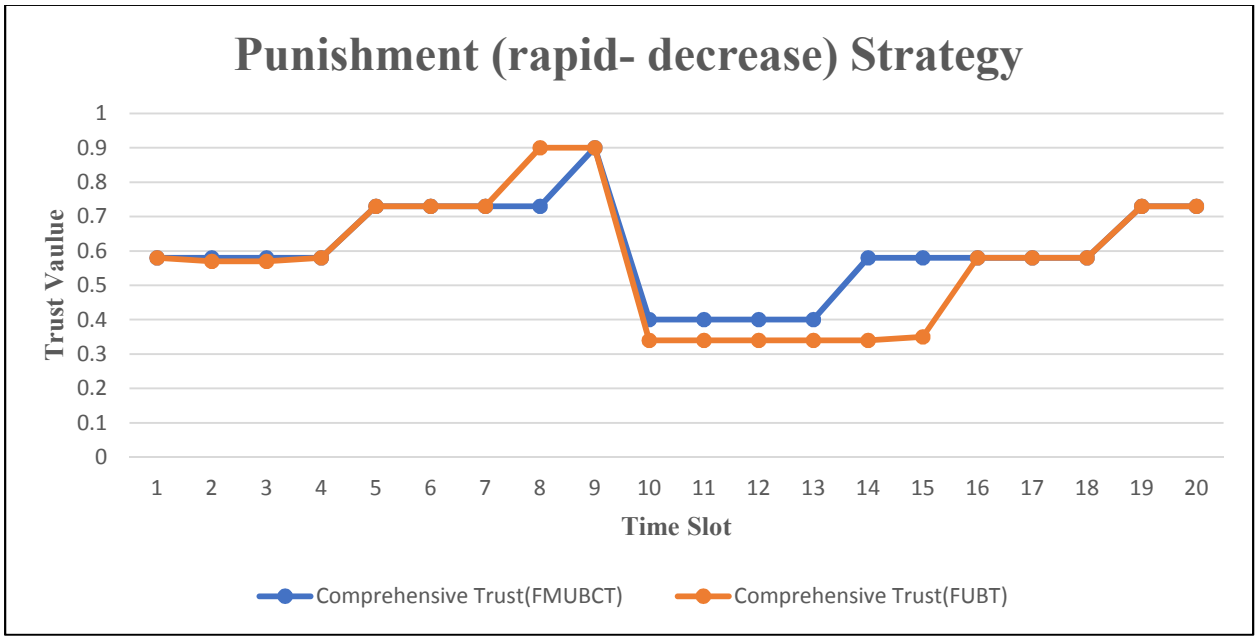

Figure 38. Evaluating the punishment (rapid-decrease) strategy.

Figure 39 and Table 31 illustrate a user continuing to repeat malicious behavior from times 18 to 20, thus the FMUBCT punishes the user by reducing the trust value using equations 16-17 in Section 3.4.3. In equation 17, we use 0.3 as the value for γ. The FUBT punishes a user by utilizing the strategy described in Section 4.2.3.2. The FMUBCT and FUBT punish users by decreasing the trust value until a user reaches the distrust degree, leading to the user being denied access to the cloud.

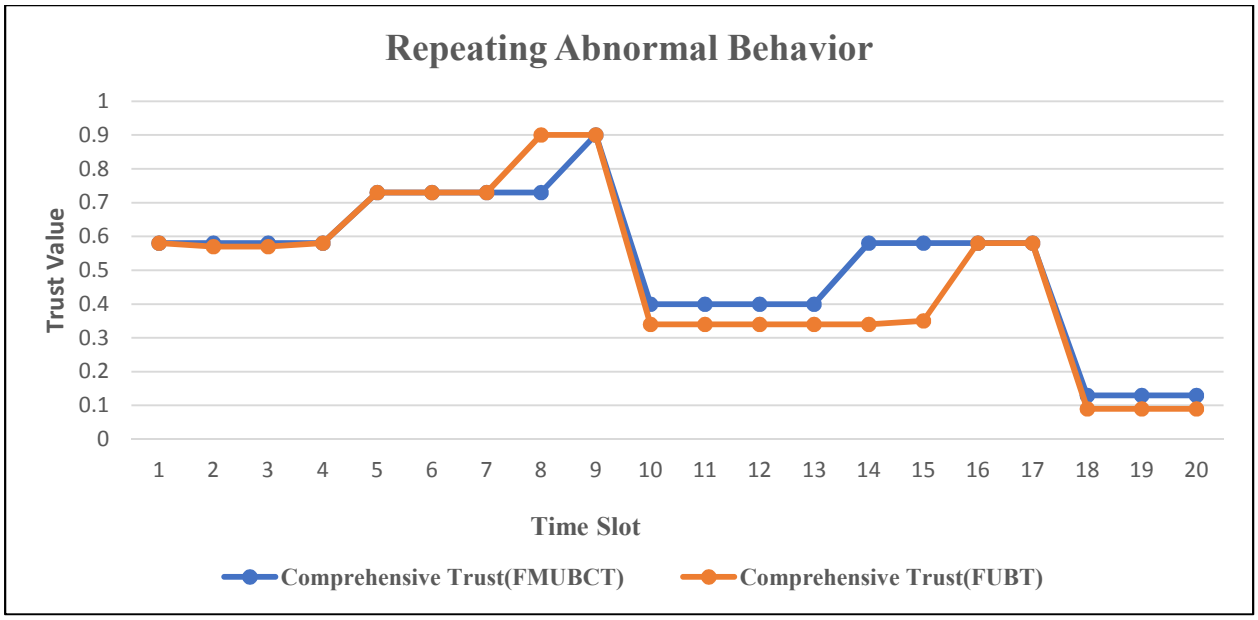

Figure 39. Evaluating repeated abnormal behavior.

Table 31. Evaluating repeated abnormal behavior.

Time Slot	Comprehensive Trust (FMUBCT)	Comprehensive Trust (FUBT)
1	0.58	0.58
2	0.58	0.57
3	0.58	0.57
4	0.58	0.58
5	0.73	0.73
6	0.73	0.73
7	0.73	0.73
8	0.73	0.9
9	0.9	0.9
10	0.4	0.34
11	0.4	0.34
12	0.4	0.34
13	0.4	0.34
14	0.58	0.34
15	0.58	0.35
16	0.58	0.58
17	0.58	0.58
18	0.13	0.09
19	0.13	0.09
20	0.13	0.09

5.2.1.4. The Fourth Case: Changing Different Trust Types

User-behavior characteristics vary. While historical behavior data can introduce a collection of trusted behaviors, the user's behavior does not solely rely on the historical behavior. Thus, only using historical data to evaluate the users' behavior is too indiscriminate. Therefore, in the FMUBCT and FUBT, we consider different trust types to compute the users' trust value.

In the FMUBCT, the changing trend for the three trust types is shown in Figure 40 and Table 32. At time 16 when the user logs in to the cloud at the usual time and spends more time there than usual, the direct trust decreases to reach a suspicious level at 0.49. Then, based on the fuzzy rule described in Section 3.3.2.4, even though the user's history trust is at the trust level, the comprehensive trust is 0.41 as a suspicious user. In addition, Figure 40 shows how our models consider the more recent behavior (direct trust) and have more influence on the trust value. The symbols in the tables present the membership functions: N is normal; S is suspicious; G is general; T is trusted; and HT is High Trusted.

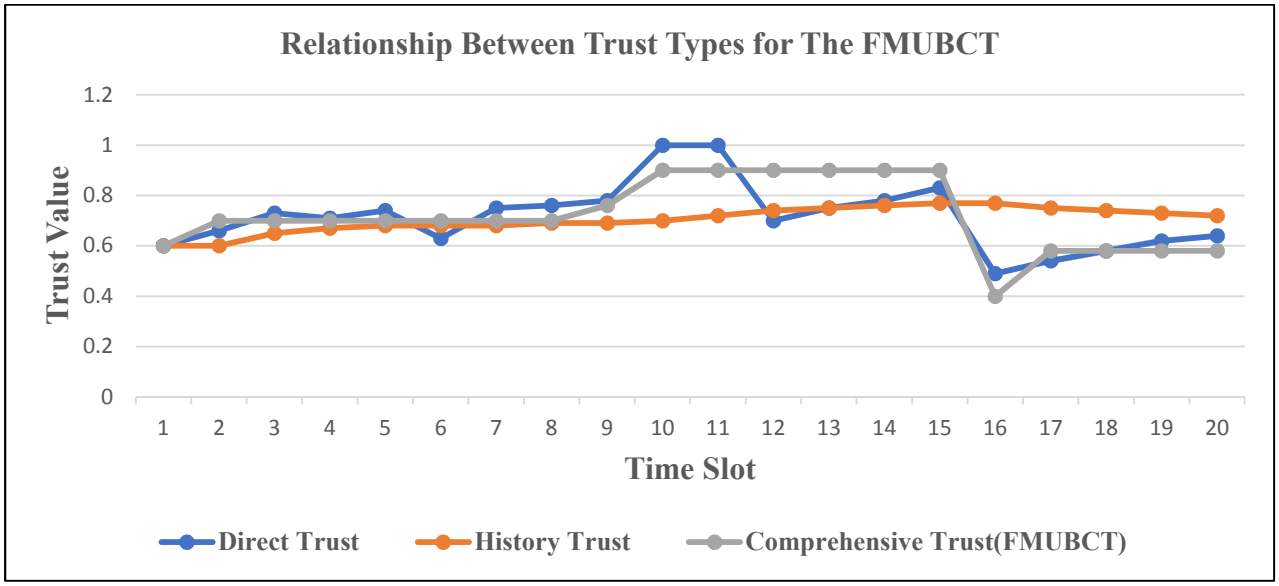

Figure 40. Changing the direct and history trust affect the comprehensive trust.

Table 32. Changing the direct and history trust affect the comprehensive trust.

