Journal of Computer and Communications, 2022, 10, 133-157
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Experience-Based Access Control in UbiComp: A New Paradigm

Nalini A. Mhetre[1*], Arvind V. Deshpande[2], Parikshit N. Mahalle[3]

[1]Department of Computer Engineering, Sinhgad College of Engineering, Pune, India
[2]Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Pune, India
[3]Department of AI and Data Science, Vishwakarma Institute of Information Technology, Pune, India
Email: *nalini.mhetre@gmail.com, principal.skncoe@sinhgad.edu, aalborg.pnm@gmail.com

How to cite this paper: Mhetre, N.A., Deshpande, A.V. and Mahalle, P.N. (2022) Experience-Based Access Control in UbiComp: A New Paradigm. *Journal of Computer and Communications*, **10**, 133-157.
https://doi.org/10.4236/jcc.2022.101007

Received: December 28, 2021
Accepted: January 27, 2022
Published: January 30, 2022

Abstract

Experience is a sociological concept and builds over time. In a broader sense, the human-centered equivalents of experience and trust apply to D2D interaction. Ubiquitous computing (UbiComp) embeds intelligence and computing capabilities in everyday objects to make them effectively communicate, share resources, and perform useful tasks. The safety of resources is a serious problem. As a result, authorization and access control in UbiComp is a significant challenge. Our work presents experience as an outcome of history (HI), reliability (RL), transitivity (TR), and Ubiquity (UB). This experience model is easily adaptable to a variety of self-regulating context-aware access control systems. This paper proposes a framework for Experience-Based Access Control (EX-BAC) with all major services provided by the model. EX-BAC extends attribute-based access control. It uses logical device type and experience as context parameters for policy design. When compared with the state-of-the-art, EX-BAC is efficient with respect to response time.

Keywords

Access Control, Experience-Based Access Control, Experience Model, History, Reliability, Transitivity, Ubiquitous Computing, Ubiquity

1. Introduction

The vast majority of connected devices in UbiComp can process information; however, some are just designed to monitor data and transmit it to another device for computation. Consider a smart thermostat that senses external climate and triggers the Air Conditioner unit to auto-start if it's too hot and humid or adjusts the temperature while it is running. This sort of smart and pervasive na-

ture attracts customers. These devices can interact in a sort of learning that allows them to acquire and store knowledge, as well as change user preferences. Statistics [1] predicted that, by 2027, 41 billion connected devices will be inducted into IoT mesh, it's a massive quantity. Their data collecting and processing requirements will need more efficient and reliable edge nodes. They must be capable of managing digital data created inside a safe, trustworthy, and self-regulating ecosystem of smart objects. When neither side has experience or awareness of the other's reputation, there is always some risk associated with transactions. In the UbiComp ecosystem, a typical connection with a network provider may be obsolete. It may have been substituted with a far more stringent link involving lots of new unknown associations, networking devices, and providers. Furthermore, these devices may be highly mobile. Mobility will affect the dynamics of networks, needing significant modification. To deal with this amount of uncertainty and risk in UbiComp scenarios, it is critical to establish acceptable and effective approaches. These measures should foster trust and aid communicative entities in completing secure transactions.

Experience is a sociological concept that describes what happens when two things interact in the past. As a result of every engagement, the parties involved become more aware of one another's activities (*i.e.*, whether or not they met expectations). M-Webster dictionary explains experience as "direct observation of or participation in events as a basis of knowledge". The same dictionary also says the experience is "the fact or state of having been affected by or gained knowledge through direct observation or participation". Experience is the result of collective awareness of the activity of entities, which depicts their interrelationship. The more experience one has, the more valuable it is in establishing trust.

The human-centered equivalent of experience and trust also applies to D2D interaction in a ubiquitous word. In a nutshell, the experience is subjective and environment-dependent. This implies that the behavior of a single device may vary depending on the environment in which it is used. For the same set of devices, the experience may also differ depending on the situation. When two entities do not have a history of direct communication, a trust link can be built through sentential experiences generated by the recommendations. The experience can be modeled in the ubiquitous environment utilizing device interactions/events. These interactions/events can be logged and analyzed to provide a better understanding of the experience. In our previous work, a trust and context-aware access control framework [2] and an experience model [3] were proposed.

This paper presents an access control framework based on experience and logical device type as context parameters. Logical device types considered are CONSTRAINT, SEMIPOWERFUL, and POWERFUL depending on their networking environment [4]. This model considers the human notion of trust to build experience. As a foundation for modeling experience, trust and reputation are seen as essential. However, a fundamental component or attribute of a node,

ubiquity, is given less emphasis [5]. Our approach calculates a node's experience by weighing ubiquity equally or more. As a result, our model calculates overall experience using attributes including history, reliability, transitivity, and ubiquity, which are discussed in more depth in the next sections.

The remaining sections of the paper are organized as follows: Related work and experience modeling is discussed in Section 2 and Section 3 respectively. Section 4 and Section 5 are about the proposed experience-based access control (EX-BAC) model and EX-BAC framework. Experiment scenarios and results are discussed in Section 6 and Section 7 presents an evaluation of EX-BAC. Section 8 concludes the paper.

2. Related Work

2.1. Trust Management Schemes

Numerous approaches have been developed to formalize trust in a variety of areas, including peer-to-peer networks [6] [7], pervasive computing [8], and the IoT [9]. Recent methods [10] [11] began employing an ML approach to generalize and formalize trust models, in contrast to previous methods that were context or domain-specific. These models may be divided into two types: centralized and distributed models. [12] proposes a distributed way to manage trust based on direct interactions among edge things. The model estimates trustworthiness using the beta distribution and measurement theory. To measure uncertainty in the model, a supplementary characteristic, confidence, is employed in conjunction with direct interactions. In [13], the author presented a trust management approach for IoT (TRM-IoT) sensor device collaboration and inferring decisions based on fuzzy logic. Based on direct and indirect interactions, the final trustworthiness score is generated using fuzzy logic. Models in [12], [13] lacks context-awareness factors.

A strategy for trust-based context-aware access management using fuzzy logic (FTBAC) in the IoT is outlined in [14]. They base trust on three factors: experience, knowledge, and suggestion. The author [15] presented a methodology for evaluating trust that was based on direct and indirect interactions, and knowledge. This method concentrates on the SIoT context and incorporates data from direct observations. The model also considers experience with a global reputation to encompass heterogeneous dimensions of trust. The CBSTM for the IoT [16] is intended to foster collaboration among trusted nodes while limiting malicious node interaction. The significance of the context, compute power, confidence, and feedback all influence the node transaction factor. TAS-IoT model [17]—Device-level message authentication is used in the Internet of Things (IoT) concept. Depending on who sent the communication, the device decides whether or not to authenticate it. In CTMS-SIOT [18], the author adopted a feedback system. In [19] Trust management model based on fuzzy logic is proposed which is a generic model of trust based on behavior, recommendation belief and uncertainty. Evaluation of discussed models is summarized in **Table 1**

Table 1. Summary of trust management schemes.

Ref	Details	Context-Aware	Direct Trust/exp	Transitivity	Reliability	Ubiquity
[12]	Based on Beta distribution along with Measurement theory	No	Yes	No	No	No
[13]	Fuzzy logic-based trust management (TRM-IoT)	No	Yes	No	No	No
[14]	Context-Aware Fuzzy Trust-based—FTBAC	Yes	Yes	Yes	No	No
[15]	Trust evaluation model based on reputation, experience—SIoT focused	No	Yes	Yes	Yes	No
[16]	Context-based Social Trust Model for IoT (CBSTM-IoT), context importance, computation power, confidence, and feedback	Yes	Yes	Yes	No	No
[17]	TAS-IoT—Adaptive security in the internet of things based on trust.	No	Yes	Yes	No	No
[18]	Context-based TMS for the Social IoT (CTMS-SIOT)	Yes	Yes	Yes	No	No

based on whether the following five characteristics are considered or not:

- Context-aware—It refers to the ability of a model to consider context information.
- Direct Trust—Trust based on direct interactions between nodes. This parameter checks whether the model considers historical interaction while calculating overall trust.
- Transitivity—When there is no interaction with a new node, a node may rely on its trusted peer nodes to collect transitive trust of this fresh interaction.
- Reliability—Reliability parameters confirms how reliable a subject is and is calculated based on recent interactions.
- Ubiquity—It refers to the mobile nature of nodes in an access network.

2.2. Access Control Models

Devices were primitive in the early days of computing. Security to those devices was provided using username and password for the device. Only persons with username passwords can access systems. When network-based systems were introduced, and multiple entities started accessing systems, new securities provisions were adapted based on different roles and security concerns. Traditional access control models which were introduced during evolution, such as ACM, ACL, DAC, MAC, role-based access control (R-BAC) are founded on the prin-

ciple that computers can predict who their users will be. Traditional AC techniques work well in a centralized, reasonably stable environment. In these environments subjects and objects are familiar, their permissions are rarely altered.

In an era of ubiquitous computing, the market is flooded with different types of devices, and new devices are coming at a rapid speed. Mobility in these devices adds the requirement of context-awareness like location, time, and other environmental factors to be considered while designing access control models. This type of environment created a challenge to provide a robust, flexible, scalable AC model.

