BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Journal of Computer Science Research

Volume 4 | Issue 2 | April 2022 | ISSN 2630-5151（Online）

BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Volume 4 Issue 2 • April 2022 • ISSN 2630-5151 (Online)

Journal of Computer Science Research

BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Volume 4 | Issue 2 | April 2022 | Page1-41
Journal of Computer Science Research

Contents

Articles

13 **A Mathematical Theory of Big Data**
Zhaohao Sun

24 **Animal Exercise: A New Evaluation Method**
Yu Qi Chongyang Zhang Hiroyuki Kameda

31 **Optimization of Secure Coding Practices in SDLC as Part of Cybersecurity Framework**
Kire Jakimoski Zorica Stefanovska Vekoslav Stefanovski

Review

1 **A Review on Ranking of Z-numbers**
Fırat Bilgin Musa Alci

BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Journal of Computer Science Research
https://ojs.bilpublishing.com/index.php/jcsr

REVIEW

A Review on Ranking of Z-numbers

Fırat Bilgin[1*] Musa Alci[2]

1. Graduate School of Natural and Applied Science, Ege University, Izmir, 35040, Turkey
2. Electrical and Electronics Engineering Department, Ege University, Izmir, 35040, Turkey

ARTICLE INFO

Article history
Received: 14 March 2022
Accepted: 19 April 2022
Published Online: 29 April 2022

Keywords:
Fuzzy Z-numbers
Ranking discrete Z-numbers
Ranking of Z-numbers
Relative entropy based ranking

ABSTRACT

There are numerous studies about Z-numbers since its inception in 2011. Because Z-number concept reflects human ability to make rational decisions, Z-number based multi-criteria decision making problems are one of these studies. When the problem is translated from linguistic information into Z-number domain, the important question occurs that which Z-number should be selected. To answer this question, several ranking methods have been proposed. To compare the performances of these methods, benchmark set of fuzzy Z-numbers has been created in time. There are relatively new methods that their performances are not examined yet on this benchmark problem. In this paper, we worked on these studies which are relative entropy based Z-number ranking method and a method for ranking discrete Z-numbers. The authors tried to examine their performances on the benchmark problem and compared the results with the other ranking algorithms. The results are consistent with the literature, mostly. The advantages and the drawbacks of the methods are presented which can be useful for the researchers who are interested in this area.

1. Introduction

L. Zadeh introduced the Z-number concept to the literature in 2011 [1]. Actually, he was working on the topics combining fuzzy and probabilistic information such as probability measures with fuzzy events [2], fuzzy random variables [3], fuzzy sets and information granularity [4] before put forward the Z-number theory. He claims that Z-numbers can represent the rational decision making ability of the humans under uncertain conditions. Thus, a Z-number contains an uncertainty degree in addition to fuzzy information. A Z-number notation can be shown as

$$Z = (A, B) \text{ or } Z = (X, A, B) \quad (1)$$

X is a set of random variables, A is the restriction part on X and B is the reliability of A. (X, A) is similar for fuzzy researchers, because it is exactly the same with Type-I Fuzzy Logic. And addition of B part makes it Z-number. There is also an extension on Z-number shown as Z^+-number. Whereas Z-number has reliability degree B on A, Z^+-number has probability distribution of reliability degree, B on A.

In recent times, the concept of Z-number is gaining

*Corresponding Author:
Fırat Bilgin,
Graduate School of Natural and Applied Science, Ege University, Izmir, 35040, Turkey;
Email: firat.bilgin@ege.edu.tr

DOI: https://doi.org/10.30564/jcsr.v4i2.4499

much popularity among the researchers. Firstly, generating Z-number was an open issue. For solving this, ordered weighted averaging operators(OWA) based and logistic regression based studies were made [5,6]. Although there were studies about generating Z-number, most of them about Z-number were linguistic [7-9]. We know that the fuzzy systems are good for translating linguistic knowledge into mathematics [10]. After creating Z-number fuzzy if-then rules from the linguistic information, Z-number based calculations play a vital role. For doing these calculations, R. Aliev et al. showed up the formulas for basic algebraic operations such as addition, subtraction etc. [11]. All Z-numbers can be translated into linguistic expressions or vice versa. In this situation, each Z-number contains linguistic or mathematical restrictions. Combining these restrictions with probabilistic restrictions, the probability distribution of the Z-number can be obtained [12]. After having a command of fundamental Z-number terms and calculations, some linguistic based studies have been done. Using Z-number in control problem was one of these studies. In 2018, R. Abiyev et al. controlled an omnidirectional soccer robot with linguistic Z-number rule base[9]. In 2019, M. Shalabi et al. modelled and controlled automotive air-spring suspension system with Z-number based fuzzy system [13]. In 2020, M. Abdelwahab et al. worked on trajectory tracking of a mobile robot with Z-number [14]. As another branch of work, W. Jiang et al. proposed a novel method combining Z-number with Dempster - Shafer evidence theory and they made an application in sensor data fusion problem [15]. As a clustering/classifying problem, Z-number was used with fuzzy c-means and k-means clustering, respectively [16,17]. The effectiveness of the proposed methods was shown on well-known datasets such as iris dataset, wine dataset etc.

As the Z-number based studies are examined in the literature, we make calculations with Z-numbers according to the rule base and we get another Z-number in final. To use final Z-number, B. Kang et al. proposed a method for converting Z-number into classical fuzzy number [18]. Lastly, the researchers have said that converting Z-number into a crisp number may cause information loss. Therefore, using it as Z-number form is more preferable [19]. For doing this, we need Z-number based if-then rules and Z-number based inference engine. At the moment, there is not any study about Z-number inference without converting Z-number into crisp number. Instead of this, ranking Z-numbers are more popular. And there are studies on multi-criteria decision making problems by ranking Z-numbers [20,21]. The results of these applications are mostly consistent with the studies done by mathematical and classical fuzzy operations. But, the issue about this type of works is that there are so many different decision making problems. So, comparing the performances of the proposed ranking methods is impossible. To make this possible, a Z-number fuzzy set has been created. Thus, the researchers can try their proposed method on this set and compare the results with the other methods. As in other Z-number applications, there are two approaches for ranking of Z-numbers. First one is converting Z-number into classical fuzzy, then ranking obtained fuzzy number. The second one is done without converting Z-number into classical fuzzy. In 2014, D. Mohamad et al. converted Z-number into generalized fuzzy number(GFN) because of simpler calculations and they used the standard deviation of GFNs to order them [22]. In 2015, A. Bakar and A. Gegov called their work as multi-layer decision methodology. According to them, conversion process, from Z-number to fuzzy number, is realized in the first layer and in the second layer ranking process is realized. For ranking process, they used centroid point in addition to the spread, called CPS [23]. In 2017, S. Ezadi and T. Allahviranloo proposed a method to rank fuzzy numbers. The method is based on hyperbolic tangent function and convex combination. They turned Z-numbers into generalized normalized fuzzy numbers(GNFS) with B. Kang's formula, and then tried to rank converted fuzzy numbers with their proposed methods [24]. Later, S. Ezadi et al. proposed another method to rank fuzzy numbers by using the similarity between hyperbolic tangent and sigmoid function. By converting the Z-numbers into GNFS, they adapted their method into ranking of Z-numbers [25]. In 2017, Jiang W. et al. proposed a novel method to ranking GFNs. According to this method, a score function is produced based on the centroid of the membership function, spread and Minkowski degree of fuzziness. And the ranking process is realized with produced score value. For ranking Z-numbers, they made some assumptions. According to them, the constraint part of Z-number is more important than the reliability part and it must be the main part of a Z-number. Therefore, the weight of constraint part should be greater than the weight of the reliability. In addition, the information of Z-numbers should be retained without converting into fuzzy or crisp numbers. In the light of these assumptions, they obtained scores for both constraint and reliability part via their proposed method, and they combined these scores with a formula by considering the distance between scores and a reference point [26]. In 2020, R. Chutia proposed a method to rank GFNs according to the concept of value and ambiguity. They obtained values and ambiguities for both constraint and reliability part. After that, they combined the scores as in the method of Jiang W. et al. [27].

There are new methods which are relative entropy of Z-numbers [28] and a method for ranking discrete Z-numbers [29]; but, these methods were not tried on the benchmark set of fuzzy Z-numbers before. In this paper, we examine the performances of these methods on ordering Z-numbers. According to the results, we want to present the drawbacks and advantages of these methods.

2. Materials and Methods

2.1 Materials

The materials of this study are Z-numbers. Let Z_1 and Z_2 be two Z-numbers defined as

$Z_1 = (A_1, B_1)$ and $Z_2 = (A_2, B_2)$ (2)

$$A_s = \frac{\mu_{A_s}(u_1)}{u_1} + \frac{\mu_{A_s}(u_2)}{u_2} + \ldots + \frac{\mu_{A_s}(u_m)}{u_m}$$
$$B_s = \frac{\mu_{B_s}(v_1)}{v_1} + \frac{\mu_{B_s}(v_2)}{v_2} + \ldots + \frac{\mu_{B_s}(v_n)}{v_n} \quad (3)$$

In Equations (2) and (3), A_s and B_s are the membership functions of a Z-number where A defines fuzzy part of a variable which are $u_1, u_2, \ldots u_m$ and B defines the reliability of A. Since we have two Z-numbers, s=1, 2. $\mu_{A_s}(u_i)$, $\mu_{B_s}(v_j) \in [0,1]$ are the membership degree of given variables whose indices are i=1,2,…,m and j=1,2,…n.

Instead of Equation (3), we will use more compact expression to show Z-numbers. Most of time, the fuzzy membership functions are triangular, trapezoidal, Gaussian etc. And the benchmark fuzzy sets in this work only consist of triangular and trapezoidal membership functions. For example, a triangular membership function of reliability, B, can be described as given in Equation (4).

B = (0.6, 0.8, 1.0; 1.0) (4)

The first three component of B describes the critical values and the last component of B, describes the peak membership value of B as seen in the Figure 1.

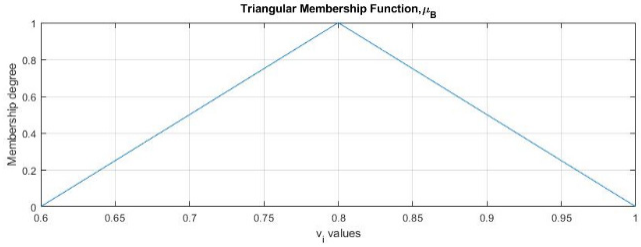

Figure 1. Triangular membership function given in Equation (4).

As in triangular membership function, sample trapezoidal membership functions can be written as in Equation (5).

B = (0.4, 0.8, 1.0; 1.0) (5)

As in Equation (4), the first four components of Equation (5) give the critical values and the last component of B, corresponds to the peak membership value of B as seen in the Figure 2.

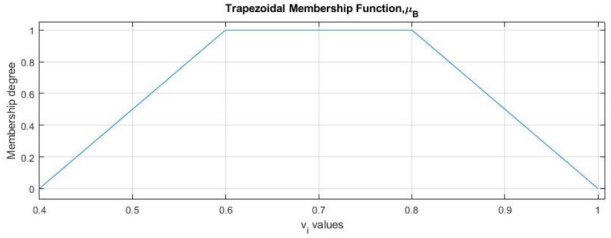

Figure 2. Trapezoidal membership function given in Equation (5).

2.2 Relative Entropy of Z-numbers

L. Yangsue et al. suggested to use relative entropy for ranking Z-numbers. Because the paper is based on the entropy, the underlying probability distributions of the Z-numbers must be found before attempting to find relative entropy. We do not know the underlying probability distributions; but, according to L. Zadeh, there are some restrictions about Z-numbers, called Z-restriction, such as

$$Prob(X\ is\ A_s)\ is\ B_s. \quad (6)$$

This information can be formulated as given in Equation (7).

$$B_s = \int \mu_{A_s}(u).\, p_s(u).\, du \quad (7)$$

The second restriction is given in Equation (8) if the centroids of μ_{A_s} and p_s are coincident.

$$\int u.\, p_s(u).\, du = \frac{\int u.\, \mu_{A_s}(u).\, du}{\int \mu_{A_s}(u).\, du} \quad (8)$$

And we know the probability restrictions as in Equation (9).

$$\sum_{i=1}^{m} p_s(u_i) = 1$$
$$0 \le p_s(u_i) \le 1 \quad (9)$$

There can be more than one solution that satisfies these restrictions. So, the solution that gives maximum Shannon entropy is chosen.

$$max: H(p_s(u_i)) = -\sum_{i=1}^{n} p_s(u_i) \log(p_s(u_i)) \quad (10.a)$$

Or in other words,

$$min: H(p_s(u_i)) = \sum_{i=1}^{n} p_s(u_i) \log(p_s(u_i)) \quad (10.b)$$

In continuous form, it is hard to solve this equation

subject to given equalities and inequalities. To overcome this issue, some assumptions are made such as discretization or having Gaussian probability distributions. But, we do not have any information whether the probability distribution is Gaussian, uniform etc. So, by making the calculations discrete form, we hope that the obtained solution is close to continuous probability distribution with admissible error. Note that there will be *n* solutions for p_1, because there are *n* pieces of *v* value in Equation (3).

After getting the probability distributions p_1 and p_2, the relative entropy must be calculated as step two. For calculating this, the authors used Kullback - Leibler divergence. The divergence is defined in Equation (11.a) and (11.b).

$$D(p_1||p_2) = \sum_{i=1}^{n} p_1(u)\ log_2 \frac{p_1(u)}{p_2(u)} \tag{11.a}$$

$$D(p_2||p_1) = \sum_{i=1}^{n} p_2(u)\ log_2 \frac{p_2(u)}{p_1(u)} \tag{11.b}$$

In meaning, the divergence, $D(p_1||p_2)$, gives the information gain if p_1 is used instead of p_2. As third step, the authors proposed to create another fuzzy subset with obtained relative entropy values as follows.

$$C_1 = \frac{\mu_{B_1}(v_1) + \mu_{B_2}(v_1)}{D_1(p_1||p_2)} + \ldots + \frac{\mu_{B_1}(v_n) + \mu_{B_2}(v_n)}{D_n(p_1||p_2)} \tag{12}$$

$$C_2 = \frac{\mu_{B_1}(v_1) + \mu_{B_2}(v_1)}{D_1(p_2||p_1)} + \ldots + \frac{\mu_{B_1}(v_n) + \mu_{B_2}(v_n)}{D_n(p_2||p_1)} \tag{13}$$

We mentioned about there are n probability distributions for each p_s. And here, $D_i(p_1||p_2)$ is the relative entropy between i^{th} probability distributions of p_1 and p_2.

The centroid of these subsets will be the score of each Z-number. According to this score, ranking process will be realized. The Z-number with greater score will have a better rating in ranking.

2.3 A Method for Ranking Discrete Z-numbers

Gong Y. et al convert the Z-number into classical fuzzy set, and then try to rank different Z-numbers. Unlike the other works in the literature, they proposed a different method to convert Z-number into classical fuzzy instead of B. Kang's conversion formula. According to them, the reliability part, B, is the weight of the constraint part A. But, not just the values of B affect the weight, other criteria such as range (cardinality), linguistic order of the fuzzy set should be important to determine the weight. At the first step, measure of uncertainty, $E(B_s)$, is calculated as follows.

$$E(B_s) = \frac{|V_{B_s}| - 1 + \sum_{i=1}^{n} \min\left(B_s(v_i), B_s^C(v_i)\right)}{|V_{B_s}| - 1 + \sum_{i=1}^{n} \max\left(B_s(v_i), B_s^C(v_i)\right)} \tag{14}$$

Here $|V_{B_s}|$ is the number of elements (cardinality) that B_s have. B_s^C is the complementary set of B_s as in Equation (15).

$$R(B_s) = (1 - E(B_s)) \frac{Order\ of(B_s)}{|L_P|} \tag{15}$$

According to the authors, there are linguistic fuzzy clusters both for reliability and constraint part. Here, $|L_P|$ is the number of these fuzzy clusters for reliability part and $Order\ of(B_s)$ is the order of (B_s) in L_P. To illustrate, assume a fuzzy set L_P.

$$L_P = \{never, rarely, sometimes, usually, always\} \tag{16}$$

Here, L_P =4 and $Order\ of(rarely)$=1.

We know that $E(B_s)$ is the measure of uncertainty, $(1 - E(B_s))$ gives the measure of certainty, analogously.

At the second step, the constraint part can be weighted with $R(B_s)$, and the new fuzzy set, A_S^*, occurs as a result of this process.

$$A_S^* = R(B_s).A \tag{17}$$

For ranking discrete fuzzy sets, H. Basirzadeh et al. proposed a method in 2012 [30]. Gong Y. et al. followed the same procedure to rank A_S^* in their work. According to this method, the regions where the membership function increases or decreases are checked. And an α-value is defined as follows.

$$\begin{aligned} &\alpha - \text{value} \triangleq \alpha \in [0,1] \text{ such that} \\ &\alpha \leq \min\left(A_S^*(u_1), A_S^*(u_2), \ldots, A_S^*(u_m)\right) \end{aligned} \tag{18}$$

For both of the regions, increasing and decreasing, a score is calculated. For increasing part, the score is, Q_{inc}^{α}. For decreasing part, the score is, Q_{dec}^{α}. They are given below equations.

$$Q_{inc}^{\alpha} = u_1.\left(\mu_{A_S^*}(u_1) - \alpha\right) + \sum_{i=2}^{m} u_i(\mu_{A_S^*}(u_i) - \mu_{A_S^*}(u_{i-1})) \tag{19}$$

$$Q_{dec}^{\alpha} = u_m.\left(\mu_{A_S^*}(u_m) - \alpha\right) + \sum_{i=1}^{m-1} u_i(\mu_{A_S^*}(u_i) - \mu_{A_S^*}(u_{i+1})) \tag{20}$$

And the total score of Z-number is the sum of Q_{inc}^{α} and Q_{dec}^{α}.

$$Q^{\alpha} = Q_{inc}^{\alpha} + Q_{dec}^{\alpha} \tag{21}$$

As in relative entropy based ranking method, the Z-number with greater score will have a better rating.

The main disadvantage of this method is that it needs $Order\ of(B_s)$ and number of fuzzy clusters, L_P. If we want to rank just two Z-numbers, the multiplier will be 0.5 and 1.0 from Equation (15). Even if the reliability parts

of Z-numbers have small differences, the difference between multipliers will be enormous. Moreover, assume two Z-numbers that have B parts, B_1=(0.4,0.6,0.8) and B_2=(0.4,0.5,0.7,0.8). These are so similar and they may be called with same linguistic information "sometimes", although they are different membership functions. In this situation, how we will decide to order of these Z-numbers is another important issue. Thus, we proposed an extension to overcome this drawback. It is known that the range of membership functions is between [0,1]. Even if it is not in this range, it can be scaled by normalization.

As a first step, we will divide [0,1] into 20 parts with 0.05 steps. Thus, the $|L_P|$ will be equal to 20 whatever the Z-number is. To determine $Order\ of(B_s)$, the centroid method will be used, centroid of B_s can be found via Equation (22). To which region the centroid is closer, the number of that region corresponds to the $Order\ of(B_s)$.

$$Centroid\ of\ B_s = \frac{\sum_{i=1}^{n} v_i \cdot \mu_{B_s}(v_i)}{\sum_{i=1}^{n} \mu_{B_s}(v_i)} \tag{22}$$

$Order\ of(B_s)$ is used in Equation (15) to calculate the weight, $R(B_s)$. By arranging the $Order\ of(B_s)$ in the method proposed from Gong Y. et al., we have aimed to obtain more accurate weight term. With this improvement, we are expecting the minor differences that roots from the reliability membership functions can be distinguished and can be taken into account. Actually, the calculations continue to both get a new fuzzy set and produce a score from this new fuzzy set after determining $R(B_s)$. Therefore, the effect of this improvement may not be observed directly from the scores given in Equation (21). However, when Equation (15) is examined, both $Order\ of(B_s)$ and L_P are the multiplier of $(1 - E(B_s))$ which is a measure of certainty. Given the importance of using knowledge in uncertain conditions, any improvements in certainty will have a positive effect on the results.

Optimization of the number of fuzzy membership functions is an important topic in fuzzy applications. Increasing the number of membership functions may cause the system lose the capability of generalization and may require large computation time. At the same time, using few membership functions may cause incomplete modeling and inaccurate results [31]. At this stage, converting linguistic information to the fuzzy Z-number accurately with a sufficient number of membership functions can be another work topic. In this study, L_P was chosen as 20 and the results were given in Results section. Getting a better ranking performance will be possible by optimizing L_P.

