Turkish Journal of Computer and Mathematics Education Vol.12 No. 5 (2021), 1414-1428

Research Article

NdRAdAC: Need based Access Control Framework for an Emergency Response System

Kriti Srivastava[a], Dr. Narendra Shekokar[b], Pratik Aher[c]

[a]Research Scholar, D.J. Sanghvi College of Engineering, India.
[b]Professor, D.J. Sanghvi College of Engineering, India.
[c]Software Engineer, J.P Morgan chase & co.

Article History: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract: Access control is easy to implement in a static system with resource-role mapping and known policies. It becomes challenging if the system is dynamic and volatile, which means there are unpredictability in the workflow. Existing role based and attribute- based access control systems are very efficient in static and predictable situations. But they are not effective in a dynamic situation. Researchers over the last two decades have tried to propose various probabilistic based, machine learning based and decision theory-based access control to prove adaptability in their access control methods. But there are existing gaps in operational needs and proposed adaptability methods. Under regular scenario access control system may work based on the policies or decided roles. Only if there is a genuine need, then access control should switch to adaptable solutions. Also, a true adaptable system should not allow human intervention, the system should be able to understand the genuineness of the requester and take decisions whether access should be granted or not. In this paper with the help of a disaster management case study, a need-based access control framework – NdRAdAC is proposed. It evaluates the genuineness of the requester and acts appropriately. An ontology-based access control for an emergency response system is developed, which can help the disaster management system to coordinate with different hospitals and help in transferring patient data from one hospital to another if needed. It ensures that data requester is authenticated with the help of access control module. The framework is tested for three main parameters: Adaptability, Consistency and Computational Efficiency. It was found that framework was accurately adaptable, consistent with all the different types of cases and computationally efficient.

Keywords: Risk Adaptive Access Control, Ontology, Inference Engine, Emergency Response System, Heterogeneous System.

1. Introduction

One can never stop a natural disaster. Many times, the impact is so huge that it takes years to recover from it. But using technologies we can always try to be prepared for efficient disaster management. The main objectives of these disaster management systems are to save the affected persons lives. For this the victims should immediately be sent to the nearby hospitals for treatment. Hospitals and doctors play a very important role during the time of disaster. There are limited resources such as beds, ventilators or specialized doctors. Based on the research on different natural disasters such as Kerala flood [1] [2] [3], floods in Leh [4] floods in Sumbawa regency [5] Japan's Tsunami [6] [7] [8] or latest Covid-19 pandemic [9] [10], one of the common thing in different types of disasters is, to be able to communicate with different nearby hospitals and be able to exchange information without any threat of information misuse. Following are the challenges in implementing a disaster management system:

- Matching the right specialist to the case
- Requirement
- Matching the right hospital based on bed and other infrastructure
- Inadequate patient information / medical history viz: chronic conditions, underlying co-morbidities
- Lack of correct testing methods.
  All of the above leads to significant delay in providing timely and the right quality of healthcare leading to fatalities or long-term chronic conditions. Motivation of this research work lies in the possible solutions for the above-mentioned problem.
- Solution for matching of specialist to the case requirement - if there is a Centralized Emergency Response System, which takes the request from these local hospitals and contacts appropriate specialist (remotely available). The specialist can guide the local doctors to treat the patient.
- Solution for matching of bed availability and infrastructure - if Emergency Response System will have the information which nearby hospitals have a live position of available beds / other infrastructure so hospitals can contact them and transfer the patients there.
- Solution for the third problem - victims' medical information may not be available with the hospital. If various hospitals have mutual agreement with the Emergency Response System to share patient's historical

information, then the right treatment can be given faster.

Hence there is a need for an emergency response system which has two main features. First there should be a user interface where information about the resources such as beds, ventilators, specialist doctors are updated with timestamp. The second feature is need-based access control system which is adaptable and is compatible with heterogenous sources. This research work is based on the second feature of the emergency response system. In this paper, an access control mechanism for communicating and sharing information with different hospitals during disaster management is proposed. Section 2 discusses a detailed literature survey to explain the need of such a system. Section 3 explains the framework modules and its workflow. Section 4 discusses the ontology definitions. Section 5 shows the result analysis and section 6 is the conclusion.

2. Related Work

2.1. Dynamic Access Control

For a heterogeneous data layer with uncertainty in the situation access control methods should be dynamic. Researchers have proposed many improvements on robust role-based access control method. Khalid Zaman Bijon, Ram Krishnan and Ravi Sandhu had proposed a framework where each risk is calculated for each new role created and the value is mapped with the threshold [10]. It helps in deciding whether to approve the new role or not.

J. P. Cruz, Y. Kaji and N. Yanai, had proposed a role-based access model with Ethereum smart contract [11]. Blockchain concept is used to represent the trust and endorsement relationship which is useful for Role based access control. Though the prototype was created and tested but the security aspect testing was missing. Task role-based methods were proposed where, tasks-based role hierarchy was introduced, and dynamic access control was provided based on task and role concepts [12] [13]. But the disadvantage here lies in the flexibility. It became more flexible than required by audit. This could lead to security issues. Some authors have discussed the importance of operational need for developing risk adaptive access controls [14] [15]. A probabilistic based framework was introduced for both static and dynamic data, using purpose forest concept. Probabilistic based access control requires administrator's intervention which cannot be justified as completely adaptable.

Research work done by Amar A Rasheed on healthcare access control while performing an automatic operation, it was performed without the knowledge of the integrity status of the underlying software component [16]. With the help of attestation-based technology it calculated the sensitivity score and calculated the risk. Ahmed Al Faresi, in his thesis had proposed an access control system for healthcare [17]. He had used a probabilistic based model for access control but here more emphasis was given to data encryption part. Authors have worked on contexed aware calculation [18] [19] [20], Topology aware [21] and risk calculations methods [22]. Context aware is based on the static scenarios and risk calculations methods were also not able to generalize well. Giuseppe Petracca, Frank Capobianco, Christian Skalka, and Trent Jaeger have proposed a risk estimation function which could be used for access control implementation [23]. One of the very important usage of dynamic access control is in smart home. E. Fernandes, J. Jung and A. Prakash, had derived a risk assessment module for smart homes, mwhich could be used for access control as well.[24] Open sources are also vulnerable to unauthorized access. A. A. Malik, H. Anwar and M. A. Shibli, have proposed a role evolution mechanism based on genetic algorithm, access control role for open computing environment, were decided [25]. In this paper only role generation was focused. Adaptability is still an issue.

2.2. Machine Learning

Another set of work had been done in the field of access control using machine learning algorithms. Since machine learning algorithms can find patterns in the data, learning new patterns and acts appropriately, this could be used to develop adaptable access control. Baris Yuce and Yacine Rezgui used ANN and genetic algorithm to provide adaptability [26] [27]. Recursive neural Network and deep learning concepts are also used for rule extraction [28]. The purpose of access control and Intrusion detection system is to identify the genuineness. Some researchers have worked in the area of intrusion detection for finding the correctness of the inquirer [29] [30] [31] [32] [33]. These works were very helpful to get an insight of various methods of identifying the genuineness of the requester. G Rushin, C Stancil, M Sun, S Adams and P Beling had used three different machine learning algorithms to find credit card fraud [34]. Finding anomaly through machine learning approaches gave us the confidence that we can work in similar lines to develop access control for our purpose. Similar kind of work had been proposed by E. L.

