BDFIS: Binary Decision Access Control Model Based On Fuzzy Inference Systems

Diogo Domingues Regateiro[1], Óscar Mortágua Pereira[2], Rui L. Aguiar[3]
Instituto de Telecomunicações
DETI, University of Aveiro
Aveiro, Portugal
{diogoregateiro[1], omp[2], ruilaa[3]}@ua.pt

***Abstract*—Access control is a ubiquitous feature in almost all computer systems, and as data becomes more and more of an important asset for organizations, so do the associated access control policies. However, with the increase in the amount of data being produced, e.g. in IoT and social networks, the interest in simpler access control is increasing as well since more subjects (public, researchers, etc.) are now requesting access to it. Defining the exact conditions to allow each subject to access the data can be difficult, especially when vaguely defined conditions such as "expertise of a researcher" come into play. Fuzzy Inference Systems (FIS) allow to process these vague conditions and enables access control mechanisms to be more easily applied. The contribution of this paper lies in showing how a FIS can be used to output binary access control decisions (grant/deny) and what are the differences in the inference process that stems from restricting the output to these two output values.**

Keywords-fuzzy systems, vague knowledge, information security, access control.

I. Introduction

Access control has always been an important feature in any system, be it physical or digital, as it restricts access to a resource in a controlled and selective manner [1]. The most successful access control models are usually those that mimic real-world ways of managing permissions within the context of their application, of which the Role-based Access Control (RBAC) [2] model is a key reference. RBAC is a classical model that maps subjects trying to access some resource to a role, a meaningful category within the context of the system being protected, and crisp access control rules define the resources a subject playing a given role may access to successfully complete its tasks. Other classical access control models operate in a similar manner, using crisp rules that clearly define which resources each subject may access.

However, with the advent of big data and social networks, the quantity and complexity of data available that needs to be stored and processed have increased considerably. Classical access control models are ill-suited to handle these scenarios, as they require tight mappings between subjects, objects, and permissions. This means new subjects must be manually assigned to their permissions before they can access the data, which introduces delays and adds security management loads.

Additionally, the real world is not always as unambiguous as the classical access control models require it to be in their policies. For example, some documentation from European projects may not be publicly available but could be disclosed to experts researching in some area related to a project. There is no hard definition of what makes someone an expert, so normally it would have to be checked manually on a case-by-case basis. In such situations, the fuzzy set theory is an appropriate solution since it can handle vague concepts, such as expertise of the subject, without requiring crisp values within its rules. Thus, allowing policy rules to be richer in meaning and flexibility.

The theory of fuzzy logic [3] aims to capture how human perception and cognition interpret the world, which is not unambiguous all the time. To this end, it uses relative graded memberships between a subject and a vague concept. Thus, fuzzy logic permits the inclusion of vague human assessments in computing problems, which proved to be an effective way to deal with multi-criteria problems [4]. Solutions known as fuzzy inference systems (FIS) were then designed to map a set of inputs to outputs, using fuzzy logic and fuzzy sets to define vague conditions. These characteristics allowed fuzzy logic and fuzzy set theory to find a lot of uses in various areas, such as medicine [5]–[7], computer security [8]–[14], networking [15], [16], aeronautics [17], stock trading [18], and many others [4], [19]–[22].

Consider a community managed public data, like a wiki, where only subjects that are experts (to some degree) on the contents of a page would be able to modify it. Since wiki pages are generally given categories related to their contents and other related tags, the access control system could access services such as Scopus to retrieve the number of publications, their keywords, citations, etc. This information could then be fed into a fuzzy inference system to determine the level of expertise of the user and help the access control system to make its access control decisions. Thus, leading to fewer modifications by users with malicious intents and more modifications with quality content.

Thus, the contribution of this paper lies in proposing an access control model that uses a FIS to make binary access control decisions, herein known as BDFIS; the presentation of application scenarios where such a system could be useful; and what are its benefits/issues when compared to other access control mechanisms.

The rest of this paper is organized as follows: section II will

This work is funded by FCT/MEC through national funds and when applicable co-funded by FEDER – PT2020 partnership agreement under the project UID/EEA/50008/2019 and SFRH/BD/109911/2015.

DOI reference number: 10.18293/SEKE2019-039

provide some of the state of the art in regards to the application of fuzzy set theory in access control; section III will provide the analysis made to each step made during the output generation process of the BDFIS; section IV will present a proof of concept of the proposed model; and section V will provide a short discussion while addressing the issues found.