3. Results

R. Chutia divided the benchmark set of fuzzy Z-numbers into three examples in his work. We will follow this tradition and we will give the results in this order. In example 1, there are 6 fuzzy sets that have same restriction part, A which is given in Equation (23). This means that the information, e.g. a sensor data, is same for each Z-number, but the reliabilities of information will differ with changing B parts which are given Equations (24), (25), and (26).

$$A = (0.1,0.3,0.5;1.0) \tag{23}$$

In Set 1,

$$B_1 = (0.1,0.3,0.5;1.0)$$
$$B_2 = (0.2,0.3,0.4;1.0) \tag{24}$$

In Set 2,

$$B_1 = (0.1,0.2,0.4,0.5;1)$$
$$B_2 = (0.1,0.3,0.5;1.0) \tag{25}$$

In Set 3,

$$B_1 = (0.1,0.3,0.5;0.8)$$
$$B_2 = (0.1,0.3,0.5;1.0) \tag{26}$$

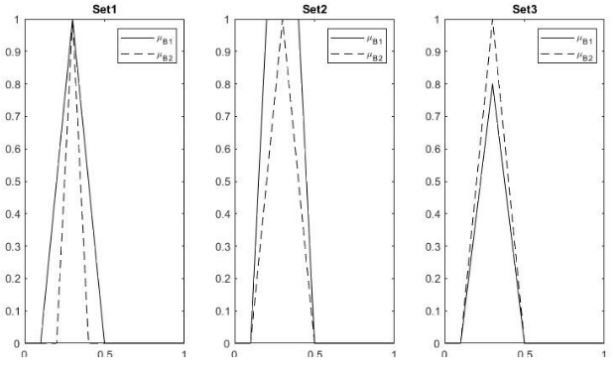

Figure 3. Reliability membership functions of first three sets of example 1

When first three sets of example 1 are examined, the reliability membership functions differ in shape, wideness and peak value. In Set 1, the peak values and shape of the membership functions are same. But, B_1 is wider than B_2. We can infer that B_1 is more fuzzy than B_2. So, we will be able to see the effect, when the fuzziness of the reliability changes. In Set 2, the membership functions differ in shape where the first one is triangular membership

function and the latter is trapezoidal. Although their peak values, centroids and wideness are same, we will be able to examine the effect of the membership function's shape. Different from the first two sets, we can predict a result for this set intuitively that the membership function with higher peak value is more preferable than the other. From this point of view, Set 3 can be more distinctive while comparing the ranking methods.

In Set 4,

$$B_1 = (0.1,0.2,0.4,0.5;1)$$
$$B_2 = (0.1,0.3,0.5;0.8) \quad (27)$$

In Set 5,

$$B_1 = (0.1,0.2,0.4,0.5;1)$$
$$B_2 = (0.3,0.3,0.3;1) \quad (28)$$

In Set 6,

$$B_1 = (0.1,0.3,0.5;1)$$
$$B_2 = (0.3,0.3,0.3;1) \quad (29)$$

The above membership functions in Equations (27), (28), (29) which belongs the Set 4, Set 5 and Set 6 can be seen in the Figure 4. All these sets try to measure the effects of membership functions that differ in shape. Set 4 tries to measure the effect of peak values, additionally.

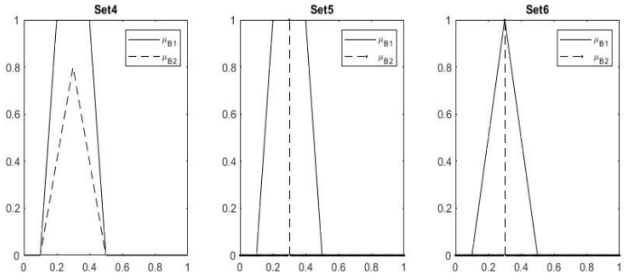

Figure 4. Reliability membership functions of last three sets of example 1

The results from the literature and the methods examined in this study for example 1 are given in Table 1. It is seen from the Table 1 that some of the proposed methods are failed to rank Z-numbers. Although the differences between membership functions in Set 1 are clear, the produced scores are exactly same. It can be said that the performances of the methods producing same scores for different membership functions are poor. These methods were proposed by Mohamad et al. [22], Bakar and Gegov [23], Ezadi and Allahviranloo [24], Ezadi et al. [25].

Jiang et al., R. Chutia, Yongsue et al. and Gong et al. succeed to rank these Z-numbers and their ranking results are same. Another important point about the study of R. Chutia et al., the score is inversely proportional to the rank. From decision making aspect, the score can be interpreted as cost criterion instead of benefit criterion.

Different from Table 1, R. Chutia orders the Z-numbers contrarily to Jiang et al. and Gong et al. in Table 2. Another difference from the previous results that Yangsue et al. do not differ the two Z-numbers. Their score for each Z-number is zero. Because the method of Yangsue et al. is based on the underlying probability distributions, they obtain same probability distributions when the range of reliability and the constraint membership functions are same. Thus, the relative entropy turns zero when the two probability distributions are same.

Table 1. Results for Set 1 of Example 1

Set 1			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0508	0.0508	$Z_1 \sim Z_2$
Jiang et al.	0.1953	0.2024	$Z_1 < Z_2$
Mohamad et al.	0.0706	0.0706	$Z_1 \sim Z_2$
Ezadi et al.	0.5224	0.5224	$Z_1 \sim Z_2$
Ezadi and Allahviranloo			
$\alpha = 0.0$	0.3564	0.3564	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.2996	0.2996	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.0897	0.0897	$Z_1 \sim Z_2$
Ezadi et al.			
$\alpha = 0.0$	0.5921	0.5921	$Z_1 \sim Z_2$
$\alpha = 0.5$	0.5719	0.5719	$Z_1 \sim Z_2$
$\alpha = 1.0$	0.5224	0.5224	$Z_1 \sim Z_2$
R. Chutia			
$\alpha = 0.1$	0.0648	0.0557	$Z_1 < Z_2$
$\alpha = 0.5$	0.0333	0.0286	$Z_1 < Z_2$
$\alpha = 0.9$	0.0018	0.0016	$Z_1 < Z_2$
Yangsue et al.	0.0105	0.0110	$Z_1 < Z_2$
Gong et al.	0.2375	0.2672	$Z_1 < Z_2$

Table 2. Results for Set 2 of Example 1

Set 2			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0508	0.0508	$Z_1 \sim Z_2$
Jiang et al.	0.2049	0.1953	$Z_1 > Z_2$
Mohamad et al.	0.0706	0.0706	$Z_1 \sim Z_2$
Ezadi et al.	0.5224	0.5224	$Z_1 \sim Z_2$
Ezadi and Allahviranloo			
$\alpha = 0.0$	0.3564	0.3564	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.2821	0.2821	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.0897	0.0897	$Z_1 \sim Z_2$

Table 2 continued

Set 2			
Methods	Z_1	Z_2	**Result**
Ezadi et al.			
$\alpha = 0.0$	0.5921	0.5921	$Z_1 \sim Z_2$
$\alpha = 0.5$	0.5719	0.5719	$Z_1 \sim Z_2$
$\alpha = 1.0$	0.5224	0.5224	$Z_1 \sim Z_2$
R. Chutia			
$\alpha = 0.1$	0.0906	0.0648	$Z_1 < Z_2$
$\alpha = 0.5$	0.0571	0.0333	$Z_1 < Z_2$
$\alpha = 0.9$	0.0101	0.0018	$Z_1 < Z_2$
Yangsue et al.	0.0000	0.0000	$Z_1 \sim Z_2$
Gong et al.	0.3117	0.2375	$Z_1 > Z_2$

Jiang et al., R. Chutia and Gong et al. ranked the Z-numbers successfully in Table 3. We say they ranked successfully, because the ranking, $Z_1 < Z_2$, can be done intuitively, too. When the Set 3 is examined, it can be seen that the membership functions of reliability are same in range and shape. They only differ in maximum membership value, so one can expect that the membership function with greater value takes greater order in ranking.

Table 3. Results for Set 3 of Example 1

Set 3			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0508	0.0508	$Z_1 \sim Z_2$
Jiang et al.	0.1916	0.1953	$Z_1 < Z_2$
Mohamad et al.	0.0706	0.0706	$Z_1 \sim Z_2$
Ezadi et al.	0.5224	0.5224	$Z_1 \sim Z_2$
Ezadi and Allahviranloo			
$\alpha = 0.0$	0.3564	0.3564	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.2821	0.2821	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.0897	0.0897	$Z_1 \sim Z_2$
Ezadi et al.			
$\alpha = 0.0$	0.5921	0.5921	$Z_1 \sim Z_2$
$\alpha = 0.5$	0.5719	0.5719	$Z_1 \sim Z_2$
$\alpha = 1.0$	0.5224	0.5224	$Z_1 \sim Z_2$
R. Chutia			
$\alpha = 0.1$	0.2649	0.2970	$Z_1 < Z_2$
$\alpha = 0.5$	0.1945	0.2250	$Z_1 < Z_2$
$\alpha = 0.9$	0.1261	0.1530	$Z_1 < Z_2$
Yangsue et al.	0.0000	0.0000	$Z_1 \sim Z_2$
Gong et al.	0.2215	0.2375	$Z_1 < Z_2$

In Table 4, $Z_1 > Z_2$ according to Jiang et al., R. Chutia and Gong et al. Set 4 looks like Set 2 from many perspectives. Differently, the reliability membership value of in Set 4 is less than the Z_2 in Set 2. While the decisions of Jiang et al. and Gong et al. are staying same as expected, R. Chutia changes his order preference as $Z_1 > Z_2$ for Set 4 comparing to the decision in Set 2.

Table 4. Results for Set 4 of Example 1

Set 4			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0508	0.0508	$Z_1 \sim Z_2$
Jiang et al.	0.2049	0.1916	$Z_1 > Z_2$
Mohamad et al.	0.0706	0.0706	$Z_1 \sim Z_2$
Ezadi et al.	0.5224	0.5224	$Z_1 \sim Z_2$
Ezadi and Allahviranloo			
$\alpha = 0.0$	0.3564	0.3564	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.2821	0.2821	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.0897	0.0897	$Z_1 \sim Z_2$
Ezadi et al.			
$\alpha = 0.0$	0.5921	0.5921	$Z_1 \sim Z_2$
$\alpha = 0.5$	0.5719	0.5719	$Z_1 \sim Z_2$
$\alpha = 1.0$	0.5224	0.5224	$Z_1 \sim Z_2$
R. Chutia			
$\alpha = 0.1$	0.2970	0.2649	$Z_1 > Z_2$
$\alpha = 0.5$	0.2250	0.1945	$Z_1 > Z_2$
$\alpha = 0.9$	0.1530	0.1261	$Z_1 > Z_2$
Yangsue et al.	0.0000	0.0000	$Z_1 \sim Z_2$
Gong et al.	0.3117	0.2215	$Z_1 > Z_2$

For Table 5 which gives the results for Set 5 of example 1, Jiang et al., R. Chutia, Yangsue et al., Gong et al. positioned above Z_1. Because the range of reliability part membership functions are different, the method of Yangsue et al. produced a meaningful output for this set.

Table 5. Results for Set 5 of Example 1

Set 5			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0508	0.0508	$Z_1 \sim Z_2$
Jiang et al.	0.2049	0.2554	$Z_1 < Z_2$
Mohamad et al.	0.0706	0.0706	$Z_1 \sim Z_2$
Ezadi et al.	0.5224	0.5224	$Z_1 \sim Z_2$
Ezadi and Allahviranloo			
$\alpha = 0.0$	0.3564	0.3564	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.2821	0.2821	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.0897	0.0897	$Z_1 \sim Z_2$
Ezadi et al.			
$\alpha = 0.0$	0.5921	0.5921	$Z_1 \sim Z_2$
$\alpha = 0.5$	0.5719	0.5719	$Z_1 \sim Z_2$
$\alpha = 1.0$	0.5224	0.5224	$Z_1 \sim Z_2$
R. Chutia			
$\alpha = 0.1$	0.0906	0.0521	$Z_1 < Z_2$
$\alpha = 0.5$	0.0571	0.0268	$Z_1 < Z_2$
$\alpha = 0.9$	0.0101	0.0015	$Z_1 < Z_2$
Yangsue et al.	0.0735	0.0813	$Z_1 \sim Z_2$
Gong et al.	0.3117	0.7125	$Z_1 < Z_2$

Set 6 has the same membership function for as Set 5 from other respects except for the shape. So, Jiang et al., R. Chutia, Yangsue et al., Gong et al. give same answer, $Z_1 < Z_2$ in Table 6.

Table 6. Results for Set 6 of Example 1

Set 6			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0508	0.0508	$Z_1 \sim Z_2$
Jiang et al.	0.1953	0.2554	$Z_1 < Z_2$
Mohamad et al.	0.0706	0.0706	$Z_1 \sim Z_2$
Ezadi et al.	0.5224	0.5224	$Z_1 \sim Z_2$
Ezadi and Allahviranloo			
$\alpha = 0.0$	0.3564	0.3564	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.2821	0.2821	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.0897	0.0897	$Z_1 \sim Z_2$
Ezadi et al.			
$\alpha = 0.0$	0.5921	0.5921	$Z_1 \sim Z_2$
$\alpha = 0.5$	0.5719	0.5719	$Z_1 \sim Z_2$
$\alpha = 1.0$	0.5224	0.5224	$Z_1 \sim Z_2$
R. Chutia			
$\alpha = 0.1$	0.0648	0.0521	$Z_1 < Z_2$
$\alpha = 0.5$	0.0333	0.0286	$Z_1 < Z_2$
$\alpha = 0.9$	0.0018	0.0015	$Z_1 < Z_2$
Yangsue et al.	0.0741	0.0823	$Z_1 < Z_2$
Gong et al.	0.2375	0.7125	$Z_1 < Z_2$

In example 2, there are 3 fuzzy sets that have same constraint part A. This means that the example 2 will try to rank Z-numbers according to the changing reliability parts as in example 1. The equations of the given fuzzy sets can be seen in Equations (30), (31), (32) and (33).

$$A = (0.1,0.4,0.6;1) \tag{30}$$

In Set 1,

$$B_1 = (0.1,0.4,0.5;1)$$
$$B_2 = (0.2,0.3,0.6;1) \tag{31}$$

In Set 2,

$$B_1 = (0.1,0.4,0.7;1)$$
$$B_2 = (0.2,0.3,0.5,0.6;1) \tag{32}$$

In Set 3,

$$B_1 = (0.2,0.3,0.4,0.5;1)$$
$$B_2 = (0.2,0.3,0.5,0.6;1) \tag{33}$$

The sets of example 2 consist of the membership functions that differ in critical values. The critical values term can be defined as the limit and the peak values of the piecewise continuous function. Considering that the right side is more reliable, a result can be predicted for Set 3, intuitively. But, estimating the result for Set 1 and Set 2 is looking hard. The results for example 2 can be seen in the following tables: Tables 7, 8 and 9.

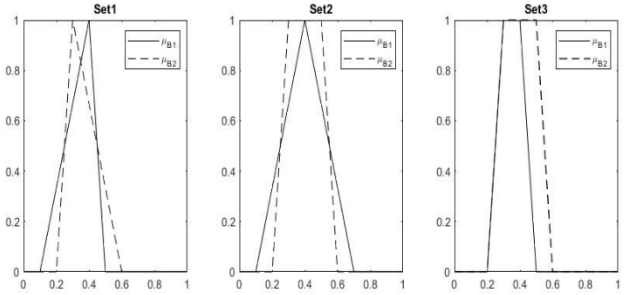

Figure 5. Reliability membership functions of example 2

Table 7. Results for Set 1 of Example 2

Set 1			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0680	0.0736	$Z_1 < Z_2$
Jiang et al.	0.2303	0.2597	$Z_1 < Z_2$
Mohamad et al.	0.0987	0.1422	$Z_1 < Z_2$
Ezadi et al.	0.5332	0.5399	$Z_1 < Z_2$
Ezadi and Allahviranloo			
$\alpha = 0.0$	0.5062	0.5257	$Z_1 < Z_2$
$\alpha = 0.0$	0.4081	0.4300	$Z_1 < Z_2$
$\alpha = 0.0$	0.1325	0.1586	$Z_1 < Z_2$
Ezadi et al.			
$\alpha = 0.0$	0.6358	0.6420	$Z_1 < Z_2$
$\alpha = 0.5$	0.6066	0.6130	$Z_1 < Z_2$
$\alpha = 1.0$	0.5332	0.5399	$Z_1 < Z_2$
R. Chutia			
$\alpha = 0.1$	0.3854	0.3960	$Z_1 < Z_2$
$\alpha = 0.5$	0.2945	0.3000	$Z_1 < Z_2$
$\alpha = 0.9$	0.0756	0.0760	$Z_1 < Z_2$
Yangsue et al.	0.0602	0.0640	$Z_1 < Z_2$
Gong et al.	0.0888	0.1035	$Z_1 < Z_2$

All of the methods in the literature order $Z_1 < Z_2$.

Table 8. Results for Set 2 of Example 2

Set 2			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0736	0.0736	$Z_1 \sim Z_2$
Jiang et al.	0.2420	0.2597	$Z_1 < Z_2$
Mohamad et al.	0.1422	0.1422	$Z_1 \sim Z_2$
Ezadi et al.	0.5399	0.5399	$Z_1 \sim Z_2$

Table 8 continued

Set 2			
Methods	Z_1	Z_2	**Result**
Ezadi and Allahviranloo			
$\alpha = 0.0$	0.5257	0.5257	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.4300	0.4300	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.1586	0.1586	$Z_1 \sim Z_2$
Ezadi et al.			
$\alpha = 0.0$	0.6420	0.6420	$Z_1 \sim Z_2$
$\alpha = 0.5$	0.6130	0.6130	$Z_1 \sim Z_2$
$\alpha = 1.0$	0.5399	0.5399	$Z_1 \sim Z_2$
R. Chutia			
$\alpha = 0.1$	0.0972	0.1094	$Z_1 > Z_2$
$\alpha = 0.5$	0.0500	0.0658	$Z_1 > Z_2$
$\alpha = 0.9$	0.0028	0.0104	$Z_1 > Z_2$
Yangsue et al.	0.0121	0.0128	$Z_1 < Z_2$
Gong et al.	0.1294	0.1553	$Z_1 < Z_2$

As in the other sets, the results of Jiang et al., Yangsue et al. and Gong et al. are consistent and $Z_1 < Z_2$. Unlike these results, the results of R. Chutia give $Z_1 > Z_2$ for all α-levels.

Table 9. Results for Set 3 of Example 2

Set 3			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0. 0736	0. 0736	$Z_1 \sim Z_2$
Jiang et al.	0.2577	0.2597	$Z_1 < Z_2$
Mohamad et al.	0.1422	0.1422	$Z_1 \sim Z_2$
Ezadi et al.	0.5399	0.5399	$Z_1 \sim Z_2$
Ezadi and Allahviranloo			
$\alpha = 0.0$	0.5257	0.5257	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.4300	0.4300	$Z_1 \sim Z_2$
$\alpha = 0.0$	0.1586	0.1586	$Z_1 \sim Z_2$
Ezadi et al.			
$\alpha = 0.0$	0.6420	0.6420	$Z_1 \sim Z_2$
$\alpha = 0.5$	0.6130	0.6130	$Z_1 \sim Z_2$
$\alpha = 1.0$	0.5399	0.5399	$Z_1 \sim Z_2$
R. Chutia			
$\alpha = 0.1$	0.0806	0.1094	$Z_1 > Z_2$
$\alpha = 0.5$	0.0415	0.0658	$Z_1 > Z_2$
$\alpha = 0.9$	0.0023	0.0104	$Z_1 > Z_2$
Yangsue et al.	0.0164	0.0165	$Z_1 < Z_2$
Gong et al.	0.1294	0.1553	$Z_1 < Z_2$

For Set 3, Jiang et al., Yangsue et al. and Gong et al. ranked the Z-numbers as $Z_1 < Z_2$. R. Chutia ranked as different from the other researchers. We were expecting the result, $Z_1 < Z_2$, intuitively. For the methods that fail to rank this set, we can interpret that as the fuzziness increases, the ranking performances decrease.