Paula, M Laderia, R N Caravalho and T Marzagao, using deep learning methods for anti-money laundering [35]. After going through all these literature surveys in the field of machine learning we worked with machine learning algorithms to check if we can get an adaptable access control method for our problem definition [36]. Through we got good results for the datasets which we had used, but there were some gaps with respect to our objectives. First gap: Neural Network, Auto encoders and Random Forest all the methods used in the proposed work added bias to the model. Second gap: The dataset used was an example of imbalanced dataset. Third gap: Crisp values are used to identify risk in the situation. Whereas risk is subjective. Depends on situations. There is a chance where during emergency system needs to modify its decision. So, in the next section we did our survey in the field of fuzzy and Decision making.

2.3. Decision Making and Fuzzy

One of the gaps identified was working with crisp values. Adaptability is dependent on real life situations and these situations are fuzzy in nature. Also, there is a great need of adding expert advice before taking the decisions instead of totally being dependent on data. There are different kinds of decision-making system. One of them is Topsis. There are a lot of work done with Topsis decision making system.

P. K. Parida and S. K. Sahoo, in their research work had used Topsis for multi criteria decision making [37]. It was helpful to understand if there were many parameters in the system then how Topsis helped analysing the situation with respect to the parameters and gave solutions. As discussed in the previous section working with crisp value will not be able to give realistic decision. Researcher had actually used fuzzy Topsis methods for different applications to support realistic decisions [38] [39] [40]. Hence one aspect is clear that decision making system will help in giving solutions which are relatable. But if we refer our problem definition again, it needs a realistic and adaptable access control for heterogenous infrastructure. With decision making there are gaps in providing solutions which can be adaptable and work in a heterogenous infrastructure. In this paper we would like to address following research gaps:

- There is a lack of dynamic access control to accommodate the diverse hosting of information.
- Inability to dynamically handle access privileges, given uncertainty of subject or information being requested.
- Lack of visibility and access control governance due to inadequate access control policy

3. NdRAdAC: Need based Access Control

As discussed in the above section, to identify the genuineness of the requestor, a complete situation has to be analyzed. Each requestor and current situations combination will be different. Requesters can be of different categories, such as junior. doctor, specialist doctor, admin, nurses and admin staff, hence its combination with the current situation will be many. We need an intelligent access control mechanism to decide the genuineness of the requester. Ontology is an intelligent way to represent complex relations and rules in the relations [41] [42] [43]. The idea to represent access control using ontology is to access data in an intelligent manner and be able to handle any unknown or new request with high accuracy. As per best of our knowledge we propose a novel ontology-based access control framework for emergency response system which, can be useful during a disaster management for information sharing between different kinds of hospitals and medical camps.

This ontology-based solution is for managing patient’s treatment through nearby hospitals and camps. First, the case study in discussed in detail and then the proposed solution. Generally, the disaster management team goes to the location, rescues the victims and sends them to nearby hospitals. Here if there is a system, where the disaster management team can find which nearby hospital has how many available beds then from the location itself, the disaster management team can send the affected people to the hospitals appropriately. This is the first requirement of a disaster management system, which is not the scope of this work. Another issue in such case is that a patient reaches the hospital in time, but a specialist doctor is not available. Also, to start the treatment patient’s previous history is required which may be stored in another hospital. In such cases available doctors should be able to access patient’s data from another hospital. Every hospital has its own access control policy to access stored information. In the above-mentioned case, the situation is different. Information needs to be exchanged between two different hospitals. Both may have different data storage and different access control policies. Accessing patient’s data from a different hospital becomes challenging.

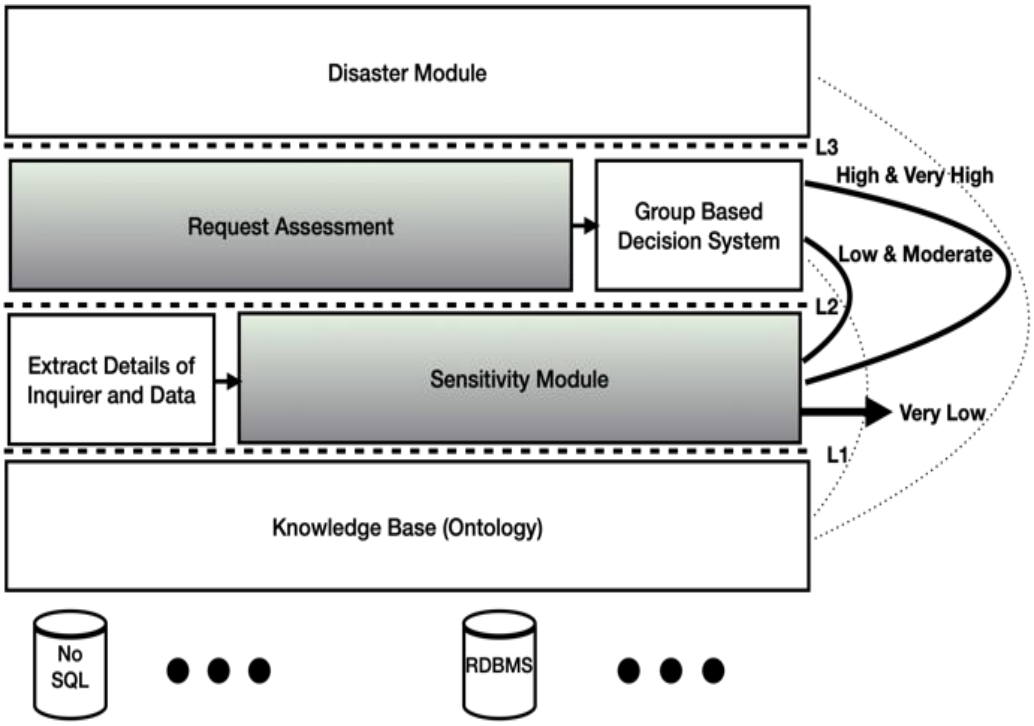


Figure 1. Architecture of Need Based Access Control Model

The proposed framework is motivated from the challenges in accessing data from a heterogeneous and distributed data storage. As shown in figure 1, there are four main modules in this framework. Knowledge base, Sensitivity Module, Request Assessment and Disaster Modul. The scope of the ERS system is as follows:

- Each hospital can have different types of data storage.
- Hospitals need to have an agreement with the ERS to share information in the desired format. In return ERS will provide access control.
- ERS will have each doctors, admins and patients, ID and Biometric details only. Rest all the information will be stored with the hospitals.
- ERS will have its own ontology based on Inquirer (Doctor, Staff and Admin) and data (personal and medical).
- There were many hospital related ontologies, but their relations were not the same as we wanted. Hence, we decided to create our own Ontology.

3.1. Details of the Base System

Risk adaptive access control (RAdAC) is an access control system which identifies the need of the requester as well as identifies the criticality of the situation. After a well analyzed process it gives its decision of providing access. Defense, airport surveillance and hospital management system are few systems where we need to have risk adaptive access control system. We have considered hospital management system as the base system for this work. The reason why RAdAC is needed in HMS (hospital management system) is during any kind of emergency, if the assigned doctor is not available then system has to take decision whether to provide access to another doctor or not. In such situation regular system will not allow the new doctor to access information. Hence, we need a system, which sense that this doctor is genuine and allow access. But this may not always be true. Sometimes someone may pretend to be a doctor and try to access patient's information illegally. In such situation the system shall be able to identify intruders and deny access.

