A Mathematical Model to Calculate Data Sensitivity in Hadoop Platform Using the Analytic Hierarchy Process Method

Hafsa Ait idar, Hicham Belhadaoui and Reda Filali

***Abstract*—The quick growth of smart innovations has allowed providing vast amounts of data across the web. As the amount of data continues to grow up, companies need to be able to process it rapidly, efficiently, and securely. Since sensitive information is often included among the company's data, safeguarding this sensitive data has always been a challenging requirement for many organizations. Therefore, the proposed Dynamic Data Sensitivity Access Control (D2SAC) framework in [1] focuses on protecting sensitive data in Hadoop Platform. To continue improving our framework, in this paper, we concentrate on the Sensitivity Estimator Module (SEM) which is responsible for calculating the data sensitivity value. For this reason, we propose a mathematical model based on the Analytic Hierarchy Process (AHP) method in order to derive the priority vector of data attributes then the empirical average is used to calculate the final data sensitivity value a way to provide an automated calculation of sensitivity without any intervention of the Data Owner.**

***Index Terms*—Analytic Hierarchy Process, Sensitive Data, HDFS, Hadoop, Big Data.**

I. Introduction

WE are surrounded by data. Data from social media, sensors, organization applications, mobile devices, and several more sources are part of our day-to-day basis [2]. The emergence of Big Data technologies is due to the novel information revolution, prompting a great change in the way that companies store, treat and provide their services [3]. Thus, for many businesses, Big Data is becoming ubiquitous and vital to enhance their overall activity and raise their revenues [3][4].

Thanks to Big Data, companies analyze better collected data across the web in order to understand perfectly their customer's needs [2]. Sensitive data represents a significant part of the collected data. It refers to data that must be kept out of the reach of unauthorized access in order to protect individuals and organizations information [5]. Sensitive information includes all type of data that must be protected against illegitimate admissions, such as:

- Personal Information, also known as Personally Identifiable Information (PII) or Sensitive Personal Information (SPI), is data that can be used to identify a single individual and, if revealed, could lead to severe harm to that individual. This information includes unique identifiers, biometric data, medical information, Personally Identifiable Financial Information (PIFI), etc.
- Sensitive Business Information includes any data that may present a danger to the company if discovered by undesired persons. Such information includes financial data, trade secrets, customer and supplier information, and so on.

Hadoop [6] has emerged as an important open platform to deal with the main Big Data challenges. It is mainly based on distributed storage and computing, parallel processing, cost-efficient and high availability of data [7]. Hadoop includes various components: core components (Hadoop Distributed File System (HDFS), MapReduce and YARN) along with numerous tools (HBase, Hive, Sqoop, Zookeeper, Ambari, etc.) designed to allow users to leverage the potential of data [8].

Hadoop is increasingly deployed to manage a high amount of data. Data residing within Hadoop may contain sensitive information of financial or insurance companies, healthcare or government services. Such companies, which are handling large amounts of sensitive data, are facing several risks due to various types of security vulnerabilities, including data-theft, data leakage, data breach [9][10] .

When security violations occurred within an organization, the consequences can be an extreme hardship for this latter. The costs associated with data violations such as data recovery fees and financial penalties can differ depending on the significance of the breach. Besides, the worst consequence for any organization is customer dissatisfaction and reputation loss, which can cause great longterm damage to the affected organization. However, to avoid all these severe effects, companies dealing with sensitive data are under more pressure to keep data contained in Hadoop protected.

Regardless of different methods and approaches provided to improve security and privacy concerns in big data [9] [10]. Sensitive data is prone to many threats that must be solved efficiently and strongly. To address these security and privacy challenges, we proposed in our previous work the Dynamic Data Sensitivity Access Control (D2SAC) framework with the aim to protect sensitive data stored in Hadoop [1]. In the continuity of our previous work, our primary goal in this paper is to calculate the data sensitivity in an automated way without further intervention from the data owner. To this end, we select the Analytic Hierarchy Process (AHP) method to provide the weights of each data attributes, named also factors or criteria, which are represented as priority vectors then the empirical average is used to derive the final data

Manuscript received July 11, 2019; revised January 21, 2020.

H. Ait idar is with the RITM Laboratory, National High School for Electricity and Mechanics, Hassan II University, Casablanca, Morocco. E-mail: (hafsa.aitidar93@gmail.com).

H. Belhadaoui is with the RITM Laboratory, National High School for Electricity and Mechanics, Hassan II University, Casablanca, Morocco. E-mail: (belhadaoui_hicham@yahoo.fr).

R. Filali is with the RITM Laboratory, National High School for Electricity and Mechanics, Hassan II University, Casablanca, Morocco. E-mail: (filalihilalireda@gmail.com).

sensitivity value in order to keep sensitive data protected according to its degree of sensitivity.

The rest of this paper is organized as follows: Section 2 provides an overview of Multi-Criteria Decision-Making Methods. Section 3 presents the previous D2SAC framework. In Section 4, the proposed mathematical model is explained in detail. Section 5 contains an example of calculating the sensitivity of the data. A discussion of our work is included in Section 6. Section 7 concludes the paper and offers a future research direction.

II. MULTI-CRITERIA DECISION-MAKING METHODS

The Multi-Criteria Decision-Making (MCDM) is a branch of the area of Operations Research (OR) that is interested in structuring problems including multiple criteria and solving complicated decisions in order to support decision-makers facing such issues [11] [12]. The MCDM approach provides an efficient technique to facilitate the decision-making process by selecting the optimum alternative among an important set of alternatives involving numerous criteria [13]. The MCDM is widely applied to deal with various problems, like service selection, investment decision, transportation planning field, product evaluation, staff assessment, and other personal issues [12].

The Multi-Criteria Decision-Making approach can be divided into two categories: Multi-Attribute Decision-Making (MADM) and Multi-Objective Decision-Making (MODM) [14]. In MADM, the selection of the best alternative is based on evaluating the predetermined alternatives which are described with different attributes [12], while MODM involves analysis that relies on multiple and competitive objectives [13]. Figure 1 presents a taxonomy of the MCDM methods.

There are different methods of MCDM approach, such as Analytical Hierarchy Process (AHP) [15], Analytic Network Process (ANP) [16], Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [17], Elimination and Choice Expressing Reality (ELECTRE) [18], Preference Ranking Organization METHod of Enrichment Evaluations (PROMETHEE) [19], Simple Additive Weighting (SAW) [20], and so on. Each method has its own characteristics, strong and weak points [12]. Therefore, selecting the appropriate method requires the knowledge of the addressed problem besides the context in which the problem is being fixed. Typically, the common working principle of any MCDM method is the same [11][13]:

1) Selection of criteria,
2) Selection of alternatives,
3) Selection of weighing or outranking methods,
4) Selection of aggregation method.