In example 3, there are 3 fuzzy sets again. The constraint parts of these fuzzy sets are same with example 1 and example 2. As in the other examples, we will try to measure the effect of the change in reliability on the ranking. The related z-numbers and their membership functions (Figure 6) are given below.

$$A = (0.1,0.4,0.6;1.0) \tag{34}$$

The reliability part in Set 1,

$$B_1 = (0.1,0.3,0.5;1.0)$$
$$B_2 = (0.3,0.5,0.7;1.0) \tag{35}$$

In Set 2,

$$B_1 = (0.1,0.2,0.4,0.5;1)$$
$$B_2 = (1.0,1.0,1.0;1.0) \tag{36}$$

In Set 3,

$$B_1 = (0.4,0.5,1.0;1.0)$$
$$B_2 = (0.4,0.7,1.0;1.0) \tag{37}$$
$$B_3 = (0.4,0.9,1.0;1.0)$$

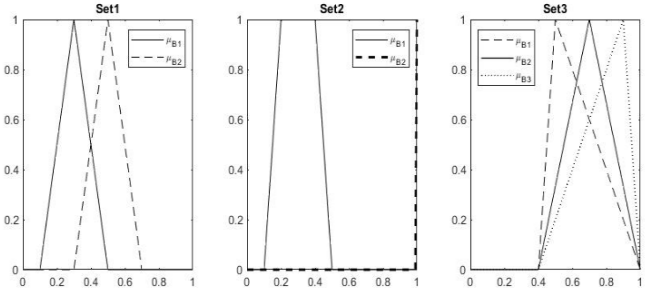

Figure 6. Reliability membership functions of example 3

When the right side is considered more reliable, the sets of example 3 can be interpreted, intuitively. And this is an advantage while comparing the performances of the methods. From the perspective of algorithms, varying fuzziness can be challenging in Set 2. And for Set 3, the place of the peak values is different, although the limit values of the membership functions are same. This situation can also be challenging for the methods. The results of Set 1 from example 3 are given in the Table 10.

Table 10. Results for Set 1 of Example 3

Set 1			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0650	0.0809	$Z_1 < Z_2$
Jiang et al.	0.2220	0.2774	$Z_1 < Z_2$
Mohamad et al.	0.0715	0.1987	$Z_1 < Z_2$
Ezadi et al.	0.5299	0.5498	$Z_1 < Z_2$
Ezadi and Allahviranloo $\alpha = 0.0$ $\alpha = 0.0$ $\alpha = 0.0$			
	0.4962	0.5541	$Z_1 < Z_2$
	0.3969	0.4621	$Z_1 < Z_2$
	0.1194	0.1973	$Z_1 < Z_2$
Ezadi et al. $\alpha = 0.0$ $\alpha = 0.5$ $\alpha = 1.0$			
	0.6328	0.6511	$Z_1 < Z_2$
	0.6034	0.6225	$Z_1 < Z_2$
	0.5299	0.5498	$Z_1 < Z_2$
R. Chutia $\alpha = 0.1$ $\alpha = 0.5$ $\alpha = 0.9$			
	0.3653	0.4308	$Z_1 < Z_2$
	0.2768	0.3264	$Z_1 < Z_2$
	0.0701	0.0827	$Z_1 < Z_2$
Yangsue et al.	0.2379	0.2523	$Z_1 < Z_2$
Gong et al.	0.0757	0.1514	$Z_1 < Z_2$

All of the methods in the literature give $Z_1 < Z_2$. It is seen from the Set 1 that has a membership function closer to the right. The results are not surprising, when the right side is considered more reliable.

In Table 11, the result of Yangsue et al. is different contrary to other methods. It may root from that Set 2 is less fuzzy than the other sets. And this may be compelling to find possible underlying distributions that represent the real distributions. We know that the number of underlying probability distributions will decrease, as the number of v_i decreases.

Table 11. Results for Set 2 of Example 3

Set 2			
Methods	Z_1	Z_2	**Result**
Bakar and Gegov	0.0650	0.1067	$Z_1 < Z_2$
Jiang et al.	0.2309	0.5799	$Z_1 < Z_2$
Mohamad et al.	0.0506	0.5623	$Z_1 < Z_2$
Ezadi et al.	0.5299	0.5987	$Z_1 < Z_2$
Ezadi and Allahviranloo $\alpha = 0.0$ $\alpha = 0.0$ $\alpha = 0.0$			
	0.4962	0.6774	$Z_1 < Z_2$
	0.3969	0.6043	$Z_1 < Z_2$
	0.1194	0.3799	$Z_1 < Z_2$

Table 11 continued

Set 2			
Methods	Z_1	Z_2	**Result**
Ezadi et al. $\alpha = 0.0$ $\alpha = 0.5$ $\alpha = 1.0$			
	0.6328	0.6951	$Z_1 < Z_2$
	0.6034	0.6682	$Z_1 < Z_2$
	0.5299	0.5986	$Z_1 < Z_2$
R. Chutia $\alpha = 0.1$ $\alpha = 0.5$ $\alpha = 0.9$			
	0.3653	0.6346	$Z_1 < Z_2$
	0.2768	0.4808	$Z_1 < Z_2$
	0.0701	0.1218	$Z_1 < Z_2$
Yangsue et al.	2.2010	0.5341	$Z_1 > Z_2$
Gong et al.	0.1192	0.9083	$Z_1 < Z_2$

For Set 3, the method of Yangsue et al. does not produce meaningful output and we have already mentioned about its reasons in previous sets. Except this, the results that all the methods give the same result with $Z_1 < Z_2 < Z_3$ can be seen in Table 12. It was an expected result that the membership functions was closer to the reliable region from Z_1 to Z_3.

Table 12. Results for Set 3 of Example 3

Set 3				
Table 6	Z_1	Z_2	Z_3	**Result**
Bakar and Gegov	0.0892	0.0929	0. 0929	$Z_1 < Z_2 < Z_3$
Jiang et al.	0.3084	0.3295	0.3507	$Z_1 < Z_2 < Z_3$
Mohamad et al.	0.2695	0.3293	0.3774	$Z_1 < Z_2 < Z_3$
Ezadi et al.	0.5629	0.5695	0.5761	$Z_1 < Z_2 < Z_3$
Ezadi and Allahviranloo $\alpha = 0.0$ $\alpha = 0.0$ $\alpha = 0.0$				
	0.5899	0.6071	0.6236	$Z_1 < Z_2 < Z_3$
	0.5030	0.5227	0.5418	$Z_1 < Z_2 < Z_3$
	0.2481	0.2729	0.2974	$Z_1 < Z_2 < Z_3$
Ezadi et al. $\alpha = 0.0$ $\alpha = 0.5$ $\alpha = 1.0$				
	0.6632	0.6691	0.6750	$Z_1 < Z_2 < Z_3$
	0.6349	0.6411	0.6472	$Z_1 < Z_2 < Z_3$
	0.5630	0.5695	0.5761	$Z_1 < Z_2 < Z_3$
R. Chutia $\alpha = 0.1$ $\alpha = 0.5$ $\alpha = 0.9$				
	0.4552	0.5082	0.5639	$Z_1 < Z_2 < Z_3$
	0.3388	0.3850	0.4339	$Z_1 < Z_2 < Z_3$
	0.0834	0.0976	0.1127	$Z_1 < Z_2 < Z_3$
Yangsue et al.	0.0000	0.0000	0.0000	$Z_1 \sim Z_2 \sim Z_3$
Gong et al.	0. 1817	0.2023	0.2168	$Z_1 < Z_2 < Z_3$

4. Conclusions

There has been a lot of study about using Z-numbers in multi-criteria decision making problems since the day they were introduced. Z-numbers are important for this kind of problems, because the idea at the root of their emergence is that better decisions can be made by imitating the hu-

man decision making ability. However, after the linguistic information had been converted into Z-number, an important issue occurred about which Z-number was better. To obtain an answer to this question, several ranking methods have been proposed in time. The performances of some of these ranking methods were measured on the benchmark problem, some of them were not. In this paper, we examined the performance of two Z-number ranking methods whose performances are not examined yet. We tried to rank Z-numbers in the benchmark problem and presented the advantages and disadvantages of these methods. The first method was relative entropy of Z-numbers by L. Yangsue et al. Their method was entropy based and it was bounded to probability distributions of Z-numbers which have probabilistic and fuzzy restrictions. The main disadvantage of this method is that the underlying probability distributions are same for the Z-numbers which have same constraint membership function and reliability membership function with the same range. When the probability distributions are same, the relative entropy cannot differ given Z-numbers. The results of this method are consistent with the literature for example 1 and example 2. For the Set 2 from example 3, this method produced an output contrary to the other methods; this may root from the fuzziness of this set. When one examines this set, the membership function looks precise, so this may reduce the possible underlying probabilities and may cause incorrect ordering. As an advantage, the method makes ranking process without converting Z-numbers into fuzzy numbers. Considering converting Z-numbers leads to loss of information, the method can be beneficial for critical applications. The second method was for ranking of discrete Z-numbers. As the name implies, the method is for discrete Z-numbers. Extending the method for continuous Z-numbers may be considered for the future works. To obtain a better ranking performance, an improvement is made by setting the number of reliability functions as constant. The purpose of doing this was to avoid inaccurate ranking while ordering the similar Z-numbers. As the ranking results are examined, the scores are consistent with the other methods in the literature. Therefore, it can be mentioned as an improvement. However, studies may be done on optimizing this constant number for better results in the future. The outputs of this method are same with the results of Jiang et.al. In spite of their good performances, these methods have to convert Z-numbers. So, these methods should be used where a small amount of information loss can be tolerated.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] Zadeh, L.A., 2011. A note on Z-numbers. Information Sciences. 181(14), 2923-2932.
DOI: https://doi.org/10.1016/j.ins.2011.02.022

[2] Zadeh, L.A., 1968. Probability measures of fuzzy events. Journal of Mathematical Analysis and Applications. 23(2), 421-427.
DOI: https://doi.org/10.1016/0022-247X(68)90078-4

[3] Puri, M.L., Ralescu, D.A., Zadeh, L.A., 1993. Fuzzy random variables. Readings in Fuzzy Sets for Intelligent Systems. pp. 265-271.
DOI: https://doi.org/10.1016/B978-1-4832-1450-4.50029-8

[4] Zadeh, L.A., 1979. Fuzzy sets and information granularity. Advances in Fuzzy Set Theory and Applications. 11, 3-18.
DOI: https://doi.org/10.1142/9789814261302_0022

[5] Tian, Y., Kang, B., 2020. A modified method of generating Z-number based on OWA weights and maximum entropy. Soft Computing. 24(20), 15841-15852.
DOI: https://doi.org/10.1007/s00500-020-04914-8

[6] Bilgin, F., Alcı, M., 2021. Generating Z-number by logistic regression. 7th International Symposium on Electrical and Electronics Engineering (ISEEE). pp. 1-6.
DOI: https://doi.org/10.1109/ISEEE53383.2021.9628654

[7] Azadeh, A., Saberi, M., Atashbar, N.Z., et al., 2013. Z-AHP: A Z-number extension of fuzzy analytical hierarchy process. 2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST). pp. 141-147.
DOI: https://doi.org/10.1109/DEST.2013.6611344

[8] Khalif, K.M.N., Gegov, A., Abu Bakar, A.S., 2017. Z-TOPSIS approach for performance assessment using fuzzy similarity. 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). pp. 1-6.
DOI: https://doi.org/10.1109/FUZZ-IEEE.2017.8015458

[9] Abiyev, R.H., Akkaya, N., Gunsel, I., 2019. Control of omnidirectional robot using Z-Number-based fuzzy system. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 49(1), 238-252.
DOI: https://doi.org/10.1109/TSMC.2018.2834728

[10] Peide, L., Fei, T., 2015. An extended TODIM method for multiple attribute group decision making based on intuitionistic uncertain linguistic variables. 29(2), 701-711.
DOI: https://doi.org/10.3233/IFS-141441

[11] Aliev, R.A., Huseynov, O.H., Aliyev, R.R., et al., 2015. The arithmetic of Z-numbers: theory and applications. Singapore: World Scientific.

DOI: https://doi.org/10.1142/9575

[12] Patel, P., Khorosani, E.S., Rahimi, S., 2015. Modeling and implementation of Z-number. Soft Computing. 20(4), 1341-1364.
DOI: https://doi.org/10.1007/s00500-015-1591-y

[13] Shalabi, M.E., Hussieny, H., Abouelsoud, A.A., 2019. Control of automotive air-spring suspension system using Z-number based fuzzy system. IEEE International Conference on Robotics and Biomimetics (ROBIO). pp. 1306-1311.
DOI: https://doi.org/10.1109/ROBIO49542.2019.8961492

[14] Abdelwahab, M., Parque, V., Elbab, A.M.F., et al., 2020. Trajectory tracking of wheeled mobile robots using z-number based fuzzy logic. IEEE Access. 8, 18426-18441.
DOI: https://doi.org/10.1109/ACCESS.2020.2968421

[15] Jiang, W., Xie, C., Zhuang, M., et al., 2016. Sensor data fusion with Z-Numbers and its application in fault diagnosis. Sensors. 16(9), 1509.
DOI: https://doi.org/10.3390/s16091509

[16] Aliev, R.A., Pedrycz, W., Guirimov, B.G., et al., 2020. Acquisition of Z-number-valued clusters by using a new compound function, IEEE Transactions on Fuzzy Systems. 30(1), 279-286.
DOI: https://doi.org/10.1109/TFUZZ.2020.3037969

[17] Tian, Y., Mi, X., Cui, H., et al., 2021, Using Z-number to measure the reliability of new information fusion method and its application in pattern recognition. Applied Soft Computing. 111, 107658.
DOI: https://doi.org/10.1016/j.asoc.2021.107658

[18] Kang, B., Wei, D., Li, Y., et al., 2012. A method of converting Z-number to classical fuzzy number. Journal of Information & Computational Science. 9(3), 703-709.

[19] Banerjee, R., Pal, S., Pal, J.K., 2021. A decade of the Z-numbers, IEEE Transactions on Fuzzy Systems.
DOI: https://doi.org/10.1109/TFUZZ.2021.3094657

[20] K. Shen and J. Wang, 2018, Z-VIKOR method based on a new comprehensive weighted distance measure of Z-Number and its application, IEEE Transactions on Fuzzy Systems, 26(6), 3232-3245.
DOI: https://doi.org/10.1109/TFUZZ.2018.2816581

[21] Qiao, D., Shen, K., Wang, J., 2020. Multi-criteria PROMETHEE method based on possibility degree with Z-numbers under uncertain linguistic environment. Journal of Ambient Intelligence and Humanized Computing. 11, 2187-2201.
DOI: https://doi.org/10.1007/s12652-019-01251-z

[22] Mohamad, D., Shaharani, S.A., Kamis, N.H., 2017. Ordering of Z-numbers. AIP Conference Proceedings, AIP Publishing LLC. 1870(1).
DOI: https://doi.org/10.1063/1.4995881

[23] Bakar, A.S.A., Gegov, A., 2015. Multi-layer decision methodology for ranking Z-numbers. International Journal of Computational Intelligence Systems. 8(2), 395-406.
DOI: https://doi.org/10.1080/18756891.2015.1017371

[24] Ezadi, S., Allahviranloo, T., 2017. New multi-layer method for Z-number ranking using hyperbolic tangent function and convex combination. Intelligent Automation & Soft Computing. pp. 1-7.
DOI: https://doi.org/10.1080/10798587.2017.1367146

[25] Ezadi, S., Allahviranloo, T., Mohammadi, S., 2018. Two new methods for ranking of Z-numbers based on sigmoid function and sign method. International Journal of Intelligent Systems. 33(7), 1476-1487.
DOI: https://doi.org/10.1002/int.21987

[26] Jiang, W., Xie, Ch.H., Luo, Y., et al., 2017. Ranking Z-numbers with an improved ranking method for generalized fuzzy numbers. Journal of Intelligent & Fuzzy Systems. 32(3), 1931-1943.
DOI: https://doi.org/10.3233/JIFS-16139

[27] Chutia, R., 2021, Ranking of Z-numbers based on value and ambiguity at levels of decision making. International Journal of Intelligent Systems. 36(1), 313-331.
DOI: https://doi.org/10.1002/int.22301

[28] Li, Y.X., Pelusi, D., Deng, Y., et al., 2021, Relative entropy of Z-numbers. Information Sciences. pp. 1-17.
DOI: https://doi.org/10.1016/j.ins.2021.08.077

[29] Gong, Y., Li, X., Jiang, W., 2020. A new method for ranking discrete Z-number. 2020 Chinese Control and Decision Conference (CCDC). pp. 3591-3596.
DOI: https://doi.org/10.1109/CCDC49329.2020.9164654

[30] Basirzadeh, H., Farnam, M., Hakimi, E., 2012. An approach for ranking discrete fuzzy sets. Journal of Mathematics and Computer Science. 2(3), 584-592.
https://scik.org/index.php/jmcs/article/view/90

[31] Chen, M.S., Wang, S.W., 1999, Fuzzy clustering analysis for optimizing fuzzy membership functions. Fuzzy Sets and Systems. 103(2), 239-254.
DOI: https://doi.org/10.1016/S0165-0114(98)00224-3

BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Journal of Computer Science Research
https://ojs.bilpublishing.com/index.php/jcsr

ARTICLE

A Mathematical Theory of Big Data

Zhaohao Sun*

Department of Business Studies, Papua New Guinea University of Technology, Lae 411, Morobe, Papua New Guinea

ARTICLE INFO

Article history
Received: 21 April 2022
Accepted: 19 May 2022
Published Online: 31 May 2022

Keywords:
Big data
Big data analytics
Fuzzy logic
Similarity
Discrete mathematics

ABSTRACT

This article presents a cardinality approach to big data, a fuzzy logic-based approach to big data, a similarity-based approach to big data, and a logical approach to the marketing strategy of social networking services. All these together constitute a mathematical theory of big data. This article also examines databases with infinite attributes. The research results reveal that relativity and infinity are two characteristics of big data. The relativity of big data is based on the theory of fuzzy sets. The relativity of big data leads to the continuum from small data to big data, big data-driven small data analytics to become statistical significance. The infinity of big data is based on the calculus and cardinality theory. The infinity of big data leads to the infinite similarity of big data. The proposed theory in this article might facilitate the mathematical research and development of big data, big data analytics, big data computing, and data science with applications in intelligent business analytics and business intelligence.

1. Introduction

Big data has become one of the most important frontiers for innovation, research, and development in data science, computer science, artificial intelligence, industry, and business [1,2]. Big data has also become a strategic asset for nations, organizations, industries, enterprises, businesses, and individuals [3,4]. Big data technology including big data analytics has been successfully used to explore business insights and data intelligence from big data [2,5]. Mathematics researchers have paid increasing attention to the dramatic development of big data and its impact on mathematics by offering courses and holding workshops to develop the mathematics of big data [6,7]. However, there is no literature on a mathematical theory of big data based on the search using Google Scholar and Scopus (accessed on April 28, 2022). This indicates that a mathematical theory of big data has lagged far behind the big intelligence, big service, and big market opportunity resulting from big data [8]. The above brief analysis implies that the followings are still big issues for big data toward the establishment of a mathematical theory.

- What is a mathematical theory of big data?
- How does a social networking platform become an outstanding contributor to big data?

This article addresses these two issues by providing a mathematical theory of big data. This mathematical theo-

*Corresponding Author:
Zhaohao Sun,
Department of Business Studies, Papua New Guinea University of Technology, Lae 411, Morobe, Papua New Guinea;
Email: zhaohao.sun@pnguot.ac.pg; zhaohao.sun@gmail.com

DOI: https://doi.org/10.30564/jcsr.v4i2.4646

ry covers the cardinality approach, fuzzy logic approach, similarity approach, and logical approach due to the space limitation. More specifically, it proposes "big" as an operation and presents a cardinality approach to the big volume of big data, the latter is one of the ten big characteristics of big data [5], a fuzzy logic-based approach to big data, a similarity-based approach to big data, and a logical approach to the marketing strategy of big data-driven social networking services.

The remainder of this article is organized as follows. Section 2 presents a cardinality approach to big data. Section 3 looks at searching for big data using the set theory. Section 4 applies a fuzzy-logic approach to big data from a relativity perspective. Section 5 proposes a similarity-based approach to big data. Section 6 looks at a logical approach for promoting big data-driven social networking services. The final sections discuss the related work and end this article with some concluding remarks and future work.

It should be noted that this article does not directly address the issues related to how to efficiently manage, analyze, mine, and process big data although the proposed mathematical theory can be applied to each of them.

2. A Cardinality Approach to Big Data

This section examines cardinality of big data as a mathematical approach to the big volume of data based on discrete mathematics and cardinal number theory in real analysis. It addresses what the biggest volume of big data is.

Definition 1

Let S be a set. The cardinality of S is m, denoted by $|S| = m$, if there are exactly m distinct elements in S, where m is a non-negative integer [9].

Example 1.

Let $S = \{a, b, c, d, e, f, g\}$, then $|S| = 7$.