3.2. Ontology based Knowledge Base

It has been discussed in many researches works that in a static scenario mapping requester with appropriate resources provides accurate access control. The idea of Need based access control is to assess the need of the requester if access is denied. This means under regular and normal scenario the system will work in a predictable manner. Hence an ontology-based knowledge repository with regular mapping of requester and resources is provided as the first module of the system. The main classes of the ontology were decided keeping the access control logic in mind. There is a requester (Inquirer) who requests to get access of an information (Data), which

belongs to an owner (Patients). All three of them either belong to same location or to different locations. Hence in the ontology we have Inquirer, Data, patient and Hospital as four main classes shown in figure 2. Inquirer has various hospital staff as inquirers, patients are categorized as Elite, Regular and Donor subclasses. Data can be either personal or medical and Hospital can be regular hospital where data will be stored or medical camos where patients are admitted during a disaster.

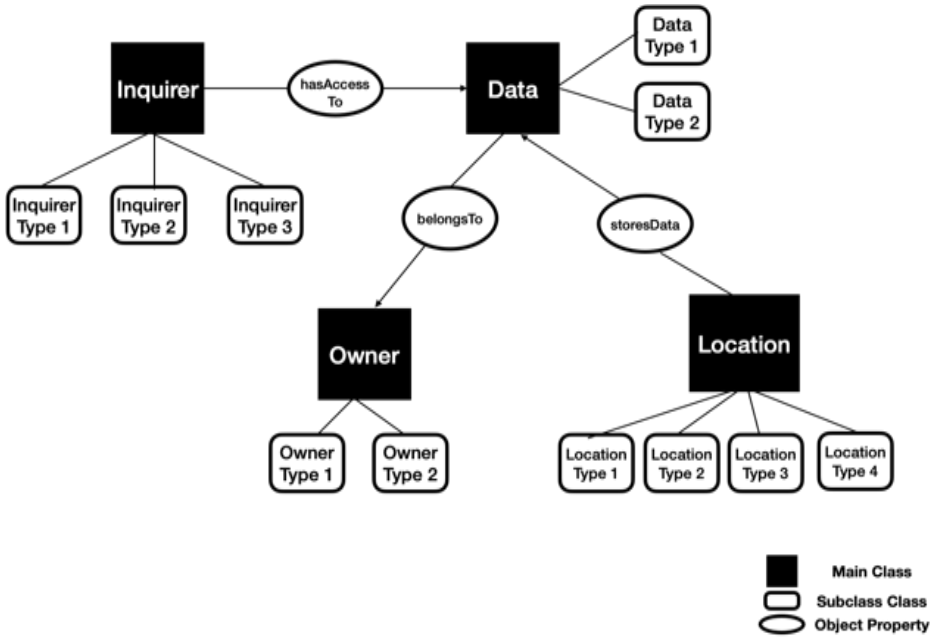


Figure 2. Skeleton of Knowledgebase

Various object properties were used to connect different classes and subclasses. For example: "belongsTo" object property is used to have a relation between class data and owner. This is an inverse of "hasData", which says owner has data. There were 26 object properties with one symmetric, 1 inverse and 11 disjoint.

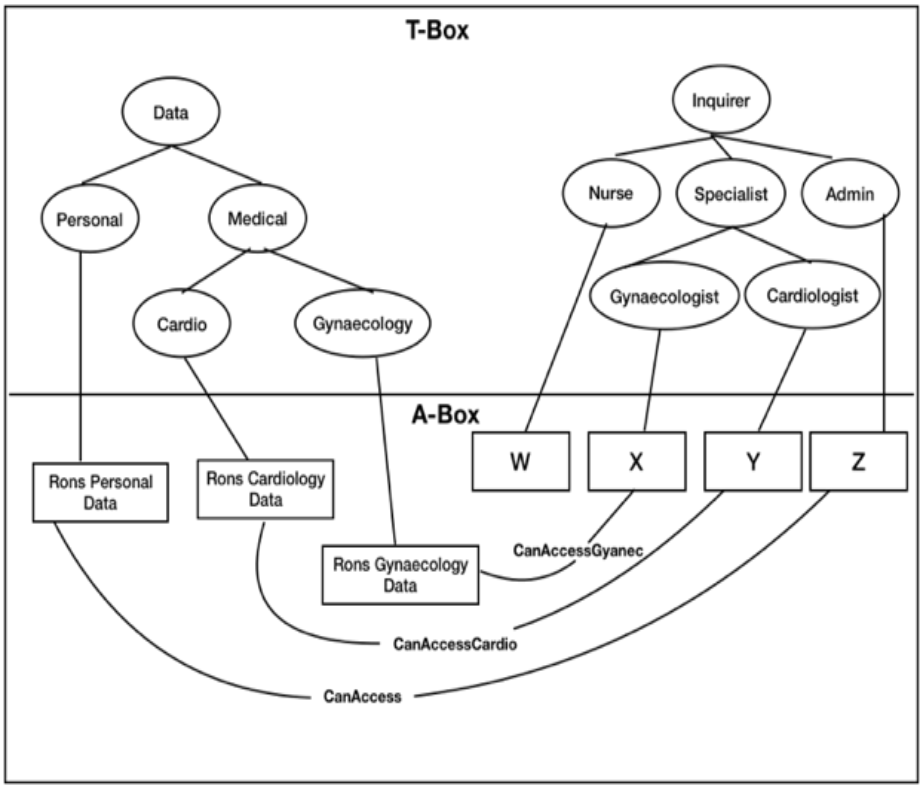


Figure 3. T-Box and A-Box

There were 286 axioms, 36 data properties and 59 individuals to test if the ontology is consistent. The correctness of the ontology can be shown in figure 3 as a sample. T box are all the classes and the properties and A box is the representation of the individuals. For example: if a class cardiologist "hasAccessTo" to class cardio then with the same property, individuals belonging to cardiologist will be mapped with individual belonging to cardio class. This knowledge base consists of the relationship between each class. Under regular and static scenario this knowledge base can be considered for access management. It can also provide basic knowledge to all the other modules for decision making purpose.

3.3. Sensitivity of Data

Based on the requesters and data id the system can fetch all the information about the type of requester, location of requester, type of data, owner of data, owner category of data, location of stored data and location of owner, from the knowledge repository. These entities help us build next module which calculates the sensitivity of information asked.

$$\text{Sensitivity} = \text{Product}(\alpha.\beta.\gamma) \quad (1)$$

Sensitivity is a product of alpha, beta and gamma. Alpha discusses the relation between owner and data. Beta is a relation between inquirer and data. Gamma is about the owner category. Sensitivity module is a domain

Table 1. Sample of Sensitivity Score

Owner-Data	Description	Score	Inquirer-Data	Description	Score	Owner Category	Score
XX	Owner & Data both belongs to same data source	1	AAA	Owner, Data & Inquirer both belong to same data source. Inquirer has access to Owners data.	1	Important	2.5
XY	Owner & Data both belongs to different data source	10	ABA	Owner, Data & Inquirer both belong to same data source. Inquirer has no access to Owners data.	2.5	Specific	2
			AB * C	(Owner, Data) & Inquirer both belong to different data source.	5	Regular	1

XX means same and XY means data and owner belongs to different location. There are three columns here first one has maximum weightage and will be distributed between 1 to 10. So if XX then 1 and if XY then critical so 10. Second column has less weightage than first so the weights will be between 1 to 5. If Inquirer, Owner and data belongs to same location (AAA) then weights will be 1. If Inquirer, Owner and data belongs to same location but Inquirer is not allowed to access data (ABA) then the weight will be 2.5. Third case is not possible in this use case. And the third category is Owner category which can be distribute between 1 to 2.5. As we move from one column to another the weights are reducing. When we calculate the score and normalize then we get the sensitivity which can be mapped as very low, low, moderate, high and very high.