Due to paper size constraints, we cannot afford to go into the detail of each method. In this work, we are interested in the Analytic Hierarchy Process (AHP).

A. AHP Method

The Analytic Hierarchy Process (AHP) is a method, proposed by Satty [15], which addresses the decision-making problems [21]. The AHP provides an effective technique to structure and analyze complex decisions and problems,

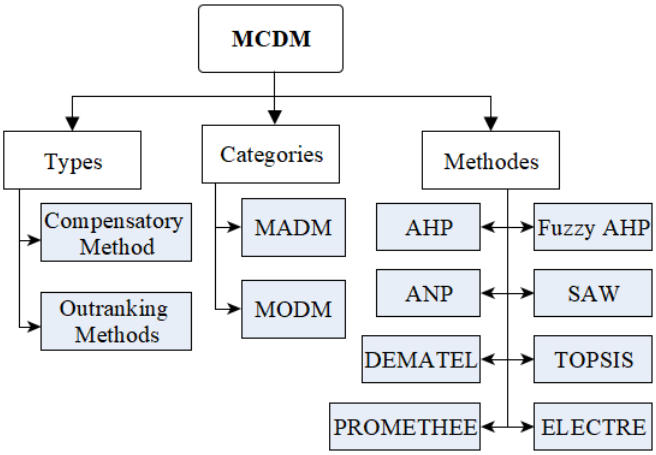

Fig. 1. Taxonomy of MCDM Methods.

including problems with factors that are difficult to quantify and problems with a high number of decision criteria [22].

The AHP method is well known for its natural simplicity [23]. It requires a pairwise comparison of factors in order to determine the weight of each one, considering both tangible (e.g., price) and intangible (e.g., comfort) factors [23][22].

The AHP decomposes the studied problem as a hierarchy to identify the priority vectors in order to select the best alternative, which corresponds to the highest rank among the available alternatives [22]. The AHP method was widely applied in various decision situations in areas such as business, government, education, industry, as well as healthcare [11][21]. The AHP involves the following steps [23]:

Step 1: Build a hierarchy model for the decision: decompose the decision problem into a hierarchy of objective, criteria, and alternatives as represented in Figure 2.

Step 2: Determine weights or priorities for the criteria: the experts make their judgments based on the importance of criteria, which are compared pairwise, then the consistency of the judgments is checked.

Step 3: Determine local priorities for alternatives with respect to each criterion (a similar process of step 2 is followed).

Step 4: Calculate the overall priority of each alternative using the local priority of this alternative and the weight of each criterion. This step is repeated for each of the evaluated alternatives in order to select the optimum one.

Step 5: Make sensitivity analysis to know if changes in the weights of criteria might affect the obtained results.

Step 6: Take the final decision based on the overall priorities and sensitivity analysis.

III. PROPOSED DYNAMIC DATA SENSITIVITY ACCESS CONTROL FRAMEWORK

Sensitive data in Hadoop may face concerns related to privacy and unauthorized access [7] [24] [25]. These concerns can get more complicated when the number of businesses and individuals increases [7]. Numerous papers [8] [26] [27] [28] discuss the main Hadoop security issues in order to build suitable solutions to protect organizations' data. In this regard, we proposed in our previous work the Dynamic Data Sensitivity Access Control (D2SAC) framework. The primary aim of this framework is to calculate the sensitivity

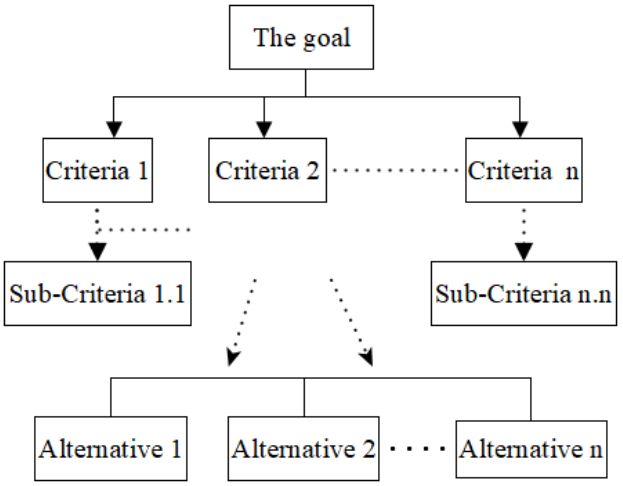

Fig. 2. AHP hierarchy model.

value of each data created in HDFS and then control access to this data as long as it resides in HDFS. Therefore, two scenarios are included within our framework, as shown in Figure 3.

Scenario 1: when the user intends to create a data in HDFS (create request). In this case, the sensitivity value of this data will be calculated using the proposed mathematical model in the next section.

Scenario 2: when the user intends to access a data that is already stored in HDFS (access request). In this case, depending on the sensitivity value of the demanded data, the user can be allowed to access this data or not.

In addition, the D2SAC consists of four main components [1] to ensure its proper functioning. These components are:

- *The Access Enforcement Module (AEM)* receives the user's request (create or access request) and returns the decision about this request (entitled to access or not).
- *The Information Extractor Module (IEM)* receives the metadata information related to the created data, and then extracts the sub-criteria used in calculating the sensitivity value of this data.
- *The Sensitivity Estimator Module (SEM)* is responsible for calculating the sensitivity value of data using the proposed mathematical model that will be explained in detail in the next section.
- *The Metadata Generator Module (MGM)* accesses the metadata files of the NameNode (FSImage and EditLog) in order to retrieve the information related to the newly created data, then it stores this metadata information in the MetaDatabase.

IV. PROPOSED MATHEMATICAL MODEL

The procedure of calculating the sensitivity value of data is related to the Sensitivity Estimator Module (SEM) as shown in Figure 4. The SEM calculates the sensitivity value of each data created in HDFS using the proposed mathematical model that is based on the second step of the AHP method [23] in order to determine the priority vector of criteria, and then, the empirical average is used to derive the sensitivity of the data. Furthermore, the proposed model includes two major phases: the pre-processing phase and the real-time phase. Each phase involves several steps to be followed.

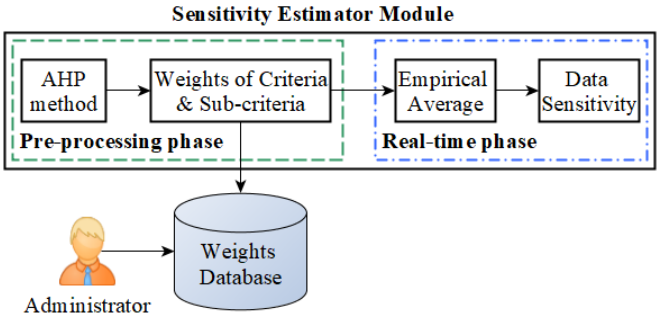

Fig. 4. Proposed SEM.