S is said to be finite if m is a non-negative integer. Otherwise, S is said to be infinite. The cardinality of an infinite set S is discussed in set theory and discrete mathematics [10-12].

A large number of research articles on big data have been published in the past decade since 2012 [8,13-16]. Most of them consider volume as the first V of big data [8]. They mentioned terabytes (TB, 2^{40} B), petabytes (PB, 2^{50} B), exabytes (EB, 2^{60} B)[16], zettabytes (ZB, 2^{70} B), and yottabyte (YB, 2^{80} B) as the big volume of big data. Google, YouTube, and other global data giants have the volume of big data at a PB level yearly, while Amazon has big data at an EB level yearly [17]. It seems that the principle of big data is that the bigger volume the big data has, the more important it is. This is true in some cases, for example, Google and Amazon with a bigger volume of big data have a big value in the market.

However, when the author asked his friend's child to count numbers, 1, 2, 3, ..., then only a few minutes later, he could not like to count numbers anymore. Then he said "infinity" as the conclusion of his counting number. What an interesting child he is! He intuitively knew the end of counting numbers is infinity. The generalization of this story is the cardinality of big data.

In entity-relationship modeling, an attribute value is the least unit for representing data [18]. An attribute value has also been the least unit for defining and manipulating data in database systems [19]. An attribute value is still the least unit for NoSQL database or web data processing. Therefore, the attribute value can be used as the least unit of big data. An attribute value can be denoted as v. For example, $v_1 = big$, $v_2 = data$, $v_3 = analytics$, $v_4 = intelligence$, $v_5 = service$, etc. These values can be considered as keywords when searching online and or stop words in natural language processing. From a linguistic viewpoint, they are the elements for constructing a sentence, a paragraph, a text, and so on. However, for searching, some attribute values, for example, *a*, *an*, *the*, *of*, can be considered negligible components. An attribute value can be any word(s) (e.g., in English) in the web text. Using number sequence and limit in mathematics [20], an attribute value sequence is $v_1, v_2, \cdots, v_n, \cdots$, and n is an integer. Now a question arises,

What does the limit of attribute value sequence v_i mean when i tend to infinity?

Let V be a set of attribute values, and U be the universe of big data. U consists of all online and offline data available to mankind. Therefore, U includes all the data, information, knowledge, experience, intelligence, and wisdom in either article or website, or multimedia form [9]. Then, the relationship between V and U is as follows.

$$V \subseteq U \tag{1}$$

In other words, V is a subset of U.

A finite attribute value sequence $v_1, v_2, \cdots, v_n$ can be used to constitute a sentence using concatenation, where n is an integer. However, from a perspective of human cognition [21], n is not a fixed integer. The corresponding attribute value sequence $v_1, v_2, \cdots, v_n$ cannot represent all the data and knowledge existing in the world. At least a countable infinite number of attribute values is required to constitute all the big data, big information, big knowledge, big intelligence, and big wisdom [22], because the number of English sentences is theoretically infinite [23]. This implies that the cardinality of V is $|V| = \aleph_0$. $\aleph_0$ is the cardinality of all integers N, then U should be at least uncountable infinity as the cardinality of all the real numbers [24], because any subset of V can be constituted to a meaningful sentence, paragraph, or text. It can also correspond to

at least a picture or a set of pictures, such as a data stream. For example, if one searches "Paul" (as a v_1), then one will find a set of texts or pictures consisting of "Paul", even using Google (see the next section). This means that an element of V corresponds to a set of elements of U. Therefore, a relationship between the cardinality of V and that of U is

$$|U| = 2^{|V|} \tag{2}$$

and

$$|U| = 2^{|V|} = 2^{\aleph_0} = c \tag{3}$$

where c is the cardinality of the real number set. Strategically, if one likes to understand the existing finite world of big data, one should "live" in the infinite world of big data. It is important to be a follower in the finite world of big data in terms of EB, ZB, and YB. It is also important to be an explorer in the infinite world of big data. For the former, we enjoy the 3G communication using a mobile phone in the 2000s, whereas for the latter one should look at what will be the next generation of smartphones using 6G or 7G communication.

It should be noted that this section is motivated by a large number of articles or books using petabytes, exabytes, and zettabytes, as well as yottabytes, to describe how big the volume of big data [17,25,26]. Therefore, it is interesting to answer how big the volume of big data is in the future. The cardinality theory [12,24] is used to develop it in some detail. In other words, this section provides an answer to the question: how big is the volume of big data eventually, using real analysis or measure theory.

3. A Set Theory for Searching Big Data

This subsection discusses searching for big data using the set theory, which is the foundation of modern mathematics [12,24].

Let $u \in U$ be a document on the web. u may be a Microsoft word file in .docx or report in pdf. Let $v \in V$ be an attribute value. v may be a word such as "data", or "intelligence", or "wisdom", or "engineering", then a search function denoted as $s: V \to U$, is defined as

$$s(v) = u, \text{ if } v \in u \tag{4}$$

For example, if one uses Google Scholar to search for "big data", denoted as v, then she or he finds 1,750,000 results (retrieved on April 26, 2022), denoted as u, each of them should include v.

More generally, a search function can be defined as

$$s(v) = F(v) \tag{5}$$

where $F(v) = \{u_i | \ v \in u_i, u_i \in U, i = \{0,1,2,\cdots,m\}\}$; $i = 0$ means that "no research results" for v. This is valid for searching practice using all the search engines online including Google, Baidu, Semantic Scholar, and Google Scholar. The core idea behind the online search is that one keyword search corresponds to at least a picture/document or a set of pictures/documents as the search results. A Google search for "big data" found 61,700,000 results (retrieved on 22 April 2018) and 398,000,000 results (retrieved on 20 April 2022). Therefore, $F(v) \subseteq U$.

Searching on the web using search engines such as Google and Baidu is an operation. Search or query in a relational database using SQL (Structured Query Language) is a data operation (data manipulation) [19]. SQL should be renamed as Structured Query Engine (SQE) based on the usage of the search engine. At least the author discussed it with his colleagues. In what follows, this section looks at the property of the search function as an operation.

Let $v_1, v_2, v_3 \in V$, using Equation (5), $s(v_1) = F(v_1)$, $s(v_2) = F(v_2)$, $s(v_3) = F(v_3)$. Then the following property of search functions holds [10].

$$s(v_1 \vee v_2) = s(v_1) \cap s(v_2) = F(v_1) \cap F(v_2) \tag{6}$$

where $\vee$ is a space operation between v_1 and v_2 to reflect the search using Google and Baidu. $\vee$ as an operator has the property of association, that is, $v_1 \vee (v_2 \vee v_3) = (v_1 \vee v_2) \vee v_3$ [27]. $\vee$ is similar to concatenation between data items in linguistics or formal language [11].

Now given an attribute value sequence $v_1, v_2, \cdots, v_n, \cdots$, then $s(v_1) = F(v_1)$, $s(v_2) = F(v_2), \ldots$, $s(v_n) = F(v_n)$, ...

Theorem 1.

In the finite world of big data, the search result with respect to operation $\vee$ is

$$s(v_1 \vee v_2 \vee \cdots \vee v_n) = \prod_1^n F(v_i) = F(v_1) \cap F(v_2) \cap \cdots \cap F(v_n) \tag{7}$$

Theorem 2.

When $n \to \infty$, the following property of search as an operation $\vee$ in the web search holds.

$$s(v_1 \vee v_2 \vee \cdots \vee v_n \vee \cdots) = \prod_1^\infty F(v_i) \tag{8}$$

Theorem 1 and Theorem 2 can be proved based on Equation (6) easily.

Equation (7) and Equation (8) are representation theorems for searching in the finite world and infinite world of big data respectively.

Many people have the experience of searching for what they expect on the web using Equations (6 and 7), although each of them has not experienced the search of the web based on the Equation (8). This is because an individual's search on the web is finite (in terms of attribute value) whereas all the human being's searches on the web should be infinite.

Based on the dual principle of the set theory, the $\vee$ operation of $v_1 \vee v_2$ motivates us to introduce $\wedge$ operation [27].

Let's look at the following example: Paul just searched for "intelligent big data analytics" using Google Scholar. However, he had not searched for what he expected, so he had to extend his search space by using "big data analytics". Let "intelligent big data analytics" be v_1 and "big data analytics" be v_2. Then Paul's extending search space means that he uses $v_1 \wedge v_2$ (a kind of intersection in set theory [24]) to search for what he expected on the web. Then the following two theorems hold corresponding to Equations (7) and (8), based on the dual principle of set theory [9].

Theorem 3.

In the finite world of big data, the search results with respect to operation $\wedge$ are

$$s(v_1 \wedge v_2 \wedge \cdots \wedge v_n) = \coprod_1^n F(v_i) = F(v_1) \cup F(v_2) \cup \cdots \cup F(v_n) \quad (9)$$

Theorem 4.

When $n \to \infty$, the following property of search as an operation $\wedge$ in the web search holds.

$$s(v_1 \wedge v_2 \wedge \cdots \wedge v_n \wedge \cdots) = \coprod_1^{\infty} F(v_i) \quad (10)$$

Equations (7), (8), (9), and (10) are a mathematical basis for searching for big data on the web.

4. A Fuzzy Logic Approach to Big Data

Fuzzy sets theory has been successfully applied in many areas including finance, database, pattern recognition, and natural language processing, to name a few [28,29]. Fuzzy logic can be applied to address the vagueness and veracity of big data [14,29,30]. This section uses fuzzy sets and fuzzy logic to examine the relativity of big data as a fundamental of big data.

Let U be the universe of big data, and $n \in N$. Then big as an attribute value is an element of U, that is, {big} $\in U$.

A fuzzy set of big in N is defined with a membership (characteristic) function $f_{big}(n)$ which associates every number of $n \in N$ with a real number in the interval [0,1] [30], that is,

$$f_{big}(n) \in [0,1]$$

If $f_{big}(n) = 1$, $n \in N$ is said to be big, otherwise, $f_{big}(x) < 1$, $n \in N$ is said to be not big. In fuzzy logic, "not big" does not mean "small" [28].

A question arises: What big does mean in big data from a perspective of fuzzy logic? To answer this question, this subsection examines an average child at 5 years old, a young person at 20 years old, and a graduate with a Bachelor of Data Science, and observes what they believe big as a term [9].

For the child, he believes that 5,000 is big, denoted as $f_{big}(n, p_1) \in [0,1]$, because he likes to have US$ 5,000, then

$$f_{big}(n, p_1) = \begin{cases} 1, & if\ n \geq 5{,}000 \\ < 1, & otherwise \end{cases}$$

This equation indicates that the child believes that any number greater than 5,000 will be "big", whereas number less than 5,000 is not big.

For the young person, he believes that 1 million is big, denoted as $f_{big}(n, p_2) \in [0,1]$, because he likes to be a millionaire, that is,

$$f_{big}(n, p_2) = \begin{cases} 1, & if\ n \geq 1{,}000{,}000 \\ < 1, & otherwise \end{cases}$$

This means that the young person believes that any number greater than 1,000,000 will be "big", whereas number less than 1,000,000 is not big.

For the graduate with the degree, he believes that 1 billion is big, denoted as $f_{big}(n, p_3) \in [0,1]$, because he likes to be a billionaire, that is,

$$f_{big}(n, p_3) = \begin{cases} 1, & if\ n \geq 10^9 \\ < 1, & otherwise \end{cases}$$

In other words, he believes that any number greater than 10^9 will be "big", whereas number less than 10^9 is not big.

The above analysis using fuzzy sets indicates that all persons unanimously agree that a number less than 5,000 is "not big", and different people have a different understanding of big as the term. However, all people unanimously have a concept of "big" in numbers motivated by their backgrounds, environments, and expectations.

Let $P = \{p_k | p_k\ is\ a\ person, k = 1, 2 \cdots, m\}$. $m \in N$ is a given natural number, it can be the total number of all the people living in the world. For every person, p_k, his or her perspective to big can be represented as a fuzzy set Bp_k with the following membership function $f_{big}(n, p_k) \in [0,1]$ and

$$f_{big}(n, p_k) = \begin{cases} 1, if\ n \geq n_k \\ < 1, otherwise \end{cases} \quad (11)$$

For example, the above child can be named a p_1, then the perspective of p_1 to "big" satisfies the following properties: $n_1 = 5{,}000$.

$$f_{big}(n, p_1) = \begin{cases} 1, if\ n \geq n_1 \\ < 1, otherwise \end{cases}$$

Based on the operations of fuzzy sets [28,30], the intersection of the all fuzzy sets Bp_k ,$k = 1, 2 \cdots, m$ with membership functions like Equation (12) is a fuzzy set C, written as $C = Bp_1 \cap Bp_2 \cap \cdots \cap Bp_M$, whose membership function $f_{big}(n, C)$ is

$$f_{big}(x, C) = Min(f_{big}(n, p_1), f_{big}(n, p_2), \cdots, f_{big}(n, p_m)) \quad (12)$$

Or, in abbreviated form

$$f_{big}(n, C) = \wedge_1^m (f_{big}(n, p_k)) \quad (13)$$

The membership function $f_{big}(n, C)$ indicates that there exists a number $K \in N$ such that for every $p_k, k = 1, 2 \cdots, m$.

$$f_{big}(n, p_k) = \begin{cases} 1, & when\ n \geq K \\ 0, & otherwise \end{cases} \quad (14)$$

where $K = \max(n_1, n_2, \cdots, n_m)$. In other words, all the people have unanimously agreed that *K* or greater is big. In terms of big data, this means that all the people have unanimously agreed that the data with *K*Bytes or greater is big data. This result conforms to the currently popular idea in the literature of big data, that is, data with Exabytes or Zettabytes are big data [13,31], whereas data with an MB cannot be considered big data anymore, although it used to be big data two decades ago [9].

The union of the all-fuzzy sets $Bp_k, k = 1, 2 \cdots, m$ with membership functions in Equation (11) is a fuzzy set *D*, written as $D = Bp_1 \cup Bp_2 \cup \cdots \cup Bp_m$, whose membership function $f_{big}(x, D)$ is

$$f_{big}(n, D) = Max(f_{big}(n, p_1), f_{big}(n, p_2), \cdots, f_{big}(n, p_m)) \quad (15)$$

Or, in abbreviated form [30]

$$f_{big}(n, D) = \bigvee_1^m (f_{big}(n, p_k)) \quad (16)$$

The membership function $f_{big}(n, D)$ indicates that there exists a number $H \in N$ such that for every $p_k, k = 1, 2 \cdots, m$.

$$f_{big}(n, p_k) = \begin{cases} 1, & when\ n \geq H \\ 0, & otherwise \end{cases} \quad (17)$$

where $H = \min(n_1, n_2, \cdots, n_m)$. Different from the above analysis based on the intersections of fuzzy sets, the union of fuzzy sets [28] indicates that for any given $J \in N$, if there exists a person who believes that J is big, that is, $J \geq H$, then all the other persons have to agree that J or greater is big. In terms of big data, this means that if there exists a person who believes that data with $Jbytes$ is big data, then all the other persons have to agree that the data with $Jbytes$ or greater is big data. If this is true, then J might be $100 \in N$, because a child might consider 100 as a big number. In other words, the statement that the data with Exabytes or Zettabytes is just big lacks evidence from the social reality based on the theory of fuzzy logic.

The above discussion implies that one characteristic of big data is relativity, that is, big is a relativity concept; the relativity of big data is a fundamental of big data. The secret behind the relativity of big data is inclusiveness, that is, we have to permit that every individual has his or her understanding of what big means in big data [9]. This inclusiveness can make big more powerful in terms of research and development of big data. This relativity of big data also brings forth that a universal benchmark does not exist for big volume, big variety, big velocity, and big veracity that define and measure the characteristics of big data [14].

Finally, the relativity of big data leads to the continuum from small data to big data, big data-driven small data analytics is of statistical significance [14].

5. A Similarity-based Approach to Big Data

The concept of similarity is a fundamental concept in mathematics, computer science, data science, and artificial intelligence [2,10]. For example, "similar problems have similar solutions" is the principle of case-based reasoning [32]. In other words, case-based reasoning is a process of discovering similarity intelligence from a case base, just as data mining is a process of discovering data intelligence from a database or big data [22,26].

Definition 2.

A binary relation *S* on a non-empty set *X* is a similarity relation if it satisfies [32]

(R) $\forall x, xSx$;

(S) If xSy, then ySx;

(T) If xSy and ySz, then xSz.

The conditions (R), (S), and (T) are the reflexive, symmetric, and transitive laws. If xSy or $< x, y > \in S$ then x and y are said to be similar, denoted as $x \approx y$.

Based on this definition, this section introduces finite similarity, infinite similarity, weak similarity, strong similarity, and limit of similarity. All these concepts of similarity are important for similarity-based reasoning for big data to discover similarity intelligence and data intelligence from big data.

5.1 Finite Similarity and Infinite Similarity

From a viewpoint of search online, a search engine cannot limit the number of text words. Different texts have different lengths in words. Therefore, the search space is infinite and consists of infinite texts or words.

As mentioned earlier, if one searches for "*v*" using a search engine (e.g. Google and Baidu), then all the search results are $s(v) = F(v)$. *F(v)* can be considered as a similarity class of *v,* denoted as [*v*], that is, [*v*] = {*y* | *y* is a search result from searching for "*v*" online}. In other words, if $y_1, y_2 \in [v]$ then y_1 and y_2 are similar, denoted as $y_1 \approx y_2$. y_1 and y_2 are the search results from searching for "*v*" online. This example implies that two entities are similar *iff* they have something in common (here, they have the same *v*).

More generally, given an attribute value sequence $v_1, v_2, \cdots, v_n, \cdots$, then searching for each of them online, then $s(v_1) = [v_1] = F(v_1)$, $s(v_2) = [v_2] = F(v_2), \ldots, s(v_n) = [v_n] = F(v_n), \ldots$. If for any given $i \in \{1, 2, 3, \ldots, n, \cdots\}$ such that $v_i \approx v_{i+1}$, then $F(v_i) \cong F(v_{i+1})$. $y_1, y_2 \in F(v_i)$ are said to be infinite similar when the given *i* is sufficiently greater in *N*.

Infinite similarity originates from the experience of searching online using Google and our early work on similarity [32], when one searches for things on the web for many times and many years, s/he find that there is a kind of similarity among what searched appearing. This kind of similarity is infinitely similar.

Remark: This infinite similarity also reflects bounded rationality: individuals make decisions, their rationality is bounded by the available information, the tractability of the decision problem, the cognitive limitations of their minds, the time, environment, and technical conditions available to make the decision [4,33].

5.2 Weak and Strong Similarity

Given that sets $A_1, A_2, \cdots, A_n$ are an attribute sequence, where n is an integer. R is a relational database schema on these sets, denoted as $R(A_1, A_2 \cdots, A_n)$. Any instance of $R(A_1, A_2 \cdots, A_n)$ is a relation [19], that is, assume that $r \in R$ is a relation, there are n attributes $a_1, a_2 \cdots, a_n$ associated with r, that is, $r(a_1, a_2 \cdots, a_n)$ is an instance of $R(A_1, A_2 \cdots, A_n)$. A relation represents a fact, a piece of information, or a piece of knowledge. When $n = 0$, r is a fact. When $n = 1$, $r(a_1)$ represents a fact with an attribute a_1. For example, r(a) can represent "Paul is tall". r denotes " x is tall", a = Paul.

In a relational database [19,36],assume that the number of attributes $A_1, A_2 \cdots, A_n$ in a relation R is finite. For example, there are more than 300 attributes used for matching the love relationship between individuals on a matching website. In some cases, n > 1,000, in order to represent a piece of knowledge or information.

Now assume $R_1(A_{11}, A_{12} \cdots, A_{1N})$ and $R_2(A_{21}, A_{22} \cdots, A_{2N})$ are two relational schemas, where N is an integer.

Definition 3.

R_1 and R_2 are said to be similar with respect to relational schema, denoted as $R_1 \approx R_2$, iff there exists an integer $0 < K_1 \leq N$ so that for any $i \leq K_1$, $A_{1i} = A_{2i}$. In this case, we call R_1 and R_2 are similar with respect to a relational schema.

This concept of similarity is at a relational database schema level. It is useful for designing a relational database [19].

Definition 4.

In a relational database, assume that $r(a_1, a_2 \cdots, a_N)$ is a relation, an instance of $R(A_1, A_2 \cdots, A_N)$. $r(v_{i1}, v_{i2} \cdots, v_{iN})$ and $r(v_{j1}, v_{j2} \cdots, v_{jN})$ are row i and row j of r. then $r(v_{i1}, v_{i2} \cdots, v_{iN})$ and $r(v_{i1}, v_{i2} \cdots, v_{iN})$ are said to be K-similar, denoted as $r(v_{i1}, v_{i2} \cdots, v_{iN} \overset{K}{\approx} r(v_{j1}, v_{j2} \cdots, v_{jN})$ *iff* for a given K, $0 < K \leq N$, such that $v_{ik} = v_{jk}$, where $k = 1,2, \cdots, K$.