II. Related Work

The fuzzy set theory is a topic that has been researched in recent years to tackle scenarios where the information that needs to be processed is vague, which can include management science, politics, social psychology, artificial intelligence, and access control, among others [4]. This capability to handle vague information is what enables it to be useful in scenarios where binary decisions must be made and the decision rules are difficult to define in a crisp manner.

Surprisingly, it was found that there is very little research done in terms of the application of fuzzy set theory in access control systems. The issue is suspected to come from the fact that by using vague conditions in the form of fuzzy sets, a fuzzy-based access control system does not explicitly state which input values would grant access to some resource which would not. This paper is focused on exploring these limitations and where fuzzy systems can be improved for this area of intervention.

In [23], the authors introduce Fuzzy Role-based Access Control (FRBAC), which uses fuzzy relations between users-roles and roles-permissions:

- USERS x ROLES → [0, 1]
- ROLES x PERMISSIONS → [0, 1]

This approach allows for users to have partial permission assignments, which are then used to calculate the access degree they have to each resource. Then, if the access degree is used directly to control access to a resource, the resource itself must have fractional access, defined using the following access function where *USERS* is the set of users, *OPS* the set of operations and *OBS* the set of objects:

- access: *USERS* x *OPS* x *OBS* → [0, 1]

Martínez-García et al. define a function that takes a threshold variable δ (i.e. a value between 0 and 1) and returns *grant* if the access degree is greater than δ or *deny* if not. However, this approach still restricts the fuzzy sets to roles. Therefore, it limits the type of access control logic that can be used. BDFIS, in contrast, does not require subjects to be mapped to the protected resources through any specific model, such as roles, allowing to abstract any mapping between them.

Another work was found where the trust level of devices is measured, so a fuzzy approach to trust-based access control could be achieved (FTBAC) [11]. This is done by capturing information about the devices to determine the vague concepts *Experience* (*EX*), *Knowledge* (*KN*) and *Recommendation* (*RC*), and several fuzzy sets (linguistic terms) were defined for each one. The following values for each concept are calculated for a context *c* between two devices *A* and *B*, used as inputs for the membership functions of the linguistic terms.

EX depends on the history of interactions v_k between *A* and *B*, where $k \in [0, n]$, incrementing or decrementing when a positive or negative interaction occurs, as shown in (1).

$$(\mathrm{EX})^{\mathrm{c}} = \frac{\sum_{\mathrm{k=1}}^{\mathrm{n}} \mathrm{v_k}}{\sum_{\mathrm{k=1}}^{\mathrm{n}} |\mathrm{v_k}|} \tag{1}$$

KN is calculated with the help of direct knowledge *d*, indirect knowledge *r*, and their respective weights ($\mathrm{W_d}$, $\mathrm{W_r}$), where $\mathrm{d}, \mathrm{r} \in [-1,1]$, $\mathrm{W_d}, \mathrm{W_r} \in [0,1]$, and $\mathrm{W_d} + \mathrm{W_r} = 1$, as shown in (2).

$$(\mathrm{KN})^{\mathrm{c}} = \mathrm{W_d} * \mathrm{d} + \mathrm{W_r} * \mathrm{r} \tag{2}$$

The *RC* is calculated by device *A* based on the summation of the RC values from *n* other devices about device *B*. $\mathrm{W_i}$ and $(\mathrm{r_c})_\mathrm{i}$ are weights assigned by device *A* to the recommendation of the $\mathrm{i^{th}}$ device and its *RC* value respectively, where $\mathrm{r_c} \in [-1,1]$ and $\mathrm{W_i} \in [0,1]$, as shown in (3).

$$(\mathrm{RC})^{\mathrm{c}} = \frac{\sum_1^{\mathrm{n}} \mathrm{W_i} * (\mathrm{r_c})_\mathrm{i}}{\sum_1^{\mathrm{n}} (\mathrm{r_c})_\mathrm{i}} \tag{3}$$

Different permissions can be mapped to different levels of trust, so depending on the level of trust the granted permissions change. This the access decisions solely based on the level of trust. If there are other access conditions, they need to be considered separately. Since BDFIS can abstract any mapping rule between subjects and resources, the level of trust can be used in the same manner. However, unlike FTBAC, other access requirements can be added without issue to the inference system.