A. Pre-processing phase

In this phase, we calculate the weights of criteria and sub-criteria using the five steps described in this section. It should be noted that all these steps are executed only once at the D2SAC implementation time. Then, we save these weights in the Weights database. However, considering the importance of this database, the administrator is the only one capable of editing this database, if necessary.

Step1: Identify the major criteria (factors) and sub-criteria (sub-factors) to be taken into account when using our model. Each criterion can be divided into several sub-criteria depending on the studied context. Let the:
$(C_1, C_2, ...C_i...C_n)$ be the criteria to be considered where i $\in$ n and n is the number of criteria.
$(C_{11}, C_{12}, ...C_{21}...C_{nj})$ be the sub-criteria where j is the number of sub-criteria of criterion n.

Step 2: Construct the criteria comparison matrix based on the numerical scale developed by Saaty [15] (Table I) to formulate the expert's judgments, which are arranged into a square matrix as shown in Table II.

TABLE I
SAATY'S PAIRWISE COMPARISON SCALE

Linguistic scale	Numeric Value
Equally important	1
Moderately important	3
Strongly important	5
Very Strongly important	7
Extremely important	9
Intermediate values between two adjacent judgments	2,4,6,8

TABLE II
PAIRWISE COMPARISON MATRIX

C_i \ C_j	C_1	C_2	...	...	C_n
C_1	a_{11}	a_{12}	...	...	a_{1n}
C_2	a_{21}	a_{22}	...	...	a_{2n}
...	...	...	...	...	...
...	...	...	...	...	...
C_n	a_{n1}	a_{n2}	...	...	a_{nn}

Where:

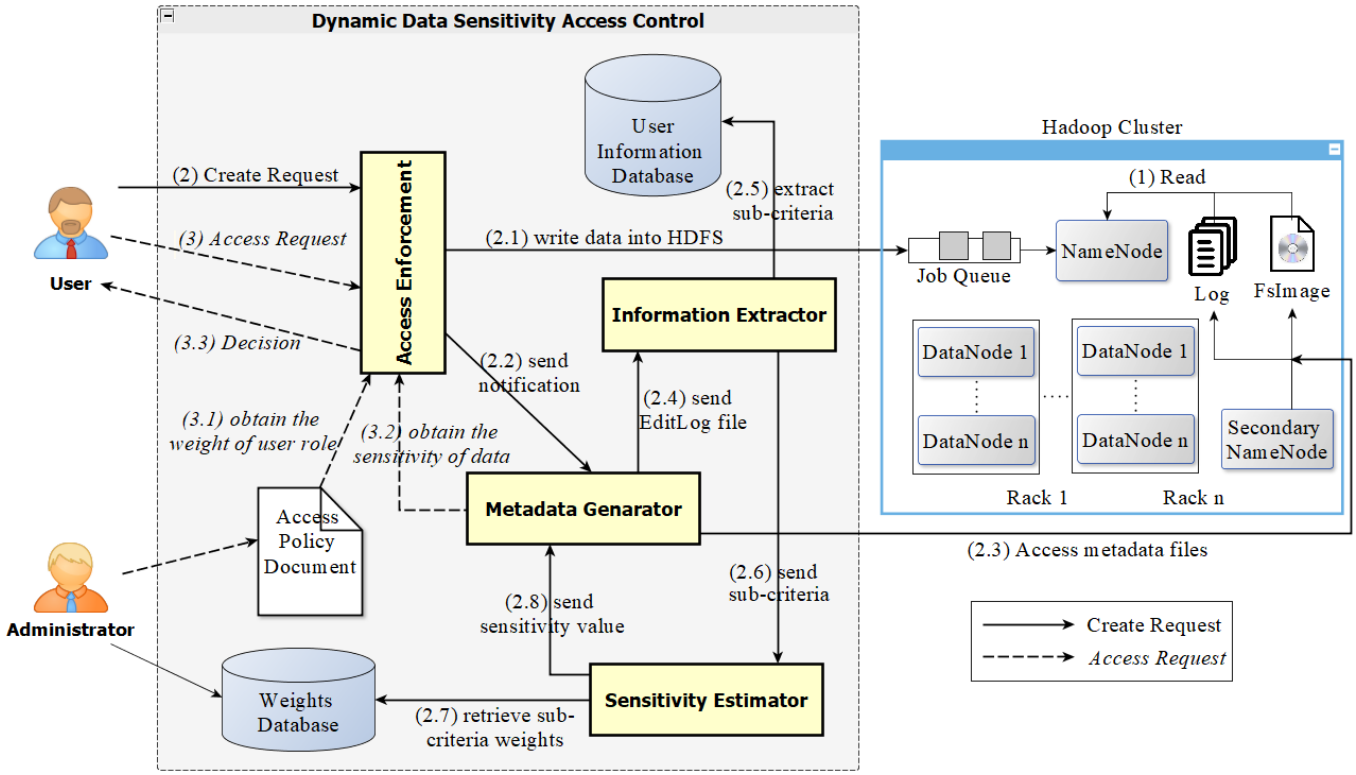

Fig. 3. Proposed D2SAC framework.

- a_{ij} represents how much criteria i is more important than criteria j.
- i , j = 1, 2, ..., n and n is the number of criteria.
- $a_{ij} = \frac{1}{a_{ji}}$ for $a_{ij} \neq 0$
- $a_{ij} = 1$ for i = j

Step 3: Determine the normalized matrix based on the pairwise comparison matrix (Table II) using Equation (1) as shown in Table III.

$$v_{ij} = \frac{a_{ij}}{(\sum_{i=1}^{n} a_{ij})} \tag{1}$$

for i,j = 1, 2, . . . n

TABLE III
NORMALIZED MATRIX

C_i \ C_j	C_1	C_2	...	...	C_n
C_1	v_{11}	v_{12}	...	...	v_{1n}
C_2	v_{21}	v_{22}	...	...	v_{2n}
...	...	...	...	...	...
...	...	...	...	...	...
C_n	v_{n1}	v_{n2}	...	...	v_{nn}

Step 4: Derive the weights or priorities for criteria based on the normalized matrix by simply using Equation (2). The priority vector of criteria is represented by Equation (3).

$$w_i = \frac{(\sum_{j=1}^{n} v_{ij})}{n} \tag{2}$$

for i = 1, 2, . . . n

$$W = \begin{pmatrix} w_1 \\ w_2 \\ \cdots \\ w_i \\ \cdots \\ w_n \end{pmatrix} \tag{3}$$

It should be mentioned that a similar process is followed in order to derive the weights or priority vector for sub-criteria (w_{ij}).