B. Cha [20] introduced the A-BAC approach, which uses characteristics to determine AC decisions. Scalability and flexibility, making it suited for distributed systems like IoT and UbiComp. Wang *et al.* [21] and Smari *et al.* [22] described the problem of using standards (A-BAC) that don't have provision for context handling and trust attribute. They extended the existing model (E-ABAC) to accommodate context and Trust, which makes the model usable for collaborative computing. Yitao and John [23] introduced key-based encryption and used physical tags to encrypt session keys. The proposed solution achieved high security and context awareness but the solution is not scalable due to the limitation of the number of keys that can be created. Trust between users is achieved as physical tags are used which are considered to be provided by a competent authority but misuse of tags is not considered in the solution.

3. Experience Modeling

3.1. Fundamental Concept

Previous work [3] presented a mathematical model for experience in the UbiComp environment. Quantifying the concept of experience will help us design solutions more efficiently. The requester or subject is the node that makes the request, whereas the service owner or object is the node that fulfills it. B denotes a requester, while A denotes an owner.

The experience score of A on B is indeed a positive integer between 0 and 1 and denoted as.

$EX_{A,B} = 1$: Indicates strong positive experience of A on B.

$EX_{A,B} = 0.5$: A had a neutral or no experience on B (default).

$EX_{A,B} = 0$: A had a negative experience on B.

3.2. Attributes of Experience Model

Figure 1 shows a model of experience calculation using parameters such as HI, RL, TR, and UB. This model includes novel parameter ubiquity. The experience score (EX) represents a node's experience with another node. A higher experience score value indicates a better probability of the node's trustworthiness. The model shows experience is a function of these four parameters: HI, RL, TR, and UB. This can be used to compute the experience as per Equation (1).

$$EX_{A,B} = f\left(HI_{A,B}, RL_{A,B}, TR_{A,B}, UB_{A,B}\right) \tag{1}$$

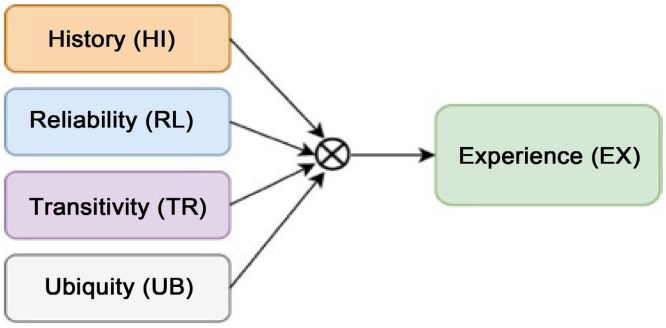


Figure 1. Components of experience calculation.

3.2.1. History (HI)

History or history attribute [3] is calculated using several previous interactions subject node B has with A. It may be assumed that a target device is more reliable if there are a large number of interactions over time that are both consistent and frequent. Take a scenario, for context c, A needs to compute the HI for B for nth interaction. B to A interactions is denoted by the symbol $e = \{e_1, e_2, \cdots, e_n\}$, and depending on the time of occurrence, are recorded as positive or negative. A calculates the aggregate of each past positive as well as a negative experience with B. It is denoted as P_B and F_B. History attribute is calculated as shown in Equation (2).

$$HI_{A,B}^{c} = cnf_{B}^{c} + \left(u_{B}^{c} \times a_{B}\right) \tag{2}$$

where cnf_B^c is confidence, $cnf_B = \dfrac{P_B}{P_B + F_B + 2}$ and is u_B uncertainty, $u_B = \dfrac{2}{P_B + F_B + 2}$.

The success of interactions will always be unclear when little or no knowledge is present. The uncertainty factor is between [0, 1]. There are fewer interactions between A and B as the level of uncertainty approaches one. There are enough interactions to calculate the HI attribute if this value is close to 0.

3.2.2. Reliability (RL)

The term "Reliability" is described as "the ability of an item to perform a required function under stated conditions for a stated time period" [24]. Reliability measures how reliable or trustworthy a gadget is based on an assessment of its prior interactions in a given time. Because sensing, processing, and transmitting information in real-time are critical in UbiComp, reliability is vital for effective communication. If Node B consistently produces successful results for A, a conclusion can be taken that Node B is reliable for node A. Node A can compute the reliability of node B based on recent interactions record between B→A. When node B interacts with node A. A total number of interactions in window W is denoted as *nr*. Reliability in context c is calculated as shown in Equation (3).

$$RL_{A,B}^{c} = \log_e\left(nr^c + 1\right) \times \frac{Pr_B^c}{nr^c} \times \frac{1}{\log_e\left(W^c\right)} \tag{3}$$

where,

W^c = Maximum number of previous interactions taken into account for calculating reputation in context c.

nr^c = Total interactions such that ($0 < nr \le W^c$).

Pr_B^c = Number of positive interactions in window W in context c.

3.2.3. Transitivity (TR)

In a given context, transitivity indicates a transitive trust link between the subject node and other reliable buddy nodes. In practice, trust is often not transitive and is only useful in limited situations. The work in [25] discussed the importance of having trust objectives that are identical and semantically consistent throughout transitive trust paths. This means that in one context, if A trusts B and in the same context B also trusts C, then only it is possible to build a trust path between A and C, and of course in the same context. If there are not enough interactions between B and A, A can query its other active peers in the network about B's experiences. Consider, peer node C then $EX_{A,C}$ represents the experience of A with C. $EX_{C,B}$ is C's experience about B. When A receives a request from B for a resource or service, A discovers that the two have never interacted before. Because of this, A sends out requests to all of its trustworthy associates, urging them to pass on their experiences to B. C revert backs to A with its EX regarding B. If A feels that this score is sufficient to grant B's request, B will be granted access. This will create a transitive trust path between A and B and will be denoted as, $TR_{A,B}$. $TR_{A,B}$ will be calculated as, $\min\left(E_{A,C}, E_{C,B}\right)$. $TR_{A,B}$ is nothing but a transitive experience score. When a node gets transitive experiences from peers $j = (1, 2, 3, \cdots, n)$, the transitive aggregate experience is determined as stated in Equation (4).

$$TR_{A,B} = \frac{\sum_{j=1}^{n} \min\left(E_{A,j}, E_{j,B}\right)}{n} \tag{4}$$

3.2.4. Ubiquity (UB)

Ubiquity refers to the degree to which a node is ubiquitous, demonstrating its movable nature. Nodes are more ubiquitous when they are moving; they are less ubiquitous when they are static or move slowly. According to [5], during an interaction, if node speed increases beyond a certain speed, packet drop happens. As a result, ubiquity can be exploited to adjust experience calculations. To illustrate, suppose that A trustor node wishes to send data to a particular location. In case the node remains static for the whole packet transfer, its experience will be computed, considering the node is non-movable. Assuming that node is now moving during packet transmission, effecting increased packet drop and lowering the experience value. Packet loss is related to the node's speed; the faster the node, the greater the packet loss. Observations show that device ubiquity affects the entire communication experience, hence the ubiquity component UB is included in the experience model.

Theoretically, if the speed of the node is between the minimum speed $V_{\min}$,

and the estimated average speed V_{avg}, the mobility factor *mf*, is set to 1. When this *mf* is set to 1, no change is made to the experience value. Thus, it can be expected that whether a node is static or moving at the permissible minimum speed, the experience value will remain unchanged. *mf* decreases as speed increases, and when the node reaches its maximum speed, V_{max}, it becomes zero, resulting in low experience. Thus, *mf* ranges between 0 and 1, ($0 \le mf \le 1$). *i.e.*

$$mf \propto \frac{1}{V_{current}}$$

where $V_{current}$ is the current speed of the node. In other words, as speed increases, the experience decreases. This means that if the node is steady, the experience will be good; if the node is moving or at a fast speed, the experience will be impacted proportionally. The mobility factor *mf* is stated in Equation (5).

$$mf = \frac{V_{max} - \min\left(\max\left(\text{avg}\left(V_{min}, V_{max}\right), V_{current}\right), V_{max}\right)}{\text{avg}\left(V_{min}, V_{max}\right) - V_{min}} \tag{5}$$

where,

V_{max} = Maximum allowed speed of node.

V_{min} = Minimum speed of node.

It is also possible to set correction factors based on access networks; for example, if connectivity is Ethernet, the node is static and is 1. Similarly, as shown in Table 2, values can be assigned to various access networks. These values can be adjusted considering application use cases. The ubiquity attribute can be calculated using Equation (6).

$$UB = mf \times cf \tag{6}$$

3.2.5. Computation of EX Score

From Equation (1), $EX_{A,B} = f\left(HI_{A,B}, RL_{A,B}, TR_{A,B}, UB_{A,B}\right)$. This function works on the weighted average of HI, RL, TR, UB. In general, the formula for weighted average is: Weighted Average = $\frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i}$ Where *x* is distribution and *w* is weight. The $EX_{A,B}$ of A with B is then computed using Equation (7).

$$EX_{A,B} = \frac{\left(W1 * HI_{A,B}\right) + \left(W2 * RL_{A,B}\right) + \left(W3 * TR_{A,B}\right) + \left(W4 * UB_{A,B}\right)}{w_1 + w_2 + w_3 + w_4} \tag{7}$$

4. Proposed EX-BAC Model

From existing literature, it is observed that there is a clear logical grouping of the different assets involved in ubiquitous systems. Based on this grouping, systems

Table 2. Ubiquity correction factors.