Another work was carried out that uses fuzzy set theory to calculate a measure of risk and applies it to enhance the access security of eHealth cloud applications [10]. To achieve this, three different inputs are used: *data sensitivity*; *action severity*; and *risk history*. Next, a set of rules is applied to calculate the level of risk associated. A crisp output value is then determined by applying a defuzzification technique, which indicates the overall level of risk as a percentage. However, the process to determine whether the access should be granted given a risk level is not detailed. Moreover, this approach is specific to the measurement of the level of risk with a given access attempt. This limits the applicability of this approach when compared to BDFIS, which can use most concepts in its policies.

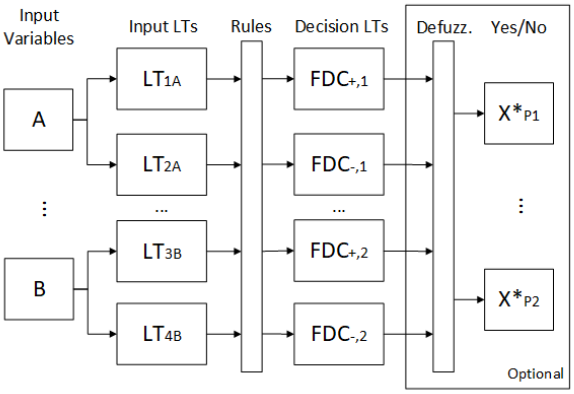


Figure 1. Conceptual BDFIS block diagram.

III. Binary Decision FIS Analysis

In this section, a FIS is analyzed regarding its applicability to binary decision making (access control grant/deny decisions) in each of its processing steps.

There are many different types of FIS [24]–[28]. However, the Mamdani-type FIS [28] was chosen for analysis since it is a type of system that is commonly available on most FIS implementation tools, has widespread use and it was found to be easily adaptable to support binary decisions. The Sugano-type FIS [27] is also commonly available in such tools, but it falls short as it has less expressive power and interpretability than the Mamdani-type FIS [29]. Fig. (1) shows the conceptual BDFIS that will emerge from the analysis made in this section and will serve as an illustrative guide to the proposed modifications.

The standard Mamdani-type FIS goes through the following set of steps during processing:

1. The determination of the set of fuzzy rules by an expert in the application context (i.e. the rules block);
2. The fuzzification of the input variable values into the input linguistic terms (LTs) using the associated membership functions;
3. The application of the fuzzy rules to establish the rule strengths to the output LTs, known as the fuzzy decision components (FDC) in the BDFIS;
4. The combination of the rule strength and the output LTs membership functions to determine the consequence functions for each output variable;
5. The combination of the consequence functions to get an output distribution function for each output variable;
6. The defuzzification step, which outputs a single crisp value for each output variable (a decision in the context of the BDFIS) given an output distribution function (required only if a crisp output is needed).

These steps will be detailed in the following subsections and how they were modified for the BDFIS.

A. Fuzzy Rule Determination

The fuzzy rule determination process involves deciding which linguistic terms are going to be used within the FIS, both for input and output variables, and how they influence each other using predefined rules.

The input linguistic terms should stay effectively the same as they still qualify the attributes available in the application context. The input linguistic terms and their membership functions are still required to be written by an expert.

The output variables and linguistic terms, however, are dependent on the decisions must be made. In the case of an access control system, the decisions are either to grant or deny a subject some permission to a resource. Thus, permissions can be declared as the output variables according to Def. (1).

Definition 1. Access permissions in a BDFIS are output variables associated with exactly two FDC linguistic terms: one for a positive decision FDC_+ (yes/grant); and one for a negative decision FDC_- (no/deny).

The rules can then take input linguistic terms from one or more input variables and establish a relation to one of the FDCs. To illustrate, consider two input variables *A* and *B*, with the linguistic terms $\mathrm{LT_A}$ and $\mathrm{LT_B}$, and an output variable *Z*. A rule can take the form "**if** *A* is $\mathrm{LT_A}$ **and/or** *B* is $\mathrm{LT_B}$ **then** *Z* is $\mathrm{FDC_\pm}$". For example, "**if** *Expertise* is *High* **and** *Activity* is *Moderate* **then** **Read** is *Granted*."

This shows how a FIS can be used to easily encode vague access conditions: given a set of vague concepts about a subject (e.g. *Expertise*, level of *Activity*, etc.), the permissions (*Read*, *Write*, etc.) to the resource are output variables that are defined by either being granted (FDC_+) or denied (FDC_-). Furthermore, the permission and decision pair are easily identifiable.