Time Slot	Direct Trust	History Trust	Comprehensive Trust (FMUBCT)
1	0.6 (G)	0.6 (G)	0.6 (G)
2	0.66 (N)	0.6 (G)	0.7 (T)
3	0.73 (N)	0.65 (G)	0.7 (T)
4	0.71 (N)	0.67 (N)	0.7 (T)
5	0.74 (N)	0.68 (N)	0.7 (T)
6	0.63 (G)	0.68 (N)	0.7 (T)
7	0.75 (N)	0.68 (N)	0.7 (T)
8	0.76 (N)	0.69 (N)	0.7 (T)
9	0.78 (N)	0.69 (N)	0.76 (T)
10	1 (N)	0.7 (N)	0.9 (HT)
11	1 (N)	0.72 (N)	0.9 (HT)
12	0.7 (N)	0.74 (N)	0.9 (HT)
13	0.75 (N)	0.75 (N)	0.9 (HT)
14	0.78 (N)	0.76 (N)	0.9 (HT)
15	0.83 (N)	0.77 (N)	0.9 (HT)
16	0.50 (S)	0.77 (N)	0.4 (S)
17	0.62 (G)	0.75 (N)	0.58 (G)
18	0.65 (G)	0.74 (N)	0.58 (G)
19	0.67 (N)	0.73 (N)	0.58 (G)
20	0.7 (N)	0.72 (N)	0.58 (G)

Figure 41 shows the simulation result with the crisp trust value. Columns 1 and 2 represent a simulated crisp value for direct trust and history trust, respectively. Column 3 represents a fuzzified trust value based on the 16 defined rules. In Figure 41, the direct trust is at general trust, and the history trust is at general trust; then, the comprehensive trust is at general trust, too.

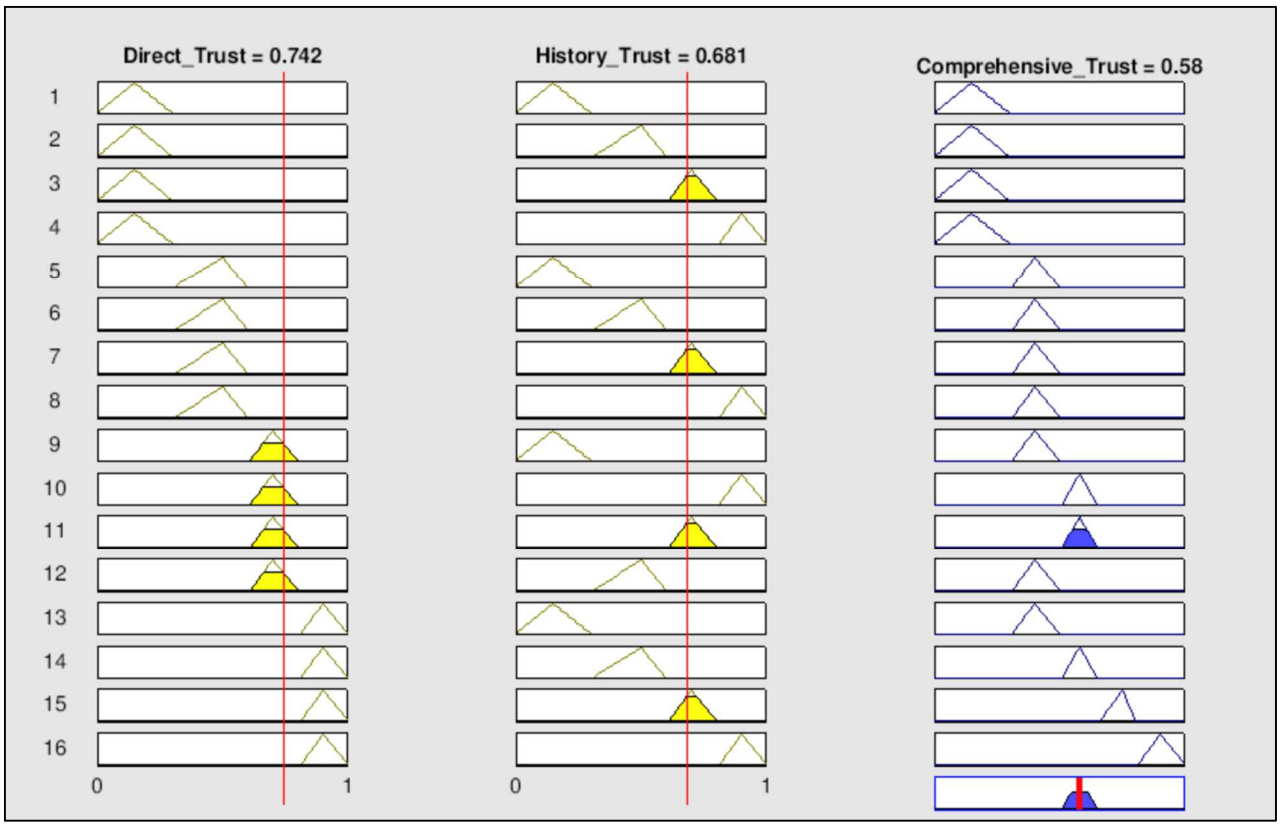

Figure 41. The FMUBCT output as a rule viewer.

For the FUBT, the changing trend with three trust types is shown in Figure 42 and Table 33. When the indirect trust values decreases at time 15, the history and direct trust are at the trust level, based on the FUBT model, the comprehensive trust is affected. Then, at time 16, when the user behaves normally in the cloud based on the direct trust value, the user receives high repetition from other users, leading to increased indirect trust; then, the comprehensive trust increases. From this experiment, we conclude that it is important to consider more than one trust type in order to evaluate the users' behavior in the cloud. The symbols in the tables present the membership functions: N is normal; S is suspicious; G is general; D is distrust; T is trusted; and HT is High Trusted.

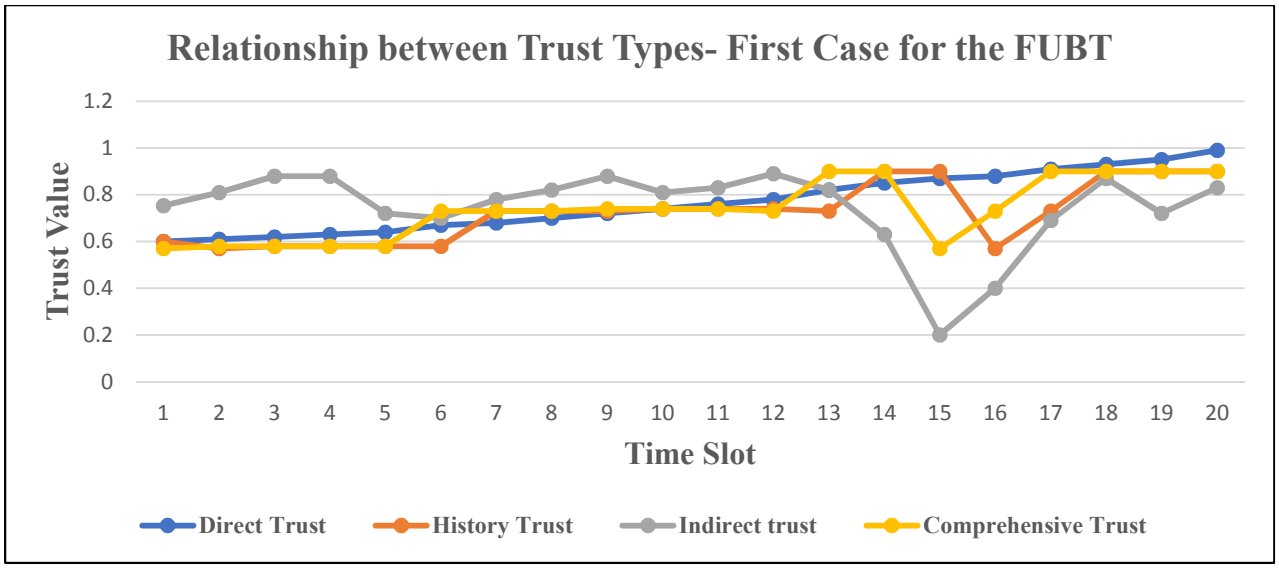

Figure 42. Relationship between trust types- first case for the FUBT.

Table 33. Relationship between trust types: first case for the FUBT.

Time Slot	Direct Trust	History Trust	Indirect trust	Comprehensive Trust
1	0.6 (G)	0.6 (G)	0.75 (T)	0.57 (G)
2	0.61 (G)	0.57 (G)	0.81 (T)	0.58 (G)
3	0.62 (G)	0.58 (G)	0.88 (T)	0.58 (G)
4	0.63 (G)	0.58 (G)	0.88 (T)	0.58 (G)
5	0.64 (G)	0.58 (G)	0.72 (T)	0.58 (G)
6	0.67 (T)	0.58 (G)	0.7 (T)	0.73 (T)
7	0.68 (T)	0.73 (T)	0.78 (T)	0.73 (T)
8	0.7 (T)	0.73 (T)	0.82 (T)	0.73 (T)
9	0.72 (T)	0.73 (T)	0.88 (T)	0.74 (T)
10	0.74 (T)	0.74 (T)	0.81 (T)	0.74 (T)
11	0.76 (T)	0.74 (T)	0.83 (T)	0.74 (T)
12	0.78 (T)	0.74 (T)	0.89 (T)	0.73 (T)
13	0.82 (HT)	0.73 (T)	0.82 (T)	0.9 (HT)
14	0.85 (HT)	0.9 (HT)	0.63 (T)	0.9 (HT)
15	0.87 (HT)	0.9 (HT)	0.2 (D)	0.57 (G)
16	0.88 (HT)	0.57 (G)	0.4 (S)	0.73 (T)
17	0.91 (HT)	0.73 (T)	0.69 ((T)	0.9 (HT)
18	0.93 (HT)	0.9 (HT)	0.87 (T)	0.9 (HT)
19	0.95 (HT)	0.9 (HT)	0.72 (T)	0.9 (HT)
20	0.99(HT)	0.9 (HT)	0.83 (T)	0.9(HT)

We run the second case to demonstrate that the value for comprehensive trust is affected when the value for any trust type decreases. Figure 43 and Table 34 show that the user behaves normally in the cloud until time 14 when the user behaves abnormally in the cloud by carrying a virus to the cloud; then, direct trust decreases. Thus, comprehensive trust decreases based on the proposed fuzzy rules in the FUBT. Figure 43 and Table 34 show how our models consider that the more recent behavior (direct trust) has more influence on the trust value. The symbols in the tables present the membership functions: S is suspicious; G is general; T is trusted; and HT is High Trusted.