This is one of the major contributions of this work. A significant impact will be visible in the existing work if they use sensitivity of module as a parameter in their evaluation. Discussing the workflow of the system, sensitivity module is at L1 level and all the requests which belongs to very low sensitivity will be granted access at this level, which means no further investigation is required for low sensitivity data. Rest all the request will be send for evaluation to the other module which is discussed in next section.

Algorithm 1:

Input: Inquirer Id and IP
Output: True or false

```
1 center_point = ['lat', 'lng']
2 test_point = ['lat', 'lng']
3 lat1 = center_point[0]['lat']
4 lon1 = center_point[0]['lng']
5 lat2 = test_point[0]['lat']
6 lon2 = test_point[0]['lng']
7 lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])
8 dlon = lon2 - lon1
9 dlat = lat2 - lat1
10 a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
11 c = 2 * asin(sqrt(a))
12 r = 6371
13 ph count from the knowledge base
14 if c*r <= THRESHOLD && ph <= 2
15     return True
16 else:
17     return False
```

Here haversine formula is used to calculate the radius of the requesters ip and using the knowledge bases, previous history (ph) is retrieved. If radius and ph both are within the specified range, then request is sent forward else rejected. Request Assessment is another minor contribution of this work, which is domain independent. It can be used for various use cases. As per the architecture if the query is passed through the request assessment phase then based on the information sensitivity it is evaluated either by group-based decision system or by disaster module, which is discussed in the following section.

4. Ontology Definitions and Inference Rules

All the queries which is passed through the request assessment module and have low or moderate sensitivity will be sent to group-based decision system for evaluation. The idea of this module is to provide adaptability to the system if there is a need. In order to evaluate the need this module suggests two definitions.

Definition 1: Same Group

Data: a; Inquirer 1: x, Inquirer 2: y, Patient: b, Hospital: H
∃a∃x∃y∃b∃H: storedAt(a, H) ^ belongsTo(a, b) ^ employeedBy(x, H) ^ employeedBy(y, H) ^canAccess(x,a) ^ hasAccess(a,y) ^differentFrom(x, y) => sameGroup(x)

If the inquirer is not available to access the situation then any other inquirer who could access the same information will belong to sameGroup class. For example, if cardiologist is not available then a junior cardiologist or any other cardiologist who does not have the access of same information can be grouped in the sameGroup class.

Definition 2: Emergency Group

Data: a; Specialist: x, Admin: y, Patient: b, Hospital: H
∃a∃x∃y∃b∃H: storedAt(a, H) ^ belongsTo(a, b) ^ admittedTo (b, H) ^ employeedBy(x, H) ^ employeedBy(y, H) ^ hasAdminsPermission(x,y) ^ canAccess(y, a) ^ Emergency(e) ^ differentFrom(x, y)=> EmergengyGroup(x)

Another definition is for EmergencyGroup which says that if there is an emergency then admin can grant permission to a specialist to access data. For this the specialist has to belong to EmergencyGroup. While taking the inputs there will be another field in the user interface which will tell if there is an emergency or not. Hence in this module based on the definitions two new equivalent classes of Inquirers, will be created. If the query passes through the request assessment module and the sensitivity of the information is high or very high, then the query

goes to the L3 level and it is evaluated through the disaster module. In this level based on the requirement of the query three equivalent classes of inquirer will be created. DiffLocSameGrp, DiffLocDiffGrp and Donor class.

Algorithm 2:

Input: Inquirer-Data Table ID; Data-Patient Table DP; Inquirer-Hospital Table IH; Patient-Hospital Table PH; Data-Hospital Table DH
Output: Inference Rule for data access in rare scenarios
1: Get Data_sub, Patient_sub, Hospital_sub and Inquirer_sub for Data, Patient, Hospital and Inquirer classes

---

/*Create an inference rule for Inquirer class*/
2: for each subclass iq_Sub[I] E Inquirer_sub do
3: Get the subclass label of iq_sub [I]
4: Create Enumerated Class for Inquirer
/*Create Rules for SameGroup*/
5: for each relation rel[k] E ID do
6: if d_sub[I] = rel[j] E DP
7: if range of h_sub[I] in IH =range of h_sub[j] in DH && range of h_sub[l] in PH
8: Obtain the SameGroup s from rel[k]
9: Find Ontology instance ID for SameGroup s from Inquirer_sub and add it to Enumerated Class using <owl:oneOf>
10: endif
11: endif
12: Create <owl: allValuesFrom> restriction and add EnumeratedClass to this restriction using isAccessibleBy object property
/* Create Rules for EmergencyGroup*/
13: if d_sub[I] = rel[j] E DP
14: if range of h_sub[I] in IH =range of h_sub[j] in DH && range of h_sub[l] in PH
15: If iq_sub [n] = Specialist && hasAdminsPermission (n, m)
16: Obtain the emergencyGroup e from rel[k]
17: Find Ontology instance ID for EmergencyGroup s from Inquirer_sub and add it to EnumeratedClass using <owl:oneOf>
18: endif
19: endif
20: endif
21: Create <owl: allValuesFrom> restriction and add EnumeratedClass to this restriction using isAccessibleBy object property
/* Create Rules for SameGroupDifferentHospital*/
22: if d_sub[I] = rel[j] E DP
23: if range of h_sub[I] in IH notequalTo range of h_sub[j] in DH
24: for each relation rel[k] E ID && range of h_sub[I] in IH noequalto range of h_sub[j] in DH
25: if there exist rel(m) where d_sub[m] notequalto d_sub[I]
26: Obtain the DisasterSameGroup dsd from rel[k]
27: Find Ontology instance ID for DisasterSameGroup dsd from Inquirer_sub and add it to EnumeratedClass using <owl:oneOf>
28: endif
29: endif
30: endif
Similar rules for data access by DiffGroupDiffLoc and DonorGrp equivalent classes are created.

Similar way based on the definition other equivalent classes will be created. For data access seven rules were written using semantic web rule language (SWRL). A sample of rules are shown in table 2.

Definition 3: DiffLocSameGrp

Data 1: a ; Data 2: a2; Inquirer 1: x, Patient 1: b, Patient 2: b1; Hospital: H; Hospital: H1
∃a∃x∃y∃b∃H: storedAt(a,H) ^storedAt(a1, H1) ^ belongsToPatient(a, b) ^ belongsToPatient(a1, b1) ^admittedTo(b, H1) ^ employeedBy(x, H1) ^as Access(x, a2) ^ SameGroupData(a, a1) =>DiffLocSameGrp(x)

This definition is written keeping disaster scenario in mind. During such times patients are admitted to any hospital and their medical data must be in a different hospital. Inquirers of one hospital may want to access information of the patient from another hospital. If same group of data is accessed by the inquirer in his hospital, then he can belong to a class named as DiffLocSameGrp.