Medium	Mobile Network	Wi-MAX	WIFI 802.1	Wired network
Value of cf	0.7	0.9	0.95	1

can be categorized as device, network, and cloud. Each logical group in UbiComp have their own needs and security challenges compared to existing internet security challenges. UbiComp devices have unique issues that are distinct from those faced by traditional Internet clients. The diverse form factors, power requirements, and hardware characteristics of devices all provide challenges. UbiComp devices may use little power, lower bandwidth networks than current Internet systems. Often UbiComp devices connect through wireless or cellular technologies. Wireless and cellular networks have much higher latency and more disconnections than fixed networks. The protocols that are used for the world wide web are relatively high data-intensive and power-savvy for resource constraint devices. Security approaches such as encryption and certificates are problematic and, in some cases, unfeasible in tiny devices.

The proposed access control framework is logically divided into three layers of service architecture (See Figure 2).

- Discovery and classification service
- Control and enforcement service
- Data service

4.1. Computation of EX Score

This service acts as first-level security in our access control framework. This service encompasses modules like node classification, node trust establishment, request handler, response handler, and experience evaluator.

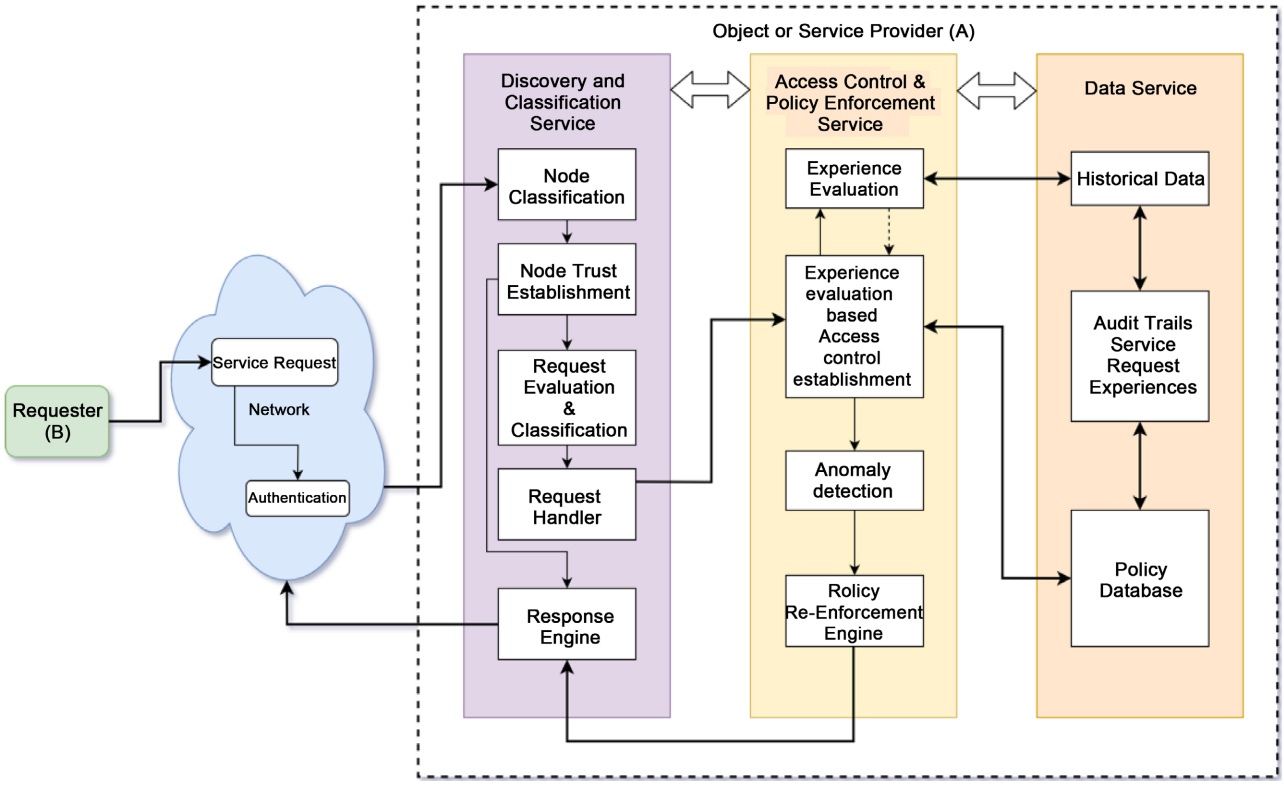


Figure 2. Proposed access control model.

4.1.1. Node Classification

When any node A wants to access the resource of other service node B, it establishes a network connection with the service node. Once the connection is established, node B gets authenticated by the standard authentication process. The authentication process is not discussed as it is assumed that it's already defined in the network during design time. Once the node is authenticated, the node will be classified based on different criteria. In previous work [4], devices were classified as CONSTRAINT, SEMIPOWERFUL, and POWERFUL based on network parameters like RSSI, Energy, Throughput, and medium. These node types are logical labels and can easily incorporate into the policy framework as one of the context parameters. Other context parameters will be extracted by this module such a device ubiquity, *i.e.*, whether the device is mobile or static as discussed in the previous section.

4.1.2. Node Trust Establishment

This module established primary Trust between A and B. It checks whether the requester is supported by the system or not. This module takes attributes from the classification module and if the node is valid then the request is passed to the request handler. If basic Trust is not formed, *i.e.* the device is not passed the basic requirement of the system, the access request is rejected from this point.

4.1.3. Request Handler

We consider, the larger scale of devices are present in the network and try to access various resources of each other. As resources are finite and can be accessed by limited requesters at a time. The request handler module checks the availability of resources at request time and if the resource is available for access, then only forwards requests for further evaluation. Request queuing and resource management is assumed to be the core functionalities that any request handler would essentially have, hence detailing out of such functionalities has not been included in this paper.

4.2. Access Control and Policy Enforcement Service

4.2.1. Access Control Module

For access control, the proposed model is leveraging the ABAC framework [26] as the base for the access control module (Refer **Figure 3**). In ABAC access control policies are formed using attributes groups such as subject, object *i.e.*, resource, action, and context attributes. Access to resources is given based on these policies, and new policies can be added dynamically whenever required. The scalability, fine-grained policy, and flexibility provide more secure access to resources than traditional access methods.

Request from request handler will be forwarded to Policy Enforcement Point (PEP). PIP converts the corresponding request in policy format and sends it to the policy decision point (PDP) for further evaluation. PDP acts as the central processing unit for policies. PDP evaluates incoming requests against stored policies and returns decisions either allow or deny. During policy evaluation,

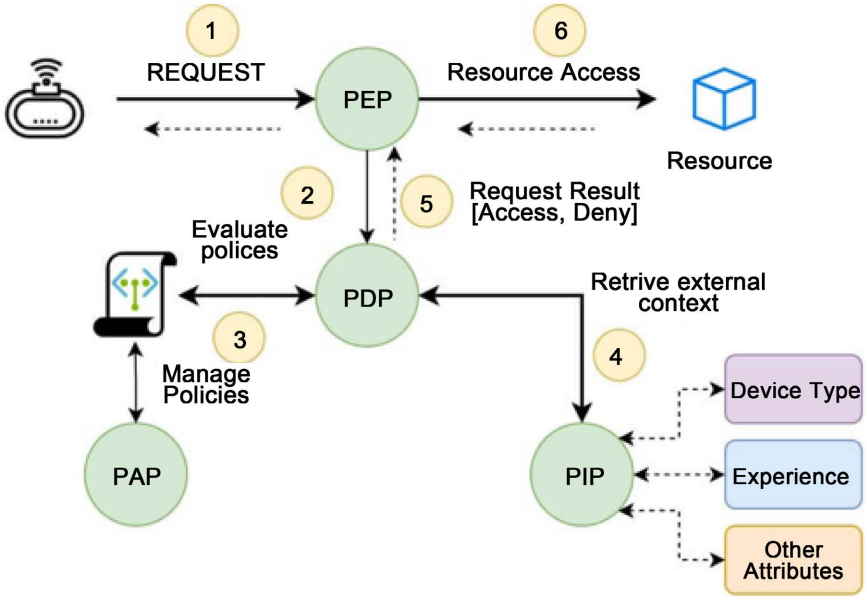


Figure 3. Access control module.

PDP initiates a request to Policy Information Point (PIP) to retrieve context information like Experience coefficient, device type, etc. Policy Administration Point manages creation (PAP), update, and deletion of policies evaluated by PDP. PAP uses a Policy database managed by a data service to store all policies. PAP is part of the Policy Re-Enforcement Engine.

4.2.2. Experience Evaluation

When PDP requests PIP for context attributes for given Access Control Object, PIP communicates with the experience evaluation module to calculate the experience of the subject. Experience will be calculated as mentioned in the experience model and sent back to PIP. Experience evaluation module access data service to get historical device data as well as communicates with peer nodes if required for transitive experience.

4.2.3. Policy Re-Enforcement

When PDP evaluates policy and creates a response, it sends a response via a policy enforcement engine.