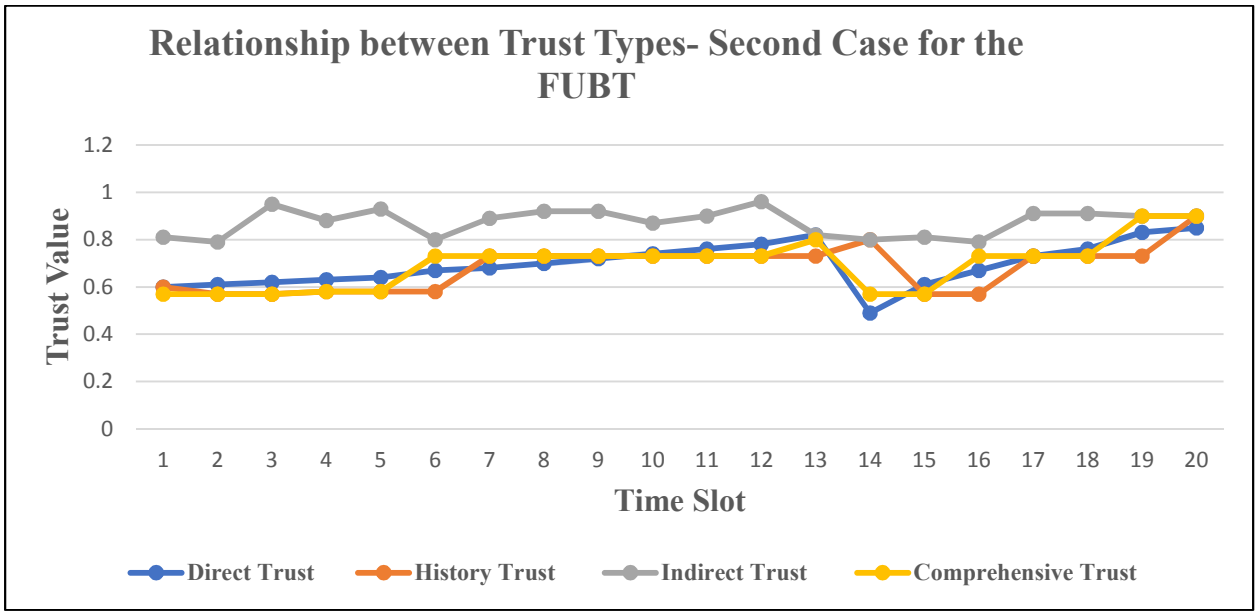

Figure 43. Relationship between trust types: second case for the FUBT.

Table 34. Relationship between trust types: second case for the FUBT.

Time Slot	Direct Trust	History Trust	Indirect Trust	Comprehensive Trust
1	0.6 (G)	0.6 (G)	0.81 (T)	0.57 (G)
2	0.61 (G)	0.57 (G)	0.79 (T)	0.57 (G)
3	0.62 (G)	0.57 (G)	0.95 (T)	0.57 (G)
4	0.63 (G)	0.58 (G)	0.88 (T)	0.58 (G)
5	0.64 (G)	0.58 (G)	0.93 (T)	0.58 (G)
6	0.67 (T)	0.58 (G)	0.80 (T)	0.73 (T)
7	0.68 (T)	0.73 (T)	0.89 (T)	0.73 (T)
8	0.7 (T)	0.73 (T)	0.92 (T)	0.73 (T)
9	0.72 (T)	0.73 (T)	0.92 (T)	0.73 (T)
10	0.74 (T)	0.73 (T)	0.87 (T)	0.73 (T)
11	0.76 (T)	0.73 (T)	0.90 (T)	0.73 (T)
12	0.78 (T)	0.73 (T)	0.96 (T)	0.73 (T)
13	0.82 (HT)	0.73 (T)	0.82 (T)	0.8 (T)
14	0.49 (S)	0.8 (T)	0.80 (T)	0.57(G)
15	0.61(G)	0.57 (G)	0.81 (T)	0.57(G)
16	0.67(T)	0.57(G)	0.79 (T)	0.73 (T)
17	0.73 (T)	0.73 (T)	0.91 (T)	0.73 (T)
18	0.76 (T)	0.73 (T)	0.91 (T)	0.73 (T)
19	0.83 (HT)	0.73 (T)	0.90 (T)	0.9 (HT)
20	0.85(HT)	0.9 (HT)	0.90 (T)	0.9 (HT)

5.2.1.5. The Fifth Case: Fault Propagation

In this case, the models were tested based on idea of fault propagation from software engineering concepts [86]. we manually created 5 users to simulate behavior for 5 types of trust degree (distrust, suspicious, general trust, trusted, and High Trusted). For every user, we manually generated 20 records. Each record has 12 factors (user Id, the result of comparison history Ip with current Ip, the result of comparison history login time with current login time, the result of comparison history services with current service, the result of comparison history action

with current action, the result of comparison history duration with current duration, does user scan important port ?does user carry virus? dose user connect use illegal connection? dose user type a sensitive keyword? how much does user utilize CPU? how much does user utilize memory? how much does user utilize disk space.

Table 35 illustrate test cases for distrust user. The user login to system with normal behavior until time 6 user scans an important port then at time 7 user repeat malicious behavior by scanning an important port. Based on FMUBCT and FUBT, trust value decrease to suspicious level Which is (0.4) in The FMUBCT , and (0.34) in the FUBT , then by repeating the malicious behavior the trust value must decrease to distrust level Which is (0.13) in THE FMUBCT , and (0.09) in the FUBT . Thus, if the user's is distrust, the FMUBCT and FUBT denied user access to the cloud.

Table 35. Fault propagation for distrust user.

Case #	Record Details	Comprehensive Trust (FMUBCT)			Comprehensive Trust (FUBT)		
		Actual	Expect	P/F	Actual	Expect	P/F
1	(1,1,1,1,1,1,1,1,1,1,0.15,0.20,0.10)	0.58	0.58	P	0.58	0.58	P
2	(1,1,1,1,1,1,1,1,1,1,0.18,0.20,0.10)	0.58	0.58	P	0.58	0.58	P
3	(1,1,1,1,1,1,1,1,1,1,0.15,0.17,0.14)	0.58	0.58	P	0.58	0.58	P
4	(1,1,1,1,1,1,1,1,1,1,0.15,0.08,0.10)	0.73	0.73	P	0.73	0.73	P
5	(1,1,1,1,1,1,1,1,1,1,0.20,0.20,0.10)	0.73	0.73	P	0.73	0.73	P
6	(1,1,1,1,1,1,0,1,1,1,0.15,0.20,0.10)	0.4	0.4	P	0.34	0.34	P
7	(1,1,1,1,1,1,0,1,1,1,0.15,0.20,0.20)	0.13	0.13	P	0.09	0.09	P

Table 36 illustrate test cases for suspicious user. From time 1 to time 17 user behave normally on the cloud however, at time 18 user access unusual service at usual login time, the FMUBCT and FUBT are able to detect this behavior then reduce the trust level to suspicious level. The suspicious degree is (0.4) in THE FMUBCT , and (0.34) in the FUBT At time 19

user is back to behave normally to cloud however based on slow rise and punishment strategies the trust level is still on suspicious level then increase slowly within the time.

Table 36. Fault propagation for suspicious user.

Case #	Record Details	Comprehensive Trust (FMUBCT)			Comprehensive Trust (FUBT)		
		Actual	Expect	P/F	Actual	Expect	P/F
1	(2,1,1,1,1,1,1,1,1,1,0.15,0.20,0.10)	0.58	0.58	P	0.58	0.58	P
2	(2,1,1,1,1,1,1,1,1,1,0.18,0.20,0.10)	0.58	0.58	P	0.58	0.58	P
3	(2,1,1,1,1,1,1,1,1,1,0.15,0.17,0.14)	0.58	0.58	P	0.58	0.58	P
4	(2,1,1,1,1,1,1,1,1,1,0.15,0.08,0.10)	0.73	0.73	P	0.73	0.73	P
5	(2,1,1,1,1,1,1,1,1,1,0.20,0.20,0.10)	0.73	0.73	P	0.73	0.73	P
6	(2,1,1,1,1,1,1,1,1,1,0.15,0.20,0.10)	0.73	0.73	P	0.73	0.73	P
7	(2,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.15)	0.73	0.73	P	0.73	0.73	P
8	(2,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.20)	0.73	0.73	P	0.73	0.73	P
9	(2,1,1,1,1,1,1,1,1,1,0.18,0.0.25,0.15)	0.73	0.73	P	0.74	0.74	P
10	(2,1,1,1,1,1,1,1,1,1,0.14,0.11,0.25)	0.73	0.73	P	0.74	0.74	P
11	(2,1,1,1,1,1,1,1,1,1,0.28,0.0.5,0.15)	0.73	0.73	P	0.73	0.73	P
12	(2,1,1,1,1,1,1,1,1,1,0.09,0.0.5,0.15)	0.73	0.73	P	0.9	0.9	P
13	(2,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.23)	0.73	0.73	P	0.9	0.9	P
14	(2,1,1,1,1,1,1,1,1,1,0.10,0.14,0.20)	0.9	0.9	P	0.9	0.9	P
15	(2,1,1,1,1,1,1,1,1,1,0.14,0.12,0.25)	0.9	0.9	P	0.9	0.9	P
16	(2,1,1,1,1,1,1,1,1,1,0.10,0.14,0.20)	0.9	0.9	P	0.9	0.9	P
17	(2,1,1,1,1,1,1,1,1,1,0.14,0.12,0.25)	0.9	0.9	P	0.9	0.9	P
18	(2,1,0,0,1,1,1,1,1,1,0.14,0.12,0.25)	0.4	0.4	P	0.34	0.34	P
19	(2,1,1,1,1,1,1,1,1,1,0.15,0.12,0.13)	0.4	0.4	P	0.34	0.34	P
20	(2,1,1,1,1,1,1,1,1,1,0.14,0.15,0.25)	0.4	0.4	P	0.34	0.34	P

Table 37 illustrate test cases for general trust user. From time 1 to time 18 user behaves normally on the cloud however, at time 19 user spend more time than the usual time, the FMUBCT and FUBT are able to detect abnormal behavior then reduce the trust level to general trust level.

Table 37. Fault propagation for general trust user.