Step 5: Check the consistency of the judgments matrix (Table II) through the calculation of the Consistency Ratio (CR) that is defined by Equation (4).

$$CR = \frac{CI}{RI} \tag{4}$$

Where: CI is the Consistency Index calculated by Equation (5) based on the maximum eigenvalue λ_{max} and RI is the Random Consistency Index provided by Saaty [15] for matrices of different sizes (n), as shown in Table IV.

$$CI = \frac{(\lambda_{max} - n)}{(n-1)} \tag{5}$$

TABLE IV
RANDOM CONSISTENCY INDEX VALUES

n	1	2	3	4	5	6	7	8	9	...
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	...

According to Saaty [15] the judgment matrix is considered to be consistent if the corresponding CR is equal to or less than 10%. If CR is greater than 10%, it is required

to reconsider the judgments to correct the cause of the inconsistency in order to avoid conflicting results.

Figure 5 shows the code for calculating the weight of each criteria and sub-criteria used in our model.

- The first function *initializeMatrix (int nbCriteria)* creates a matrix with the specified number of criteria and initializes it with 1.
- The second function *constructComparisonMatrix(int critA, int critB, double comparisonScale*) establishes the criteria comparison matrix based on the numerical scale developed by Saaty.
- The last function *obtainWeights()* illustrates the process of determining the weights of criteria.

```
public void initializeMatrix(int nbCriteria){
   criteriaMatrix = new
   Array2DRowRealMatrix(nbCriteria,nbCriteria);
   for (int i = 0; i < nbCriteria; i++) {
      for (int j = 0; j < nbCriteria; j++) {
          criteriaMatrix.setEntry(i, j, 1.0);
      }
   }
}
public void constructComparisonMatrix(int
    critA, int critB, double comparisonScale)
          throws Exception {
   if (criteriaMatrix == null)
      throw new Exception("The Criteria Matrix
      must be initialitzed first.");
   if (critA != critB) {
       criteriaMatrix.setEntry(critA, critB,
           comparisonScale);
       criteriaMatrix.setEntry(critB, critA,
           1.0/comparisonScale);
   }
 }
public double[] obtainWeights() throws
    Exception {
   if (criteriaMatrix == null)
      throw new Exception("The Criteria Matrix
      must be initialized and constructed.");
   else {
      ed = new
          EigenDecomposition(criteriaMatrix);
      eIdx = 0;
      for(int i = 0; i <
          ed.getRealEigenvalues().length;
          i++) {
            eIdx = (ed.getRealEigenvalue(i) >
                  ed.getRealEigenvalue(eIdx))
                      ? i : eIdx;
      }
      double sum = 0.0;
      RealVector vector =
          ed.getEigenvector(eIdx);
      for (double j : vector.toArray()) {
            sum += j;
      }
      double[] W = new
          double[vector.getDimension()];
      for (int i = 0; i <
          vector.getDimension(); i++) {
             W[i] = vector.getEntry(i) / sum;
      }
      return W;
    }
}
```

Fig. 5. Part of code to get the weights of criteria and sub-criteria.

Figure 6 gives the code for checking the Consistency Index, and the Consistency Ratio that should be less than 10% to continue the process of calculating the sensitivity.

```
public double getConsistencyIndex() {
   return
       (ed.getRealEigenvalue(eIdx)-(double)
       criteriaMatrix.getColumnDimension())/
       (double)(criteriaMatrix.
        getColumnDimension() - 1);
}
public double getConsistencyRatio() {
  if(criteriaMatrix.getColumnDimension()> 2){
      return (getConsistencyIndex() /
          RI[criteriaMatrix.
            getColumnDimension()]) * 100.0;
  }
  else return 0.0;
}
```

Fig. 6. Part of code to check the CI and CR.

B. Real-time phase

After calculating and saving the weight of each criterion (and sub-criteria) during the first phase. In this phase, once the user sends a request to create a data in HDFS, the following steps are used to calculate the sensitivity of this data.

Step 6: Calculate the global weight for the selected sub-criteria. Global sub-criteria weight is calculated by multiplying the weight of the sub-criteria with the weight of the criterion to which it belongs, as shown in Equation (6).

$$GW_{ij} = w_{ij} \times w_i \tag{6}$$

Where: j represents the selected sub-criteria which belongs to criterion i.

Step 7: Calculate the final data sensitivity value (S_d) by utilizing the empirical average as shown in Equation (7).

$$S_d = \frac{(\sum_{i=1}^{n} GW_{ij})}{n} \tag{7}$$

Where:

- n is the number of criteria.
- j represents the selected sub-criteria which belongs to criterion i.

Figure 7 shows the code for calculating the sensitivity of each data created in HDFS. The function *calculateSensitivity()* receives the identified sub-criteria, retrieves the weight of each sub-criteria(*double subCriteriaWeight*), and the weight of the criterion to which it belongs(*double criteriaWeight*) , then calculates the global weight for this sub-criteria (*double globalWeight*). Finally, the empirical average is used to calculate the final sensitivity of data.

Figure 8 summarizes the proposed mathematical model steps. In the pre-processing phase, we identify the criteria and sub-criteria that will be used in our model. We then calculate their weights and check the consistency ratio. During the real-time phase, when the D2SAC receives a request to create a data in HDFS, the SEM is invoked to calculate the sensitivity of this data.

```
public double calculateSensitivity(String
    role, String dep, String purpose, String
    time, String location) {
  String[] subCriteriaTab =
      {role,dep,purpose,time,location};
   double sommeOfGlobalWeight = 0.0;
   int nbrOfSubCriteria =
       subCriteriaTab.length;
   for(int i=0; i<nbrOfSubCriteria; i++) {
     SubCriteriaTable subCriteria = new
         SubCriteriaTable();
     subCriteria = SubCriteriaRepository.
                findByName(subCriteriaTab[i]);
     double subCriteriaWeight =
         subCriteria.getWeight();
     CriteriaTable criteriaID =
         subCriteria.getIDCriteria();
     CriteriaTable criteria = new
         CriteriaTable();
     criteria= CriteriaRepository.
            findById(criteriaID.getId());
     double criteriaWeight =
         criteria.getWeight();
     double globalWeight = subCriteriaWeight *
                           criteriaWeight;
     sommeOfGlobalWeight += globalWeight;
   }
  dataSensitivity = sommeOfGlobalWeight /
                    nbrOfSubCriteria;
  return dataSensitivity;
}
```

Fig. 7. Part of code to calculate the sensitivity of data.

V. EXAMPLE OF CALCULATING THE SENSITIVITY OF DATA

Financial institutions are among the most important sectors that handle a large quantity of sensitive data about their customers. These institutions are increasingly adopting advanced Big Data solutions to gain benefits provided by these solutions. Therefore, in this section we aim to calculate the sensitivity of data used in the financial sector in order to evaluate the proposed mathematical model.