B. Input Fuzzification And Rule Strength

The input fuzzification process and rule strength determination are steps that qualify the input variables in terms of the defined linguistic terms. Given a set of linguistic terms $\mathrm{T_i}$ for an input variable i, the membership degree of a subject s to each linguistic term $t \in T_i$ is obtained by applying that linguistic term membership function μ_t, as shown in (4).

$$\mu_t(s)\colon t \to [0,1], \mathrm{t} \in \mathrm{T_i} \tag{4}$$

To illustrate, if $\mu_t(x) = x/20, 0 \leq x \leq 20$ is used to define the “high” linguistic term for the “number of publications” input variable, then if a subject has 15 publications it has a membership degree of $15/20 = 0.75$ to that linguistic term. The membership functions μ are defined by an expert in the application context the BDFIS is to be deployed on, since vague concepts like *Expertise* can change slightly depending on the context. After a membership degree is calculated for each input linguistic term, the rule strength for each output FDC can be determined. This is usually done by applying the fuzzy logic operators as dictated by the rules (AND, OR, and NOT). If more than one rule applies to the same FDC, the rule strength of each such rule is unified by typically applying the OR operator. These operators have several different implementations for fuzzy logic that can be used, but they always satisfy the De Morgan’s Laws.

To reiterate, there are only two possible output linguistic terms: the $\mathrm{FDC_-}$ and the $\mathrm{FDC_+}$. This makes it clear for which outcome a rule is being used for instead of having something more abstract, such as the user expertise level, and lets the system make access control decisions based on it.

C. Consequence Determination

The next step is to determine the consequence of the rules and to do so it is necessary to think about what the output is intended to be.

The goal is to have a FIS that can make a binary access control decision for each permission (i.e. grant or deny). As such, each decision is defined by two linguistic terms, the $\mathrm{FDC_+}$ and $\mathrm{FDC_-}$ output linguistic terms previously introduced. These represent the positive and negative decisions for a single output, respectively. This way, a subject attempting to access some piece of information or service is mapped automatically through rule strengths to each FDC.

Since each FDC is also a fuzzy set, each can have any membership function that the security expert chooses. However, a simpler approach is proposed to reduce the complexity of the calculations and the potential performance bottleneck that running a FIS can introduce. Instead of a user-defined output membership function, each FDC will have a predefined singleton function (see Def. (2)) instead.

Definition 2. A given function $\mathrm{f(x)}$ is a singleton function if its output is always 0 except for a single input value $\mathrm{x_0}$, for which its output is 1 as shown in (5).

$$\mathrm{f(x)} = \begin{cases} 1, & \text{if } \mathrm{x} = \mathrm{x_0} \\ 0, & \text{if } \mathrm{x} \neq \mathrm{x_0} \end{cases} \tag{5}$$

Since it is expected for the BDFIS to output a decision, it makes sense that they are each associated with a single value. This is partially the reason why the membership function for each FDC is proposed to be a singleton function. This is done by setting the single function $\mathrm{x_0}$ value to 0 for the $\mathrm{FDC_-}$ and to the value 1 for the $\mathrm{FDC_+}$. These values were carefully chosen since they simplify the defuzzification step considerably, which will be shown in section III.D.

Thus, the membership functions of the $\mathrm{FDC_-}$ ($\mu_-(\mathrm{x})$) and $\mathrm{FDC_+}$ ($\mu_+(\mathrm{x})$) are singletons that are defined as shown in (6) and (7), respectively.

$$\mu_-(\mathrm{x}) = \begin{cases} 1, & \text{if } \mathrm{x} = 0 \\ 0, & \text{if } \mathrm{x} \neq 0 \end{cases} \tag{6}$$

$$\mu_+(\mathrm{x}) = \begin{cases} 1, & \text{if } \mathrm{x} = 1 \\ 0, & \text{if } \mathrm{x} \neq 1 \end{cases} \tag{7}$$

Finally, the process of determining the consequence consists of truncating the output membership function, i.e. both $\mu_-(x)$ and $\mu_+(x)$. Def. (3) shows how this process is accomplished.

Definition 3. The process of truncating a given function *f* at the value $\mathrm{y} = \mathrm{y_0}$ generates a new function *g* that has the same output as *f* except that any output value greater than $\mathrm{y_0}$ becomes $\mathrm{y_0}$, as shown in (8).

$$\mathrm{g(x)} = \min(\mathrm{f(x)}, \mathrm{y_0}) \tag{8}$$

Determining the consequence from these singleton functions also becomes easier for the following reasons:

1. Singleton functions only have one input value ($\mathrm{x} = \mathrm{x_0}$) for which the output is not 0, thus only one value may have to be truncated;
2. The FDC singleton membership functions always output 1 for the value $\mathrm{x} = \mathrm{x_0}$;
3. The rule strength applicable to each FDC always lies within the range $[0,1]$ since it is the result of the application of fuzzy logic.