Definition 4: DiffLocDiffGrp

Data 1: a; Data 2: a2; Specialist: x, Admin: y, Patient 1: b, Patient 2: b1; Hospital: H; Hospital: H1
∃a∃x∃y∃b∃H: storedAt(a, H) ^ storedAt(a2, H1) ^ employeedBy(x, H1) ^ admittedTo(b, H1) ^ belongsToPatient(a, b) ^ hasAdminsPermission(x, y) ^ employeedBy(y, H1) ^canAccessdatainEmergency (x, a2) ^ SameGroupData(a, a1) =>DiffLocDiffGrp(x)

Definition 5: DonorGrp

Donor_Data: a; Data 2: a1; Inquirer 1: x, Donor: b, Patient 2: b1; Hospital: H; Hospital: H1
∃a∃x∃y∃b∃H: storedAt(a, H1) ^ belongsTo(a, b) ^ employeedBy(x, H) ^ storedAt(a1, H) ^ hasAccessTo(x, a1) ^ belongsToPatient (a1, b1) ^ needDonor(b1, a) => DonorGrp(x)

If there is a need of donor data from another hospital, then the system will assess if the inquirer has access to patient data who needs donor data. If yes, then a new equivalent class is created named as DonorGrp. If the inquirer belongs to this group, then he will be able to access donors data from another hospital based on the rules.

Table 2. Sample Rules

Owner (Patient)	Data	Inquirer	Rule Type	Generation Level
Regular [A]	Medical 1(Cardio) [A]	Inquirer 1 (Cardiologist) [A]	Allowed	L1
Regular [A]	Medical 1(Cardio) [A]	Inquirer 11 (Junior Cardiologist) [A]	SameGroup Rule	L2
Regular [A]	Medical 1(Cardio) [A]	Inquirer 2 (Surgeon) [A]	EmergencyGroup	L2
Regular [A]	Personal 1 [A]	Admin [A]	Allowed	L1
Elite [A]	Medical 1(Cardio) [A]	Inquirer 1 [A]	Allowed	L1
Elite [A]	Medical 1(Cardio) [A]	Admin [A]	Elite Rule	L2
Regular [A]	Medical 1(Cardio) [B]	Inquirer 1 (Cardiologist) [A]	DiffLocSameGrp Rule	L3
Regular [A]	Medical 1(Cardio) [B]	Inquirer 11 (Junior Cardiologist) [A]	DiffLocSameGrp Rule	L3
Regular [A]	Medical 1(Cardio) [B]	Inquirer 2 (Surgeon) [A]	DiffLocDiffGrp Rule	L3
Elite [A]	Medical 1(Cardio) [B]	Admin [A]	Elite Rule	L3
Donor [A]	Medical (Kidney) [A]	Inquirer (Urologist) [B]	Donor Rule	L3
Regular [C]	Medical 1(Cardio) [B]	Inquirer 1 (Cardiologist) [C]	DiffLocDiffGrp Rule	L3

5. Performance Evaluation and Discussion

5.1. Experimental Setup

The knowledge base is developed using protégé 5.2 and reasoning was implemented using HermiT reasoner. HermiT uses hypertableau reasoning to match the rules with working memory elements. Each rule has at least 11 elements and there are 256 axioms created. Matching will result in a conflict set which is resolved using LEX method based on specificity. The output of this resolve step will be a selected rule. This selected rule will make

modifications in the working memory element. This is known a rule inference which is implemented using Drool Inference Engine which uses modified RETE algorithm to update the working memory elements. This is a detail explanation of the ontology knowledge base experimental setup.

Once ontology is tested and reasoned well the next step is to connect this knowledge base with other modules and query the knowledge base. A small user interface was created using Django Python framework. Ontology was connected to this user interface using SPARQL. Different types of data bases can also be connected. For testing purpose, a Mongo DB and a Postgre SQL were added to the existing setup as shown in figure 4.

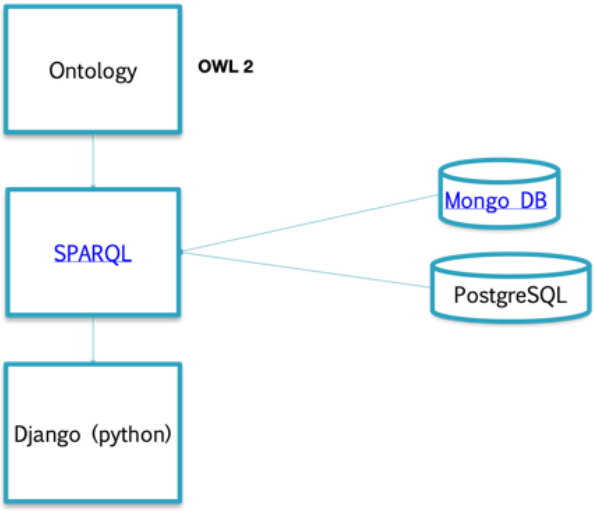


Figure 4. Experimental Setup

5.2. Querying Result Discussion

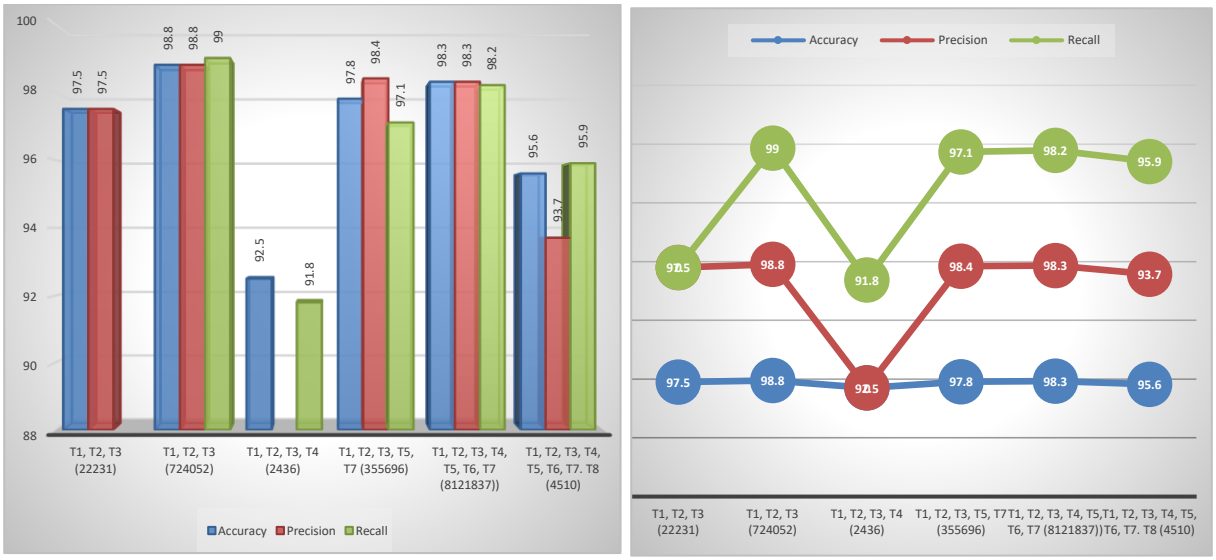
Querying was divided into various task category for evaluation. Tasks are various possible scenarios. For example T1: Roles assigned to resources, T2: Request from same group Inquirer, T3: Requester from different group but situation is emergency, T4: Request for an elite group patient, T5: Under a disaster situation requesting for information stored in another location but the requester belongs to same group, T6: Under a disaster situation requesting for information stored in another location but the requester belongs to different group, T7: Requesting for a donor data and T8: Request is made from a medical camp where no data is stored.