A. *Factors used*

Several factors are involved in the calculation of data sensitivity value. Thus, it is important to identify the major factors that would have the greatest impact on the sensitivity value. Based on the analysis of the literature review and the support of the financial experts, the main factors were selected. Each expert provided us with a list of the most important criteria related to the sensitivity of financial data. Consequently, the common factors on the lists were retained, namely user role and department, login time and location, and data purpose. More precisely, we determine the criteria used in our mathematical model by addressing the following questions: who will create the data, when, where, and why?

User Role: In this work, a user's job is more significant than the user's identity. For this reason, users are classified according to their roles within the organization. Moreover, users will be assigned to the most appropriate role to accomplish their activities. According to experts'judgments, the role of the user who creates the data is extremely important to calculate the sensitivity of this data. For instance, we assume that data created by the project manager may be more sensitive than data created by the technical staff.

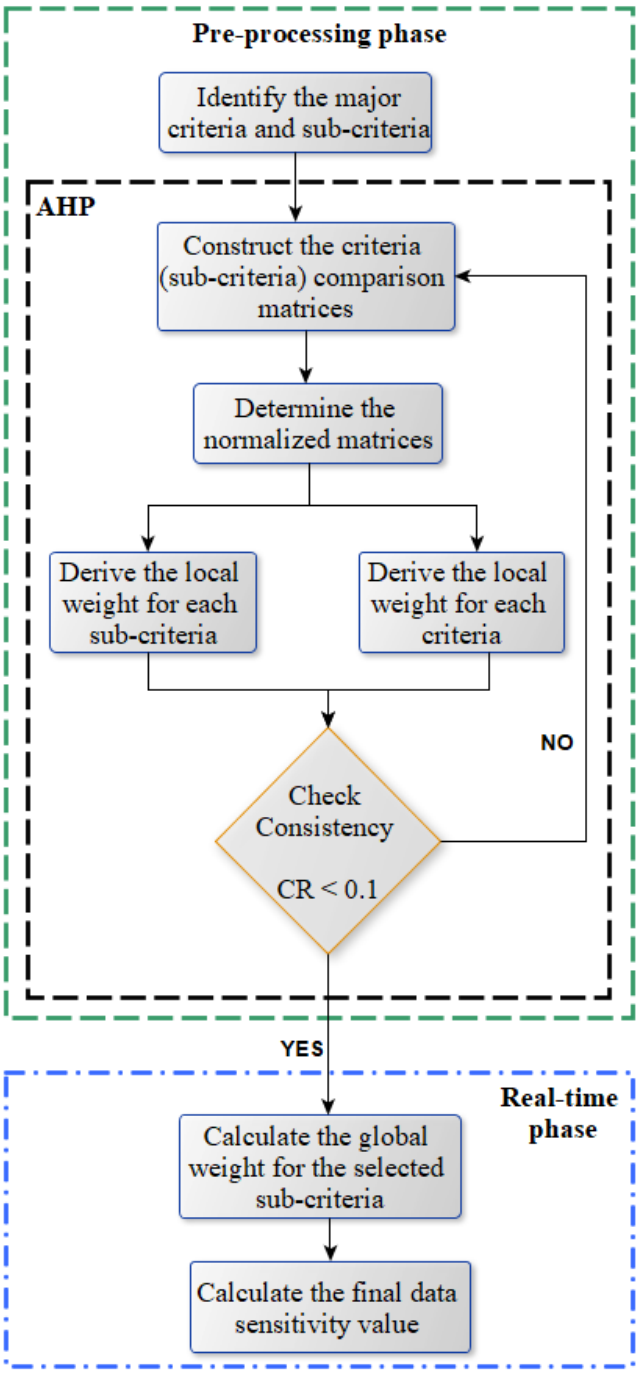

Fig. 8. Proposed mathematical model.

User department: An organization may consist of several departments that contribute to its overall missions and objectives. To fulfill these missions, each department contains a set of roles. Thus, identifying the department and the role of the user is extremely important to calculate the sensitivity of data that will be created in HDFS.

Login time: In the financial sector, the time of creating data is a crucial factor that must always be taken into consideration. For example, institutions dealing with highly sensitive data can allow creating sensitive data only at the workplace during working hours. Otherwise, the demand to create data is directly detected and reported to administrators.

Login location: Similarly, the login location should be taken into account to calculate data sensitivity. Financial companies can limit creating sensitive data by indicating a specific workplace, for example, authorizing the creation of sensitive data only from the Intranet, otherwise, the demand is rejected.

Data purpose: The purpose of creating a data is important in the calculation of sensitivity. The data created with the aim to improve decision-making is more sensitive than the data created to perform an internal report, as an example. Thus, the importance of considering this factor in order to keep sensitive data protected from any external violations.

B. Pre-processing phase

During the pre-processing phase, we calculate all weights of identified criteria and sub-criteria, and then, we save these weights in the Weights Database. The detailed explanation of this phase is as follows.

Step 1: The factors used to calculate the sensitivity value of data are previously defined. By applying all these factors to the financial context, we obtain the following Table V.

TABLE V
IDENTIFIED FACTORS AND SUB-FACTORS

Criteria	Sub-Criteria
C_1: User role	$C_{1.1}$: Chief Executive Officer $C_{1.2}$: Financial Analyst $C_{1.3}$: Security & Fraud Specialist $C_{1.4}$: Business Technology Specialist
C_2: User department	$C_{2.1}$: Administrative and Management $C_{2.2}$: Human Resource $C_{2.3}$: Finance $C_{2.4}$: IT Support $C_{2.5}$: Marketing
C_3: Data purpose	$C_{3.1}$: Decision Info $C_{3.2}$: Engineering Info $C_{3.3}$: Financial Info
C_4: Login time	$C_{4.1}$: Working Hours $C_{4.2}$: Off-Hours
C_5: Login location	$C_{5.1}$: Inside Company $C_{5.2}$: Outside Company

Steps 2 and 3: Table VI summarizes Steps 2 and 3 of the proposed model. More precisely, the experts provide us with their judgments relying on the numerical scale (Table I) to construct the pairwise comparison matrix. Then, the normalized matrix is determined using Equation (1).

Step 4: Using Equation (2), we obtain the priority vector of determined criteria as follows:

$$W = \begin{pmatrix} 38.3 \\ 29.1 \\ 6.1 \\ 15.1 \\ 11.4 \end{pmatrix}$$

To validate our calculation, we use the BPMSG AHP Priority Calculator [29] and we obtain the same priority vector of criteria, as shown in Figures 9 and 10 respectively. Consider the example of criterion "Role" that has a priority of 38.3% with an uncertainty of +12.1% and -12.1%. In addition, the diagram in Figure 10 presents in green the calculated weights of each criterion, in dark and light grey the possible plus and minus variations of priorities.