K-similarity is called as a weak similarity. It is weak because it is a kind of similarity at the record (row) level. This similarity is related to the data redundancy at the attribute level, and the record level, and therefore it is of practical significance in database management systems [19].

Example 2.

r(Sex, Program, ID, Name, Age) is a relational database, illustrated in the Table 1.

Table 1. A student relational database.

Sex	Program	ID	Name	Age
M	IT	160001	John	18
M	IT	160002	Peter	19
M	IT	160003	Lee	20
F	IT	160004	Liz	19
F	IT	160005	Lana	18
F	IT	160006	Bessie	19
F	IT	160007	Grace	20

Program oriented relation P = {<john, john>, <Peter, Peter>, <Lee, Lee> <John, Peter>, <Peter, John>, <John, Lee>, <Lee, John>, <Peter, Lee>, <Lee, Peter>; <Lana, Lana>, <Bessie, Bessie>, <Grace, Grace>, <Lana, Bessie>, <Bessie, Lana>, <Lana, Grace>, <Grace, Lana>, <Bessie, Grace>, <Grace, Bessie>}. Then P is a 2-similar relation. P partitions the above table into two similarity classes, [John] = {all the male IT students} and [Lana] = {all the female IT students}. This example implies that two male undergraduate students studying IT program at a university are similar.

Theorem 5.

K-similarity relation is a similarity relation.

Prove. To prove this theorem, it only needs to prove K-similarity relation is reflexive, symmetric, and transitive, based on Definition 2.

Assume that $r(v_{i1}, v_{i2} \cdots, v_{iN})$, $r(v_{j1}, v_{j2} \cdots, v_{jN})$ and $r(v_{h1}, v_{h2} \cdots, v_{hN})$ are row i, row j and row h of r, where $i, j, j\ h \in \{1, 2, ..., N\}$ .K is an integer, $0 < K \leq N$.

1) For $k \in \{1, 2, ..., K\}$, $v_{ik} = v_{ik}$, then $r(v_{i1}, v_{i2} \cdots, v_{iN}) \overset{K}{\approx} r(v_{i1}, v_{i2} \cdots, v_{iN})$, This means that K-similarity relation is reflexive.

2) If $r(v_{i1}, v_{i2} \cdots, v_{iN}) \overset{K}{\approx} r(v_{j1}, v_{j2}$, then for any k, $1 \leq k \leq K$, $v_{ik} = v_{jk}$ or $v_{jk} = v_{ik}$. Then $r(v_{j1}, v_{j2} \cdots, v_{jN}) \overset{K}{\approx} r(v_{i1}, v_{i2} \cdots, v_{iN})$ holds. That is, K-similarity relation is symmetric.

3) If $r(v_{i1}, v_{i2} \cdots, v_{iN}) \overset{K}{\approx} r(v_{j1}, v_{j2} \cdots, v_{jN})$ and $r(v_{j1}, v_{j2} \cdots, v_{jN}) \overset{K}{\approx} r(v_{h1}, v_{h2} \cdots, v_{hN})$, then for any $1 \leq k \leq K$, $v_{ik} = v_{jk}$, and $v_{jk} = v_{hk}$, that is, $v_{ik} = v_{hk}$, then $r(v_{i1}, v_{i2} \cdots, v_{iN}) \overset{K}{\approx} r(v_{h1}, v_{h2} \cdots, v_{hN})$ holds. Therefore, K-similarity relation is transitive.

This theorem demonstrates that weak similarity is a similarity relation.

When $K = N$, then a K-similarity relation is said to be strongly similar. It is easy to prove.

Theorem 6.

If $r(v_{i1}, v_{i2} \cdots, v_{iN})$ and $r(v_{j1}, v_{j2} \cdots, v_{jN})$ are strongly similar, then $r(v_{i1}, v_{i2} \cdots, v_{iN})$ and $r(v_{j1}, v_{j2} \cdots, v_{jN})$ are K-similar,

where $0 < K \leq N$.

Remark: Weak and strong similarities can facilitate saving, updating, and deleting data in a relational database. For example, if $r(v_{i1}, v_{i2}\cdots, v_{iN})$ and $r(v_{j1}, v_{j2}\cdots, v_{jN})$ are strongly similar, then one of the two rows (records) is redundant, and should be deleted. Furthermore, weak similarity can be considered a kind of partial similarity with respect to a row, whereas strong similarity is a kind of the whole similarity with respect to a row.

5.3 Database with Infinite Attributes and Similarity-based Reasoning for Big Data

This subsection applies the concept of limit to explore searching for big data based on the limit of number sequence in calculus [34]. It also delves into similarity-based reasoning for big data. More specifically, big data can be classified into two categories: One is database-based data, and another is NoSQL data [19]. Then this subsection discusses similarity-based reasoning for database-based big data and NoSQL big data. They are the basis for non-computation-based similarity, similarity-based infinite reasoning.

NoSQL databases such as Google's BigTable and Apache's Cassandra use the key-value data model or attribute-value model [19]. The attribute-value model consists of two data elements: an attribute and a value, in which every attribute has a corresponding value or a set of values. This can be considered as a simplified table different from the traditional tables that underpin the relational database. The simplified table in the NoSQL database consists of only three columns: AttributeID, Attribute, and Attribute value. For short, it is (AID, A, V), where AID is the ID of attribute $a \in A$, A is an attribute set, V is an attribute value set. The relationship between an attribute $a \in A$ and attribute value is 1:M, that is, an attribute $a \in A$ corresponds to a number of attribute values $v \in V$. Generally, N is the set of natural numbers. For any $i \in N, N = \{1, 2, 3, ..., n, ...\}$, $a_i \in A$ is an attribute, its attribute value is $ai(vj)$. When the cardinality of A equals to that of N, that is, $|A| = \aleph_0$, then (AID, A, V) is a relational database with infinite attributes.

A relational database with infinite attributes can be also defined as follows: let R be a relation. Its sequence of attributes is $A_1, A_2\cdots, A_n\cdots$, where n is an integer, $n \in N$. When n trends to infinity, $R(A_1, A_2\cdots, A_n\cdots,)$ is a relational database schema with infinite attributes. A relational database $r \in R(A_1, A_2 \cdots, A_n \cdots,)$. $r(a_1, a_2, \cdots, a_n, \cdots)$, is a relational database with infinite attributes *iff* it has a countable infinite attribute sequence $a_i \in A$.

Definition 5.

For any given integer K, if R_1 and R_2 are always similar with respect to relational schemas based on Definition 1, then we call R_1 and R_2 are infinitely similar with respect to a relational schema.

Definition 6.

Assume that $r(v_{i1}, v_{i2}\cdots, v_{in},\cdots,)$ and $r(v_{j1}, v_{j2}\cdots, v_{jn},\cdots,)$ are row i and row j of a relational database with infinite attributes, $r(a_1, a_2\cdots, a_n,\cdots)$. Then $r(v_{i1}, v_{i2}\cdots, v_{in},\cdots)$ and $r(v_{j1}, v_{j2}\cdots, v_{jn},\cdots)$ are said to be infinitely similar *iff* for any significantly big integer $K \in N$, $a_{ik} = a_{jk}$, where $k = 1, 2,\cdots, K$.

Theorem 7.

If $r(v_{i1}, v_{i2}\cdots, v_{in},\cdots)$ and $r(v_{j1}, v_{j2}\cdots, v_{jn},\cdots)$ are infinitely similar. Then they are K_1-similar for any $K_1 \leq K$.

From a practical viewpoint, only a few dozens of attributes or a few hundreds of attributes are not enough for characterizing an entity in the age of big data. This is the reason why we introduce a relational database with infinite attributes. The finite similarity in a relational database with infinite attributes paves the way from finite similarity to infinite similarity. This is useful for searching for big data and similarity-based search for a large database with infinite attributes. This is also useful for the development of human recognition because the practice in the finite world can be used to understand the infinite similarity in the infinite world.

6. A Logical Approach for Making Social Networking Services Big

Online social networking (OSN) services generate a big volume of big data. For example, YouTube generates 263 PB of big data yearly [17]. This section presents a logical approach to making social networking services (OSN) big as a part of applying mathematics to big data.

An OSN like Meta (Facebook) launched an application "people you may know" to directly (online) acquire email addresses based on your registered email address. The principle of this acquisition is illustrated in Figure 1.

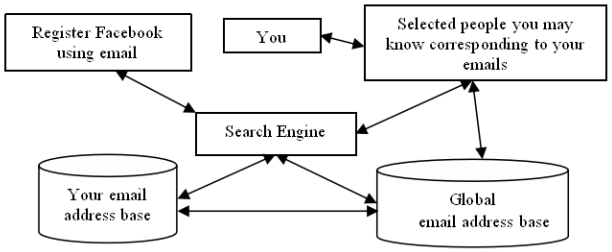

Figure 1. An intelligent technique of directly (online) acquiring email addresses based on a registered email address

As soon as one registered as a user of an OSN such as Meta (Facebook), WeChat, and LinkedIn, all of the email addresses in her or his email address base have been au-

tomatically exposed to the OSN. The OSN can automatically and regularly visit the registered email box, scan all the emails, extract information of email addresses that one has used, received, and sent, and then collect all of the email addresses away and store them in the Global email address base, as shown in Figure 1. Then the OSN can use any of the email addresses to contact the "friends" that s/he has used email to communicate with, the names have been in the email address base.

Friending is a marketing strategy of the OSN like Facebook and WeChat. It does not care if it is private to you. The friending mechanism of Facebook automatically invites your "friends" to join Facebook using "Selected people you may know". All these illustrated in Figure 1 are automatically realized using intelligent agents [2]. This is the reason why the leader of OSN or OSN services advises that you need not care about your privacy. If everyone opposes the invasion of his or her email address base for other purposes at the early time of Facebook, then Facebook would be disastrous. However, the social norm is just something that has evolved over time. One enjoys the services provided by OSN like Facebook and WeChat as well as TikTok, s/he has to sacrifice some privacy.

Assume that your email address base is E, F is your correspondence name base with respect to email communications, that is, for any name $f \in F$, there at least exists an email address $e \in E$, e is the email address of f. This means that any person that contacted you using email is your friend from a viewpoint of an OSN like Facebook, his or her email address is in E, and his or her name is in F. Now we have a virtual friendship, or email-based friendship as a binary relation, denoted as eF. eF is similar, because 1) You can email yourself. Then, eF is reflective. 2) If you can email anyone in E, then he or she can email you, then eF is symmetric. 3) If you can email your friend x, and your friend x can email his or her friend y, then you can email friend y, then eF is transitive.

The symmetry of eF makes everyone share the information in a symmetrical way. The transitivity of eF can make an OSN like Facebook market its services and acquire new customers, new Facebook friends. This is why an OSN can become globally popular within a short time.

A marketing strategy aims to acquire new customers, select customers, extend customers and retain customers profitably [4]. Now a logical approach to the marketing strategy of the OSN is presented below, based on Figure 1. Let : $P(f)$: $f \in F$ be a person. $N(e)$: e is an email address.

1) $P(f_0)$ (f_0 has registered as a member of OSN like Facebook)

2) $P(f_0) \rightarrow N(e_0)$ (f_0 submitted the email address e_0 to OSN)

3) $N(e_0)$ (The email address has been saved to OSN's global email address base) (1, 2) modus ponens

4) $\forall e(N(e_0) \rightarrow N(e))$ (Your contacted email address)

5) $N(e_0) \rightarrow N(e)$ (Remove the qualifier)

6) $N(e)$ (3, 5) (modus ponens)

7) $\forall e \exists f(N(e) \rightarrow P(f))$ (For any given email address, it corresponds to a person f)

8) $N(e) \rightarrow P(f)$ (Remove the qualifiers)

9) $P(f)$ (6, 8) (modus ponens)

10) $\exists f P(f)$ (Add the qualifier)

Then the OSN saves the information of $P(f)$ and tells you, "You may know $P(f)$". This logical approach implies that if f_0 is an OSN user, then the OSN can contact and attract all persons $P(f)$, $f \in F$, to become the *OSN* users. For example, if an average individual has 100 correspondence names. Then the OSN uses "You may know $P(f)$" to contact and attract all the corresponded persons five times one after another, exponentially, and then can attract $(100)^5 = (10)^{10}$ persons to become its users. Therefore, this automatic marketing approach brings about a bursting (exponential) increase of the OSN users just as Facebook has done in the past decade.

It should be noted that ResearchGate (https://www.researchgate.net/) has also used the technology based on the above-mentioned principle and logical approach. However, WeChat (www.wechat.com) have not mastered such a technology to attract its registered users to self-willingly expose their own privacy after submitting their own email address to WeChat. They still use traditional viral marketing for promoting their business. Viral marketing is based on a fact that a WeChat user invites his or her friends to use WeChat so that there are over 1 billion monthly active users in the WeChat world.

7. Related Work and Discussion

A number of scholarly research publications on big data have been mentioned in the previous sections. This section will focus on related work and discussion on a mathematical theory of big data.

Shannon's landmark article titled "A mathematical theory of communication" [35], provides the basis for information theory and has facilitated the lasting development of information science and technology since then. However, no articles titled a mathematical theory of big data have appeared so far. This inspires us to develop this article, which is an endeavor in this direction. This is also an extension and generalization of our early work from a mathematical foundation to a mathematical theory [9].

Google searches for "mathematics of big data" found about 32,100 results (27 November 2016, when this section was first written) and about 95,500 results (on April

20, 2022, when this section is updated). These results include courses offered by universities, workshops, and presentations on the mathematics of big data or data science. This means that mathematicians have paid attention to the dramatic development of big data and attempted to provide a mathematical approach to big data. For example, Laval has been developing a course on the mathematics of big data since 2015 [6,36]. The course provides students with mathematical techniques used to acquire, analyze, and visualize big data (e.g., using MATLAB) [37]. The workshop on mathematics in data science was held in 2015 in the USA [38]. Its objective is to explore the role of the mathematical sciences in big data as a discipline. Peter delivered a presentation on mathematics in data science at ICERM [7,41].

A Google scholar (www.scholar.google.com.au) searched for "mathematics of big data" in November 2016, when this section was first written, there were no searched article titles or book titles including "mathematics of big data". A Google scholar search for "mathematics of big data" was conducted on April 25, 2018 to update this research, and found that there were only 22 search results. A Google scholar search for "mathematics of big data" found 82 results on April 20, 2022. Four search results out of 22 are particularly worth discussing here. They are 1) Introduction to the Mathematics of Big Data [39,42]. 2) A Mathematical Foundation of Big Data [9]. 3) A Book on Applied Mathematics [40,43]. 4) The recently published book on mathematics of big data [44].

The first is a course description for the course with the same name. This course has been offered since 2015 [37,39,42]. It gives a short overview of big data and discusses the issues associated with big data with some answers.

The second [9] examines big as an operation, the cardinality of big data, and explores a mathematical approach to searching big data. However, the work of Sun and Wang [9] lacks logical approach and other mathematical approaches that are necessary for developing a mathematical theory of big data. This article updates and generalizes some of its results, and further explores infinite similarity and logical approach to online social networking platforms.

The third states that the mathematics of big data can provide theories, methods, and algorithms for processing, transmitting, receiving, understanding, and visualizing datasets [40,43].

The last is a book focusing on applications and practices of spreadsheets, databases, matrices, linear algebra, and graphs and for processing big data [44].

Big data is a market-inspired brand and research field. It seems to lack rigorous research from a perspective of mathematics. This is similar to social computing which "benefits from mathematical foundations, but research has barely scratched the surface" [45]. The above discussion implies that there is still a long way to go to develop the mathematics of big data as a discipline. This article provides an attempt to explore a mathematical theory for big data based on the work of Sun and Wang [9] and motivated by C. E. Shannon [35]. More theoretical work will be undertaken to develop a mathematical theory of big data and big data analytics.

Fuzzy sets and fuzzy logic [29,32] have been used to explore the relativity of big data and showed that one should have inclusiveness in exploring big data so that everyone can get benefit from the research and development of big data with applications. Furthermore, two big characteristics of big data are big volume and big veracity [5]. The big volume of big data is fuzzy in essence. The big veracity is related to the ambiguity and incompleteness of big data [46]. Fuzzy logic and fuzzy sets have developed a significant number of methods and techniques to address ambiguity and incompleteness of data, and therefore they will play a significant role in overcoming ambiguity and incompleteness of big data [30,32,47].

8. Conclusions

The objective of this article is to apply mathematics to treat a few fundamental problems of big data and develop a mathematical theory of big data. To this end, it explores the volume of big data with the cardinality theory. It provides a mathematical foundation for searching for big data with the set theory. It reveals the relativity of big data with fuzzy logic and fuzzy sets theory [28]. It presents a similarity-based approach to big data by investigating finite and infinite similarity, the weak and strong similarity of big data, and similarity-based infinite reasoning. It also presents a logical approach to marketing strategy for online social networking platform services. The research contributes to the literature along three dimensions: 1) Cardinality of big data is the same as the cardinality of all the real numbers. 2) The relativity and infinity are two big characteristics of big data besides the ten big characteristics of big data [8]. The relativity of big data leads to the continuum from small data to big data, big data-driven small data analytics becomes statistically significant for further research and development of big data [14]. The infinity of big data leads to the exploration of infinite similarity of big data. 3) A logical foundation for revealing the secret behind the success of Meta (Facebook) and other social networking platforms or services will lead to logical methods for big data and big data analytics besides machine learning and deep learning [2,3]. The proposed

approach in this article might facilitate the research and development of big data, big data analytics, big data computing, data science, and data intelligence.

The mathematic theory for big data, analytics, and processing is a very important issue that is worthy of paying great attention to study. A mathematical theory of big data should also include addressing the following questions: What is a fuzzy-logic theory of big data? What is a similarity-based theory of big data? What is a calculus of big data? What is the cyclic model of big data reasoning? All these require further deep investigation in the near future. We will present the calculus of big data, the calculus of analytics, and big data reasoning as research results soon.

Optimization has drawn increasing attention in the field of big data in general and big data analytics in particular [22], because it is the foundation of big data predictive analytics in general and big data prescriptive analytics. In future work, we will examine the process of optimization for big data descriptive analytics taking into account the life cycle of business process-oriented big data analytics.

Conflict of Interest

There is no conflict of interest.

References

[1] Sun, Z., Wu, Z., 2021. A Strategic Perspective on Big Data Driven Socioeconomic Development. in The 5th International Conference on Big Data Research (ICBDR).September 25-27 (pp. 35-41). Tokyo, Japan: ACM.

[2] Russell, S., Norvig, P., 2020. Artificial Intelligence: A Modern Approach (4th Edition), Upper Saddle River: Prentice Hall.

[3] Hurley, R., 2019. Data Science: A Comprehensive Guide to Data Science, Data Analytics, Data Mining, Artificial Intelligence. Machine Learning, and Big Data, Middletown, DE: Hurley.

[4] Laudon, K.G., Laudon, K.C., 2020. Management Information Systems: Managing the Digital Firm (16th Edition), Harlow, England: Pearson.

[5] Sun, Z., 2022. A Service-Oriented Foundation for Big Data. Research Anthology on Big Data Analytics, Architectures, and Applications, Hershey, PA, IGI-Global. pp. 869-887.

[6] Laval, P.B., 2015. The Mathematics of Big Data. (Online). Available: http://math.kennesaw.edu/~plaval/math4490/fall2015/mathsurvey_def_slide.pdf. (Accessed 4 Sept 2016).

[7] Peters, T.J., 2015. Mathematics in Data Science. (Online). Available: www.engr.uconn.edu/~tpeters/MaDS.pptx. (Accessed 04 Sept 2016).

[8] Sun, Z., Strang, K., Li, R., 2018. Big data with ten big characteristics. Proceedings of 2018 The 2nd Intl Conf. on Big Data Research (ICBDR 2018). October 27-29 (pp. 56-61). Weihai, China: ACM.

[9] Sun, Z., Wang, P.P., 2017. A Mathematical Foundation of Big Data. Journal of New Mathematics and Natural Computation. 13(2), 8-24.

[10] Sun, Z., Xiao, J., 1994. Essentials of Discrete Mathematics, Problems and Solutions., Baoding: Hebei University Press.

[11] Johnsonbaugh, R., 2013. Discrete Mathematics (7th Edition), Pearson Education Limited.