Synthetic DataSet: D1	**Scripps Mercy Hospital DataSet: D2**	**Practo DataSet: D5**
Number of Inquirers: 14 Types of Data Category: 7 Types of Data Owners Category: 2 Total Number of cases: 2436 Number of Cases for T1 Task category: 1068 Number of Cases for T2 Task category: 1175 Number of Cases for T3 Task category: 175 Number of Cases for T4 Task category: 980	Number of Inquirers: 15 Types of Data Category: 24 Types of Data Owners Category: 3 Total Number of cases: 22231 Number of Cases for T1 Task category: 21160 Number of Cases for T2 Task category: 28 Number of Cases for T3 Task category: 1043	Number of Inquirers: 10 Types of Data Category: 97 Types of Data Owners Category: 3 Total Number of cases: 724052 Number of Cases for T1 Task category: 468806 Number of Cases for T2 Task category: 1179 Number of Cases for T3 Task category: 254067
Heterogenous DataSet with 2 hospitals and 1 Medical Camp: D3	**Heterogeneous DataSet with Scripps Mercy Hospital & Synthetic: D4**	**Heterogeneous DataSet with Practo & Synthetic: D6**
Number of Inquirers: 21 Types of Data Category: 8 Types of Data Owners Category: 6 Total Number of cases: 4510 Number of Cases for T1 Task category: 1068 Number of Cases for T2 Task category: 1175 Number of Cases for T3 Task category: 175 Number of Cases for T4 Task category: 980 Number of Cases for T5 Task category: 510 Number of Cases for T6 Task category: 175 Number of Cases for T7 Task category: 238 Number of Cases for T8 Task category: 189	Number of Inquirers: 29 Types of Data Category: 24 Types of Data Owners Category: 3 Total Number of cases: 355696 Number of Cases for T1 Task category: 21160 Number of Cases for T2 Task category: 28 Number of Cases for T3 Task category: 1043 Number of Cases for T5 Task category: 166732 Number of Cases for T6 Task category: 166732	Number of Inquirers: 26 Types of Data Category: 97 Types of Data Owners Category: 3 Total Number of cases: 8806870 Number of Cases for T1 Task category: 468806 Number of Cases for T2 Task category: 28 Number of Cases for T3 Task category: 1043 Number of Cases for T5 Task category: 1179882 Number of Cases for T6 Task category: 715711

Figure 5. Dataset Description

Three different types of dataset under homogenous and heterogenous category each, were used for testing the querying part of the system. One synthetic dataset is developed which included all possible task categories. Dataset from Scripps Mercy Hospital another dataset where provider specialties were considered as Inquirers, Major Diagnostic categories were considered as data and EDIPN_Randomized were considered as Patient id. In this dataset there were no Inquirer category as Admin so T4 was not tested also since no donor data so T7 was also not tested. There was no access attribute here, so all the entries were considered as access provided. Third dataset is Practo dataset having attributes such as doctor specialty, patient id, time and location of access, type of data and access results. Unavailability of Admin, Donor and Medical camp T4, T7 and T8 task category was not tested with Practo dataset also. Detail description of the heterogenous and homogeneous datasets are given in figure 5.

Accuracy, precision and recall of all the six datasets were calculated and a detail analysis was conducted on the results. It was found that there is a correlation of number of task category involved and the size of the dataset. Though the Accuracy, Precision and Recall values for all the datasets were above 90%, as shown in fig 6, a significant reduction in the recall value of D1 dataset was visible also a mild dip in the accuracy, precision and recall was observed for D2 dataset. To understand the reason behind this, decrease a detail analysis on how adaptability is provided in the system through the inference rule is required. Hence in the next section a discussion on Adaptability and Consistency is done.



(a) (b)

Figure 6. (a) Accuracy Precision and Recall for Each Dataset and (b) Comparison of Accuracy, Precision and Recall.

5.3. Discussion on Adaptability

In section 3 and section 4 it was mentioned that the need is accessed for the request and based on the sensitivity of information different levels of rules were selected to provide adaptability to the system. It is very important that correct level of adaptability is implemented because is adaptability is very relaxed then it will be vulnerable to security threats and if adaptability is very strict then it will be same as static access control methods. Another major contribution of this work is a measure to calculate the adaptability. This adaptability score can be used in any system as a performance measure.

$$\text{Adaptability Score} = \left\{ \frac{\text{Observed Access - Ambiguous}}{\text{Observed Access}} \right\} * 5 \quad (2)$$

Ambiguous = mod (Actual Access -Observed Access)

Actual Access = TP + FN - FP

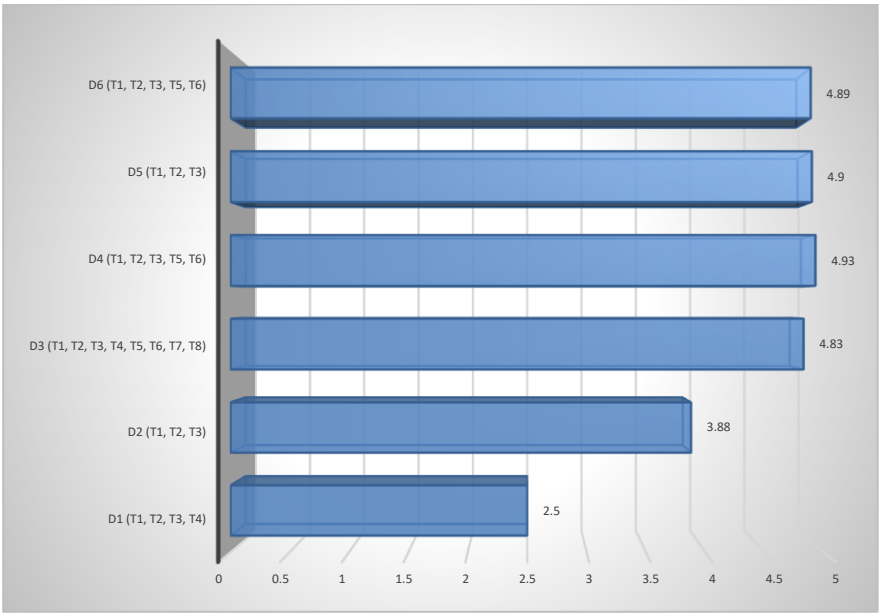


Figure 7. Adaptability Score for Various Datasets

To calculate the adaptability, score two parameters, have to be calculated first. First is what is the count of actual access provided by the system in response to the queries. This can be identified with the formula which says add the true negative values with the true positive values and subtract the false positives. This count will be the count of actual access given to the system through the developed adaptability model. Second is to calculate the ambiguity in the solution. For this take the difference of actual access and observed access. Once both these values are calculated, then adaptability score can be found by putting the values in equation 2. Accessing the adaptability score of all the six datasets it was observed that dataset 1 had adaptability score which was close to 50%. On analyzing the reason of such a low score for dataset 1 first observation was that there were less observations in the dataset as compared to other datasets. And second observation was apart from dataset 6 only dataset 1 had used task category T4. Observing the rules behind this task category it was found very strict. It said even if there is a disaster or emergency only the Admin will access elite patient's data. This reduced the adaptably quotient in the system.