TABLE VI
WEIGHTS OF CRITERIA

$C_{(i)}$ \ $C_{(j)}$	C_1	C_2	C_3	C_4	C_5	Weights
C_1: Role	1	2	4	3	3	**0.383**
C_2: Department	1/2	1	4	3	3	**0.291**
C_3 Purpose	1/4	1/4	1	1/3	1/3	**0.061**
C_4: Time	1/3	1/3	3	1	2	**0.151**
C_5 Location	1/3	1/3	3	1/2	1	**0.114**
CI	0.057					
RI	1.12					
CR	**5%**					

Cat		Priority	Rank	(+)	(-)
1	User Role	38.3%	1	12.1%	12.1%
2	User Department	29.1%	2	9.5%	9.5%
3	Data Purpose	6.1%	5	2.1%	2.1%
4	Login Time	15.1%	3	4.8%	4.8%
5	Login Location	11.4%	4	3.9%	3.9%

Fig. 9. Priorities of criteria.

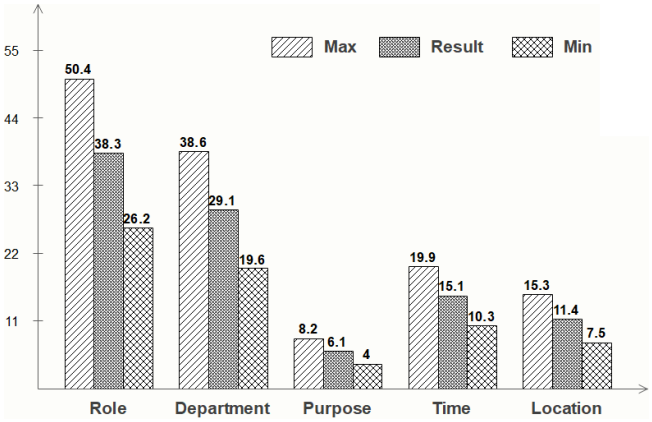

Fig. 10. Diagram of priorities.

We use the same steps to calculate the weight of each sub-criteria, Tables VII, VIII, IX, X and XI synthesize the process of deriving the priority vector of Role, Department, Purpose, Time and Location sub-criteria respectively.

Step 5: Once the judgment matrices have been entered, we check the consistency of these matrices using Equation (4). The CI, RI and CR corresponding to each matrix are given in Tables VI, VII, VIII and IX. We can notice that the

TABLE VII
PRIORITY VECTOR OF ROLE SUB-CRITERIA

$C_{1.j}$	$C_{1.1}$	$C_{1.2}$	$C_{1.3}$	$C_{1.4}$	Weights
$C_{1.1}$: Chief Executive Officer	1	3	2	4	**0.451**
$C_{1.2}$: Financial Analyst	1/3	1	1/3	3	**0.160**
$C_{1.3}$: Security & Fraud Specialist	1/2	3	1	3	**0.304**
$C_{1.4}$: Business Technology Specialist	1/4	1/3	1/3	1	**0.085**
CI	0.053				
RI	0.9				
CR	**5.8%**				

TABLE VIII
PRIORITY VECTOR OF DEPARTMENT SUB-CRITERIA

$C_{2.j}$	$C_{2.1}$	$C_{2.2}$	$C_{2.3}$	$C_{2.4}$	$C_{2.5}$	Weights
$C_{2.1}$: Administrative and Management	1	5	3	5	6	**0.494**
$C_{2.2}$: Human Resource	1/5	1	1/3	2	4	**0.134**
$C_{2.3}$: Finance	1/3	3	1	2	3	**0.215**
$C_{2.4}$: IT Support	1/5	1/2	1/2	1	4	**0.108**
$C_{2.5}$: Marketing	1/6	1/4	1/3	1/4	1	**0.049**
CI	0.088					
RI	1.12					
CR	**7.8%**					

TABLE IX
PRIORITY VECTOR OF PURPOSE SUB-CRITERIA

$C_{3.j}$	$C_{3.1}$	$C_{3.2}$	$C_{3.3}$	Weights
$C_{3.1}$: Decision Info	1	4	3	**0.614**
$C_{3.2}$: Engineering Info	1/4	1	1/3	**0.117**
$C_{3.3}$: Financial Info	1/3	3	1	**0.269**
CI	0.038			
RI	0.58			
CR	**6.5%**			

TABLE X
PRIORITY VECTOR OF TIME SUB-CRITERIA

$C_{4.j}$	$C_{4.1}$	$C_{4.2}$	Weights
$C_{4.1}$: Working Hours	1	7	**0.875**
$C_{4.2}$: Off-Hours	1/7	1	**0.125**
CR	**0%**		

Consistency Ratio of all matrices is less than 10%, meaning that our judgment matrices are adequately consistent so we can continue our calculation.

TABLE XI
PRIORITY VECTOR OF LOCATION SUB-CRITERIA

$C_{5.j}$	$C_{5.1}$	$C_{5.2}$	Weights
$C_{5.1}$: Inside Company	1	8	**0.889**
$C_{5.2}$: Outside Company	1/8	1	**0.111**
CR	**0%**		

In the case of judgment matrices with only two criteria to compare (n = 2), these matrices will always be consistent because of the corresponding CR will be equal to 0 [15]. The above Tables X and XI are given as an example.

C. Real-time phase

During this phase, the sensitivity of the data is calculated. Typically, the global weights of the retrieved sub-criteria are calculated, then the data sensitivity is obtained by averaging these weights. Figure 11 illustrates the procedure of calculating the sensitivity of a data created by Alice. In this example, we explain each step (1)–(14) as follows.

(1) To start up the Hadoop cluster, the NameNode needs to read the metadata information contained in FSImage and EditLog files.
(2) The user Alice inputs a request to create a file in HDFS.
  (2.1) Alice creates a file using the create() method of the DistributedFileSystem, which performs an RPC call to the NameNode to start creating a new file in the filesystem's namespace. The NameNode then makes several checks to ensure that this new file is not present in the file system and that Alice is allowed to create this file. If so, a new record for the file is created by the NameNode. Otherwise, the operation to create the file fails and the IOException is thrown to Alice.
  (2.2) Once a new record is created, the DistributedFileSystem returns the object FSDataOutputStream to start writing data into HDFS using the write() method.
(3) After writing the file into HDFS, the AEM sends a notification to the MGM indicating that a new file is created in the file system.
(4) Receiving the AEM's notification, the MGM accesses the metadata files, more precisely, the EditLog and FSImage files in order to obtain all the metadata information of the HDFS.
(5) The EditLog file records every action performed on the HDFS cluster after the most recent FSImage in order to have the current metadata information and the recent system status. For this reason, this file is submitted to the IEM.
(6) Once the EditLog is received, the IEM transforms this file into a readable format in order to extract the metadata information related to the data created by Alice.
(7) (7') From the readable file, we retrieve the user name (Alice-PC) and the sub-criteria that are directly extracted from this file. Namely, the login time (during working