4.3. Data Service

Data service provides interfaces for accessing historical data regarding interactions as well as acts as a storage provider for storing policies created by policy administration points. Storage media and data handing is a node-specific task hencedetails are not considered.

5. Implementation of EX-BAC

5.1. Access Control Elements

In Ex-BAC, the below elements are involved while creating a resource or service

request. JSON-based policy language is used to store and communicate different policies which are based on XACML standards.

- Subject: A subject is something that requests access to resources or services. A subject can be anything either user or application or any device in the environment.
- Resource: The object or service which is requested by the subject. These resources are governed by various policies.
- Action: An action that can be performed on the subject when policy executes
- Context: Set of parameters that handle the dynamic aspect of the access control situation.

In Ex-BAC, defined policies contain any condition on one or more attributes of the above for elements. For example, for the subject, the condition can be defined as providing access to a resource only if a node is "Node A" or node name starts with "NODE" or ".*" means any node. If these conditions are met, the PDP will produce an access decision based on an evaluation process.

5.2. Experimental Setup

All experiments are conducted on the system Intel i5@3.4 GHz with 8 GB of memory. The Proposed model is implemented using python with the help of various packages like pandas, py-abac, bottle, pymongo, json, scikit-learn. Various device configurations considered for experimentation are listed in Table 3. Constraint: Only rssi, energy values are considered for determining device type. Experiment simulation is carried out with controlled values as well as random values for various parameters like speed, signal, and movement of the subject towards the resource. Also, for battery-operated devices, simulation is carried out for battery charge and discharge. The Experiment used devices like PC with direct power and with both communication mediums *i.e.*, Ethernet and WIFI. Elements Involved in Ex-BAC Evaluation.

5.2.1. Subjects

In the experiment, three types of devices like pc, laptops, and mobiles are used. These devices are running either with an uninterrupted power supply or with

Table 3. Devices configurations.

Name	Type	Initial RSSI*	Speed	Energy	Connectivity	Energy Source
Node 1+	PC	45	0	100	Ethernet	Direct supply
Node 2	PC	45	0	100	Ethernet	Direct supply
Node 3	LAPTOP	45	0	20	Ethernet	Direct supply
Node 4	LAPTOP	5	1	10	WIFI	Battery
Node 5	MOBILE	40	30	90	Cellular	Battery
Node 6	MOBILE	30	10	60	WIFI	Battery

*: For rssi values refer to Table 7, +: Resource owner for experiment.

batteries. Compared to PC, devices like laptops that are running on batteries and communicating using WIFI are considered semi-powerful devices, while mobiles with cellular connectivity and low battery are considered as constrained devices. Both cellular and WIFI-connected devices are affected by environmental changes like changes in signal strengths and speed of the device itself, while devices which are connected to Ethernet are static and show strong connectivity. Power supply to fixed devices is uninterrupted while devices running on batteries have constraints that limit their usage of resources.

5.2.2. Resources

Three types of resources considered for experimentation are listed in Table 4. Video resources are always heavy resources when accessed consume heavy bandwidth and energy. On the other hand, audio resources are less power and bandwidth savvy compared to video resources. Text resources show the lowest consumption of bandwidth as well as energy.

Table 5 shows two context attributes, device type and experience. Device type context is derived using contribution two and experience is derived using contribution three. Polices are based on values of these context values. As shown in Table 5, to access resource Video.mp4, which is a video resource, device type must be POWERFUL, and the minimum experience score required is 0.4, while for the same resource if the device is SEMI_POWERFUL, the experience score required is 0.6.

Table 4. List of resources.

#	Resource	Owner	Type
1	Video.mp4	Node 1	Video File
4	Audio.mp3	Node 1	Audio File
8	File1.txt	Node 1	Text File

Table 5. Context attributes and experience values to resource access.

#	Resource	Device Type	Experience (Min Required)
1	Video.mp4	POWERFUL	0.4
2	Video.mp4	SEMI_POWERFUL	0.6
3	Video.mp4	CONSTRAINED	0.8
4	Audio.mp3	POWERFUL	0.47
5	Audio.mp3	SEMI_POWERFUL	0.5
6	Audio.mp3	CONSTRAINED	0.6
7	File1.txt	POWERFUL	0.45
8	File1.txt	SEMI_POWERFUL	0.45
9	File1.txt	CONSTRAINED	0.45

5.2.3. Access Network Connectivity

Devices can used access networks technologies for connection such as Ethernet, WIFI, and Cellular as listed in Table 6.

Some devices can use multiple connections based on their locations. For example, PC and Laptops either can be connected to Ethernet or WIFI and they choose a medium based on availability. Similarly, laptops and mobiles may use WIFI when available and mobiles can connect to a cellular network when on move.

Table 7 shows a mapping of actual RSSI signals values normalized in the range between 0 - 45 [27].

5.3. Default Policies in the Experimental Setup

The considered policy attributes and their respective values are given in Table 8.

5.3.1. Sample Policy Format in JSON

The following snapshot shows how a single policy can be created using simple JSON.

```
        "condition": "Equals",
        "value": "Video.mp4"
      }
    },
    "action": {
      "$.method": {
        "condition": "Equals",
        "value": "get"
      }
    },
    "context": {
      "$.experience": {
        "condition": "Lt",
        "value": 0.5
      },
      "$.devicetype": {
        "condition": "Equals",
        "value": "CONSTRAINED"
      }
    }
  },
  "priority": 0
}
```

Table 6. Access networks for device connectivity.

#	Access Network	Device Details
1	ETHERNET	PC, LAPTOP
2	WIFI	LAPTOP, MOBILE
3	CELLULAR	MOBILE

5.3.2. Access Request for PDP

PEP creates an object that represents an access request. This object provides all of the data necessary for the PDP to evaluate policies and make access choices. Sample policy can be visualized in JSON format as shown below.

```
    {
      "subject": {
        "id": "",
        "attributes": {
          "name": "Node2"
        }
      },
      "resource": {
        "id": "",
        "attributes": {
          "name": "Video.mp4"
        }
      },
      "action": {
        "id": "",
        "attributes": {
          "method": "get"
        }
      },
      "context": {
        "devicetype": "SEMI_POWERFUL",
        "experience": 0.6
      }

}
```

6. Results

After the implementation of the EX-BAC model, we executed three distinct experimental scenarios for performance analysis.

6.1. Scenario 1

In scenario one, Node 2 requests a video file to Node 1. Node 2 is a static node and connectivity is ethernet, clustering algorithm maps this node to the "POWERFUL" cluster and hence logical device type of Node 2 is labeled as "POWERFUL". There is no previous interaction history between Node 2 to

Table 7. RSSI mapping.

RSSI (in dBm)	Mapped RSSI	Verbal Meaning	Details (Signal, Data Speed)
≥ −70	>45	Excellent	Strong, Max
−80 to −85	40 - 45	Very Good	Very Good, Max
−70 to −80	30 - 39	Good	Strong, Good
−86 to −100	20 - 29	Fair	Fair, Data with drop-outs
< −100	10 - 19	Poor	Performance drops radically
−110 dBm	0 - 9	No signal	Very low, Disconnection

Table 8. Policy attributes and values to access resources.

No	Resource	Device Type	Ex Condition	Experience	File Type	Access
1	Video.mp4	CONSTRAINED	Lt	0.8	Video	deny
2	Video.mp4	SEMI_POWERFUL	Lt	0.6	Video	deny
3	Video.mp4	POWERFUL	Lt	0.4	Video	deny
4	Video.mp4	CONSTRAINED	Gte	0.8	Video	allow
5	Video.mp4	SEMI_POWERFUL	Gte	0.6	Video	allow
6	Video.mp4	POWERFUL	Gte	0.4	Video	allow
7	Audio.mp3	CONSTRAINED	Lt	0.6	Audio	deny
8	Audio.mp3	SEMI_POWERFUL	Lt	0.5	Audio	deny
9	Audio.mp3	POWERFUL	Lt	0.47	Audio	deny
10	Audio.mp3	SEMI_POWERFUL	Gte	0.5	Audio	allow
11	Audio.mp3	POWERFUL	Gte	0.47	Audio	allow
12	Audio.mp3	CONSTRAINED	Gte	0.6	Audio	allow
13	File1.txt	CONSTRAINED	Lt	0.45	Text	deny
14	File1.txt	SEMI_POWERFUL	Lt	0.45	Text	deny
15	File1.txt	POWERFUL	Lt	0.45	Text	deny
16	File1.txt	CONSTRAINED	Gte	0.45	Text	allow
17	File1.txt	SEMI_POWERFUL	Gte	0.45	Text	allow
18	File1.txt	POWERFUL	Gte	0.45	Text	allow

Node 1. In scenario1 weight for the UB, the factor is 0.4 hence it is directly reflected in EX. Experiment results show, how experience grows when interaction grows. Results of scenario 1 are plotted in a graph as shown in Figure 4.

The graph in Figure 4 visually validates our mathematical model presented in Equation (7). For this scenario to calculate RL, W = 20, *i.e.*, Maximum 20 number of previous interactions taken into account for calculating reputation.

Figure 5 shows how HI and RL contribute to experience calculation. With positive HI and RL scores, experience grows logarithmically. After the interaction count passes the W window, the experience moves proportionally with history and reliability.