Since the output membership function of each FDC is either 0 or 1 as stated in reasons (1) and (2) and the rule strength is a value within the range $[0,1]$ as stated in reason (3), the consequence function *C* is determined by simply replacing the output value 1 with the rule strength RS for that FDC, as shown in (9) for FDC_- and (10) for FDC_+.

$$\mathrm{C_-(x)} = \begin{cases} \mathrm{RS_-}, & \text{if } \mathrm{x} = 0 \\ 0, & \text{if } \mathrm{x} \neq 0 \end{cases} \tag{9}$$

$$\mathrm{C_+(x)} = \begin{cases} \mathrm{RS_+}, & \text{if } \mathrm{x} = 1 \\ 0, & \text{if } \mathrm{x} \neq 1 \end{cases} \tag{10}$$

These consequence functions can then be used in the defuzzification step to generate a single, crisp output value. This explains in part how using singleton functions simplifies the computation of the final decisions. However, further benefits from this approach will be explored in the following subsection.

D. Consequence Combination And Defuzzification

The next step in the process involves combining the consequence functions of both FDCs ($\mathrm{C_-}$ and $\mathrm{C_+}$) into an output distribution function for each output decision. This allows to apply a defuzzification method, an inverse transformation to the fuzzification step, that outputs a crisp output value for each output decision.

These steps are optional and depend on the level of information the system requires to reach a decision. If the system requires more than a crisp output value, it can use the rule strengths given to each FDC and make a more informed decision this way. For example, a use case could require an access control system to ask a human to manually grant or deny access if the rule strengths of both FDCs are close to one another.

However, if the system requires a crisp value between 0 and 1, then the consequence functions can be combined into an output distribution function (Def. (4)) and a defuzzification step may be used.

Definition 4. Given the consequence functions $\mathrm{C_-}$ and $\mathrm{C_+}$ of an output variable O, the output distribution function θ associated with O is the result of applying an accumulative function S, as shown in (11).

$$\theta(\mathrm{x}) = \mathrm{S}(\mathrm{C_-}, \mathrm{C_+}) \tag{11}$$

The approach used in this paper uses the maximum accumulative function. Consider that both $\mathrm{FDC_-}$ and $\mathrm{FDC_+}$ have $\mathrm{RS_-}$ and $\mathrm{RS_+}$ rule strengths respectively. The resulting output distribution function for each output decision Z (θ_Z) is shown in (12).

$$\theta_{\mathrm{Z}}(x) = \begin{cases} RS_-, & \text{if } \mathrm{x} = 0 \\ RS_+, & \text{if } \mathrm{x} = 1 \\ 0, & \text{if } \mathrm{x} \neq 0 \land \mathrm{x} \neq 1 \end{cases} \tag{12}$$

Therefore, all output distribution functions are the combination of two singleton functions, one for the $\mathrm{FDC_-}$ on the x value 0 and the other for the $\mathrm{FDC_+}$ on the x value 1.

A defuzzification method can then be applied to each output distribution function generated this way for each output variable. To show why the selected x values for each singleton were chosen, the commonly used center of gravity for singletons (COGS) defuzzification technique will be applied to θ_Z. The general COGS formula for a given output distribution function

θ_Z is given in (13).

$$\mathrm{COGS}(\theta_Z) = \frac{\sum_x x * \theta_Z(x)}{\sum_x \theta_Z(x)} \tag{13}$$

Note that if the output distribution function θ_Z is not generated from singleton membership functions then the formula for the center of gravity is a division of two primitives. Fortunately, since the proposed output distribution functions are always the combination of two singletons, which are also always defined on the x values 0 and 1, the formula (14) follows:

$$\mathrm{COGS}(\theta_Z) = \frac{0 * \theta_Z(0) + 1 * \theta_Z(1)}{\theta_Z(0) + \theta_Z(1)} = \frac{\theta_Z(1)}{\theta_Z(0) + \theta_Z(1)} \tag{14}$$

As it can be seen, the COGS formula was simplified to a simple fraction of the rule strength of the FDC_+ to the sum of the rule strengths of the FDC_- and FDC_+, reducing its complexity.