[12] Enderton, H., 1977. Elements of Set Theory, Academic Press Inc.

[13] McAfee, A., Brynjolfsson, E., 2012. Big data: The management revolution. Harvard Business Review. 90(10), 61-68.

[14] Sun, Z., Huo, Y., 2021. The spectrum of big data analytics. Journal of Computer Information Systems. 61(2), 154-162.

[15] Sallam, R., Friedman, T., 2022. Top Trends in Data and Analytics. (Online). Available: https://www.gartner.com/doc/reprints?id=1-29ML-60N2&ct=220405&st=sb. (Accessed 21 April 2022).

[16] Minelli, M., Chambers, M., Dhiraj, A., 2013. Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Wiley & Sons (Chinese Edition 2014).

[17] National Research Council, 2013. Frontiers in Massive Data Analysis, Washington, DC: The National Research Press.

[18] Clissa, L., 2022. Survey of Big Data sizes in 2021. (Online). Available: https://arxiv.org/abs/2202.07659. (Accessed 11 March 2022).

[19] Chen, P.P., 1976. The Entity-Relationship Model-Toward a Unified View of Data. ACM Transactions on Database Systems. 1(1), 9-36.

[20] Coronel, C., Morris, S., Rob, P., 2020. Database Systems: Design, Implementation, and Management (14th edition), Boston: Course Technology, Cengage Learning.

[21] Courant, R., 1961. Differential and Integral Calculus Volume I, Glasgow: Blackie & Son, Ltd.

[22] Kelly, J.E., 2015. Computing, cognition and the future of knowing. (Online). Available: http://www.research.ibm.com/software/IBMResearch/multimedia/Computing_Cognition_WhitePaper.pdf. (Accessed 13 September 2016).

[23] Sun, Z., Pambel, F., Wu, Z., 2022. The Elements

of Intelligent Business Analytics: Principles, Techniques, and Tools. Handbook of Research on Foundations and Applications of Intelligent Business Analytics, Z. Sun and Z. Wu, Eds. pp. 1-20.

[24] Halevy, A., Norvig, P., Pereira, F., 2009. The Unreasonable Effectiveness of Data. IEEE Intelligent Systems. pp. 8-12.

[25] Jech, T., 2003. Set Theory: The Third Millennium Edition, Revised and Expanded., Springer.

[26] Manyika, J., Chui, M., Bughin, J.E.A., 2011. Big data: The next frontier for innovation, competition, and productivity. (Online). Available: http://www.mckinsey.com/business-functions/business-technology/our-insights/big-data-the-next-frontier-for-innovation.

[27] Sharda, R., Delen, D., Turban, E., et al., 2018. Business Intelligence, Analytics, and Data Science: A Managerial Perspective (4th Edition), Pearson.

[28] Lang, S., 2002. Algebra, Graduate Texts in Mathematics 211 (Revised third ed.), New York: Springer-Verlag.

[29] Zimmermann, H., 2001. Fuzzy set theory and its applications (4th edition), Boston: Kluwer Academic Publishers (Springer Seience+Business Media New York).

[30] Zadeh, L.A., 1979. Fuzzy sets and information granularity. Advances in Fuzzy Sets Theory and Applications, Horth-Holland, New York, Elsevier. pp. 3-18.

[31] IGI, 2015. Big Data: Concepts, Methodologies, Tools, and Applications.

[32] Zadeh, L.A., 1965. Fuzzy sets. Information and Control. 8(3), 338-353.

[33] Sun, Z., Sun, L., Strang, K., 2018. Big Data Analytics Services for Enhancing Business Intelligence. Journal of Computer Information Systems (JCIS). 58(2), 162-169.

[34] Finnie, G., Sun, Z., 2002. Similarity and metrics in case-based reasoning. International Journal Intelligent Systems. 17(3), 273-287.

[35] Gigerenzer, G., Selten, R., 2002. Bounded Rationality: The Adaptive Toolbox., MIT Press.

[36] Sun, Z., Pinjik, P., Pambel, F., 2021. Business case mining and E-R modeling optimization. Studies in Engineering and Technology. 8(1), 53-66.

[37] Larson, R., Edwards, B.H., 2010. Calculus (9th ed.), Brooks Cole Cengage Learning.

[38] Shannon, C.E., 1948. A mathematical theory of communication. The Bell System Technical Journal. 27, 379-423, 623-656.

[39] Laval, P.B., 2015. MATH 7900/4490 Math The Mathematics of Big Data (Syllabus). [Online]. Available: https://math.kennesaw.edu/~plaval/BigData/syllabus.pdf. (Accessed 4 Sept 2016).

[40] Laval, P.B., 2015. Introduction to the Mathematics of Big Data. (Online). Available: http://math.kennesaw.edu/~plaval/math4490/fall2015/mathsurvey_def.pdf. (Accessed 04 September 2016).

[41] ICERM, 2015. Mathematics in Data Science. (Online). Available: https://icerm.brown.edu/topical_workshops/tw15-6-mds/.

[42] Laval, P.B., 2017. Introduction to the Mathematics of Big Data. (Online). Available: http://ksuweb.kennesaw.edu/~plaval/math4490/fall2017/mathsurvey_def.pdf. (Accessed 25 4 2018).

[43] Chui, C.K., Jiang, Q., 2013. Applied Mathematics: Data Compression, Spectral Methods, Fourier Analysis, Wavelets, and Applications, Springer.

[44] Kepner, J., Jananthan, H., 2018. Mathematics of Big Data: Spreadsheets, Databases, Matrices, and Graphs, MIT Press.

[45] Chen, Y., Ghosh, A., Kearns, M., 2016. Mathematical foundations for social computing. CACM. 59(10), 102-108.

[46] IBM, 2015. The Four V's of Big Data. (Online). Available: http://www.ibmbigdatahub.com/infographic/four-vs-bigdata.

[47] Kantardzic, M., 2011. Data Mining: Concepts, Models, Methods, and Algorithms, Hoboken, NJ: Wiley & IEEE Press.

BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Journal of Computer Science Research
https://ojs.bilpublishing.com/index.php/jcsr

ARTICLE

Animal Exercise: A New Evaluation Method

Yu Qi[1*] Chongyang Zhang[1] Hiroyuki Kameda[2]

1. The Graduate School of Bionics, Computer and Media Sciences, Tokyo University of Technology, Japan
2. The School of Computer Science, Tokyo University of Technology, Japan

ARTICLE INFO

Article history
Received: 27 May 2022
Accepted: 10 June 2022
Published Online: 15 June 2022

Keywords:
Motion transfer
Animal exercise
Evaluation method
Monkeys
Target scale normalization

ABSTRACT

At present, Animal Exercise courses rely too much on teachers' subjective ideas in teaching methods and test scores, and there is no set of standards as a benchmark for reference. As a result, students guided by different teachers have an uneven understanding of the Animal Exercise and cannot achieve the expected effect of the course. In this regard, the authors propose a scoring system based on action similarity, which enables teachers to guide students more objectively. The authors created QMonkey, a data set based on the body keys of monkeys in the coco dataset format, which contains 1,428 consecutive images from eight videos. The authors use QMonkey to train a model that recognizes monkey body movements. And the authors propose a new non-standing posture normalization method for motion transfer between monkeys and humans. Finally, the authors utilize motion transfer and structural similarity contrast algorithms to provide a reliable evaluation method for animal exercise courses, eliminating the subjective influence of teachers on scoring and providing experience in the combination of artificial intelligence and drama education.

1. Introduction

Animal Exercise [1] is a method of training acting created by the Soviet dramatist Stanislavsky, and it is now mostly seen in the basic courses of acting majors. Usually, in their freshman year, students take a 16-week Animal Exercise course. The learning content of the Animal Exercise course is to observe and imitate the actions of animals such as "walking, eating, sleeping, hunting", etc. During the exercises, students will imitate a large number of animals, such as clever monkeys, ferocious tigers, aggressive roosters, etc. [2]. Through the vivid imitation of animals, the flexibility of the limbs is exercised, and the body and mind can be relaxed on the stage. Some students with poor physical shape cannot meet the requirements, and it is difficult to accurately express the external characteristics of animals. At this time, a lot of physical exercises and the guidance of teachers are needed to make the animal images created by the students realistic and credible on the stage.

At present, the teaching of Animal Exercise courses is mainly based on teachers passing on the course con-

*Corresponding Author:
Yu Qi,
The Graduate School of Bionics, Computer and Media Sciences, Tokyo University of Technology, Japan;
Email: d21200029a@edu.teu.ac.jp

DOI: https://doi.org/10.30564/jcsr.v4i2.4759

tent to students according to the theories in the textbooks and their own teaching experience. This kind of teaching method based on oral teaching will inevitably bring about teaching deviations. And due to teachers' personal preferences, there is also the problem of unfair scoring [3]. When students practice, the guidance given by different teachers is different, which will make students' thinking confused. Therefore, we need to provide a unified evaluation standard for the course, so that different teachers have a unified guideline in teaching, and provide students with an introspection environment, students can analyze the difference between their own imitation and animals, so as to get the ability to promote. We use motion transfer [4] to achieve this, transferring the original video motions of the animals to the students' own target videos.

Motion transfer technology refers to transferring the motion of the initial object in the initial motion video to the target object to produce the target motion video. Berkeley researchers have proposed a method to transfer human actions in different videos, which requires only two simple given videos, the target person, we want to synthesize his performance; the other is the original video, we want to combine his actions Transfer to the target person. However, these motion transfer are not suitable for quadrupeds, and they are all carried out in a standing posture.

In this study, we aim to provide an objective evaluation criterion for Animal Exercise courses for acting professional learners through the image similarity metric in motion transfer. Migration between humans and animals is very challenging due to several problems during the study. First of all, let's choose monkeys, which have obvious characteristics and appear relatively frequently in animal practice courses as the subject of the experiment. We utilize the coco dataset [5] to detect keypoints for students, but in the object detection dataset, we did not find monkey-related datasets. Second, in human-to-human action transfer, the mismatch of the scale of the source and target characters can cause the target characters to appear large relative to the background or surrounding objects or appear to be suspended. To address this issue, Chan et al. [6]. Devised a method for global pose normalization. They use the four coordinates of the source video character's near point, far point, nose, and ankle as the benchmark to correct the position of the target video character to make the generated target video more realistic. However, this method is only suitable for standing human posture specification, and the normalization effect for other movements such as crawling is not ideal. To solve the above two problems, we found 8 monkey videos on the open-source video website, marked 1428 monkey keypoint pictures, and created a monkey keypoint dataset named QMonkey. In order to deal with the second difficulty, we use the method of image scale filling to standardize the size of the target scale.

We summarize our contributions as follows: 1) We have created a dataset of monkey keypoints, which can provide a data basis for target detection for subsequent researchers. 2) Our experiments demonstrate that motion transfer between humans and quadrupeds is also possible. 3) We provide an evaluation standard for performance teachers when conducting animal practice tutoring.

2. Related Works

2.1 Motion Transfer

Human motion transfer refers to transferring the motion of objects in the source motion video to the target object and generating the target motion video [6]. At present, the most effective motion transfer technologies are inseparable from four steps: human pose estimation [7], training the source image generation model [8], posture normalization, and using the model to generate the person's movement in the target image. Based on Chan et al. [6], this research carried out an innovation suitable for the motion transfer between humans and monkeys.

2.1.1 Human Posture Estimation

Human posture estimation (HPE) refers to obtaining the posture of the human body from given sensor input. [9] We use OpenPose [7] to estimate the pose of monkeys and human limbs. OpenPose is a bottom-up pose estimation method. It first detects all the keypoints in the image and then uses PAF (Partial Affinity Fields) model to associate the keypoints with obtaining the correct image of the keypoints of the limbs. This method does not need to detect the object first and is very suitable for detecting keypoints of the monkey's limbs.

2.1.2 Generative Model

Since the creation of GANs by Goodfellow et al. [10], many interesting variants have emerged, some of them can transfer the style of two images [11], and some can generate high-resolution images from low-resolution images [12]. And pix2pixHD GANs [8] can generate source images with target actions. It is a variant of Conditional GANs [13] and redesigns the generation network based on pix2pix GANs [14], enabling the algorithm to complete high-quality image conversion.

2.1.3 Posture Normalization

In the human motion transfer, when the scale of the

person in the sources image and the person in the target image do not match, the target person may appear large relative to the background or surrounding objects or appear to be floating. To solve this problem, Chan et al. [6] designed a method of global posture normalization. They use the four coordinates of the source video person's near point, far point, nose, and ankle as benchmarks to correct the position of the target video person to make the generated target video more realistic. However, this method is unsuitable for non-standing monkeys, so we propose a new posture normalization method.

2.2 Dataset

COCO is large-scale object detection, segmentation, and captioning dataset [5]. It has 25,000 annotated human images, including labels for 17 keypoints such as nose, wrist, and knee. We use the COCO dataset to train the openpose model to recognize human poses. Since we also need to recognize the monkey's pose, we created a miniature monkey dataset for this experiment.

2.3 Image Similarity Index

LPIPS [15] uses depth features as a perceptual metric to judge the similarity of images. It is different from the widely used SSIM [16] and PSNR [17] evaluation indicators. It can evaluate the similarity of two images in a more human-like perceptual way. Since animal exercise is a way to improve performance, it will eventually be shown to the audience in a performance. At this time in our research, the visual similarity is significant, so we choose LPIPS as the evaluation benchmark for animal exercise.

3. Method and Experiment

3.1 Method

This experiment uses the same OpenPose parameters as Cao et al. [7] and selects the vgg19 [18] network structure. This combination is widely used in human posture detection and has a good detection effect. We use a pre-trained model for human pose detection to save experiment time. Since the QMonkey dataset (details are described in 3.2) is small, it is prone to overfitting when training monkey posture detection. We will terminate the training when the loss value reaches 0.006, taking 20,000 iterates and about 70 hours. When training the motion transfer model, parameters are the same as those of Chan et al. [6]. All training processes are performed on the Ubuntu16.04 operating system and a TITAN RTX graphics card. When using LPIPS for image similarity comparison, we choose the vgg19 network structure as the comparison parameter, which is consistent with the network structure of the OpenPose.

The evaluation system process is as follows:

1) First, use the COCO dataset to train an OpenPose model that can recognize human poses. On the other hand, use the QMonkey dataset to train an OpenPose model that can recognize monkey poses.

2) Perform posture estimation recognition on human videos and monkey videos to obtain images and posture images. We use monkey images and its posture images as the source dataset and human images and its posture images as the target dataset.

3) Train the generative model using the monkey images and its posture images.

4) Using the monkey posture images as the standard, normalization the human posture images. Simultaneously process human images.

5) Using normalized human posture images and generative models, generate human-action-based videos of monkey movements.

6) Comparing the structural similarity between the generated image and the posture image, respectively, to obtain an evaluation.

3.2 Monkey Posture Dataset

It is well known that OpenPose is very mature and accurate for human attitude recognition. Although the model trained on the human dataset can detect the pose of a few monkeys, the probability of such detection is too low to meet the needs of this experiment. Therefore, we created a new dataset of monkeys in an effective way to solve monkey pose recognition. We created a mini dataset QMonkey for monkeys. It is described in the format of the COCO dataset but only contains human-like pose information. The data from 8 different monkey videos with 1,428 images, as shown in Figure 1. Since the current dataset is small, only a few monkeys can be effectively identified, we will make it public after expanding the dataset.

3.3 Target Scale Normalization

We found from the existing animal videos that the distance between the animal and the camera is uncontrollable. When filming, the animals don't make the movements we want and don't walk within our designed range. Smaller animals may fill the screen, and larger animals may be very far from the camera. The method in [6], the everybody method, can no longer satisfy this experiment, so for the problem of mismatched target size ratios, we propose a target scale normalization method applicable in both standing and non-standing situations.

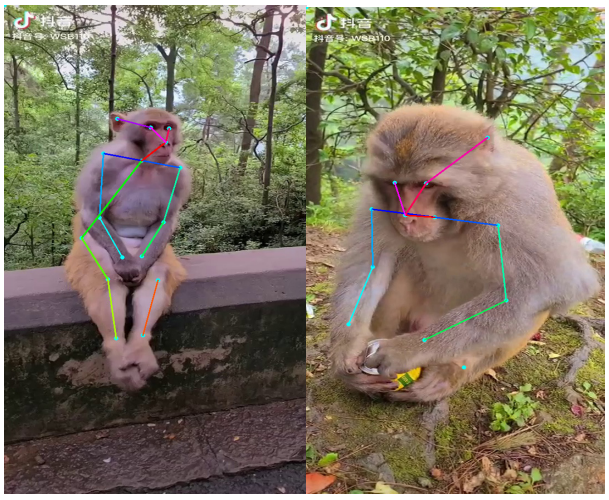

Figure 1. An Example of Images in the Qmonkey dataset

As shown in Figure 2, we record the four maximum points on the monkey posture image frame's top, bottom, left, and right. Then calculate the average width and height of the monkey posture, which are recorded as Mwidth and Mhigh, respectively. Similarly, the average width and height of the pose in the human pose map are calculated and recorded as Pwidth and Phigh, respectively. We calculate the ratio that needs to be filled or enlarged.

$$Wscale = (Pwidth / Mwidth) - 1$$

$$Hscale = (Phigh / Mhigh) - 1$$

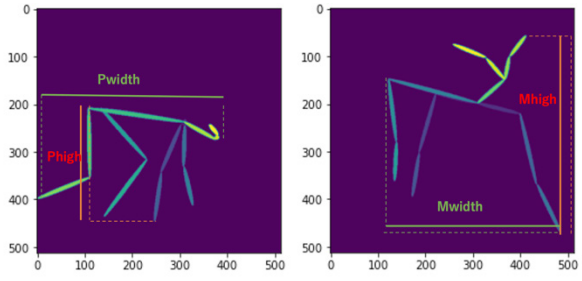

Figure 2. Human (left) and monkey (right) posture image, and their width and high

If neither Wscale nor Hscale is negative, select the larger value as the fill scale of the human posture image. If both Wscale and Hscale are negative numbers, we choose the smaller value and take the absolute value as the reduction ratio of the human posture image. When you use human images as source data and monkey images as target data to train the generative model, in that case, you can swap the human width and height with the monkey width and height in the formula.

3.4 Animal Exercise Evaluation

Here we construct the evaluation method using two sets of comparison graphs. The first group is the posture image of humans and monkeys, as shown in Figure 2, which can accurately reflect the direction of each limb and whether the degree of joint bending is similar. However, since humans and monkeys have different body proportions, we also need a second set of comparison images. Figure 3 consists of the original image of the monkey and the generated image of the human imitating the monkey. We use LPIPS to compare the two sets of images' similarity and then sum and average the scores to obtain the reference value.

Figure 3. Original image (left) and generated image of the monkey (right)

3.5 Animal Exercise Evaluation

3.5.1 Posture Normalization

We compare our scale normalization method and the everybody method of Chan et al. [6] with human images as source and monkey images as target data. The images in Figure 4 are respectively the posture image of a human imitating monkey, monkey posture image, monkey posture image normalized by our method, and monkey posture image normalized by the everybody method.

Our method downscales the monkey pose map to bring the pose scale closer to the pose images of humans imitating monkeys. The everybody method chooses the enlargement process, so that the scale of the monkey pose map and the human-imitation monkey pose image become larger, and part of the pose information is lost. It can be seen that our method is more suitable for standardized processing in non-standing postures.

3.5.2 Evaluation System

We selected two images from a video imitating a monkey for comparison in the evaluation experiments. During the monkey's walking, the hand and leg on the same side have two states, the leg moves forward close to the arm,

and the hand moves forward away from the leg.

In Figure 5, the experimenter on the left paid attention to the walking order of the limbs when imitating the monkey's walking, which was basically the same as the monkey's walking posture. The experimenter on the right walks clumsily when imitating the monkey's walking, which is not the same as the monkey's posture. After comparing the structural similarity between the two, the average scores of the generated map and the pose map are 0.309 and 0.366, respectively.