5.4. Discussion on other Significant Parameters

Adaptability is a quality which is added into a system to provide dynamicity. It is necessary to check whether introducing adaptability is consistent or it is applicable on certain elements only. Hence whenever we evaluate adaptability it is a good practice to discuss consistency parallelly. In brief consistency is an evaluation parameter which checks if there is a rule or condition it is abdicable to all the elements in the same scenario. For example, if the rule says that cardiologist is not there so other specialists can access cardio data. Now this should hold true for neurologist, urologist or pulmonologist also Sample cases are shown in table 3 and table 4.

Table 3. Specialist Who are Allowed to Access Urology Data

Urology_Data			
Inquirer	**Patient**	**Data**	**Access**
Cardiologist	Regular_Ron	Urology_Data	Allowed
Neurologist	Regular_Ron	Urology_Data	Allowed
Surgeon	Regular_Ron	Urology_Data	Allowed
Gynaecologist	Regular_Ron	Urology_Data	Allowed
Paediatrician	Regular_Ron	Urology_Data	Allowed
Urologist	Regular_Ron	Urology_Data	Allowed
Pulmonologist	Regular_Ron	Urology_Data	Allowed
GP	Regular_Ron	Urology_Data	Not Allowed
Radiologist	Regular_Ron	Urology_Data	Not Allowed
Nurse	Regular_Ron	Urology_Data	Not Allowed

Table 4. Specialist Who are Allowed to Access Cardio Data

Cardio_Data			
Inquirer	**Patient**	**Data**	**Access**
Cardiologist	Regular_Ron	Cardio_Data	Allowed
Neurologist	Regular_Ron	Cardio_Data	Allowed
Surgeon	Regular_Ron	Cardio_Data	Allowed
Gynaecologist	Regular_Ron	Cardio_Data	Allowed
Paediatrician	Regular_Ron	Cardio_Data	Allowed
Urologist	Regular_Ron	Cardio_Data	Allowed
Pulmonologist	Regular_Ron	Cardio_Data	Allowed
GP	Regular_Ron	Cardio_Data	Not Allowed
Radiologist	Regular_Ron	Cardio_Data	Not Allowed
Nurse	Regular_Ron	Cardio_Data	Not Allowed

Discussion on the computational time taken for query execution involves time taken for checking the consistency of the all the asserted axioms, time taken for generating inferred axioms through the rules and time taken for executing the query. Asserted axiom and inferred axioms are one-time activities considered as initialization process and then queries could be executed in milliseconds. Table 5 shows the details of time taken in various steps of initialization. In totality it takes approximately 30 secs for initialization. We ran queries through Mongo DB database as well as Postgre database. On an average Mongo Bd takes 167 microseconds and Postgre takes 171 micro seconds to execute one query.

Table 5. Time Taken for Initialization

Reasoner (HermiT) & Rule Engine (DROOL)	**Time (ms)**
Time take for Class - Satisfiability	11
Time taken for Object Property - Satisfiability	25
Time taken to export 1085 Axioms to Rule Engine	459
Time taken to generate 273 Inferred Axioms	27931
Time taken to transfer Inferred Axioms to Owl	7
Total Time	**28473**

Complexity of the ontology is high as it has in total 1085 axioms. Ontology has lot of clarity as reasoning is provided as each step and meta data information is also provided. For emergency response system all the nodes in the ontology are well connected and ERS is tested for all possible task category, hence the ontology is complete.

6. Conclusion and Future Scope

We have successfully developed a need-based access control system (NdRAdAC) which can work with any heterogenous data source. Hospitals can be added to the ERS system as plugins. Need- Based access control is not dependent on underlying data storage structure. Access control framework was tested with respect to accuracy, adaptability, consistency and computational Efficiency. Need-based access control method was able to understand the genuineness of the requester and acted appropriately. Unlike other task-based methods this framework has a balanced approach toward adaptability. Adding any new hospital will take approximately 30 sec for initialization then the queries will be accessed in milliseconds. No SQL and SQL both the kind of database were tested and found that the system is compatible with both. As shown in the result analysis section that increasing the number of observations does not impact the performance of the system.

In the future we would like to develop a web-based application for this NbRAdAC system and let hospitals connect to the application. Also, we would like to add a forecasting-based module which can help predict how many resources will be required if similar calamity takes place in a particular region.

References

1. Minga-León, S., Gómez-Albores, M.A., Bâ, K.M., Balcázar, L., Manzano-Solís, L.R., Cuervo-Robayo, A.P., & Mastachi-Loza, C.A. (2018). Estimation of water yield in the hydrographic basins of southern Ecuador. Hydrology and Earth System Sciences Discussions, 1-18. https://doi.org/10.5194/hess-2018-480
2. Sherpa, S.F., Shirzaei, M., Ojha, C., Werth, S., & Hostache, R. (2020). Probabilistic Mapping of August 2018 Flood of Kerala, India, Using Space-Borne Synthetic Aperture Radar. IEEE Journal of Selected