The application of the COGS defuzzification method to each output distribution function results in a crisp output value, which can used to arrive at a final decision. The simplest way to achieve this is to set a fixed threshold, such as 0.5, and if the crisp output is lower than the threshold then the decision is negative (*deny*, in the access control context), otherwise, it is positive (*grant*, in the access control context). The threshold value can be increased or decreased to fine-tune the system as required by an expert. Any other method to arrive at a decision is valid, such as the maximum method which takes the x value that maximizes the output distribution function and depends only on the use case.

E. BDFIS Analysis

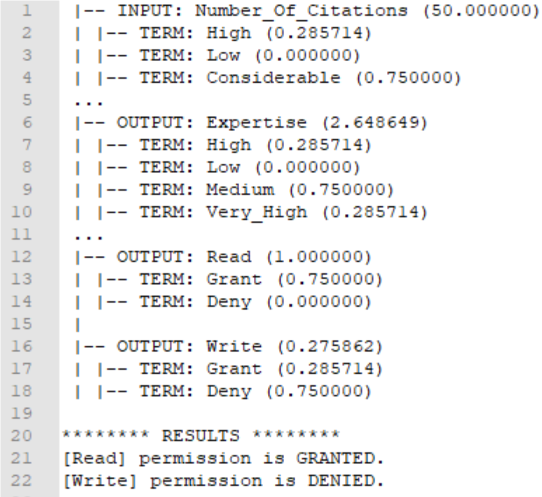
A modified Mamdani-type FIS called BDFIS has been proposed and detailed in this paper. However, upon further analysis, it was found that the consequence determination lacks some of its former expressibility. The reason behind this comes from the fact that the output linguistic terms and distribution functions are now fixed, i.e. FDCs and singleton functions respectively, while these functions could be defined freely in a standard FIS. This forces the rule strengths for the FDCs to be calculated directly from the input linguistic terms, where abstract concepts such as *Expertise* could be defined in a standard FIS instead. Thus, the ability to define abstract concepts is hindered to some degree.

While this fact may not impact use cases with simpler access control policies, it can impact the interpretability of the defined rules in others. Consider a rule that defines the vague concept of *Expertise*, such as “**if** *Number_of_Publications* is *High* **then** *Expertise* is *High*". If someone is newly hired to manage a system that uses this rule, it is clear what it is expressing: the expertise of the subject.

As is, the BDFIS would need the input variables to be mapped directly to the output decision variables, meaning that the *Expertise* vague concept cannot be explicitly defined or used in the mapping to the FDCs. Thus, a new security expert would need more detailed external documentation to understand what the rules are meant to represent to master the system, to write new rules, etc. However, it is possible to add a second layer of rules and intermediate variables to allow this (or more for additional abstraction). The input linguistic terms are mapped to these intermediate variables, which can include vague concepts like *Expertise*, and then these variables are mapped to the FDCs. This approach allows for rules to remain easily interpretable by humans at the cost of some processing.

IV. Proof of Concept

In this section, a proof of concept of the BDFIS will be shown. The prototype of the BDFIS used for this proof of concept uses JFuzzyLogic[30][31] and is available at github.com/Regateiro/FuzzyAC/tree/master/java/BDFIS. The *academic.fcl* file defines a BDFIS that makes access control decisions based on the expertise of a subject, using the Fuzzy Control Language (FCL)[32]. The BDFIS also comprises of two blocks of variables and rules, the first calculates the degree of expertise of the subject and the second the access control permissions. Fig. (2) shows the output of the proof of concept BDFIS implementation using the provided FCL file.



```
 1   |-- INPUT: Number_Of_Citations (50.000000)
 2   | |-- TERM: High (0.285714)
 3   | |-- TERM: Low (0.000000)
 4   | |-- TERM: Considerable (0.750000)
 5   ...
 6   |-- OUTPUT: Expertise (2.648649)
 7   | |-- TERM: High (0.285714)
 8   | |-- TERM: Low (0.000000)
 9   | |-- TERM: Medium (0.750000)
10   | |-- TERM: Very_High (0.285714)
11   ...
12   |-- OUTPUT: Read (1.000000)
13   | |-- TERM: Grant (0.750000)
14   | |-- TERM: Deny (0.000000)
15   |
16   |-- OUTPUT: Write (0.275862)
17   | |-- TERM: Grant (0.285714)
18   | |-- TERM: Deny (0.750000)
19
20  ******** RESULTS ********
21  [Read] permission is GRANTED.
22  [Write] permission is DENIED.
```

Figure 2. Sample BDFIS implementation output.