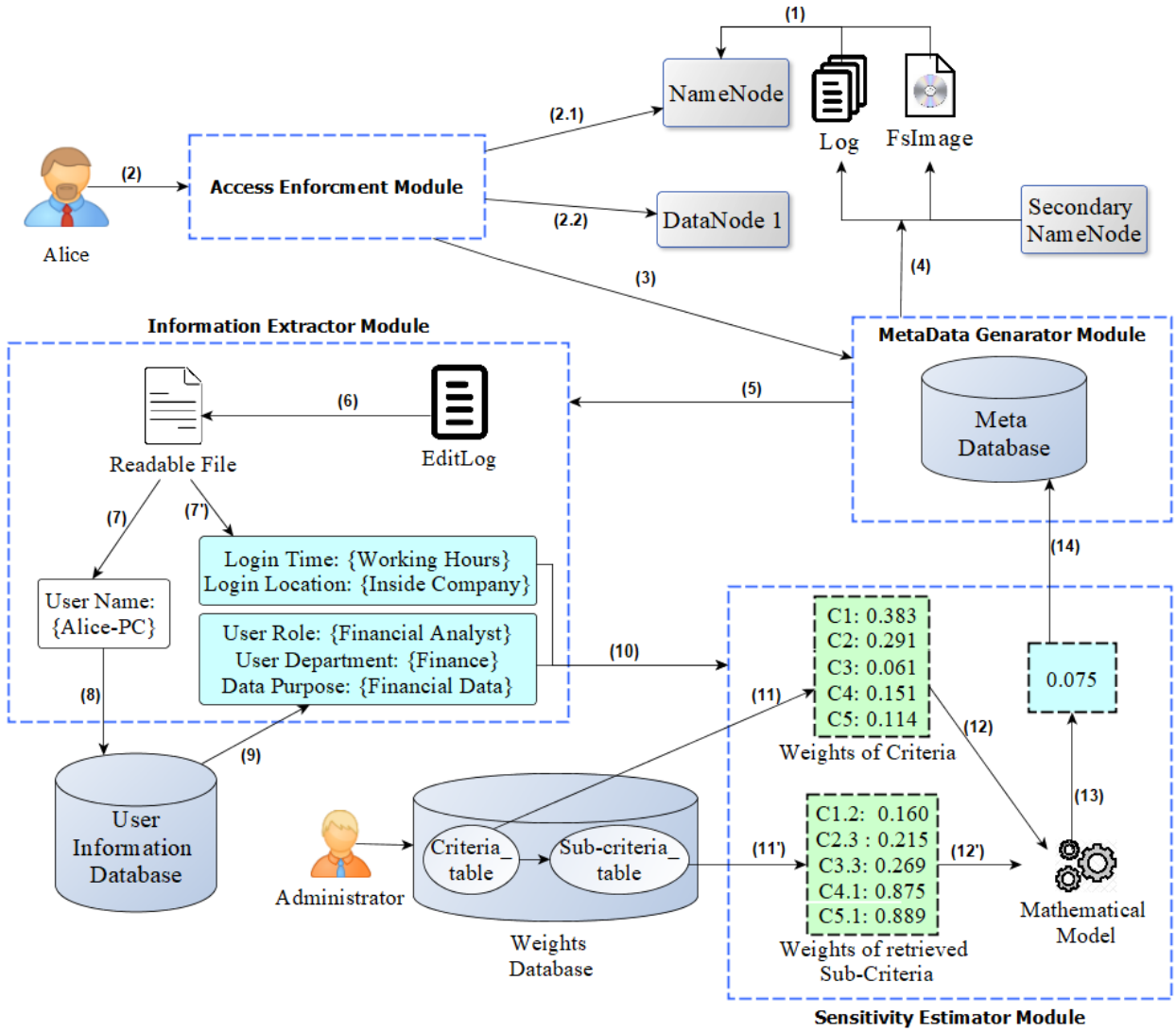

Fig. 11. Example of calculating the sensitivity of Alice's data.

hours ”9:00 AM”) and the login location (from the F2 office).

(8) Using the User Information Database, we retrieve the rest of the sub-criteria used in the calculation of data sensitivity based on the user name.

(9) We precisely retrieve the user role (Financial Analyst), the user department (Finance) and the data purpose (Financial Data).

(10) Thereafter, the IEM sends all the retrieved sub-criteria to the SEM in order to calculate the sensitivity of Alice's data.

(11) (11’) During the pre-processing phase, all weights of criteria and sub-criteria used in our model are calculated and saved in the Criteria table and Sub-criteria table. In the real-time phase, the SEM extracts the weights of the received sub-criteria along with the weights of the criterion to which they belong.

(12) (12’) All these weights are submitted to the Mathematical Model in order to calculate the global weights.

(13) Then, we average the global weights to obtain the sensitivity value of Alice's data.

(14) Finally, the sensitivity value is submitted to the MGM that stores this value into the MetaDatabase along with all the metadata information related to this data.

VI. Discussion

Sensitive data constitutes a significant part of data used in a wide variety of sectors, such as bank, healthcare, government, and others. As Hadoop is increasingly deployed to manage a high amount of sensitive data, the protection of this data has become a crucial issue due to multiple breaches targeting this data. Thus, the proposed D2SAC framework aims to provide a dynamic and strong access control to secure sensitive data contained in HDFS regardless of its sector.

To reach this goal, the D2SAC protects the data since its creation in HDFS. On one hand, the SEM calculates the sensitivity value for each newly created data using the proposed mathematical model. This sensitivity value is expressed as a numerical value and ranges between 0 to 1. In our model, we consider that values closer to 1 indicate that the data is sensitive and values closer to 0 indicate that the data is less sensitive. On the other hand, when a user intends to access a data that already exists in the HDFS. The AEM may or may not authorize access to the demanded data depending on its sensitivity value. The process of given access to users will be discussed in our future work.

In this paper, we focus on how the SEM calculates the sensitivity value of a data. For this, the SEM uses a mathematical model including two phases. In the first phase, the AHP method is used to determine the weights of all identified criteria and sub-criteria, these weights are then saved in the Weights Database. In the second phase, when the AEM receives a request to create a data in HDFS, we extract the sub-criteria related to this data, we then send them to the SEM which uses the empirical average to derive the sensitivity of the data. Figure 11 gives a detailed description of this second phase. In addition, all information related to this data is submitted to the MGM in order to be saved in the Metadatabase and thereby used in the access control process.

VII. Conclusion

A huge amount of sensitive data is generated across the web per day. e.g., Personally Identifiable Information (PII), Personal Health Record (PHR), financial data, etc. This sensitive data is vulnerable to many threats, especially threats related to security violations and unauthorized access. In order to fulfill these gaps, we proposed in our previous paper the D2SAC framework with the purpose of protecting sensitive data in Hadoop Platform.