Case #	Record Details	Comprehensive Trust (FMUBCT)			Comprehensive Trust (FUBT)		
		Actual	Expect	P/F	Actual	Expect	P/F
1	(3,1,1,1,1,1,1,1,1,1,0.15,0.20,0.10)	0.58	0.58	P	0.58	0.58	P
2	(3,1,1,1,1,1,1,1,1,1,0.18,0.20,0.10)	0.58	0.58	P	0.58	0.58	P
3	(3,1,1,1,1,1,1,1,1,1,0.15,0.17,0.14)	0.58	0.58	P	0.58	0.58	P
4	(3,1,1,1,1,1,1,1,1,1,0.15,0.08,0.10)	0.73	0.73	P	0.73	0.73	P
5	(3,1,1,1,1,1,1,1,1,1,0.20,0.20,0.10)	0.73	0.73	P	0.73	0.73	P
6	(3,1,1,1,1,1,1,1,1,1,0.15,0.20,0.10)	0.73	0.73	P	0.73	0.73	P
7	(3,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.15)	0.73	0.73	P	0.73	0.73	P
8	(3,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.20)	0.73	0.73	P	0.73	0.73	P
9	(3,1,1,1,1,1,1,1,1,1,0.18,0.0.25,0.15)	0.73	0.73	P	0.74	0.74	P
10	(3,1,1,1,1,1,1,1,1,1,0.14,0.11,0.25)	0.73	0.73	P	0.74	0.74	P
11	(3,1,1,1,1,1,1,1,1,1,0.28,0.0.5,0.15)	0.73	0.73	P	0.73	0.73	P
12	(3,1,1,1,1,1,1,1,1,1,0.09,0.0.5,0.15)	0.73	0.73	P	0.9	0.9	P
13	(3,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.23)	0.73	0.73	P	0.9	0.9	P
14	(3,1,1,1,1,1,1,1,1,1,0.10,0.14,0.20)	0.9	0.9	P	0.9	0.9	P
15	(3,1,1,1,1,1,1,1,1,1,0.14,0.12,0.25)	0.9	0.9	P	0.9	0.9	P
16	(3,1,1,1,1,1,1,1,1,1,0.10,0.14,0.20)	0.9	0.9	P	0.9	0.9	P
17	(3,1,1,1,1,1,1,1,1,1,0.14,0.12,0.25)	0.9	0.9	P	0.9	0.9	P
18	(3,1,1,1,1,1,1,1,1,1,0.14,0.12,0.25)	0.9	0.9	P	0.9	0.9	P
19	(3,1,1,1,1,1,1,1,1,1,0.15,0.12,0.13)	0.58	0.58	P	0.58	0.58	P
20	(3,1,1,1,1,1,1,1,1,1,0.14,0.15,0.25)	0.58	0.58	P	0.58	0.58	P

Table 38 illustrate test cases for trust user. From time 1 to time 15 user behaves normally on the cloud, however. However, this user doesn't have many records to reach high level trust. Thus, the trust level increase from general trust (0.58) to trust (0.73) but doesn't reach high level trust.

Table 38. Fault propagation for trustworthy user.

Case #	Record Details	Comprehensive Trust (FMUBCT)			Comprehensive Trust (FUBT)		
		Actual	Expect	P/F	Actual	Expect	P/F
1	(5,1,1,1,1,1,1,1,1,1,0.15,0.20,0.10)	0.58	0.58	P	0.58	0.58	P
2	(5,1,1,1,1,1,1,1,1,1,0.10,0.12,0.10)	0.58	0.58	P	0.58	0.58	P
3	(5,1,1,1,1,1,1,1,1,1,0.15,0.15,0.20)	0.58	0.58	P	0.58	0.58	P
4	(5,1,1,1,1,1,1,1,1,1,0.15,0.0.16,0.20)	0.58	0.58	P	0.58	0.58	P
5	(5,1,1,1,1,1,1,1,1,1,0.10,0.18,0.20)	0.58	0.58	P	0.58	0.58	P
6	(5,1,1,1,1,1,1,1,1,1,0.17,0.15,0.15)	0.58	0.58	P	0.58	0.58	P
7	(5,1,1,1,1,1,1,1,1,1,0.18,0.15,0.15)	0.58	0.58	P	0.58	0.58	P
8	(5,1,1,1,1,1,1,1,1,1,0.17,0.15,0.23)	0.58	0.58	P	0.58	0.58	P
9	(5,1,1,1,1,1,1,1,1,1,0.14,0.15,0.25)	0.73	0.73	P	0.73	0.73	P
10	(5,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.15)	0.73	0.73	P	0.73	0.73	P
11	(5,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.20)	0.73	0.73	P	0.73	0.73	P
12	(5,1,1,1,1,1,1,1,1,1,0.18,0.0.25,0.15)	0.73	0.73	P	0.74	0.74	P
13	(5,1,1,1,1,1,1,1,1,1,0.14,0.11,0.25)	0.73	0.73	P	0.74	0.74	P
14	(5,1,1,1,1,1,1,1,1,1,0.28,0.0.5,0.15)	0.73	0.73	P	0.73	0.73	P
15	(5,1,1,1,1,1,1,1,1,1,0.09,0.0.5,0.15)	0.73	0.73	P	0.9	0.9	P

Table 39 illustrate test cases for high trust user. Because of the user behaves normally on the cloud ,the trust level increase from general trust (0.58) to trust (0.73) to reach (0.9) high trust level.

Table 39. Fault propagation for high trustworthy user.

Case #	Record Details	Comprehensive Trust (FMUBCT)			Comprehensive Trust (FUBT)		
		Actual	Expect	P/F	Actual	Expect	P/F
1	(5,1,1,1,1,1,1,1,1,1,0.15,0.20,0.10)	0.58	0.58	P	0.58	0.58	P
2	(5,1,1,1,1,1,1,1,1,1,0.10,0.12,0.10)	0.58	0.58	P	0.58	0.58	P
3	(5,1,1,1,1,1,1,1,1,1,0.15,0.15,0.20)	0.58	0.58	P	0.58	0.58	P
4	(5,1,1,1,1,1,1,1,1,1,0.15,0.0.16,0.20)	0.58	0.58	P	0.58	0.58	P
5	(5,1,1,1,1,1,1,1,1,1,0.10,0.18,0.20)	0.58	0.58	P	0.58	0.58	P
6	(5,1,1,1,1,1,1,1,1,1,0.17,0.15,0.15)	0.58	0.58	P	0.58	0.58	P
7	(5,1,1,1,1,1,1,1,1,1,0.18,0.15,0.15)	0.58	0.58	P	0.58	0.58	P
8	(5,1,1,1,1,1,1,1,1,1,0.17,0.15,0.23)	0.58	0.58	P	0.58	0.58	P
9	(5,1,1,1,1,1,1,1,1,1,0.14,0.15,0.25)	0.73	0.73	P	0.73	0.73	P
10	(5,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.15)	0.73	0.73	P	0.73	0.73	P
11	(5,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.20)	0.73	0.73	P	0.73	0.73	P
12	(5,1,1,1,1,1,1,1,1,1,0.18,0.0.25,0.15)	0.73	0.73	P	0.74	0.74	P
13	(5,1,1,1,1,1,1,1,1,1,0.14,0.11,0.25)	0.73	0.73	P	0.74	0.74	P
14	(5,1,1,1,1,1,1,1,1,1,0.28,0.0.5,0.15)	0.73	0.73	P	0.73	0.73	P
15	(5,1,1,1,1,1,1,1,1,1,0.09,0.0.5,0.15)	0.73	0.73	P	0.9	0.9	P
16	(5,1,1,1,1,1,1,1,1,1,0.18,0.0.5,0.23)	0.73	0.73	P	0.9	0.9	P
17	(5,1,1,1,1,1,1,1,1,1,0.10,0.14,0.20)	0.9	0.9	P	0.9	0.9	P
18	(5,1,1,1,1,1,1,1,1,1,0.14,0.12,0.25)	0.9	0.9	P	0.9	0.9	P
19	(5,1,1,1,1,1,1,1,1,1,0.10,0.18,0.18)	0.9	0.9	P	0.9	0.9	P
20	(5,1,1,1,1,1,1,1,1,1,0.10,0.18,0.25)	0.9	0.9	P	0.9	0.9	P

5.2.2. Comparative Analysis with Existing Models

In this section, we compare our models with two models, a model from the literature [5] and UBADAC [54]. We run different cases to evaluate and to compare our models with the models from [5] and [54].

5.2.2.1. The First Case: The Models Compared Based on the Slow-Rise Strategy

Figure 44 illustrates that the FMUBCT and the FUBT slowly and gradually increase the trust value when the user behaves normally in the cloud. Our models start from general trust and gradually reach high trust. However, the UBADAC violates the slow-rise principle by giving the

user a high trust value with a small number of interactions in the cloud. For this comparison, we conclude that the FMUBCT and FUBT models are effective with solving the trust-fraud problem and improving cloud security because the two models adopt the slow-rise strategy.

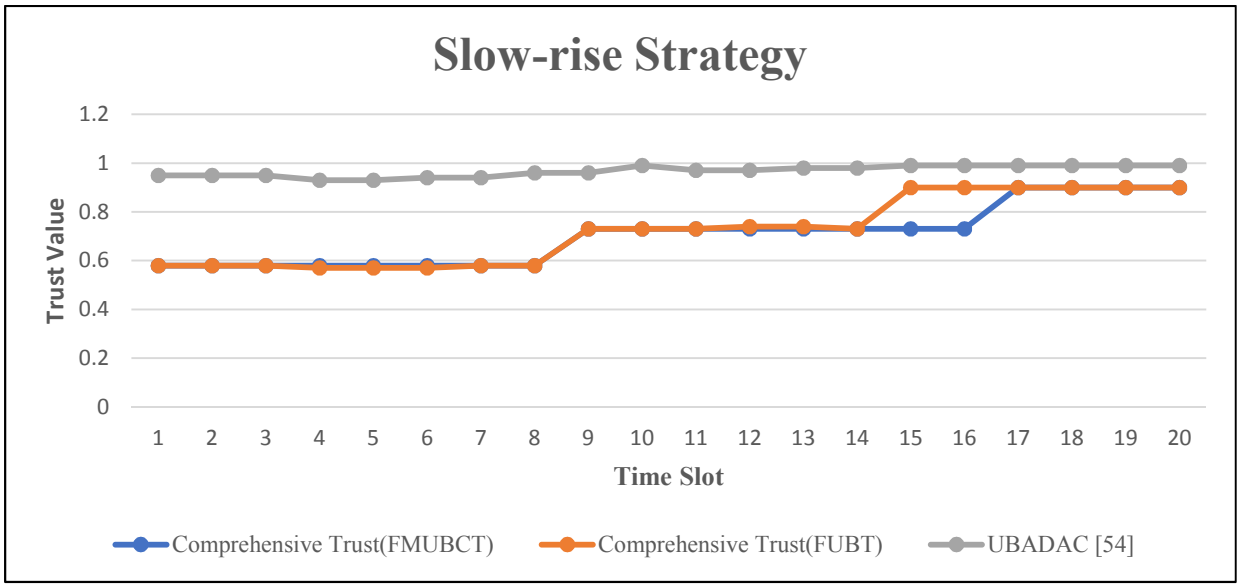

Figure 44. Model comparison based on the slow-rise strategy.