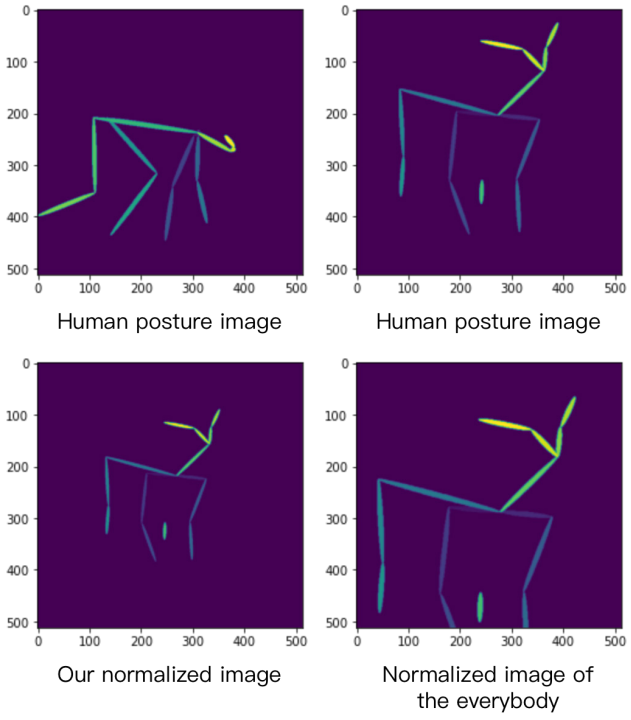

Figure 4. Comparison between our method and the everybody method

Figure 5. A comparison image of imitating the same monkey action

We surveyed 20 drama education teachers and asked them to rate 50 sets of comparison images. The teachers range in age from 28 to 70 years old, and the teaching age ranges from 1 to 41 years. Each set of comparison images contains one monkey image and ten images of humans imitating monkeys for 510 images. The evaluation scores are S, A, B, C, D from high to low. By analyzing the survey results, we came up with the evaluation criteria in Table 1.

Table 1. Scope of the proposed evaluation scope derived from the survey results

Numerical value	Evaluation
<0.3	S
0.3-0.33	A
0.34-0.37	B
0.37-0.4	C
>0.4	D

4. Results and Discussion

We found that the ability of a generative network to generate realistic images depends not only on whether the actions are mimicked the same but also on whether the limb proportions between the source and target data are similar. Although the generative network has a specific anti-interference ability and can generate images by lengthening or shortening limbs of different sizes, the movement transfer of different species increases the difficulty of generating realistic images. This research uses the target scale normalization method suitable for non-standing posture, normalizes the original data and target data to an approximate scale, and minimizes the influence of different limb scales on the model. Since we still cannot overcome the problem of limb scale changes due to camera angle, we compare the pose map and the generated map simultaneously to improve the reliability of the comparison results.

The Animal Exercise evaluation system provides teachers with a standard reference benchmark for teaching or examination, so we have adopted a grading method after research. Of course, since there is currently no action transfer between humans and animals that performs well, we cannot yet use the scores as a direct reference benchmark.

We found that whether the generation network can generate real images depends not only on whether the actions imitated are the same but also on whether the proportions of the limbs between the targets are similar. The generation network has a certain degree of anti-interference. It can stretch or shorten limbs of different sizes to generate, but this will also affect the quality of the generated image. Here we use the method of normalizing the target proportion to

make the target in the same size, try to eliminate the influence of different body proportions. However, we cannot overcome the problem of body proportion changes caused by the camera angle, as shown in Figure 5. Therefore, we compare the key point map and the generated map together to improve the reliability of the comparison result.

5. Conclusions

At present, the combination of drama education and artificial intelligence is still in its infancy, and there are a large number of technical blank areas. Many technologies can only be used from research in other fields and cannot completely solve the existing problems in drama education. The evaluation system of Animal Exercise courses fills the shortage of uneven teaching levels among teachers and too subjective teaching evaluation for Animal Exercise courses. To solve the problems, we proposed a new AI-based posture evaluation method. The Animal Exercise course evaluation system only provides teachers with an evaluation range for teaching and examination and does not entirely solve the problem of students' introspection. This is also our future work direction. Next, we will focus on developing the motion transfer algorithm between humans and monkeys to make the motion generated by the model more realistic, simplify the workflow, reduce the generation time, and facilitate the use in teaching.

Author Contributions

Yu Qi: Conceptualization; Methodology; Writing the initial draft.

Chongyang Zhang: Resources; Data Curation; Writing - Review & Editing.

Hiroyuki Kameda: Supervision.

Conflict of Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service, or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, "Animal Exercise: A New Evaluation Method".

Funding

This research received no external funding.

References

[1] Stanislavski, C., 1989. An actor prepares, Routledge.

[2] Adler, S., 2000. The Art of Acting, ed. Howard Kissel (New York: Applause, 2000).

[3] Qi, Y., Zhang, C., Kameda, H., 2021. Historical summary and future development analysis of animal exercise. ICERI2021 Proceedings, 14th annual International Conference of Education, Research and Innovation. pp. 8529-8538, IATED.

[4] Aberman, K., Wu, R., Lischinski, D., et al., 2019. Learning character-agnostic motion for motion retargeting in 2d. arXiv preprint arXiv:1905.01680.

[5] Lin, T.Y., Maire, M., Belongie, S., et al., 2014. Microsoft coco: Common objects in context. European conference on computer vision. pp.740-755.

[6] Chan, C., Ginosar, S., Zhou, T., et al., 2019. Everybody dance now. Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5933-5942.

[7] Cao, Z., Simon, T., Wei, S.E., et al., 2017. Realtime-multi-person2d pose estimation using part affinity fields. Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7291-7299.

[8] Wang, T.C., Liu, M.Y., Zhu, J.Y., et al., 2018. High-resolution image synthesis and semantic manipulation with conditional gans. Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798-8807.

[9] Andriluka, M., Pishchulin, L., Gehler, P., et al., 2014. 2d human pose estimation: New benchmark and state of the art analysis. Proceedings of the IEEE Conference on computer Vision and PatternRecognition. pp. 3686-3693.

[10] Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al., 2014. Generative adversarial nets. Advances in neural information processing systems. 27.

[11] Yoo, D., Kim, N., Park, S., et al., 2016. Pixel-level domain transfer. European conference on computer vision, Springer. pp. 517-532.

[12] Ledig, C., Theis, L., Huszár, F., et al., 2017. Photo-realistic single image super-resolution using a generative adversarial network. Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681-4690.

[13] Mirzaand, M., Osindero, S., 2014. Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784.

[14] Isola, P., Zhu, J.Y., Zhou, T., et al., 2017. Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125-1134.

[15] Zhang, R., Isola, P., Efros, A.A., et al., 2018. The unreasonable effectiveness of deep features as a perceptual metric. Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586-595.

[16] Wang, Z., Bovik, A.C., Sheikh, H.R., et al., 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing. 13(4), 600-612.

[17] Hore, A., Ziou, D., 2010. Image quality metrics: Psnr vs. Ssim. 2010 20th international conference on pattern recognition. IEEE. pp. 2366-2369.

[18] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Journal of Computer Science Research
https://ojs.bilpublishing.com/index.php/jcsr

ARTICLE

Optimization of Secure Coding Practices in SDLC as Part of Cybersecurity Framework

Kire Jakimoski[1] Zorica Stefanovska[1*] Vekoslav Stefanovski[2]

1. Faculty of Informatics, AUE-FON University, Skopje, Republic of North Macedonia
2. Sourcico, Tel Aviv, Israel

ARTICLE INFO

Article history
Received: 3 November 2021
Accepted: 13 June 2022
Published Online: 21 June 2022

Keywords:
Cybersecurity
Security risks
Secure SDLC
SQL injection
Broken authentication
Broken access control
Mitigation practices

ABSTRACT

Cybersecurity is a global goal that is central to national security planning in many countries. One of the most active research fields is design of practices for the development of so-called highly secure software as a kind of protection and reduction of the risks from cyber threats. The use of a secure software product in a real environment enables the reduction of the vulnerability of the system as a whole. It would be logical to find the most optimal solution for the integration of secure coding in the classic SDLC (software development life cycle). This paper aims to suggest practices and tips that should be followed for secure coding, in order to avoid cost and time overruns because of untimely identification of security issues. It presents the implementation of secure coding practices in software development, and showcases several real-world scenarios from different phases of the SDLC, as well as mitigation strategies. The paper covers techniques for SQL injection mitigation, authentication management for staging environments, and access control verification using JSON Web Tokens.

1. Introduction

Software is the transformation of an idea that becomes a reality in the form of a software solution to a specific real-world problem [1]. The international standard ISO/IEC/IEEE12207-2008 [1] which defines the working framework for all activities that are part of a software life cycle indicates that the software starts with an idea, i.e. with a precisely defined need for a certain type of software product. The software product is a set of computer programs accompanied by appropriate documentation, which was designed and developed for commercial purposes, i.e. sales. Everyday life imposes a great need for new software products that should, above all, be quality, but also safe. If secure coding is not applied during the development of new software, the possibility of a weakness of the software solution remains, i.e. the solution itself becomes a vulnerability of the system in general. Practice shows that such vulnerabilities are often the result of insufficient testing of the security aspect of the code, insufficient education of

*Corresponding Author:
Zorica Stefanovska,
Faculty of Informatics, AUE-FON University, Skopje, Republic of North Macedonia;
Email: zstefanovska@yahoo.com

DOI: https://doi.org/10.30564/jcsr.v4i2.4048

new IT specialists on the term secure coding, differences in security rules in different programming languages, use of free software (open source) and the like. On the other hand, due to insufficient information of people about cyber threats and in the absence of basic cyber-awareness in the common man, we are witnessing the massive use of unverified and non-validated software created with insecure coding which is one of the many vulnerabilities of systems.

There are many concepts for developing high quality and functional software [2]. The challenge of any good team of developers is to create a secure and high quality software solution that will meet security practices and measures while not being a bottleneck for the of software's functionality. Finding the optimal solution that will meet both conditions: Security and Functionality, is considered one of the most challenging tasks in the life cycle of software solution development. Achieving "ideally secure" software requires new mechanisms in coding, raising the expertise of developers to write secure code, investing in their additional education in the field of IT security, implementing additional security specifics in writing code and the like [3].

The purpose of this paper is to point out good practices and tips to be used in software development to integrate secure coding at all stages of the development cycle. This paper can help software product engineers anticipate and recognize the challenges in cyberspace that would be a vulnerability to the product they create. At the same time, this paper will contribute to raising awareness of cyber attacks among young developers and the need to write secure code, which will be subject to various types of testing in the first phase of SDLC. In other words, this paper presents the optimization of secure coding in the development of software applications using practices to improve the quality of software solutions from a security perspective, while offering the user an optimal security solution, and thus approaching the ideal security functional software.

2. Related Works

In recent years, the number of different vulnerabilities of different software products has been increasing. Software vulnerabilities are constantly growing, but the search for new ways and practices to improve software products is also growing. The emergence of vulnerable software over a period of 20 years is illustrated in Figure 1.

There are several definitions of software product vulnerability, including that of the IETF (IETF RFC 4949): "A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy" [5].

To overcome software development vulnerabilities that contribute to the creation of vulnerable software, different methods of software development have been identified in practice. Recently, most popular are the agile methods for developing software products that have incorporated the security issue in each stage of their SDLC. Namely, more and more software companies in the development of new software use secure coding practices and test the security of the software at every stage of development, while respecting the principles of the standard SDLC [6].

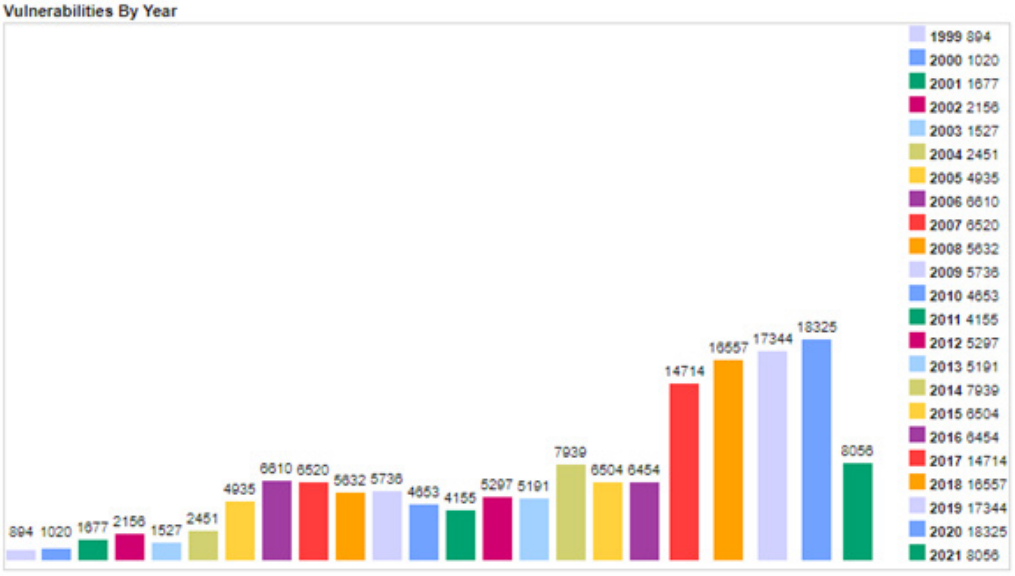

Figure 1. Number of reported vulnerable software in the CVE (Common Vulnerabilities and Exposures) database from 1999 to 2021. [4]

The term secure coding has attracted a great deal of attention in recent years. There are several ways to define this term. Most intuitively, we can define secure coding as a way of writing a secure program that will be as resistant as possible to illegal operations by malicious programs or people. By illegal operations we mean operations that compromise the security of the data and the application as a whole. If errors occur in the program that contributes to the program not fulfilling its functionalities, but these errors do not have a security implication, then we must not declare the program as unsafe.

It is necessary to distinguish between a functional application of sufficient quality that is not safe, and a safe but not sufficiently functional application. Secure coding helps protect user data from theft, corruption and malicious use. Security is not something that can be added to the software in the end as a finishing touch. In order to integrate security into the software itself, the natures of the threats must first be identified and accordingly security coding practices must be included during the software planning phase.

Without going into the reasons why malware attacks software applications, we must be aware that even the slightest vulnerability of a system is an open vector for attack and data theft. Attacks can be automated and replicated, but any vulnerability, no matter how small, is a real threat to the system as a whole, noting that no platform is immune to cyber attacks today.

It should be noted that secure coding is important for all types of software, from even the small everyday scripts that developers often write for themselves, to the largest commercial applications intended for public use.

Practices and tips for secure coding can be suggested by any experienced software team. Whether they will be implemented in software solution development usually depends on the management team leading the software development project.

Guided by the idea of avoiding the additional financial implications that would result from additional steps in the classic SDLC, the implementation of security in the early stages of development is often avoided. Practice, on the other hand, has shown that saving on security compromises application data protection. Namely, it has happened many times that the financial loss caused by a cyber attack on a web application is much greater than the finances that would be needed to include security in all phases of the SDLC. The SDLC security proposed by Microsoft is a model that includes 12 practices that need to be implemented and with their proper implementation, the security of the application is achieved in the most economical way.

These are the following practices for secure SDLC by Microsoft [7]:

- Provide Training.
- Define Security Requirements.
- Define Metrics and Compliance Reporting.
- Perform Threat Modeling.
- Establish Design Requirements.
- Define and Use Cryptography Standards.
- Manage the Security Risk of Using Third-Party Components.
- Use Approved Tools
- Perform Static Analysis Security Testing (SAST).
- Perform Dynamic Analysis Security Testing (DAST).
- Perform Penetration Testing.
- Establish a Standard Incident Response Process.

3. Top 10 Web Application Security Risks

Every development team would like to know in advance what the possible attack risks are for the application they are developing. There are several methods to anticipate possible security risks that, if not addressed in a timely manner, could result in a vulnerable and unsafe application. The security risk analysis, according to the OWASP① Methodology, is treated with four metrics to determine the level of risk for the software solution - Usability, Frequency, Lightness and Technical Impact [8]. Risk analysis provides recommendations and tips that can successfully detect if an application is vulnerable, as well as tips and suggested practices on how to protect ourselves from these risks. The tips and recommendations that we will point out are of great importance for the development teams that are trying to develop secure code. If they are implemented and followed at all stages of SDLC when developing a software product, it is very likely that you will get Secure SDLC. In that way, the end goal of highly secure and functional software can be achieved in the most economical way. In essence, by creating your own model for secure coding according to the advice and methodology of OWASP, each organization can achieve optimization of secure coding in the development of software solutions in both the private and public sector. The followings are the top 10 web application security risks:

Injection. Injection occurs when untrusted data (usually provided by the user) is sent to command or query interpreter. Commonly used vectors are databases (SQL Injection) and operating system shells (OS Command Injection). The malicious agent can use a specially crafted input that will be sent to the underlying interpreted, executing unintended commands or accessing unauthorized data [9,10].

① The Open Web Application Security Project® (OWASP) is a nonprofit foundation that works to improve the security of software.

Broken Authentication. Authentication and overall session management is easy to implement in a functional and insecure manner. This means that while the application is functioning correctly for regular users, it is possible for malicious agents to compromise passwords, keys, or session tokens [11].

Sensitive Data Exposure. Many web applications use sensitive and personally identifiable information, so those data must be stored on the server, transferred to the browser, and used during the browser session. Each of those sites is a possible exposure risk, a place where attacker can steal or modify data. This can result in credit card fraud, identity theft and other crimes.

XML External Entities (XXE). XML processors provide the option of evaluating external entity references. While this is a useful feature, it can be an attack vector if used with untrusted data. Possible issues include disclosure of internal server files and file shares, scanning and access of internal ports, remote code execution, and denial of service attacks via XML bombs.

Broken Access Control. Another issue that is commonly implemented in a functional and insecure way is authorization and access control. Abuse of these features would enable a malicious user to escalate its privileges – which could lead to access of other user's data, and in extreme cases, changing of access rights and overtaking of the system [12].

Security Misconfiguration. This is not a single issue, but a result of a flawed application deployment process. The most common issues are insecure default configurations, incomplete or ad hoc configurations, open cloud storage, verbose error messages, etc. Misconfiguration can happen at any level of an application stack, as well as on all the interfaces between different levels of the stack, both technical and human. It is common to have a vulnerability because of miscommunication of responsibilities.

Cross-Site Scripting XSS. XSS is an injection-based attack that focuses on the front-end of a web application. It happens when user-provided data is not properly validated and sanitized. Commonly it is used in a stored way, with the attacker injecting data in the web-site's database, which is later viewed by a victim. This can lead to session hijacking, data leaks or redirects to malicious sites.

Insecure Deserialization. A common approach to passing information between the client and the server is to exchange a state object, e.g. in a cookie. This state object is serialized and encoded using some scheme when in transit and is then deserialized on the client and on the server. If an attacker is able to deserialize the serialized state, he can access the application data inside, or even tamper with it. This could lead to remote code execution, privilege escalation, session hijacking, and other breaches.

Using Components with Known Vulnerabilities. Any non-trivial application will use third-party libraries, frameworks, packages, and other software modules as part of its code base.

All the parts run with the same privileges the application itself is running with, which means that any vulnerability of a component is a vulnerability of the application. These kinds of attacks are also lucrative for the attackers, as finding a vulnerability inside a heavily used component can allow access to multiple sites.

Insufficient Logging & Monitoring. If despite all our efforts, a breach does occur, it is extremely important that we have the necessary tools to detect it and mitigate it. With some attacks, like DDoS, proper detection is crucial in the defense of our site. Also, a common scenario is that once an attacker successfully overtakes a system, that system can be used as a foothold in attacking connected systems.

4. Example and Tips for Secure Coding

4.1 Injection into SQL Expressions

Technique overview. Most RDBMS② are using SQL as the querying and command language and the application build over them communicate with the database by constructing and sending SQL commands. The database does not know if the queries are malicious or not, and if they are valid, they will be executed. SQL Injection is an attack technique that will trick the application server into constructing a malicious command and getting the database to simply execute it. One often used type of attack is on applications where the construction of an SQL command is done with string concatenation. If the application concats unverified and unsanitized user input, the user can basically short-circuit the SQL Expression, and attach another of his own [13].