Figure 6 shows a snap of the first 24 interactions of the scenario. With positive interactions, reliability increases logarithmically. After 20 successful interactions, RL reaches 1, which indicates node 2 is fully reliable.

The graph also indicates, positive experience builds with the increase in HI, RL factors. In scenario 1, the device is stationary hence Ubiquity factor (UB) is always maximum *i.e.*, 1. For first interaction contexts for access control module are DeviceType = "POWERFUL" and EX = 0.56. When PIP sends a request with these contexts to PDP for evaluation, policy results in "GRANTED" as it has "allow" permission for the requested resource.

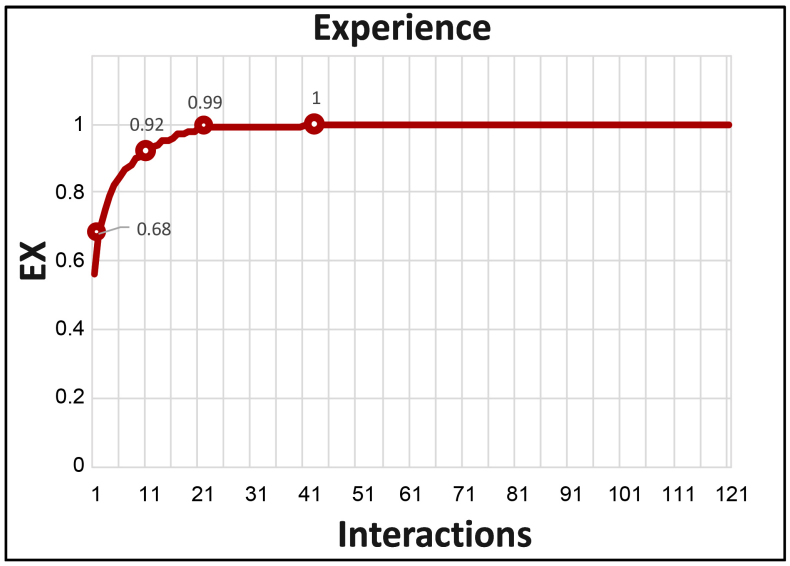


Figure 4. Number of interactions vs. experience.

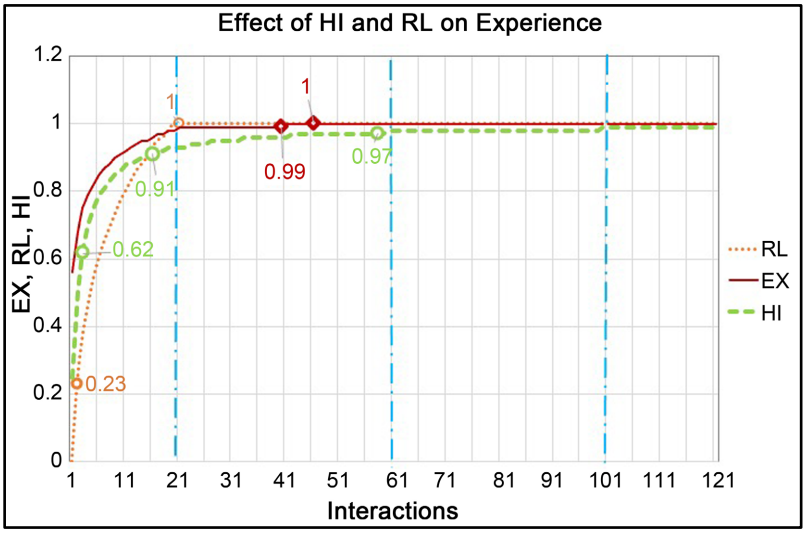
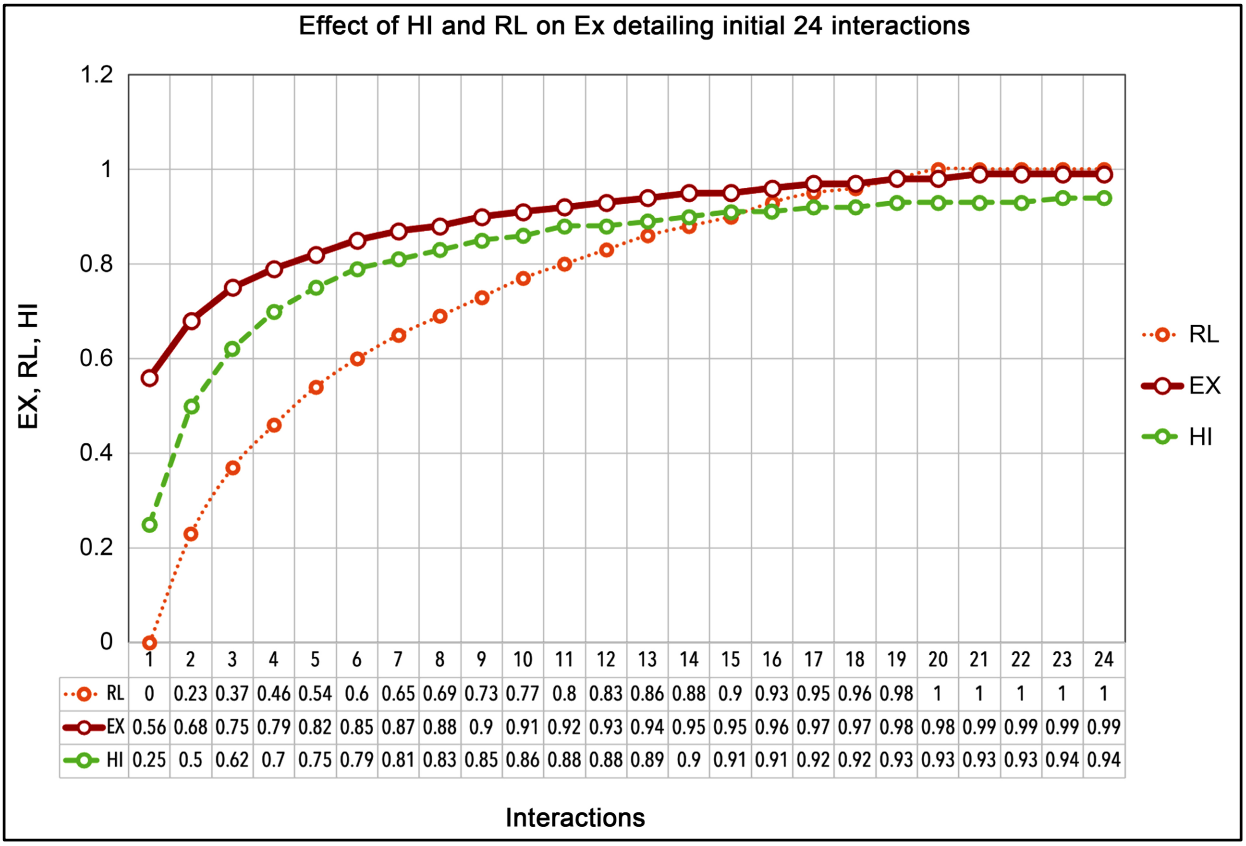


Figure 5. Effect of HI and RL on EX.

6.2. Scenario 2

In scenario two, Node 4 requests a video file to Node 1. Node 4 is a semi-mobile node and connected to a WIFI network, clustering algorithm maps this node to the "SEMI_POWERFUL" cluster and hence logical device type of device is labeled as "SEMI_POWERFUL". There is no previous history between node 4 to node 1. In this scenario, UB and RL directly affected the EX.

Figure 7 shows the first 122 interactions of the scenario. For the first 17 interactions, Node 4 falls in the "SEMI_POWERFUL" cluster, and experience ranges between 0.5 to 0.46, hence, according to the policy matrix resource request is "denied". On the 18th interaction, the Node 4 moved from SEMI_POWERFUL to POWERFUL cluster with an experience score of 0.46. For this interaction, PDP returns' allowed' response, hence resource is granted to Node 4. On 40th



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
RL	0	0.23	0.37	0.46	0.54	0.6	0.65	0.69	0.73	0.77	0.8	0.83	0.86	0.88	0.9	0.93	0.95	0.96	0.98	1	1	1	1	1
EX	0.56	0.68	0.75	0.79	0.82	0.85	0.87	0.88	0.9	0.91	0.92	0.93	0.94	0.95	0.95	0.96	0.97	0.97	0.98	0.98	0.99	0.99	0.99	0.99
HI	0.25	0.5	0.62	0.7	0.75	0.79	0.81	0.83	0.85	0.86	0.88	0.88	0.89	0.9	0.91	0.91	0.92	0.92	0.93	0.93	0.93	0.93	0.94	0.94

Figure 6. Effect of HI and RL on EX detailing initial 24 interactions.