5.2.2.2. The Second Case: The Models Compared Based on Excluding the Expiration Trust

One principle for evaluating the user's behavior is that expiration records should not be considered for the trust evaluation. When the behavior records are out of date and very old, the user stopped accessing the cloud or has not accessed it recently. This user should be evaluated as a strange user. In this case, we compare the FMUBCT and FUBT with models from [5] and [54] by providing models case for genuine user stops accessing the system for a long period. The FMUBCT and FUBT set the user as a strange user with a trust value of 0.5. Also, the oldest history-trust records are squeezed to the negative range. Changing the user's trust value is shown in Figure 45.

The user's trust values are at a trusted level for times 1 to 12; because the user stopped accessing the cloud at times 13 and 14, the trust value decreases to 0.5 as a general-trust user.

Time 15 is when the genuine user is back using the cloud. Then, for times 15 to 20, the trust value increases to the trusted level because the user keeps accessing the cloud and behaves normally. However, the models in the literature [5,54] do not exclude expiration trust, and the user keeps the previous history trust even if he/she stops accessing the cloud for a long period of time. By comparison, we conclude that FMUBCT and FUBT are more effective than the literature models [5, 54] because the FMUBCT and FUBT methodologies protect the cloud infrastructure as well as its users if a genuine user's account is hacked or if a genuine user changes his/her habits.

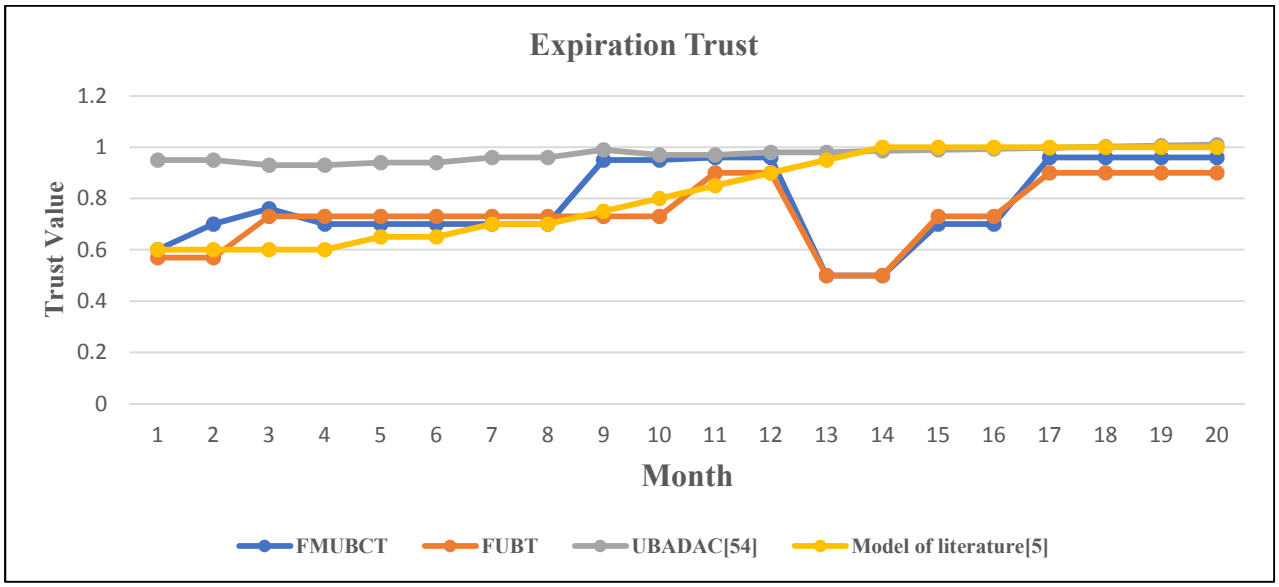

Figure 45. Model comparison based excluding the expiration trust records.

5.2.2.3. The Third Case: The Effectiveness of Detecting Malicious Behavior

To evaluate the effectiveness of detecting abnormal behavior, we test our models, model [5], and model [54] by providing abnormal cases. In the first case, the user was a trusted user until time 12 when the user types sensitive keywords in the cloud. Figure 46 shows that the security-evidence value in the FMUBCT and the FUBT rapidly decreases, affecting the direct trust. Also, the UBADAC [54] can detect abnormal behavior. However, the literature model [5]

cannot detect abnormal behavior because this model does not consider security evidence to calculate the trust value. Figure 46 shows the importance of evaluating the user's behavior based on multiple evidence types and how one type can affect the direct-trust value. In conclusion, Figure 46 demonstrates that the FMUBCT and FUBT models successfully detect abnormal behavior, leading to improved security for cloud computing.

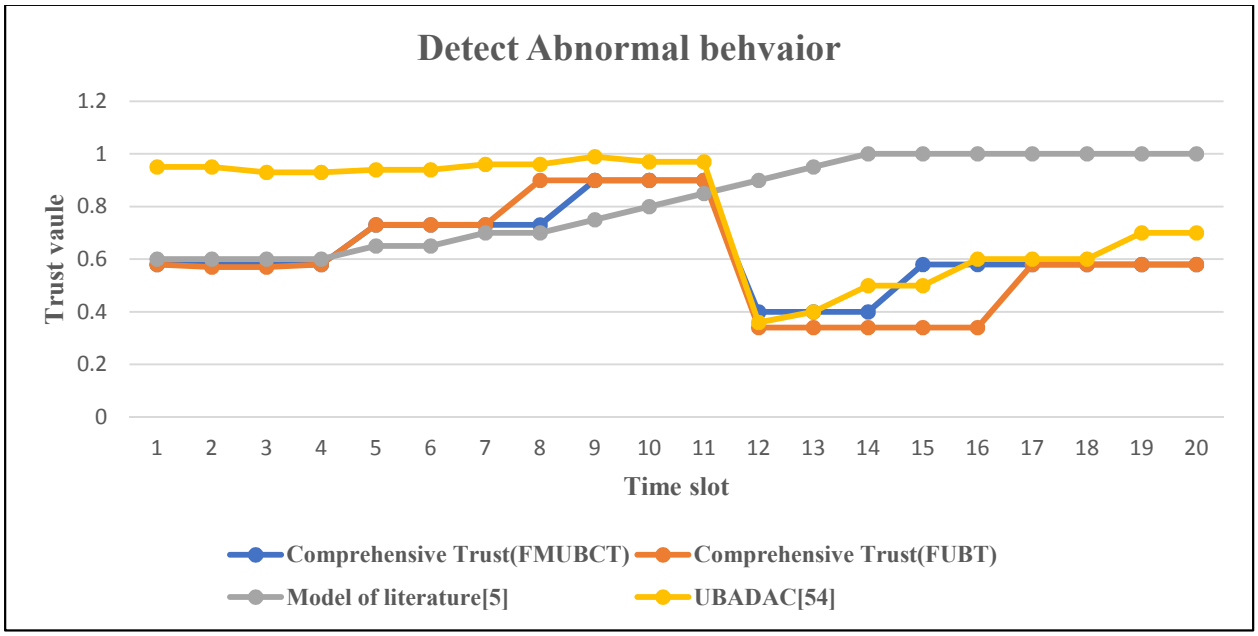

Figure 46. Model comparison for detecting malicious behavior.

5.2.2.4. The Fourth Case: Compute the Abnormal Detection Rate for the Models

Figure 47 shows the abnormal detection rate for the FMUBCT, the FUBT, the literature [5] and UBADAC [54] models. We run 250 experiments to evaluate the effectiveness of detecting abnormal behavior. The first 50 experiments have 20 abnormal behaviors: 10 operation evidence, 5 login evidence, and 5 reliability evidence. Because the FMUBCT and the FUBT do not consider reliability evidence, the FMUBCT and the FUBT detect 75% of the abnormal behavior. The UBADAC [54] and literature [5] models detect 50% because they do not consider login and reliability evidence. In 100 experiments, we provide 30 abnormal behaviors: 10 operation evidence, 5 login evidence, 5 reliability evidence, and 10 security evidence. The

detection rate is 83% for the FMUBCT and the FUBT, 66% for the UBADAC [54], and 33% for the literature models [5].

In 150 experiments, we provide 40 abnormal behaviors: 15 operation evidence, 10 login evidence, 5 reliability evidence, and 10 security evidence. The detection rate is 77% for the FMUBCT and the FUBT, 62% for the UBADAC [54], and 25% for the literature models [5]. In 200 experiments, we provide 50 abnormal behaviors: 15 operation evidence, 10 login evidence, 5 reliability evidence, 10 security evidence, and 10 performance evidence. The detection rate is 70% for the FMUBCT because the model does not consider performance evidence, 90% for the FUBT, 50% for the UBADAC [54], and 20% for the literature models [5].

In 250 experiments, we provide 60 abnormal behaviors: 15 operation evidence, 10 login evidence, 5 reliability evidence, 20 security evidence, and 10 performance evidence. The detection rate is 75% for the FMUBCT, 91% for the FUBT, 50% for the UBADAC [54], and The detection rate is 75% for the FMUBCT, 91% for the FUBT, 50% for the UBADAC [54], and 16% for the literature models [5]. Figure 47 and Table 35 indicate that the detection rate for our models is higher than the UBADAC [54] and the literature [5] models.