The defined BDFIS takes two input variables: the *number of publications* (omitted); and the *number of citations* (lines 1-4). These are used to determine the *Expertise* of the subject in the first set of rules (lines 6-10). The *Expertise* is then applied in the second set of rules to calculate the FDCs for a *Read* and *Write* permissions (lines 12-18). These are necessary to calculate whether the subject is granted each permission, by checking if the final defuzzified value is greater than 0.5 (grant) or not (deny). This value can be modified by an expert to require subjects to have lower or higher membership degrees to be granted access.

The numbers to the right of the input and output variables denote the crisp value associated with them (either provided as input or calculated from defuzzification). The number to the right of the terms denote their associated membership degrees. These are calculated from piecewise linear membership functions and rules defined by an expert in the *academic.fcl* file using the steps and methods explained in this paper. The BDFIS implementation automatically passes the *Expertise* value calculated by the first set of rules to the second set of rules as an input. Finally, since the *Read* permission has the output value 1.0 > 0.5, it is granted, and the *Write* permission with the output value 0.275862 < 0.5 is denied.

V. Discussion

In this paper, a FIS that can make binary access control decisions was proposed. Through the analysis made to each processing step, several changes were introduced that allowed a Mamdani-type FIS to specialize in the output of binary decisions. Furthermore, some proposed modifications optimized the output generation process, making this type of systems to have the potential for deployment in access control scenarios that deal with vague knowledge. Given the broad spectrum of areas where FIS are used, it shows that they are useful if correctly defined. By creating the BDFIS as closely as possible to a standard FIS, it is expected that it can be just as effective in access control scenarios.

One might question the appropriateness of applying fuzzy set theory to model access control since its inherent vagueness prevents an expert from easily knowing if a subject can access the protected data or not. Furthermore, auditing such a system is not easy given that new subjects may request access at any time.

Due to the vague nature of the fuzzy access control rules, the exact ranges of input values that grant access to the data is not clear. Thus, the possibility that an unexpected combination of input values granting access to the data may exist. It is important to note, however, that such a fuzzy access control model would still be deterministic. Furthermore, since vague conditions are applicable to a range of input values (with varying membership degrees), fewer rules are needed when compared to classic models, which generally requires each combination of input values to be written down as a separate rule. Nonetheless, fuzzy-based access control models are not easily accepted to manage sensitive data (e.g. hospital patient data), understandably due to the issues related to its inherent vagueness.

For future work, it is intended to build a system that can audit a BDFIS for correctness, an important feature for any access control system to have, and the calibration of the threshold values. Since the subject parameters used as input values are fuzzified, it is hard to determine from the fuzzy rules if they have access to a resource or not. Thus, being able to determine which input values grant or deny access is an important step towards making BDFIS a viable part of an access control system.

References

[1] R. Shirey, "Internet Security Glossary, Version 2," Aug. 2007.

[2] D. Ferraiolo and D. Kuhn, "Role-based access controls," *15th Natl. Comput. Secur. Conf.*, 1992, pp. 554–563.

[3] L. a. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, 1965, pp. 338–353.

[4] H. Singh *et al.*, "Real-Life Applications of Fuzzy Logic," *Adv. Fuzzy Syst.*, vol. 2013, 2013, pp. 1–3.

[5] N. H. Phuong and V. Kreinovich, "Fuzzy logic and its applications in medicine," *Int. J. Med. Inform.*, vol. 62, no. 2–3, Jul. 2001, pp. 165–173.

[6] M. A. Ghahazi, M. H. Fazel Zarandi, M. H. Harirchian, and S. R. Damirchi-Darasi, "Fuzzy rule based expert system for diagnosis of multiple sclerosis," in *2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, 2014, pp. 1–5.

[7] S. SushilSikchi, S. Sikchi, and A. M. S., "Fuzzy Expert Systems (FES) for Medical Diagnosis," *Int. J. Comput. Appl.*, vol. 63, no. 11, Feb. 2013, pp. 7–16.

[8] S. Berenjian, M. Shajari, N. Farshid, and M. Hatamian, "Intelligent Automated Intrusion Response System based on fuzzy decision making and risk assessment," in *2016 IEEE 8th International Conference on Intelligent Systems (IS)*, 2016, pp. 709–714.

[9] S. Al Amro, F. Chiclana, and D. A. Elizondo, "Application of Fuzzy Logic in Computer Security and Forensics," in *Studies in Computational Intelligence*, vol. 394, 2012, pp. 35–49.