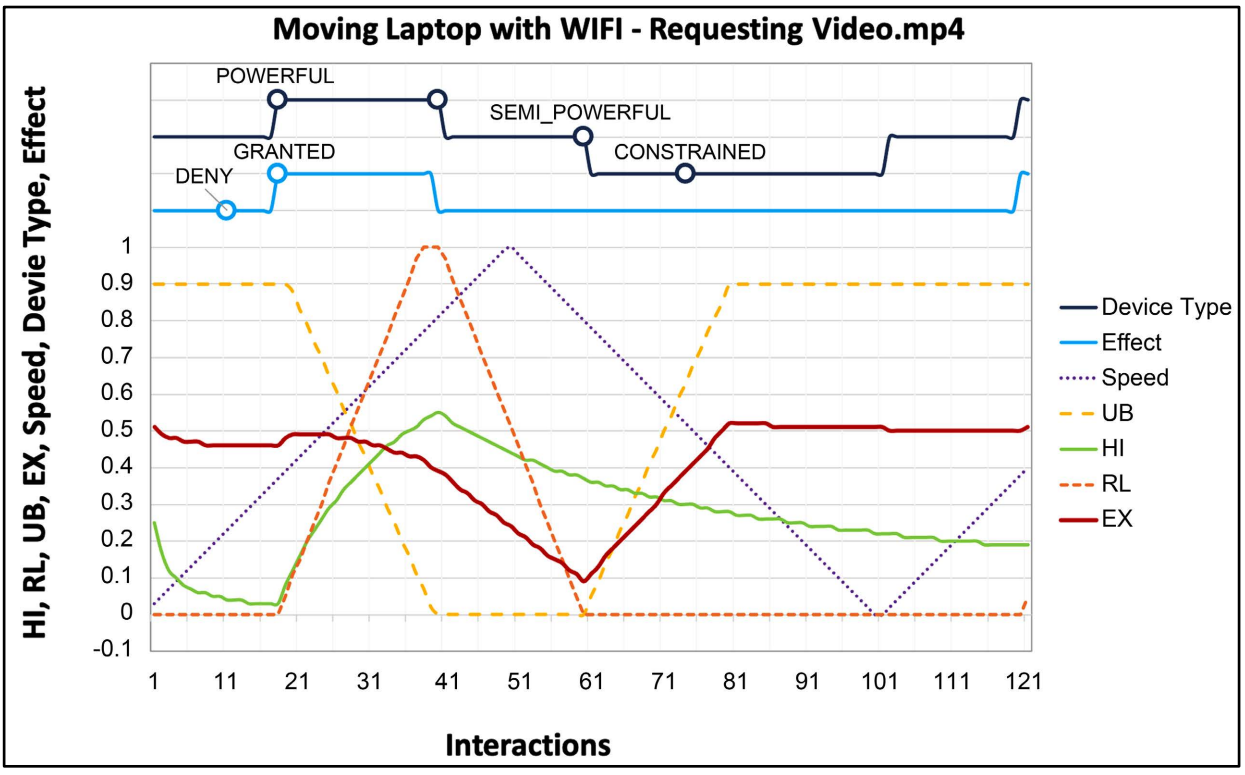


Figure 7. Scenario 2: Semi-mobile node with WIFI connectivity.

interaction, the speed of Node4crosses 80 which is the maximum allowed speed. Even though Node 4 was POWERFUL, the experience score lowered below 0.4, and thus PDP "denied" the request. On 62nd interaction, device signal drops below −110 dBm *i.e.*, 10, Node 4 get labeled as "CONSTRAINED" and policies for constrained devices with low experiences are applied by PAP. When Node 4 returned back to the POWERFUL cluster in interaction 121 with an experience value of 0.5, the resource is "granted".

6.3. Scenario 3

In scenario three, Node 5 requests a text file to Node 1. Node 5 is a mobile node and connected to a cellular network, clustering algorithm maps this node to the "POWERFUL" cluster because of good signal strength and energy. Hence, the logical device type of Node 5 is "POWERFUL". There are no previous interactions between Node 5 and Node 1. For this scenario weight of the Ubiquity attribute is adjusted to 0.4 which directly affects experience calculation.

Figure 8 shows the results of scenario 3, *i.e.*, the status of resource requests and the impact of environmental attributes like mobility, a communication network. Based on the values of HI and RL, PDP grants access to the resource. With the increase in device mobility and speed, experience decreases due to the UB which carries high weight in experience calculation. On the 25th interaction when Node 5 crosses max speed, the UB attribute becomes 0 and PDP denies request for the resource. Thus, it can be concluded that, reliability and mobility factors in experience calculation will play a crucial role due to a large number of mobile devices in UbiComp systems.

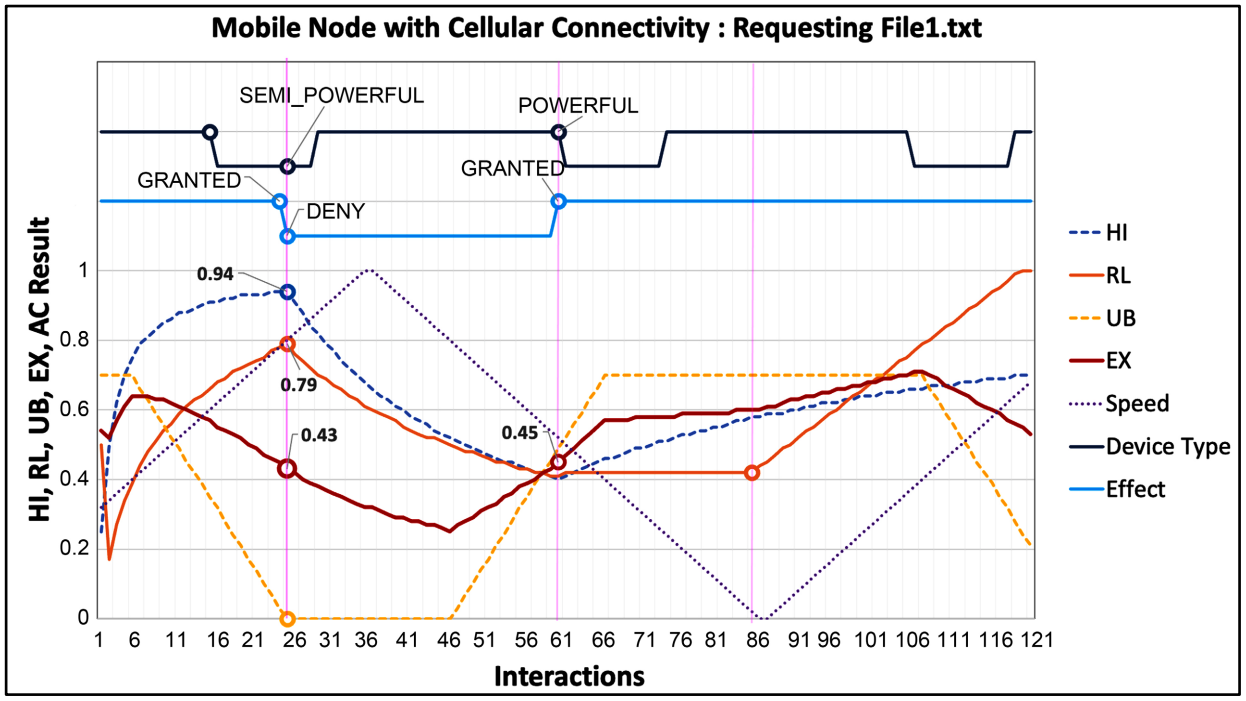


Figure 8. Scenario 3: mobile node with cellular connectivity.

7. Discussion and Analysis

In this section performance analysis of the EX-BAC model is discussed in three different ways as follows.

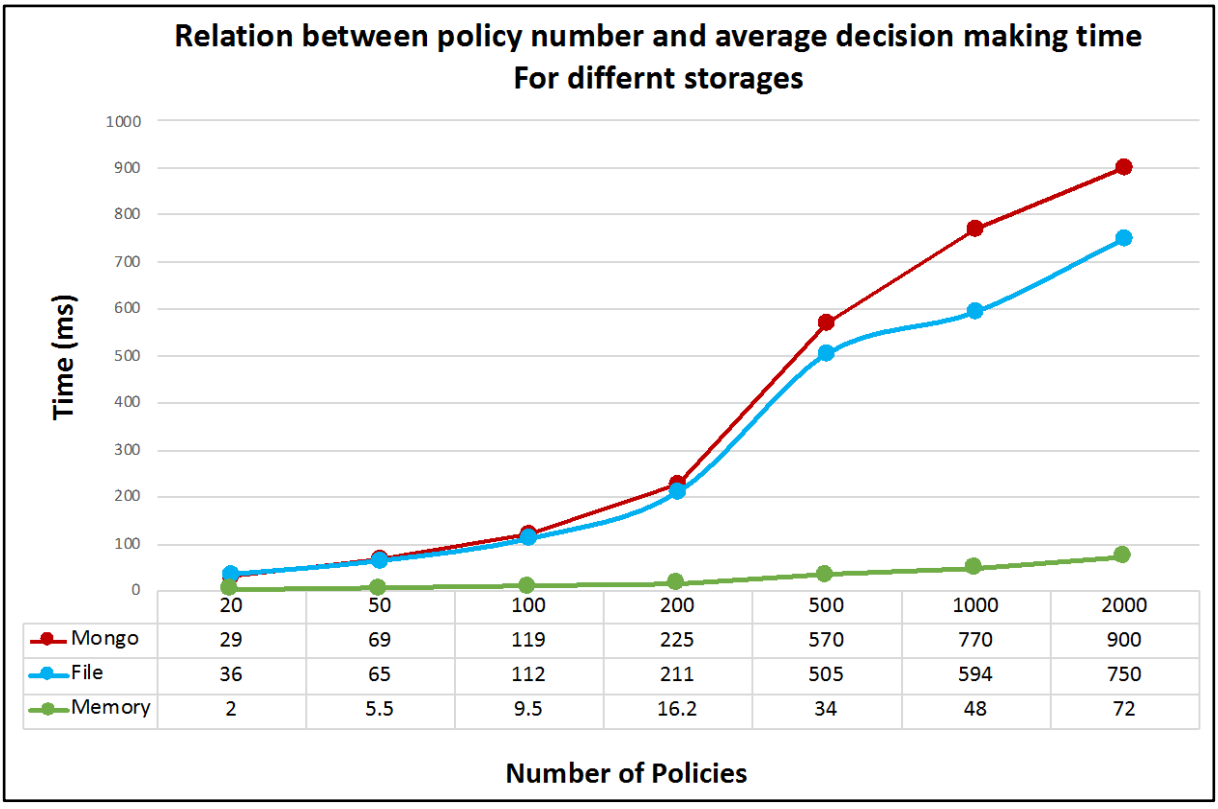
- Access request processing time: Effect on access request processing time when the number of policies are increased for different storage types.
- Policy decision-making time: EX-BAC model is compared with T-ABAC in policy selection.
- Experience calculation time: Total time required to calculate experience value with an increase in a number of history records.