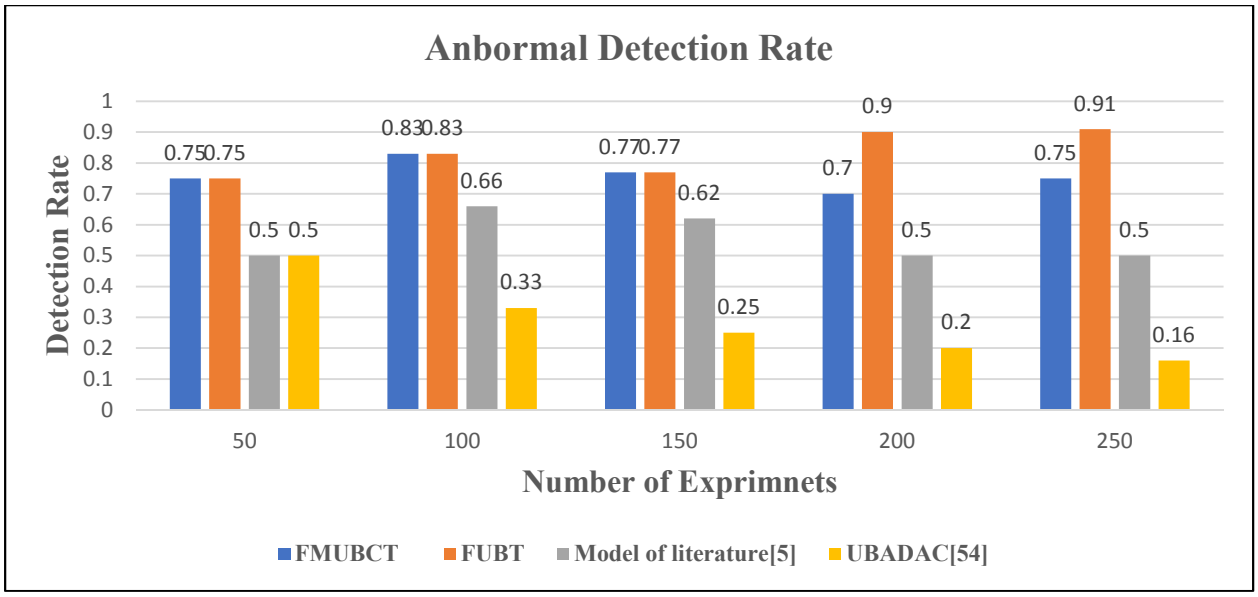

Figure 47. Abnormal detection rate.

Table 40. Abnormal detection rate.

Total Records	FMUBCT	FUBT	Model of literature [5]	UBADAC [54]
50	0.75	0.75	0.5	0.5
100	0.83	0.83	0.66	0.33
150	0.77	0.77	0.62	0.25
200	0.7	0.9	0.5	0.2
250	0.75	0.91	0.5	0.16

5.2.2.5. The Fifth Case: Comparison the Models' Performance

In this case, we run 3K records from dataset to compute the user's trust value; then, we compute the mean, variance, and standard deviation. From Figure 48 and Table 41, we conclude that our proposed models have a lower standard deviation than the UBADAC [54] and the literature model [5].

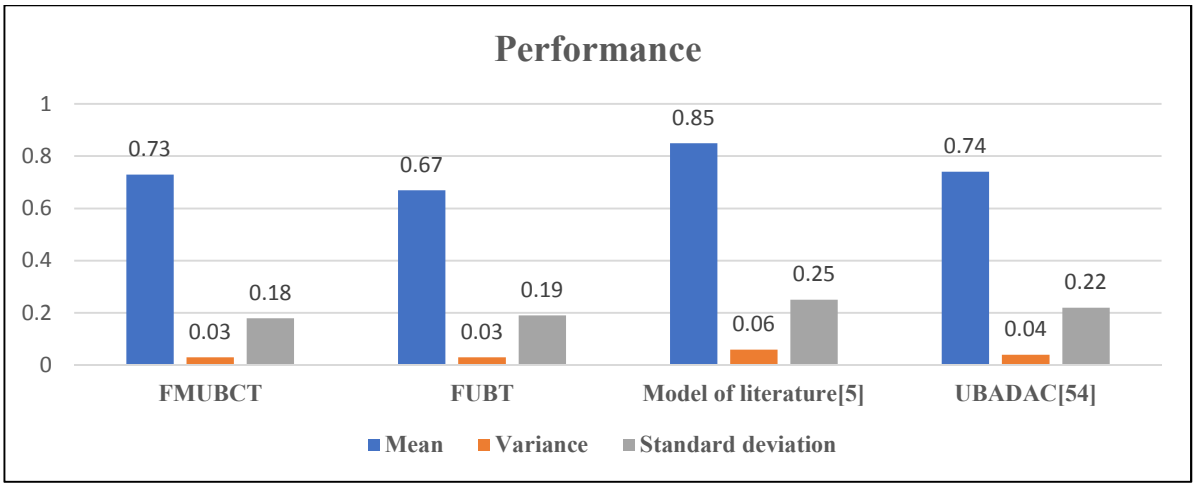

Figure 48. Comparison the models' performance.

Table 41. Comparison the models' performance.

Method	FMUBCT	FUBT	Model in the Literature [57]	UBADAC [53]
Mean	0.73	0.67	0.85	0.74
Variance	0.03	0.03	0.06	0.04
Standard Deviation	0.18	0.19	0.25	0.22

CHAPTER 6. CONCLUSION AND FUTURE WORK

6.1. Conclusion

With cloud computing increasing in popularity and providing a massive number of services, such as resources and data centers, the number of attacks in the cloud is increasing. The cloud's platform security becomes an important factor with cloud development. Basic security protection, such as traditional access control (TAC), is not able to satisfy the security requirements with the expansion of cloud computing. For example, TAC cannot prevent a malicious user from accessing the cloud. The trust-based access-control model is an efficient mechanism for security in cloud computing. Regarding the trust issues and security problems with cloud computing, and the drawbacks of the current user-behavior trust based on cloud computing, this dissertation contributes to designing and improving the current user-behavior trust models.

The dissertation contributes to the FMUBCT design to detect abnormal user behavior by extracting common user behavior and then creating user-behavior history patterns. This pattern could be used to detect insiders by comparing the user's runtime activities with historical patterns. With FMUBCT, we consider three evidence types: security, login, and operation. We consider the most important factors regarding the security evidence: the number of times an important cloud port is scanned by a user, the number of times a virus is installed in the cloud by a user, the number of times a user connects to the cloud through illegal connections, and the number of times a user types sensitive keywords. Each user exhibits a habit behavior that can help to distinguish between the user's current and historical behavior. Regarding the login evidence, we consider the user's typical IP address and preferred login time. For operating

evidence, we consider common access to the cloud service, common action in the cloud, and the user's maximum usage duration.

The user's interaction would continue to increase with the realization of time interactions. These interactions form the basis of creating the user-behavior pattern. Because the evidence's importance would decrease over time, we decided to use the sliding-window technique to describe the evidence timelines, to decrease the complexity, and to increase the speed for producing common user behavior. We established three windows: positive, uncertain, and negative. The size of each window is based on the time. Thus, for the positive windows, we filled out all behaviors from two weeks prior to the current time to the current time. The size for the uncertain windows is based on the two weeks prior to the first time in the positive time. The size of the negative windows starts from the time prior to the previous time in the uncertain windows. When the positive windows are full, the interaction shifts to the uncertain windows. Similarly, when the uncertain windows are full, the interaction is squeezed to the negative windows. In the pattern, we use the positive windows to reflect the principle that recent behaviors play a more important role in trust evaluation. To consider the creditability and slow-rise principles, we utilize uncertain windows to ensure that there is sufficient historical information and to prevent trust deception. Having sufficient interactions suggests that they exceed the uncertain windows to the positive windows. We use negative windows to exclude the expiration records.

Moreover, the FMUBCT successfully considers two types of trust—direct and history—to calculate a user's credibility. To calculate the direct trust, we parametrize the user's behavior evidence and utilize it in functional expressions to provide the basis for the trust models' framework that captures the relative importance of the behaviors in evaluating trust. We

calculate the weight for each behavior in equations 6-9. The example for the important behavior is in equation 6; when a user scans an important port, the weight for the SP factor is greater than the other factors. After the direct trust computation, the results are compared with the history trust based on fuzzy logic.

Another contribution is designing the FUBT which is improving the FMUBCT model by considering performance evidence additionally to three types of evidence: login, security, and operation to evaluate the cloud users. The performance evidence, utilizing resources such as memory, disk space, and CPU, is an important factor to distinguish between normal and malicious users. For example, the malicious user utilizes the entire memory by creating an extreme number of mail messages. For performance evidence, we consider the three factors: the user's memory occupancy rate, the user's disk-space occupancy rate, and the user's CPU occupancy rate. Because the user-behavior characteristics vary and while historical behavior data can introduce a collection of trusted behavior, user-behavior does not solely rely on historical behavior. Thus, only using historical data to evaluate the user's behavior is too indiscriminate. Therefore, we should consider different trust types. With the FUBT model, we consider four trust types to compute the user's trust value: direct, history, indirect, and comprehensive trust.

An additional contribution is based on the lack data from real system-audit logs, especially in mission-critical and senior industries such as healthcare, banking, and the military. Consequently, we built an algorithm-based probability theory using SAS (Statistical Analysis System) 9.4 to generate a dataset for use validating the dissertation's models. Because our models are based on evaluating user behavior in the cloud, we analyze the audio log from AWS API to obtain the information recorded in the cloud. For this dissertation, we created an event dataset to simulate users' real data in the cloud, where each record has the following attributes:

user ID, user packet (IP address, login date, login time, service, action, and duration), security factors (virus, illegal connection, and scanning port), and usage (memory, CPU, and disk space). In our work, we used 14 attributes to evaluate a user's behavior in the cloud. Because some data in the audit log are categorical data, we encode the categorical data as a number. There are various types of continuous probability distribution, such as normal distribution, exponential distribution, and generalized Bernoulli distribution. These types can be used to indicate the attribute values' demand distribution. In our model, we use uniform and normal distribution to produce random data. In order to obtain more accurate random events, we have used the bootstrap resampling technique.

The other contribution is the experiments. We divide the experiments into two case studies in order to demonstrate the functionality, performance, and accuracy of the proposed models. The case studies demonstrate that our proposed models effectively avoid the issue of trust fraud with the concept of a "slow–rise" strategy. We use the sliding-window technique to check the number of interactions. We have three windows: positive, uncertain, and negative. The window size is based on the number of interactions. To consider the creditability and slow-rise principles, we use uncertain windows to ensure that there is sufficient historical information and to prevent trust deception. The interactions exceed the uncertain windows to the positive windows mean we have sufficient historical information. If the number of interactions is under uncertain windows and user behaves normally, we start the trust value at 0.5 and then increase it gradually by using equation 14. If the user behaves abnormally, we retain the trust value from equations 6-9. Our proposed models can respond promptly to malicious conduct by constantly aggravating the penalty approach (principle of "rapidly decline"), efficiently preventing malicious conduct and malicious users who trust value is below the threshold. In addition, our

proposed models are capable of reflecting the user's latest credibility with the expired trust update policy and the latest trust calculation. Moreover, our proposed models evaluate users based on a large amount of evidence, such as security, operation, login, and performance; ensuring that the user's trust value is stable. Thus, the proposed models are effectively evaluating the users and successfully detecting abnormal behavior, leading to improved security for cloud computing.