[10] J. Li, Y. Bai, and N. Zaman, "A Fuzzy Modeling Approach for Risk-Based Access Control in eHealth Cloud," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013, pp. 17–23.

[11] P. N. Mahalle, P. A. Thakre, N. R. Prasad, and R. Prasad, "A fuzzy approach to trust based access control in internet of things," in *Wireless VITAE 2013*, 2013, pp. 1–5.

[12] H. Takabi, M. Amini, and R. Jalili, "Enhancing Role-Based Access Control Model through Fuzzy Relations," in *Third International Symposium on Information Assurance and Security*, 2007, no. 500, pp. 131–136.

[13] M. Botha and R. von Solms, "Utilising fuzzy logic and trend analysis for effective intrusion detection," *Comput. Secur.*, vol. 22, no. 5, 2003, pp. 423–434.

[14] V. C. V. Hu, D. F. Ferraiolo, and D. R. Kuhn, "Assessment of access control systems," *Nistir 7316*, 2006, p. 60.

[15] V. Henn, "Fuzzy route choice model for traffic assignment," *Fuzzy Sets Syst.*, vol. 116, no. 1, Nov. 2000, pp. 77–101.

[16] S. N. Shiaeles, V. Katos, A. S. Karakos, and B. K. Papadopoulos, "Real time DDoS detection using fuzzy estimators," *Comput. Secur.*, vol. 31, no. 6, 2012, pp. 782–790.

[17] J. Luo and E. Lan, "Fuzzy Logic Controllers for Aircraft Flight Control," in *Fuzzy Logic and Intelligent Systems*, vol. 3, Dordrecht: Springer Netherlands, 1995, pp. 85–124.

[18] S. Othman and E. Schneider, "Decision making using fuzzy logic for stock trading," in *2010 International Symposium on Information Technology*, 2010, pp. 880–884.

[19] Ying Bai, H. Zhuang, and D. Wang, *Advanced Fuzzy Logic Technologies in Industrial Applications*. London: Springer London, 2006.

[20] W. Liu and H. Liao, "A Bibliometric Analysis of Fuzzy Decision Research During 1970–2015," *Int. J. Fuzzy Syst.*, vol. 19, no. 1, Feb. 2017, pp. 1–14.

[21] Y. Wardhana, B. Hardian, G. Guarddin, and H. Rasyidi, "Context aware door access control on private room using fuzzy logic: Case study of smart home," in *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2013, pp. 155–159.

[22] A. C. F. Guimarães and C. M. F. Lapa, "Fuzzy inference to risk assessment on nuclear engineering systems," *Appl. Soft Comput.*, vol. 7, no. 1, Jan. 2007, pp. 17–28.

[23] C. Martínez-García, G. Navarro-Arribas, and J. Borrell, "Fuzzy Role-Based Access Control," *Inf. Process. Lett.*, vol. 111, no. 10, 2011, pp. 483–487.

[24] F. Qian, S. Sen, and O. Spatscheck, "[JJ]Characterizing resource usage for mobile web browsing," *MobiSys '14*, 2014, pp. 218–231.

[25] R. Werneck, J. Setubal, and A. da Conceicão, "(old) Finding minimum congestion spanning trees," *J. Exp. Algorithmics*, vol. 5, 2000, p. 11.

[26] M. Conti, R. Di Pietro, L. V Mancini, and A. Mei, "(old) Distributed data source verification in wireless sensor networks," *Inf. Fusion*, vol. 10, no. 4, 2009, pp. 342–353.

[27] M. Sugeno, "Industrial applications of fuzzy control," *Elsevier Sci. Pub. Co.*, 1985.

[28] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man. Mach. Stud.*, vol. 7, no. 1, 1975, pp. 1–13.

[29] A. Kaur and A. Kaur, "Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System," *Int. J. Soft Comput. Eng.*, vol. 2, no. 2, 2012, pp. 323–325.

[30] P. Cingolani and J. Alcalá-Fdez, "JFuzzyLogic: A robust and flexible Fuzzy-Logic inference system language implementation," in *IEEE International Conference on Fuzzy Systems*, 2012.

[31] P. Cingolani and J. Alcala-Fdez, "jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming," *Int. J. Comp. Int. Syst.*, vol. 6, no. sup1, 2013, pp. 61–75.

[32] IEC, "Fuzzy Control Programming (IEC 1131-7 CD1)." pp. 1–53, 1997.