7.1. Access Request Processing Time

To check the performance of the access control algorithm, the experiment checked performance with several policies created in data storage. The experiment was evaluated with two persistent storage and one in-memory storage. For the persistent storage, we selected MongoDB and the File system of the node.

Figure 9 shows comparison results for evaluation criteria. The evaluation results show that:

1) With the increase in policies, the time required to evaluate policy increases with persistent storage. Compared to the file system, MongoDB took considerably extra time when the number of policies crossed 1000. With the initial 20 policies, evaluation time for PAP was 24 ms which increased to 119 ms for 100



	20	50	100	200	500	1000	2000
Mongo	29	69	119	225	570	770	900
File	36	65	112	211	505	594	750
Memory	2	5.5	9.5	16.2	34	48	72

Figure 9. EX-BAC time evaluation for policy storage.

policies and reached 900 ms for 2000 policies. For file system-based storage, the trend shows a similar pattern but the time required after 1000 policies is lower than that of the MongoDB database. Both systems have their pros and cons, the file system is faster but the implementation and retrieval of data are complex.

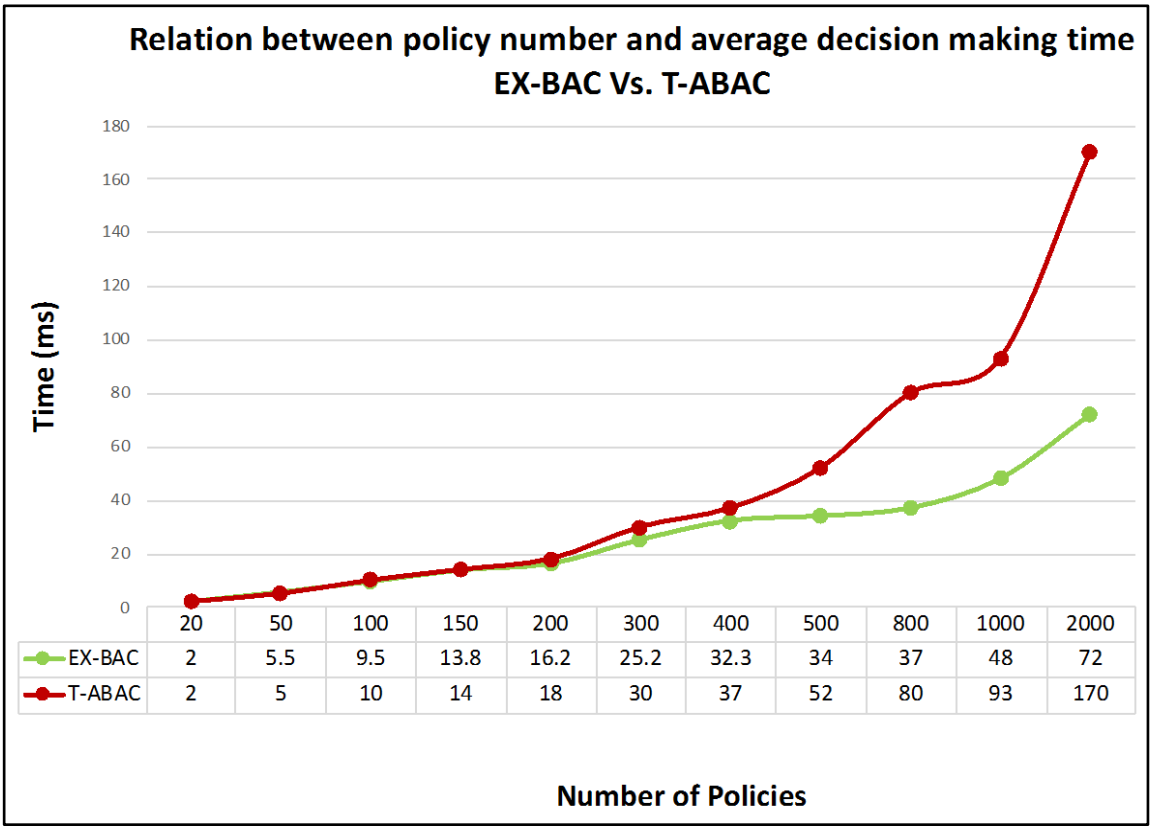
2) The experiment observed much higher performance with in-memory storage compared to persistent storage. Policy evaluation time with 20 policies to 2000 policies ranged between 4 to 5ms which shows policy evaluation algorithm is faster but performance depends on storage used to store policies. There is the scope of using in-memory databases like Redis which might give the mixed performance of in-memory and persistence storage.

7.2. Policy Decision Making Time

To demonstrate the validity of EX-BAC, the experiment results are compared to those of T-ABAC [21]. The main reason to choose T-ABAC to compare with EX-BAC is that both based on ABAC and experience also counts direct trust as one of the attributes.

Figure 10 shows the comparison for the relation between the number of policies and average decision-making time for EX-BAC and T-ABAC.

Both EX-BAC and T-ABAC show similar performance when policies are limited. Both show comparative performance till 200 polices. But EX-BAC outperforms T-ABAC when policy database size increases. For 500 policies EX-BAC



	20	50	100	150	200	300	400	500	800	1000	2000
EX-BAC	2	5.5	9.5	13.8	16.2	25.2	32.3	34	37	48	72
T-ABAC	2	5	10	14	18	30	37	52	80	93	170

Figure 10. Performance evaluation EX-BAC vs. T-ABAC.

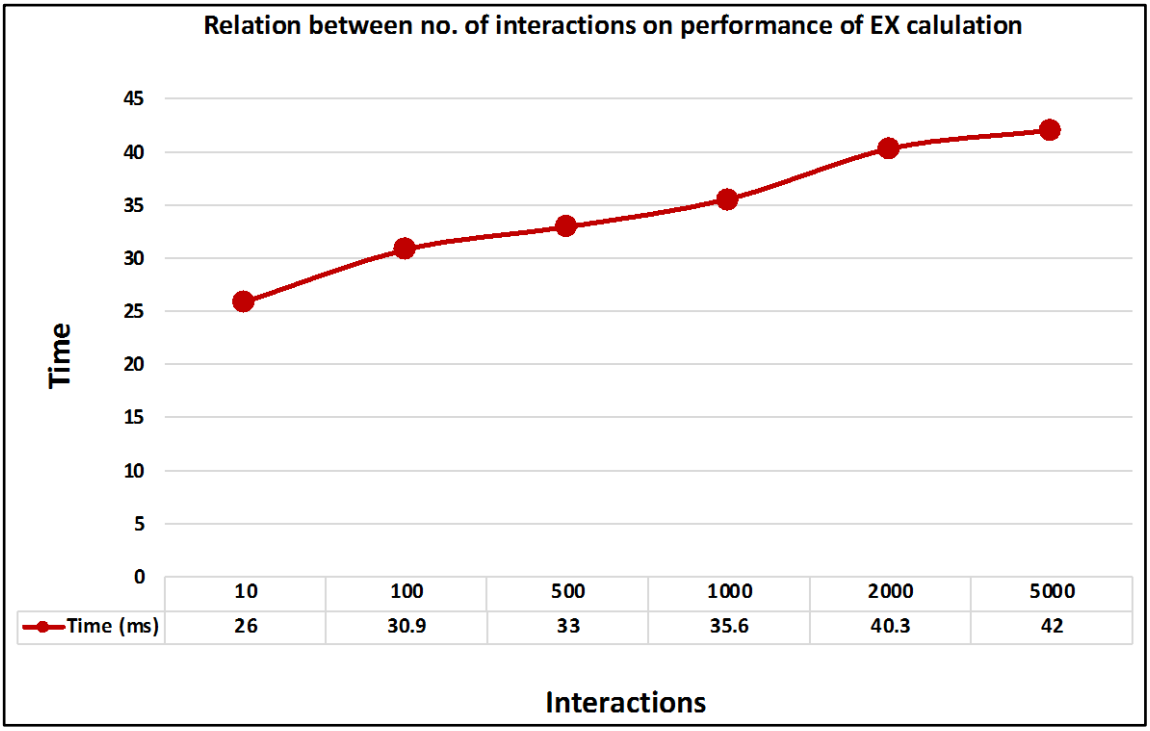


Figure 11. Time evaluation of EX-BAC for intercation count.

shows 50% more efficiency compared to T-ABAC. Though T-ABAC uses trust in the model but lacks reliability and ubiquity factor, which is proved very critical in ubiquitous scenarios by our work in this paper.