Example. In a PHP-based application, the following code is used to select values from a table called Items.

```
$sql = "SELECT Name, Status
        FROM Items
        WHERE Status != " . ITEM_DELETED_
        STATUS . "AND ID = " . $item_id . ";
```

This command is constructed with concatenating the fixed text of the command with two code-level parameters, ITEM_DELETED_STATUS and $item_id. The

② Relational Database Management System

ITEM_DELETED_STATUS is a constant that is defined in the code, so this cannot be used as an attack vector. On the other hand, $item_id is taken from a parameter of the request, using the following code:

```
$item_id = filter_var($_GET[“id”], FILTER_SANITIZE_
           STRING)
```

While it seems that the input is sanitized, the sanitization used is targeted to prevent XSS attacks. It does nothing to prevent SQL injection attacks - so from a database perspective, the value of the user input is completely raw. This endpoint could be accessed using something like the following URL:

```
http://server/item-info.php?id=123
```

In that case, the value of the $item_id variable will be “123”. The actual value of the $sql variable will be:

```
SELECT Name, Status
FROM Items
WHERE Status != 0 AND ID = 123
```

This is a valid SQL expression that when executed by the database will return the Name and Status of the item with and ID of 123. One variant of SQL injection changes the value of this parameter to an expression that will return more data than the original expression. E.g., if we use the following URL:

```
http://server/item-info.php?id=123%20OR%201=1
```

the value of the $sql variable will become:

```
SELECT Name, Status
FROM Items
WHERE Status != 0 AND ID = 123 OR 1=1
```

Since 1=1 is a condition that is always true, this command will effectively return the names and statuses of all items in the database. Another variant is to use the specifics of SQL to attach an additional statement after the intended statement. E.g. the following link includes a destructive DDL statement:

```
http://server/item-info.php?id=123;%20DROP%20
                TABLE%20Items
```

The value of the $sql variable will become:

```
SELECT Name, Status
FROM Items
WHERE Status != 0 AND ID = 123; DROP
TABLE Items
```

This actually changes our SQL statement into two statements. One is the original query, while the other is a destructive command, and will delete the Items table itself. Once this request is processed, the application will no longer have such a table, which means that, at best, the application is nonfunctional, and at worst, a major, and potentially unrecoverable data loss.

In this specific application, all database queries are run under a user that has full privileges not only on the database, but on the database server as well, so even more drastic privilege escalations are possible.

Mitigation of the example code’s vulnerability. There are multiple approaches available to this piece of code. One of the most basic ones is to limit the destructive power of an intruder, even if a successful attack occurs.

Database user privileges. The user that accesses the database should have the minimum permission that are sufficient to execute their intended operations. Usually, the user needs only to have data manipulation permissions (selecting, inserting and modifying data). This would mean that while the attack via the http://server/item-info.php?id=123%20OR%201=1 URL will succeed and leak data, the attack via the http://server/item-info.php?id=123;%20DROP%20TABLE%20Items URL will fail. An outside attacker will not be able to destroy our database, but they will still be able to extract data they should not be able to. In this specific case, the SQL command

```
REVOKE ALL ON `Database`.* FROM
‘user’@’localhost’;
GRANT SELECT, INSERT, UPDATE ON
`Database`.*
TO ‘user’@’localhost’;
```

was used to modify the accessing users’ privileges. First, all privileges were revoked, and then the user was explicitly granted only the SELECT, INSERT and UPDATE privileges. Since the application uses a technique known as soft delete, the DELETE permission was not required, so it was not granted. This approach should be used in all scenarios, as the application user should not have any extra permissions that those that are actually needed.

Type validation on user input. Another way to defend against SQL injection attacks is to ensure that the user input does conform to the type requirements of the query. In this case, the input should be an integer, so if we test the input for that, we can detect this attack and stop it before it gets to the database.

```
$item_id = filter_var($_GET[“id”], FILTER_SANITIZE_
                        STRING)

    if (!is_numeric($item_id)) {
       logError(“Invalid item_id received from client “
       . $item_id);
        die;
    }

    $sql = “SELECT Name, Status
            FROM Items
            WHERE Status != “ . ITEM_DELETED_
    STATUS
            . “ AND ID = “ . $item_id . “;”
```

This code uses the library function is_numeric to check whether the $item_id variable is either a valid number, or a string containing a valid number. If it's not, then an error is logged, and the processing of the request stops immediately. The malicious SQL is neither generated nor sent to the database.

This approach is effective, but it's not systemic. It's hard to check all the options for every single query, and the burden of implementation is on the developer.

Query parametrization using PDO. A better approach is to avoid manual generation of the SQL string completely. We can use a technique called prepared statements that is supported by most databases. In this case, we send the query using parameters, i.e. the text of the query is defined once, with placeholders at the variable parts. The values that need to specify the parameters are send separately. Since the database engine knows that is should execute a specific query format, it knows the types of the parameters, so it will not allow for any insertion of SQL statements.

In PHP there is a PDO library that supports using prepared statements. The code in our case would look like this

```
$item_id = filter_var($_GET[“id”], FILTER_SANITIZE_
            STRING)
   $statement = $pdo->prepare(“SELECT Name, Status
   FROM Items WHERE  WHERE Status != :status
   AND ID = :item_id”);
   $statement->bindValue(“:status”,ITEM_DELETED_
STATUS,
   PDO::PARAM_INT);
   $statement->bindValue(“:item_id”, $item_id,
   PDO::PARAM_INT);
   $statement->execute();
```

Since the statement knows that the :item_id parameter should only have an integer value, it will not allow any insertion of SQL inside the value.

Implicit parameterization using ORM. Instead of hand-crafting our SQL, it's quite possible to use a tool to map it for us. These kinds of tools are called Object-Relational Mappers (ORM). The most popular ORM for PHP is called Eloquent and it is part of the Laravel framework. Using it, we can describe the shape of our database using a model. Then, instead of creating SQL statements, we use regular language concepts to specify the data we need, and the SQL query is generated by the ORM automatically. This has the benefit that we are protected from SQL injection attacks by design, as there is no SQL to concatenate in an unintended way [14-16].

The code would look like this:

```
use Illuminate\Database\Eloquent\Model;
class Items extends Model
{
   protected $primaryKey = ‘ID’;
   protected $fillable = [
      ‘Name’,’Status’
   ];
}
const item = Items::where([
   [‘Status’, ‘!=’, ITEM_DELETED_STATUS],
   [‘ID’, ‘=’, $item_id],
])->first()
```

A major drawback to this approach is that, as part of a framework, it can't be easily used in isolation, as it requires significant setup effort.

Discussion. After evaluation of the different approaches we decided to implement explicit parametrization of the code. While with Eloquent any parametrization is implicit and easier to use, it requires a major refactor of the application. It was decided not to proceed with such a change. Instead, the database privileges were fixed, and in addition, all the vulnerable SQL statements were transformed into a parametrized format. In specific places, where it made sense from a user perspective, type checks were added as well, in order to be able to return user-friendly errors.

4.2 Broken Authentication

Technique overview. One of the most trivial, yet persistent security holes are authentication leaks. The issue is that quite often, they are not a purely technical problem, i.e. it's not enough to solve them through code, but they require user discipline and education.

Quite often, especially with systems with automated deployment, it is common to have a set of hardcoded system users with total access to the system. It is assumed that once the system is deployed, the operator should change the default credentials to custom ones, so that those users cannot be used by unauthorized persons. However, this is not always the case, so there are plenty of cases where an otherwise secure system was compromised using the default set of credentials.

Example. A large, distributed team of developers are developing a large, distributed system. The authentication of the system is done with a regular username/password combination, with optional two-factor authentication. Because of business reasons, it's not possible to enforce two-factor authentication across the board. The process of registration of a user involved email verification.

Since there are many changes being done to the system at a given time, using a single testing and staging environment is not practical. A procedure was developed for automated generation of ephemeral testing/staging environments. These environments are a point-in-time replica of the production environment, including databases, storages, services, cloud resources, etc.

Any subteam is able to generate such an environment, use it to test and stage a feature, and once ready, push it to production. The authorization pool that is generated per environment used a single hardcoded user with a global super-admin role. The user was preset as verified. The password for the user was stored inside a secret of the continuous integration tool, so it was not directly accessible to the developers. The intention was that only an enumerable list of people will have access to it, and that after generation of an environment, there should be a manual intervention to change the password.

That was not always the case, and, in time, most of the developers knew and used the default password.

Mitigation of the example code's vulnerability. To address the vulnerability before it became an attack, several solution scenarios were proposed.

Enforce scrubbing of data. Since the default account was used only on the ephemeral environments, the leakage will be much smaller if the data are scrubbed of any personally identifiable information. This approach would trivialize the problem, however it has some issues of its own – mainly that it's hard to guarantee and enforce a proper scrubbing procedure on a system that changes often.

These issues were considered, and it was decided that this approach, for the specific system, will create more problems than it solves, so it was not implemented.

Limit the access of the environments. Another proposed option was to limit the physical access of the environments to users within the company, instead of the general internet. This solution had the benefit that it is easy to enforce via network policies, even for remote workers, using VPN filtering or similar approaches. Also, this kind of solution was already used for things like cloud service or direct database access.

However, the business requirements are that the ephemeral environments had to be accessible to specific stakeholders who are outside of the company. While it is possible to expose the environments in a controlled manner, it would have created additional workload for the operations team, as well as disrupting the user experience of external stakeholders.

Since this approach was determined to create additional workloads, without solving the underlying problem, it was not implemented. It was decided that we might implement limited access to some ephemeral environments if we know they won't be used from outside the organization.

Code-based limitation of the hard-coded account. Since the hardcoded user should ideally be used only to create the real users that will actually use the environment, a possible solution would be to add such restrictions to the default user. For example, we could add a rule that the default user is only active some preset time after creation, or that it can only do specific actions, or we can disable it once a real super-admin user is generated, etc.

While these actions will effectively solve the problem, they would require changes and specific checks in the authentication/authorization code. This means that the hardcoded account will not only be hardcoded in the configuration of the ephemeral environments, but also in the service that processes the users.

The drawbacks are that the code will have to behave differently for different users, and that would make the system inconsistent. It will dramatically increase the need to test and verify that the authentication/authorization process is operational and secure.

This approach was dismissed because, while effective, it will increase the complexity of an already complicated system.

Implement two-factor authentication. A fourth approach was to turn on the two-factor authentication for the hardcoded account. Since this would be used by multiple

people, we will need to use an application for sharing TOTP verification codes.

This lowers the security value of the hardcoded password, since even if a malicious user knows the password, they cannot use it to login to the system, unless they have access to the shared TOTP application. And since most tools for secret sharing include centralized administration, this transfers the problem to management of the secret sharing system. This will dramatically reduce the number of people who can tamper with the system.

The only downside of this solution is that it requires a centralized secret sharing tool, but there are plenty reasonably priced solutions for that.

Add account verification. Another option was to avoid hardcoding the password for the super-admin account at all. Instead of generating a pre-verified account with a set username and password, only the username can be hardcoded, and then use an email to verify the account and set a password. Since the environment deployment process already generated an email specific to the environment, this was easy to implement, and the implementation would only change the deployment process.

The downside is that the environment is not immediately useful, as it will require a manual step of verifying the account. However, since the environment generation process is usually monitored, the person responsible for the specific environment can easily verify and set a password. If needed, they can set up two factor authentication for the specific account, or even disable the account altogether. And since the password will be generated by them, it will be unknown even to the system administrators.

Discussion. After evaluation of the different options available, it was deemed that the last two options will systematically solve the problem. Taking in mind the specific organization of the development teams, it was decided to use the last approach, as it transferred responsibility for password management on a specific environment to the team itself. A part of the solution was a training session for the team leaders on how to set and secure the password of the super-admin user.

4.3 Broken Access Control

Technique overview. Once an application knows who the user is, it's imperative to know what the operations are the user can do, and, just as important, what operations should be prohibited. Authorization is extremely complicated problem to solve, and quite often is tricky to validate. This kind of attack misuses that complexity to make the attacked system think that the malicious user has more capabilities that they should actually have.

One vector of attack, on applications that use JSON Web Tokens for authorization and access control, is to tamper with the data present in the token. A JSON Web Token consists of three parts: header, payload and signature. The signature can cryptographically verify the contents on the header and payload, so that it can be detected whether the data of the payload was tampered with. However, unless it is being done automatically, there is potential for error, and an attacker can target the endpoint of the system that does not implement correct verification.

Example. A JavaScript based server-side application is using JWT tokens for authorization and access control. It uses an external service for token generation, and the verification used the same external service. That means that the process of verifying the token's integrity was slow, and developers tended not to use it, as it was making the application sluggish.

An example of a JWT would be:

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiZW1haWwiOiJlbWFpbEBleGFtcGxlLmNvbSIsImlhdCI6MTUxNjIzOTAyMn0.
kEmXw9lLw3tO1HloDZQoORejF1RiwVFSv-73VGkbCy7Cu91ZSyuW1b7LayrNWcknl5wP3JH-9kH1err0Mx96kbrA1uHpu0RXoRmLraTY-f40krmSVLO1czYZB69BtQEkWIG3wup_wlbhDZLiKkJgyLSPx6gnhTQibSw9U7rW07Wm-CPu36-KyfgXedX--Mk-MsJqyiSBVHlhbMJmjlD-ABWJJ1fQRF2lsirug9D-16MEYFkzOshvPI1nczLH-8CBk-ls-VL5c67JPUpmOqYczEGvOth50Bymloc2Jf_l8pJUWjZzejF-Hsg4AGRHkDrYNbQELHbfGYrNKhyr_vF0j4BpquYw

It consists of three parts that are separated by the dot(.) characters. The first two sections are simply base-64 encoded strings. If we decode them, we can get their plain text quite easily. This specific token has the header of

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

and a payload of

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "email": "email@example.com",
  "iat": 1516239022
}
```

We can note that both header and payload are simple json objects, and, in the payload, the data of the user is

plainly visible. The frontend application stores this token in a cookie called ident and sends it on every request to the backend service.

The service in question is using the express framework to fulfill the requests, and an example endpoint is https://www.example.com/api/users/me which returns the full user profile for the currently logged in user, including personally sensitive information.

In order to avoid using user-specific parameters in the endpoint url, it uses the provided cookie to extract the user email, based on which the full profile is loaded from the database and returned to the client.

The code that handles the request is

```
router.get( '/users/me', (req, res) => {
    const email = getEmailFromCookie(req);
    if (!email) {
        res.sendStatus(http.forbidden);
        return;
    }

    const profile = await UserService.getProfile(email);
    return res.status(http.ok).send({ profile });
})
```

This code will extract the email from the request, and once found will query the database for the profile. This means that if a malicious user is able to successfully change the return value of the getEmailFromCookie function, he will successfully retrieve the full profile of another user.

The function getEmailFromCookie is:

```
const getEmailFromCookie = (request: HttpRequest)
=> {
    const ident: string = request.cookies.ident;
    if (!ident) {
        return undefined;
    }
    const parts = ident.split('.');
    const userData = decodeBase64(parts[1]);
    const { email } = userData;
    return email;
}
```

This code simply takes the payload from the JWT and, without running any verifications, decodes and returns the email.

This means that the user can, manually or automatically, change the value of the cookie to another with the same header and signature, but whose payload decodes to

```
{
 "sub": "1234567890",
 "name": "John Doe",
 "email": "victim@example.com",
 "iat": 1516239022
}
```

Any verification of this token would mark it as invalid, but since there is no verification, this will not be noticed. Once called, the API endpoint will treat this as a valid request from the user victim@example.com, and return the full user profile to the attacker.

Mitigation of the example code's vulnerability. The obvious solution to this issue, once identified, is to add token validation. However, this needs to fulfill some requirements:

- It should be done on every request, i.e. the developers should not be allowed to opt-out of the validation
- It should be done implicitly, with no effort on the developer side, so that it will be impossible to forget to use it
- It should be performant, as the added verification should not slow down the application more than absolutely necessary

The existing verification code failed on all three of these requirements since it was explicit and used an external service.

In order to solve the first two requirements, the verification was implemented as an express middleware that verified that the token in the ident cookie was a valid JWT token. Because of the specifics of the express framework, the middleware will be called before the actual route handler. The route handler will be invoked if and only if, the middleware ends its run with a call to the next function. If the verification is not successful, i.e. if the token has been tampered with, we stop the processing, returning a forbidden error

Note that the actual verification is done inside the verify JWT token function. That function takes the token as a parameter, and verifies it asynchronously. Once the promise is resolved, we have a single boolean with the verification result.

To satisfy the third requirement, the verify JWToken function used a two-pronged approach. It maintained a list of tokens that were already verified (along with their expiration dates), so that a known good token does not have to be verified all the time. This is needed because the verification of the signature itself takes a non-trivial processing time, even when running locally. The nature of the service is that a user will usually request several hundreds

of API endpoints in a small amount of time, so keeping a list of known good tokens can effectively short-circuit that verification, at a small memory cost.

The second prong was to run the verification process locally instead of using the external service. The service helpfully provided for a way to download the key that is being currently used for the client (along with its expiration details) in a JSON Web Key format (JWK). In order to run the validation locally, several external libraries were needed, like jsonwebtoken and jwk-to-pem. The code for the verification was:

```
const verifyJWToken = async (token:string) => {
 if (checkCache(token)) {
  return true;
 };
 const pem = jwkToPem(jsonWebKey);
 const result = await new Promise((resolve) => {
   jwt.verify(token, pem, { algorithms: ['RS256'] },
     (err, payload) => {
                if (err) {
          return resolve({success: false});
        }
        return resolve({success: true, payload});
     });
 });
 if (!result.success) {
  return false;
 }
 if (isExpired(result.payload.exp)) {
  return false;
 }
 tokenCache[token] = result.payload.exp;
 return true;
};
```

This code first checks the cache for the token. If the token is found, it returns success. If the token is not in the cache, we can verify it using the jwt library. If the verification is successful, we get the decoded payload as a result. Once we have that, we're checking for token expiration one more time, and if everything is ok, we are signaling that the verification is successful.

This approach fulfills all three requirements, as it is both performant and transparent for the end user.

5. Conclusions

Secure coding and adherence to secure SDLC is quite a difficult task, both for the developers and the other members of the project team. The recommendations and tips outlined in this paper are intended to help software companies in the public and private sectors reduce the risk of application attacks in the most cost-effective way. This goal can be achieved exclusively by using the multitude of resources offered in the literature and empirically proven to have a positive impact on cyber defense, resulting in a functional and secure software product.

To create a secure application, you first need to define the term application security. In other words, frame all the answers to the question: What is security for a software product? Such a framework should guide the development of a secure application. Most of the answers to this question come from several factors such as user requirements, the environment in which the application will be developed, the production environment, and the social environment in which the application will be implemented.

Conflict of Interest

Authors declare no conflict of interests.

References

[1] ISO/IEC/IEEE International Standard, 2008. Systems and software engineering -- Software life cycle processes. IEEE STD 12207-2008. pp. 1-138. DOI: https://doi.org/10.1109/IEEESTD.2008.4475826

[2] Vale, T., Crnkovic, I., De Almeida, E.S., et al., 2016. Twenty-eight years of component-based software engineering. Journal of Systems and Software. 111, 128-148.

[3] Gorski, P.L., Acar, Y., Lo Iacono, L., et al., 2020. Listen to Developers! A Participatory Design Study on Security Warnings for Cryptographic APIs. In-Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1-13.

[4] CVE Details, Vulnerabilities By Year (https://www.cvedetails.com/browse-by-date.php).

[5] Shirey, R., 2007. Internet Security Glossary, Version 2. DOI: https://doi.org/10.17487/RFC4949

[6] Baldassarre, M.T., Santa Barletta, V., Caivano, D., et al., 2020. Integrating security and privacy in software development. Software Quality Journal. 28(3), 987-1018.

[7] Microsoft SDL Practices (https://www.microsoft.com/en-us/securityengineering/sdl/practices).

[8] OWASP Risk Rating Methodology (https://owasp.org/www - community/OWASP_Risk_Rating_Methodology/).

[9] Alwan, Z.S., Younis, M.F., 2017. Detection and prevention of SQL injection attack: A survey. International Journal of Computer Science and Mobile Computing. 6(8), 5-17.

[10] Sinha, S., 2019. Finding Command Injection Vulnerabilities. Bug Bounty Hunting for Web Security 2019. Apress, Berkeley, CA. pp. 147-165.

[11] Nadar, V.M., Chatterjee, M., Jacob, L., 2018. A Defensive Approach for CSRF and Broken Authentication and Session Management Attack. InAmbient Communications and Computer Systems. Springer, Singapore. pp. 577-588.

[12] Petracca, G., Capobianco, F., Skalka, C., et al., 2017. On risk in access control enforcement. InProceedings of the 22nd ACM on Symposium on Access Control Models and Technologies. pp. 31-42.

[13] Tasevski, I., Jakimoski, K., 2020. Overview of SQL Injection Defense Mechanisms. In2020 28th Telecommunications Forum (TELFOR). IEEE. pp. 1-4.

[14] Budiman, E., Jamil, M., Hairah, U., et al., 2017. Eloquent object relational mapping models for biodiversity information system. In 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT). IEEE. pp. 1-5.

[15] Sinha, S., 2019. Database Migration and Eloquent. Beginning Laravel. pp. 113-166.

[16] Apress, B., Stauffer, C.A., Laravel, M., 2019. Up & running: A framework for building modern php apps. O'Reilly Media.

BILINGUAL
PUBLISHING CO.
Pioneer of Global Academics Since 1984

Tel.:+65 65881289
E-mail: contact@bilpublishing.com
Website:ojs.bilpublishing.com

2630-5151 02
9 772630 515223