In conclusion, this dissertation is a significant research work; the cloud service providers (CSPs) can use these models to improve security for cloud computing and to evaluate the trust value for all users, discriminating the user into different categories, such as high trust, trust, suspect, or malicious; then, users are authorized based on their trust value. The significance of this study is that the proposed models are flexible and scalable to consider more evidence in order to monitor and to evaluate user behavior. Thus, the CSPs can pick and choose important evidence for their systems. The other significance for this study is that no previous works successes in dynamic real-time trust computation and consider all the evaluation principles in the trust computation. Thus, the proposed models can respond to malicious behavior through dynamic, real-time trust computations and can update the user's trust value in a timely manner, re-allocating the authority degree. Finally, the significance of this study is that it uses sliding windows and limits the behavior size to reduce the complexity and to increase the speed for the dynamic, real-time trust computation.

6.2. Future Work

There are several directions for future work. Due the principle of evaluating users based on a large amount of evidence to ensure that the trust value is stable, we will add reliability evidence such as atypical data-error rate and atypical IP packet-loss rate. We also add more

factors to the evidence used in the proposed models. Another idea is to use a machine-learning technique, such as association rule mining technique, to create a user-profile pattern; then, use the pattern to detect behavior anomalies. Then, use the cluster technique to categorize users based on the current behavior compared with the history pattern from associate rule mining. One idea is to utilize deep learning in order to predicted users' behavior in the cloud based on historical behavior. Then, compare the prediction behavior with the user's current behavior to calculate the user's trust value. Because our proposed models were built with simulated data, we can evaluate our models based on the real cloud-computing environment.

REFERENCES

[1] P. Paganini, "Cost of Cybercrime Will Grow from $3 Trillion (2015) to $6 Trillion by 2021," [Online]. Available: http://securityaffairs.co/wordpress/50680/cyber-crime/global-cost-of-cybercrime.html, [Accessed 15 May. 2019].

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing." NIST special publication, 2011, pp. 2.

[3] "U.S. Federal Cloud Computing Market Forecast 2018-2023," Market Research Media, [Online]. Available: www.marketresearchmedia.com/?p=145 [Accessed: 11-Apr-2018].

[4] Q. Zhang, L. Cheng, and R. Boutaba. "Cloud Computing: State-of-the-Art and Research Challenges," Internet Serv Appl , 2010, pp .7–18, https://doi.org/10.1007/s13174-010-0007-6

[5] Y. Bendale and S. Shah, "User Level Trust Evaluation in Cloud Computing," International Journal of Computer Applications, 2013, pp .31-35, doi:10.5120/12122-8376.

[6] L. Tian, C. Lin, and Y. Ni, "Evaluation of User Behavior Trust in Cloud Computing," 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), 2010, pp .567- 572, https://doi.org/10.1109/ICCASM.2010.5620636.

[7] S. Pearson and A. Benameur, "Privacy, Security and Trust Issues Arising from Cloud Computing," 2010 IEEE 2nd International Conference on Cloud Computing Technology and Science, 2010, pp .693–702, https://doi.org/10.1109/CloudCom.2010.66.

[8] S. Almulla and C. Yeun, "Cloud Computing Security Management," 2010 2nd International Conference on Engineering System Management and Applications, 2010, pp. 1–7.

[9] Using Splunk UBA to Detect Insider Threats. Technical report, Splunk Inc., 2015. [Online]. Available: http://www.splunk.com/, [Accessed: 2016-July-29].

[10] K. Brancik, "Insider Computer Fraud; An In-depth Framework for Detecting and Defending Against Insider IT Attacks," (Boston, MA, USA: Auerbach Publications), 2007.

[11] V. Stavrou, M. Kandias, G. Karoulas, and D. Gritzalis. "Business Process Modeling for Insider Threat Monitoring and Handling." *Trust, Privacy, and Security in Digital Business*, (New York , NY, USA: Springer International Publishing), 2014, pp .19–31, https://doi.org/10.1007/978-3-319-09770-1_11.

[12] M. Bishop, H. Conboy, H. Phan, B. Simidchieva, L. Avrunin, L. Clarke, L. Osterweil, and S. Peisert,"Insider Threat Identification by Process Analysis," 2014 IEEE Security and Privacy Workshops, 2014, pp .251–64, https://doi.org/10.1109/SPW.2014.40.

[13] P. Parveen, N. Mcdaniel, Z. Weger, J. Evans, B. Thuraisingham, K. Hamlen, and L. Khan, "Evolving Insider Threat Detection Stream Mining Perspective," International Journal on Artificial Intelligence, 2013, n. p, https://doi.org/10.1142/S0218213013600130.

[14] D. Agrawal, C. Budak, A. El Abbadi, T. Georgiou, and X. Yan. "Big Data in Online Social Networks: User Interaction Analysis to Model User Behavior in Social Networks," *Databases in Networked Information Systems*, (New York ,NY,USA : Springer International Publishing), 2014, pp .1-16.

[15] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis. "The Role of Trust Management in Distributed Systems Security," Secure internet Programming, (New York, NY, USA : Springer International Publishing), 1999, pp. 185-210.

[16] R. Shaikh and M. Sasikumar, "Trust Model for a Cloud Computing Application and Service," 2012 IEEE International Conference on Computational Intelligence Computing Research (ICCIC), 2012, pp. 1-4.

[17] A. Shahzad and M. Hussain, "Security Issues and Challenges of Mobile Cloud Computing," International Journal of Grid and Distributed Computing, 2013, pp. 37-50, doi:10.14257/ijgdc.2013.6.6.04.

[18] R.Shaikh and S. Mukundan, "Security Issues in Cloud Computing: A Survey," International Journal of Computer Applications, 2012, pp. 4–10., doi:10.5120/6369-8736.

[19] "IDC Enterprise Panel: Survey 2009," [Online] Available: https://www.idc.com/ , [Accessed 15 Feb. 2019].

[20] N. Gonzalez, C. Miers, F. Redigolo, T. Carvalho, M. Simplicio, M. Naslund, and M. Pourzandi, "A Quantitative Analysis of Current Security Concerns and Solutions for Cloud Computing," 2011 IEEE 3rd International Conference on Cloud Computing Technology and Science, 2011, pp. 231–38, https://doi.org/10.1109/CloudCom.2011.39.

[21] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," 2010 Proceedings IEEE INFOCOM, 2010, pp.1–9, https://doi.org/10.1109/INFCOM.2010.5462174.

[22] C. Modi, D. Patel, B. Borisaniya, A. Patel, and M. Rajarajan, "A Survey on Security Issues and Solutions at Different Layers of Cloud Computing." The Journal of Supercomputing, 2013, pp. 561-592, http://dx.doi.org/10.1007/s11227-012-0831-5.

[23] Y. Younis, K. Kifayat, and M. Merabti, "An Access Control Model for Cloud Computing," Journal of Information Security and Applications, 2014, pp. 45-60, https://doi.org/10.1016/j.jisa.2014.04.003.

[24] M. Benantar, “Mandatory-Access-Control Mode” *Access Control Systems: Security, Identity Management and Trust Models*, (New York, NY, USA: Springer International Publishing) 2006, pp. 129-146,

[25] B. Thuraisingham, “Mandatory Access Control,” Encyclopedia of Database Systems, (New York, NY, USA: Springer International Publishing),2009, pp. 1684-1685.

[26] R.Banyal, V. Jain, and P.Jain, “Dynamic Trust Based Access Control Framework for Securing Multi-Cloud Environment,” Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies, ICTCS ’14 ,2014, pp .291–98, https://doi.org/10.1145/2677855.2677884.

[27] E. Yuan and J. Tong, “Attributed Based Access Control (ABAC) for Web Services,” IEEE International Conference on Web Services (ICWS’05), 2005, pp.569, https://doi.org/10.1109/ICWS.2005.25.

[28] M. Al-Kahtani and R. Sandhu, “A Model for Attribute-Based User-Role Assignment,” 18th Annual Computer Security Applications Conference, 2002, pp.353–62, https://doi.org/10.1109/CSAC.2002.1176307.

[29] P. Sztompka. Trust: A Sociological theory. Cambridge University Press, Cambridge shire, England, 1999.

[30] D.McKnight and N. Chervany. “Trust and Distrust Definitions: One Bite at a Time,” Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 2001, pp. 27-54.

[31] Z Yan. “Trust Management for Mobile Computing Platform.” Department of Electrical and Communications Engineering, Helsinki University of Technology, Espoo, Finland 2007.

[32] J. Anderson, “Computer Security Technology Planning Study,”[Online] Available: https://apps.dtic.mil/docs/citations/AD0758206 ,[Accessed 1 Jan. 2019].

[33] M. Blaze, J. Ioannidis, and A. Keromytis, “Experience with the Keynote Trust Management System: Applications and Future Directions,” International Conference on Trust Management ,2003, pp. 284-300.

[34] A. Abdul-Rahman and S. Hailes, “A Distributed Trust Models,” Proceedings of the 1997 Workshop on New Security Paradigms, 1998, pp. 48-60.

[35] C.Yew. “Architecture Supporting Computational Trust Formation,” [Online] Available: https://ir.lib.uwo.ca , [Accessed 15 Feb. 2019].

[36] V. Patil and R. Shyamasundar, “Trust Management for E-Transactions,” *Sadhana* Academy Proceedings in Engineering Sciences, 2005, pp.141–58, https://doi.org/10.1007/BF02706242.

[37] J. Urquhart, “he Biggest Cloud-Computing Issue of 2009 is Trust,” [Online] Available: https://www.cnet.com/news/the-biggest-cloud-computing-issue-of-2009-is-trust/ , [Accessed 15 Feb. 2019].

[38] A. Abdul-Rahman and S. Hailes, “Supporting Trust in Virtual Communities,” Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000, pp. 6007, https://doi.org/10.1109/HICSS.2000.926814.

[39] R. Shyamasundar and V. Patil, “ROADS : Role-Based Authorization and Delegation System-Authentication , Authorization and Applications ,” International Conference on Computational & Experimental Engineering and Sciences, (ICCES- 03),2003, n. p.

[40] M. Paul, “A Trust Model of Cloud Computing Based on Quality of Service,” Annals of Operations Research, 2013, pp. 1-12.

[41] A. Gholami, and M. Arani, "A Trust Model Based on Quality of Service in Cloud Computing Environment," International Journal of Database Theory and Application, 2015, pp. 161-170.

[42] Z.Tan, Y. Niu, and Y.Liu, "A Novel Trust Model Based on SLA and Behavior Evaluation for Clouds." 2016 14th Annual Conference on Privacy, Security and Trust(PST), 2016,n. p.