In this paper, we aim to continue enhancing our framework by concentrating on the SEM in order to improve the calculation of the data sensitivity value. This enhancement is due to our proposed mathematical model, which uses the data itself to calculate its sensitivity value without any involvement of the data owner. Once the priority vectors of factors and sub-factors are calculated using the AHP method then we average the global weights of the identified sub-factors in order to determine the data sensitivity value.

As a future work, we aim that our D2SAC framework could be applied in several fields to validate its effectiveness, rapidity, and data availability. For example, the D2SAC could be implemented to protect medical records in a hospital as well as it could be used to protect sensitive data issued from a business or insurance company. Besides, our second future goal is to provide access control policies to protect sensitive data depending on its degree of sensitivity. Finally, we will investigate the recalculation of data sensitivity value in order to keep sensitive data protected as long as it resides in our framework without increasing the response time for users.

References

[1] H. A. Idar, K. Aissaoui, H. Belhadaoui, and R. F. Hilali, "Dynamic Data Sensitivity Access Control in Hadoop Platform," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Marrakech, 2018, pp. 105–109.

[2] V. Pendyala, "The Big Data Phenomenon," in *Veracity of Big Data*, Berkeley, CA: Apress, 2018, pp. 1–15.

[3] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *Int. J. Inf. Manag.*, vol. 35, no. 2, pp. 137–144, Apr. 2015.

[4] N. Zanoon, A. Al-Haj, and S. M. Khwaldeh, "Cloud computing and big data is there a relation between the two: a study," *Int J Appl Eng Res*, vol. 12, no. 17, pp. 6970–6982, 2017.

[5] A. K. Tk, H. Liu, J. P. Thomas, and G. Mylavarapu, "Identifying Sensitive Data Items within Hadoop," in *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, New York, NY, 2015, pp. 1308–1313.

[6] Apache Hadoop, https://hadoop.apache.org.

[7] G. S. Bhathal and A. Singh, "Big Data: Hadoop framework vulnerabilities, security issues and attacks, *Array*, vol. 1-2, p. 100002, Jan. 2019.

[8] M. Gupta, F. Patwa, and R. Sandhu, "Object-Tagged RBAC Model for the Hadoop Ecosystem, in *Data and Applications Security and Privacy XXXI*, vol. 10359, G. Livraga and S. Zhu, Eds. Cham: Springer International Publishing, 2017, pp. 63–81.

[9] A. Gholami and E. Laure, "Security and Privacy of Sensitive Data in Cloud Computing: A Survey of Recent Developments," in *Computer Science & Information Technology (CS & IT)*, 2015, pp. 131–150.

[10] D. Hyseni, A. Luma, B. Selimi, and B. Cico, "The Proposed Model to Increase Security of Sensitive Data in Cloud Computing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 2, pp. 203–210, 2018.

[11] M. Majumder, "Multi Criteria Decision Making," in *Impact of Urbanization on Water Shortage in Face of Climatic Aberrations*, Singapore: Springer Singapore, 2015, pp. 35–47.

[12] I. M. Al-Jabri, I. Mustafa, and M. S. Sohail, "A Group Decision-Making Method for Selecting Cloud Computing Service Model," 2018.

[13] J. F. Agwa-Ejon, A. Mulaba-Bafubiandi, and J. H. Pretorius, "The Multi-Criteria Decision Making, a Tool for the Promotion of Technological Inter-Connected World through Digital Decision Making," in *2017 Portland International Conference on Management of Engineering and Technology (PICMET)*, 2017, pp. 1–7.

[14] R. Abu Taha and T. Daim, "Multi-Criteria Applications in Renewable Energy Analysis, a Literature Review," in *Research and Technology Management in the Electricity Industry*, T. Daim, T. Oliver, and J. Kim, Eds. London: Springer London, 2013, pp. 17–30.

[15] T. L. Saaty, "Decision making with the analytic hierarchy process," *Int. J. Serv. Sci.*, vol. 1, no. 1, pp. 83–98, 2008.

[16] T. L. Saaty, *The analytic network process: decision making with dependence and feedback.* Pittsburgh, Pa.: RWS Publ., 2001.

[17] E. Roszkowska, "Multi-criteria decision making models by applying the TOPSIS method to crisp and interval data," *Mult. Criteria Decis. MakingUniversity Econ. Katow.*, vol. 6, pp. 200–230, 2011.

[18] J. R. Figueira, V. Mousseau, and B. Roy, "ELECTRE methods," in *Multiple criteria decision analysis*, Springer, 2016, pp. 155–185.

[19] J.-P. Brans and B. Mareschal, "PROMETHEE methods," *in Multiple criteria decision analysis: state of the art surveys*, Springer, 2005, pp. 163–186.

[20] A. Afshari, M. Mojahed, and R. M. Yusuff, "Simple additive weighting approach to personnel selection problem," *Int. J. Innov. Manag. Technol.*, vol. 1, no. 5, p. 511, 2010.

[21] E. Chalúpková and J. Franek, "Application of the analytic hierarchy process method in a comparison of financial leasing and loans," *Ekon. Rev. - Cent. Eur. Rev. Econ. Issues*, vol. 17, no. 2, pp. 75–84, Jun. 2014.

[22] H. Li, F. Ni, Q. Dong, and Y. Zhu, "Application of analytic hierarchy process in network level pavement maintenance decision-making," *Int. J. Pavement Res. Technol.*, vol. 11, no. 4, pp. 345–354, Jul. 2018.

[23] E. Mu and M. Pereyra-Rojas, "Understanding the Analytic Hierarchy Process," in *Practical Decision Making*, Cham: Springer International Publishing, 2017, pp. 7–22.

[24] M. Behera and A. Rasool, "Big Data Security Threats and Prevention Measures in Cloud and Hadoop, in *Data Management, Analytics and Innovation*, vol. 808, V. E. Balas, N. Sharma, and A. Chakrabarti, Eds. Singapore: Springer Singapore, 2019, pp. 143–156.

[25] V. N. Inukollu, S. Arsi, and S. Rao Ravuri, "Security Issues Associated with Big Data in Cloud Computing, *Int. J. Netw. Secur. Its Appl.*, vol. 6, no. 3, pp. 45–56, May 2014.

[26] M. Gupta, F. Patwa, and R. Sandhu, "POSTER: Access Control Model for the Hadoop Ecosystem, in *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies - SACMAT 17 Abstracts*, Indianapolis, Indiana, USA, 2017, pp. 125–127.

[27] D. Das, O. OMalley, S. Radia, and K. Zhang, "Adding security to apache hadoop, *Hortonworks IBM*, pp. 26–36, 2011.

[28] P. P. Sharma and C. P. Navdeti, "Securing big data hadoop: a review of security issues, threats and solution, *Int J Comput Sci Inf Technol*, vol. 5, no. 2, pp. 2126–2131, 2014.

[29] AHP Priority Calculator, https://bpmsg.com/academic/ahp.php.