7.3. Experience Calculation Time

One of the major attributes in experience calculation is HI, *i.e.*, History of interaction. The results show that the time for the request evaluation is largely impacted by the number of interactions (See **Figure 11**).

Response time for requests change from 26 ms to 42 ms, when the number of interactions increased from 10 to 5000 for 20 policies. It is obvious that during the lifespan of the device there will millions of interactions can happen, hence there is scope for optimization for historical data storage. This optimization can be achieved by maintaining cumulative values of positive and negative interactions.

8. Conclusions

The proposed EX-BAC model shows how history, reliability, transitivity and ubiquity contribute to experience in UbiComp. This model shows device speed is inversely proportional to Ubiquity. From the results, it is observed that EX-BAC is flexible to accommodate different contexts. It is scalable also as policies can be added and removed whenever required. Model is also time efficient as compared to T-ABAC.

In the future, it will be interesting to tokenize experience using blockchain

and make it available universally because blockchain has created a new business network that combines ease of use with low costs and excellent security. We can see numerous possibilities to use the EX-BAC model in blockchain and distributed IoT.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Business Insider (n.d.) The Internet of Things 2020. https://www.businessinsider.com/internet-of-things-report?IR=T

[2] Mhetre, N., Deshpande, A.V. and Mahalle, P.N. (2022) Rethinking Access Control Mechanism for Ubiquitous Computing. In: Tuba, M., Akashe, S., Joshi, A., Eds., *ICT Systems and Sustainability*, Springer, Singapore, 157-165. https://doi.org/10.1007/978-981-16-5987-4_17

[3] Mhetre, N.A., Deshpande, A.V., Mahalle, P.N. and Thakre, P.A. (2021) Experience Modelling For Ubiquitous Computing : A Mathematical Approach. *Turkish Journal of Computer and Mathematics Education*, **12**, 5476-5488.

[4] Mhetre, N.A., Deshpande, A.V. and Mahalle, P.N. (2021) Device Classification-Based Context Management for Ubiquitous Computing Using Machine Learning. *International Journal of Engineering and Advanced Technology*, **10**, 135-142. https://doi.org/10.35940/ijeat.E2688.0610521

[5] Raikwar, A.K. (2012) Effect of Mobility on Trust in Mobile Ad-Hoc Network. In: Wyld, D., Zizka, J. and Nagamalai, D., Eds., *Advances in Intelligent and Soft Computing*, Vol. 167, Springer, Berlin, Heidelberg, 673-684. https://doi.org/10.1007/978-3-642-30111-7_64

[6] Kamvar, S.D., Schlosser, M.T. and Garcia-Molina, H. (2003) The Eigentrust Algorithm for Reputation Management in P2P Networks. *Proceedings of the* 12*th International Conference on World Wide Web*, Budapest, 20-24 May 2003, 640-651. https://doi.org/10.1145/775152.775242

[7] Tahta, U.E., Sen, S. and Can, A.B. (2015) GenTrust: A Genetic Trust Management Model for Peer-to-Peer Systems. *Applied Soft Computing Journal*, **34**, 693-704. https://doi.org/10.1016/j.asoc.2015.04.053

[8] D'Angelo, G., Rampone, S. and Palmieri, F. (2017) Developing a Trust Model for Pervasive Computing Based on Apriori Association Rules Learning and Bayesian Classification. *Soft Computing*, **21**, 6297-6315. https://doi.org/10.1007/s00500-016-2183-1

[9] Chen, I.R., Guo, J. and Bao, F. (2014) Trust Management for SOA-Based IoT and Its Application to Service Composition. *IEEE Transactions on Services Computing*, **9**, 482-495. https://doi.org/10.1109/TSC.2014.2365797

[10] Liu, X., Tredan, G. and Datta, A. (2014) A Generic Trust Framework for Large-Scale Open Systems Using Machine Learning. *Computational Intelligence*, **30**, 700-721. https://doi.org/10.1111/coin.12022

[11] Taylor, P., Barakat, L., Miles, S. and Griffiths, N. (2018) Reputation Assessment: A Review and Unifying Abstraction. *The Knowledge Engineering Review*, **33**, Article No. e6. https://doi.org/10.1017/S0269888918000097

[12] Ruan, Y., Durresi, A. and Uslu, S. (2018) Trust Assessment for Internet of Things in Multi-Access Edge Computing. *Proceedings of International Conference on Advanced Information Networking and Applications*, Krakow, 16-18 May 2018, 1155-1161. https://doi.org/10.1109/AINA.2018.00165

[13] Chen, D., Chang, G., Sun, D., Li, J., Jia, J. and Wang, X. (2011) TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things. *Computer Science and Information Systems*, **8**, 1207-1228. https://doi.org/10.2298/CSIS110303056C

[14] Mahalle, P.N., Thakre, P.A., Prasad, N.R. and Prasad, R. (2013) A Fuzzy Approach to Trust Based Access Control in Internet of Things. 2013 3*rd International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace and Electronic Systems*, *VITAE* 2013, Atlantic City, 24-27 June 2013, 1-5. https://doi.org/10.1109/VITAE.2013.6617083

[15] Truong, N.B., Lee, H., Askwith, B. and Lee, G.M. (2017) Toward a Trust Evaluation Mechanism in the Social Internet of Things. *Sensors*, **17**, Article No. 1346. https://doi.org/10.3390/s17061346

[16] Rafey, S.E.A., Abdel-Hamid, A. and El-Nasr, M.A. (2016) CBSTM-IoT: Context-Based Social Trust Model for the Internet of Things. 2016 *International Conference on Selected Topics in Mobile and Wireless Networking* (*MoWNeT*), Cairo, 11-13 April 2016, 1-8. https://doi.org/10.1109/MoWNet.2016.7496623

[17] Hellaoui, H., Bouabdallah, A. and Koudil, M. (2016) TAS-IoT: Trust-Based Adaptive Security in the IoT. *Proceedings of Conference on Local Computer Networks*, LCN, Dubai, 7-10 November 2016, 599-602. https://doi.org/10.1109/LCN.2016.101

[18] Ben Abderrahim, O., Elhedhili, M.H. and Saidane, L. (2017) CTMS-SIOT: A Context-Based Trust Management System for the Social Internet of Things. 2017 13*th International Wireless Communications and Mobile Computing Conference* (*IWCMC*), Valencia, 26-30 June 2017, 1903-1908. https://doi.org/10.1109/IWCMC.2017.7986574

[19] Mhetre, N.A., Deshpande, A.V. and Mahalle, P.N. (2016) Trust Management Model based on Fuzzy Approach for Ubiquitous Computing. *International Journal of Ambient Computing and Intelligence*, **7**, 33-46. https://doi.org/10.4018/IJACI.2016070102

[20] Cha, B., Seo, J. and Kim, J. (2012) Design of Attribute-Based access Control in Cloud Computing Environment. Vol. 120, Springer, Dordrecht, 41-50. https://doi.org/10.1007/978-94-007-2911-7_4

[21] Wang, J., Wang, H., Zhang, H. and Cao, N. (2017) Trust and Attribute-Based Dynamic Access Control Model for Internet of Things. 2017 *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (*CyberC*), Nanjing, 12-14 October 2017, 342-345. https://doi.org/10.1109/CyberC.2017.47

[22] Smari, W.W., Clemente, P. and Lalande, J.-F. (2014) An Extended Attribute Based Access Control Model with Trust and Privacy: Application to a Collaborative Crisis Management System. *Future Generation Computer Systems*, **31**, 147-168. https://doi.org/10.1016/j.future.2013.05.010

[23] Duan, Y. and Canny, J. (2005) Protecting User Data in Ubiquitous Computing: Towards Trustworthy Environments. *International Workshop on Privacy Enhancing Technologies*, Toronto, 26-28 May 2004, 167-185. https://doi.org/10.1007/11423409_11

[24] Bauer, E. and Adams, R. (2012) Reliability and Availability of Cloud Computing. John Wiley & Sons, Inc., Hoboken. https://doi.org/10.1002/9781118393994

[25] Jøsang, A. and Pope, S. (2005) Semantic Constraints for Trust Transitivity. *Conferences in Research and Practice in Information Technology Series*, **43**, 59-68.

[26] Abirami, G. and Venkataraman, R. (2019) Attribute Based Access Control Policies with Trust (ABAC-T) Mechanism in Pervasive Computing. *Journal of Advanced Research in Dynamical and Control Systems*, **11**, 699-706.

[27] Papliatseyeu, A., Osmani, V. and Mayora, O. (2010) Indoor Positioning Using FM Radio. *International Journal of Handheld Computing Research*, **1**, 19-31. https://doi.org/10.4018/jhcr.2010070102