[43] Y. Jin and S. Min, "Stadam: A New SLA Trust Model Based on Anomaly Detection and Multi-Cloud," 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI), 2016, pp.396–99, https://doi.org/10.1109/CCI.2016.7778951.

[44] T. Grandison and M. Sloman, "A Survey of Trust in Internet Applications," *IEEE Communications Surveys &Tutorials*, 2000, pp. 2–16, https://doi.org/10.1109/COMST.2000.5340804.

[45] "TCPDUMP and LIBPCAP," [Online]. Available: https://www.tcpdump.org , [Accessed 1 Jan 2019].

[46] R. Mercuri. "On Auditing Audit Trails." [Online]. Available: http://www.notablesoftware.com/Papers/auditrail.html, [Accessed 1 Jan. 2019].

[47] "Top 20 IPs by Traffic-Daily." [Online]. Available at http: //bandwidthd.sourceforge.net/demo/ 2009. 1O, [Accessed 1 Jan. 2019].

[48] "Cisco Works Software," [Online]. Available : http://www.cisco.com/public/sw-center/sw- netmgmt.shtml,2019, [Accessed 1 Jan. 2019].

[49] "NetFlow Monitor." [Online]. Available: http//netflow.cesnet.czl,201O.3, [Accessed 1 Jan. 2019].

[50] “nGenius® Probes,” [Online] Available: http: http://www.netscout.com/products/probes_home.asp, 2005, [Accessed 1 Jan. 2019].

[51] B. Dewangan and P. Shende, "The Sliding Window Method: An Environment to Evaluate User Behavior Trust in Cloud Technology,” International Journal of Advanced Research in Computer and Communication Engineering, 2013, n. p.

[52] J. Ma and Y. Zhang, “Research on Trusted Evaluation Method of User Behavior Based on AHP Algorithm,” 2015 7th International Conference on Information Technology in Medicine and Education (ITME), 2015, pp .588–92, https://doi.org/10.1109/ITME.2015.39.

[53] N. Yang, T. Liqin, S. Xueli, and G. Shukail., “Behavior Trust Evaluation for Node in WSNs with Fuzzy-ANP Method,” 2010 2nd International Conference on Computer Engineering and Technology, 2010, n. p, https://doi.org/10.1109/ICCET.2010.5486146.

[54] X. Jing, Z.Liu, S.Li, B.Qiao, and G.Tan, “A Cloud-User Behavior Assessment Based Dynamic Access Control Model,” International Journal of System Assurance Engineering and Management 8, 2017, pp .1966–75, https://doi.org/10.1007/s13198-015-0411-1.

[55] L. Jun-Jian and T. Li-Qin, “User’s Behavior Trust Evaluate Algorithm Based on Cloud Model,” 2015 5*th* International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2015, pp .556–61.

[56] A. Berrached and A. Korvin, “Reinforcing Access Control Using Fuzzy Relation Equations.,” International Conference on Security & Management ,2006, pp .489–93.

[57] R. Yang, C. Lin, Y. Jiang, and X. Chu, “Trust Based Access Control in Infrastructure-Centric Environment,” 2011 IEEE International Conference on Communications (ICC), 2011, pp .1–5, https://doi.org/10.1109/icc.2011.5963329.

[58] M. Jaiganesh, S. Mercy, K. Anupama ,and A. Kumar, “Neuro Fuzzy ART-Based User Behavior Trust in Cloud Computing,” Asian Journal of Information Technology, 2016, pp .3461-3473, https://doi.org/10.36478/ajit.2016.3461.3473.

[59] T. Junfeng and C. Xun, “A Cloud User Behavior Authentication Model Based on Multi-Partite Graphs,” 3rd International Conference on Innovative Computing Technology (INTECH 2013), 2013, pp .106–12, https://doi.org/10.1109/INTECH.2013.6653686.

[60] T. Liqin, L. Chuang, and J. Tieguo, “Quantitative Analysis of Trust Evidence in Internet,” 2006 International Conference on Communication Technology, 2006, pp .1–5, https://doi.org/10.1109/ICCT.2006.342023.

[61] W. Deng and Z. Zhou, “A Flexible RBAC Model Based on Trust in Open System,” 2012 3rd Global Congress on Intelligent Systems, 2012, pp . 400–404, https://doi.org/10.1109/GCIS.2012.79.

[62] R. Yang and X. Yu, “Research on Way of Evaluating Cloud End User Behavior’s Credibility Based on the Methodology of Multilevel Fuzzy Comprehensive Evaluation,” Proceedings of the 6th International Conference on Software and Computer Applications, (ICSCA ’17), 2017, pp .165–170, https://doi.org/10.1145/3056662.3056677.

[63] K. Reena, S. Yadav, N. Bajaj, and V. Singh, “Security Implementation in Cloud Computing Using User Behavior Profiling and Decoy Technology,” in 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), 2017, pp .471–74, https://doi.org/10.1109/ICICCT.2017.7975242

[64] Z. Chen, L. Tian, and C. Lin, “Trust Evaluation Model of Cloud User Based on Behavior Data,” International Journal of Distributed Sensor Networks, 2018, n. p doi:10.1177/1550147718776924.

[65] G. Lin, D. Wang, Y. Bie, and M. Lei, “MTBAC: A Mutual Trust-Based Access Control Model in Cloud Computing,” China Communications 11, 2014, pp .154–62, https://doi.org/10.1109/CC.2014.6827577.

[66] A. Mohsenzadeh, H. Motameni, and M. Joo-Er, “A New Trust Evaluation Algorithm Between Cloud Entities Based on Fuzzy Mathematics,” International Journal of Fuzzy Systems 18th, 2016, pp .659–72, https://doi.org/10.1007/s40815-015-0112-6.

[67] R. Alguliev and F. Abdullaeva, “User Profiles and Identifying User Behavior in the Cloud Computing Environment.” Universal Journal of Communications and Network,2014, pp .. 87-92, doi:10.13189/ujcn.2014.020501.

[68] G. Kalaskar, P. Ratkanthwar, P. Jagadale, and B. Jagadale, “FOG Computing: Preventing Insider Data Theft Attacks in Cloud Using User Behavior Profiling and Decoy Information Technology,” International Journal of Engineering Trends and Technology 32 ,2016, pp .352–55, https://doi.org/10.14445/22315381/IJETT-V32P266.

[69] M. Xiaoxue, W. Zixian, B. Jing, and L. Fei, “Trust Model Based on Rewards and Punishment Mechanism,” 2010 2nd International Workshop on Education Technology and Computer Science, 2010, pp .182–85, https://doi.org/10.1109/ETCS.2010.337.

[70] Q. Zhou, J. Yu, and F. Yu, “A Trust-based Defensive System Model for Cloud Computing,” IFIP International Conference on Network and Parallel Computing, 2011, pp.146-159.

[71] C. Lin and X-h. Peng, “Research on Trustworthy Networks,” Chinese Journal of Computer, 2005, pp. 751-758.

[72] L. Zadeh, “Is There a Need for Fuzzy Logic?” Information Sciences, 2008, pp.2751–2779, https://doi.org/10.1016/j.ins.2008.02.012.

[73] A. Ansari and N. Gupta, “Automated Diagnosis of Coronary Heart Disease Using Neuro-Fuzzy Integrated System,” 2011 World Congress on Information and Communication Technologies, 2011, pp.1379–84, https://doi.org/10.1109/WICT.2011.6141450.

[74] Y. Bai and D. Wang, “Fundamentals of Fuzzy Logic Control — Fuzzy Sets, Fuzzy Rules and Defuzzification,” Advanced Fuzzy Logic Technologies in Industrial Applications, Springer London, Uk, 2006, pp.17–36, https://doi.org/10.1007/978-1-84628-469-4_2.

[75] F. Dernoncourt, “Introduction to Fuzzy Logic.” Massachusetts Institute of Technology, Cambridge, MA, USA,2013.

[76] M. Sellitto, E. Balugani,R. Gamberini, and B. Rimini ,“A Fuzzy Logic Control Application to the Cement Industry,” IFAC-Papers On Line 51, 2018,pp.1542–47, https://doi.org/10.1016/j.ifacol.2018.08.277.

[77] M. Dattathreya, H. Singh, and T. Meitzler, “Detection and Elimination of a Potential Fire in Engine and Battery Compartments of Hybrid Electric Vehicles,” *Advances in Fuzzy Systems*, 2012, n. p, https://doi.org/10.1155/2012/687652.

[78] X. Cao and Z. Liu, “Type-2 Fuzzy Topic Models for Human Action Recognition,” IEEE Transactions on Fuzzy Systems, 2015, pp. 1581–93, https://doi.org/10.1109/TFUZZ.2014.2370678.

[79] VN. Nguyen, V. Nguyen, M. Nguyen, and T. Dang, “Fuzzy Logic Weight Estimation in Biometric-Enabled Co-Authentication Systems,” Information and Communication Technology. Springer Berlin Heidelberg ,2014, pp. 365-374https://doi.org/10.1007/978-3-642-55032-4_36.

[80] J. Kim and J-H. Lee, “A Slow Port Scan Attack Detection Mechanism Based on Fuzzy Logic and a Stepwise P1olicy,” 2008 IET 4th International Conference on Intelligent Environments, 2008, pp.1–5, doi:10.1049/cp:20081126.

[81] A. Girma, M. Garuba, J. Li, and C. Liu, “Analysis of DDoS Attacks and an Introduction of a Hybrid Statistical Model to Detect DDoS Attacks on Cloud Computing Environment.” 2015 12th International Conference on Information Technology - New Generations (ITNG), 2015, n. p, doi:10.1109/itng.2015.40.

[82] O. Achbarou, M.EL kiram, and S. El Bounaniet, “Securing Cloud Computing from Different Attacks Using Intrusion Detection Systems,” International Journal of Interactive Multimedia and Artificial Intelligence, 2017, pp. 61, doi:10.9781/ijimai.2017.439.

[83] “CloudTrail Log Event Reference.” [Online] Available : https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-event-reference.html, [Accessed 15 September 2018].

[84] “Logging IAM and AWS STS API Calls with AWS CloudTrail,” [Online] Available : https://docs.aws.amazon.com/IAM/latest/UserGuide/cloudtrail-integration.html, [Accessed 15 September 2018].

[85] “CloudTrail user Identity Element,” [Online] Available at https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-event-reference-user-identity.html, [Accessed 15 September 2018].

[86] A.Mathur,” Foundations of Software Testing”, Dorling Kindersley, India, 2013, pp.198–200.