Topics in Applied Earth Observations and Remote Sensing, 13, 896-913.
3. Ajay, A. (2019). Role of technology in responding to disasters: insights from the great deluge in Kerala. Current Science, (00113891) 116(6).
4. Gupta, P., Khanna, A., & Majumdar, S. (2012). Disaster management in flash floods in Leh (Ladakh): A case study. Indian journal of community medicine: official publication of Indian Association of Preventive & Social Medicine, 37(3), 185-190.
5. Hendra, W.Z. (2018). Community Participation in Flood Disaster Management in Sumbawa Regency (case study in Songkar Village). In E3S Web of Conferences, EDP Science, 73, 08004.
6. Kawaguchi, K., Araki, E., Hoshino, M., Yokobiki, T., Matsumoto, H., Nishida, S., & Kaneda, Y. (2014). Decision-making on seafloor surveillance infrastructure site for Earthquake and Tsunami monitoring in Western Japan. In IEEE, OCEANS 2014-TAIPEI, 1-4.
7. Eblé, M., Titov, V., Mungov, G., Moore, C., Denbo, D., & Bouchard, R. (2011). Signal-to-noise ratio and the isolation of the 11 March 2011 Tohoku tsunami in deep-ocean tsunameter records. In IEEEOCEANS'11 MTS/IEEE KONA, 1-4.
8. Goltz, J. (2017). Tsunami Generated by MJMA7. 4 (MW6. 9) Fukushima, Japan, Earthquake on November 22, 2016.
9. Corona Virus Diseas 2019 (COVID 19) Situation Report 79. https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200408-sitrep- 79-covid-19.pdf?sfvrsn=4796b143_4
10. Bijon, K.Z., Krishnan, R., & Sandhu, R. (2013). A framework for risk-aware role based access control. In 2013 IEEE Conference on Communications and Network Security (CNS), 462-469.
11. Cruz, J.P., Kaji, Y., & Yanai, N. (2018). RBAC-SC: Role-based access control using smart contract. IEEE, Access, 6, 12240-12251.
12. Wang, P., & Jiang, L. (2015). Task-role-based access control model in smart health-care system. In MATEC Web of Conferences, EDP Sciences, 2, 01011.
13. John, J.C., Sural, S., & Gupta, A. (2017). Optimal Rule Mining for Dynamic Authorization Management in Collaborating Clouds using Attribute-based Access Control. In IEEE 10th International Conference on Cloud Computing (CLOUD, 739-742.
14. Farroha, B., & Farroha, D. (2012). Challenges of operationalizing, dynamic system access control: Transitioning from ABAC to RAdAC. In IEEE International Systems Conference SysCon, 1-7.
15. Yang, Y., & Liu, S. (2014). Research on the quantification method of the operational need based on access purpose and exponential smoothing. In IEEE 7th Joint International Information Technology and Artificial Intelligence Conference, 516-522. http://doi/org/10.1109/ITAIC.2014.7065104
16. Rasheed, A.A. (2017). A trusted computing architecture for health care. In IEEE 7 International Conference on Information Networking (ICOIN), 46-50).
17. Ahmed Al Faresi. (2011). Risk based models for managing data privacy in healthcare, ProQuest LLC, Ph.D. Dissertation, George Mason University, I Management Using Environmental Knowledge," IEEE 23rd International WETICE SBN-978-1-2671-0042-9. https://eric.ed.gov/?id=ED540374
18. Mondal, A., & Goswami, R.T. (2021). Enhanced Honeypot cryptographic scheme and privacy preservation for an effective prediction in cloud security. Microprocessors and Microsystems, 81, 103719. https://doi.org/10.1016/j.micpro.2020.103719
19. Lu, Z., & Sagduyu, Y. (2016). Risk assessment based access control with text and behavior analysis for document management. In MILCOM 2016-2016 IEEE Military Communications Conference, 37-42. https://doi.org/10.1109/MILCOM.2016.7795298
20. Jia, Y.J., Chen, Q.A., Wang, S., Rahmati, A., Fernandes, E., Mao, Z.M., ... & Unviersity, S.J. (2017). ContexloT: Towards Providing Contextual Integrity to Appified IoT Platforms. In NDSS, 2(2), 2-2.
21. Tsigkanos, C., Pasquale, L., Ghezzi, C., & Nuseibeh, B. (2015). Ariadne: Topology aware adaptive security for cyber-physical systems. In IEEE/ACM 37th IEEE International Conference on Software Engineering, 2, 729-732. https://doi.org/10.1109/ICSE.2015.234
22. Fugini, M., Hadjichristofi, G., & Teimourikia, M. (2014). Dynamic security modeling in risk management using environmental knowledge. In IEEE 23rd International WETICE Conference, 429-434. https://doi.org/10.1109/WETICE.2014.42
23. Petracca, G., Capobianco, F., Skalka, C., & Jaeger, T. (2017). On risk in access control enforcement. In Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, 31-42. https://doi.org/10.1145/3078861.3078872
24. Fernandes, E., Jung, J., & Prakash, A. (2016). Security analysis of emerging smart home applications. In IEEE symposium on security and privacy (SP), 636-654. https://doi.org/10.1109/SP.2016.44
25. Malik, A.A., Anwar, H., & Shibli, M.A. (2016). Self-adaptive access control & delegation in cloud computing. In 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD),169-176. https://doi.org/10.1109/SNPD.2016.7515896

26. Mondal, A., Das, A.K., Nath, S., & Goswami, R.T. (2020). Review Study on Different Attack Strategies of Worm in a Network. Webology, 17(2), 363-375.
27. Yuce, B., & Rezgui, Y. (2015). An ANN-GA semantic rule-based system to reduce the gap between predicted and actual energy consumption in buildings. IEEE Transactions on Automation Science and Engineering, 14(3), 1351-1363.
28. Setiono, R., Baesens, B., & Mues, C. (2008). Recursive neural network rule extraction for data with mixed attributes. IEEE transactions on neural networks, 19(2), 299-307.
29. Li, L., Yu, Y., Bai, S., Hou, Y., & Chen, X. (2017). An Effective Two-Step Intrusion Detection Approach Based on Binary Classification and $ k $-NN. IEEE Access, 6, 12060-12073.
30. Lee, C. H., Su, Y. Y., Lin, Y. C., & Lee, S. J. (2017). Machine learning based network intrusion detection. In 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), 79-83.
31. Kumar, G.R., Mangathayaru, N., Narsimha, G., & Reddy, G.S. (2017). Evolutionary approach for intrusion detection. In IEEE International Conference on Engineering & MIS (ICEMIS) 1-6.
32. Shone, N., Ngoc, T.N., Phai, V.D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence, 2(1), 41-50.
33. Farahnakian, F., & Heikkonen, J. (2018). A deep auto-encoder based approach for intrusion detection system. In IEEE 20th International Conference on Advanced Communication Technology (ICACT), 178-183.
34. Rushin, G., Stancil, C., Sun, M., Adams, S., & Beling, P. (2017). Horse race analysis in credit card fraud—deep learning, logistic regression, and Gradient Boosted Tree. In IEEE systems and information engineering design symposium (SIEDS), 117-121.
35. Paula, E.L., Ladeira, M., Carvalho, R.N., & Marzagao, T. (2016). Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering. In 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 954-960.
36. Srivastava, K., & Shekokar, N. (2020). Machine Learning Based Risk-Adaptive Access Control System to Identify Genuineness of the Requester. In Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough, 129-143. https://doi.org/10.1007/978-3-030-38445-6_10
37. Parida, P.K., & Sahoo, S.K. (2013). Multiple Attributes Decision Making Approach by TOPSIS Technique. International Journal of Engineering Research & Technology, 2(11), 907-912.
38. Sodhi, B., & T V, P. (2012). A simplified description of Fuzzy TOPSIS. arXiv preprint arXiv:1205.5098.
39. Datta, D., Mishra, S., & Rajest, S.S. (2020). Quantification of tolerance limits of engineering system using uncertainty modeling for sustainable energy. International Journal of Intelligent Networks, 1, 1-8. https://doi.org/10.1016/j.ijin.2020.05.006
40. Dharmarajan, R., & Sharmila, C. (2016). The evaluation of topsis and fuzzy-topsis method for decision making system in data mining. International Research Journal of Engineering and Technology (IRJET), 3(9).
41. Li, J., Tang, J., Li, Y., & Luo, Q. (2008). Rimom: A dynamic multistrategy ontology alignment framework. IEEE Transactions on Knowledge and data Engineering, 21(8), 1218-1232. https://doi.org/10.1109/TKDE.2008.202
42. Kim, G.W., & Lee, D.H. (2019). Intelligent health diagnosis technique exploiting automatic ontology generation and Web-based personal health record services. IEEE Access, 7, 9419-9444. https://doi.org/10.1109/ACCESS.2019.2891710
43. De Giacomo, Giuseppe & Lenzerini, Maurizio. (1996). TBox and ABox Reasoning in Expressive Description Logics. Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR'96). 1996. 37-48.
44. Simperl, E.P.B., Tempich, C., & Sure, Y. (2006). Ontocom: A cost estimation model for ontology engineering. In International Semantic Web Conference, 625-639.
45. Raad, J., & Cruz, C. (2015). A survey on ontology evaluation methods. In Proceedings of the International Conference on Knowledge Engineering and Ontology Development, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. https://doi.org/10.5220/0005591001790186