Received: June 8, 2020. Revised: July 8, 2020.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Evaluating the Impact of Malware Analysis Techniques for Securing Web Applications through a Decision-Making Framework under Fuzzy Environment

Rajeev Kumar[1, 2] Mamdouh Alenezi[3] Md Tarique Jamal Ansari[1]
Bineet Kumar Gupta[2] Alka Agrawal[1]* Raees Ahmad Khan[1]

[1]*Department of Information Technology, Babasaheb Bhimrao Ambedkar University, (A Central University), Lucknow, UP, India*
[2]*Department of Computer Application, Shri Ramswaroop Memorial University, Lucknow, UP, India*
[3]*College of Computer and Information Sciences, Prince Sultan University, Kingdom of Saudi Arabia*
* Corresponding Author: alka_csjmu@yahoo.co.in

Abstract: Nowadays, most of the cyber-attacks are initiated by extremely malicious programs known as Malware. Malwares are very vigorous and can penetrate the security of information and communication systems. While there are different techniques available for malware analysis, it becomes challenging to select the most effective approach. In this context, the decision-making process may be an efficient means of empirically assessing the impact of different methods for securing the web applications. In this research study, we have used a methodology that includes the integration of Fuzzy AHP and Fuzzy TOPSIS technique for evaluating the impact of different malware analysis techniques in web application perspective. This study uses different versions of a university's web application for evaluating the impact of several existing malware analysis techniques. The findings of the study show that the Reverse Engineering approach is the most efficient technique for analyzing complex malware. The outcome of this study would definitely aid the future researchers and developers in selecting the appropriate techniques for scanning the web application code and enhancing the security.

Keywords: Web application security, Security assessment, Malware analysis, Fuzzy logic, Fuzzy-AHP, Fuzzy-TOPSIS.

1. Introduction

Information and Communication Technology (ICT) has a vital role in all most all activities of human life today. However, this phenomenal growth in the use of ICT enabled facilities is under constant threat by the cyber-attacks. Furthermore, the security threats on information technology-based infrastructure have now become increasingly complicated and destructive. So much so, that the organizations have been compelled to close the interrupted processes or systems compromised by the hackers [1]. The last few years have seen a huge rise in the number of cyber-attacks due to malware. The malicious code compromises bugs in applications to hack devices, and lets the intruders capture the confidential information of the users. Modern anti-virus (AV) market is working on inventive mechanisms to contain the menace of advanced malware which is also known as the Advanced Persistent Threat (APT).

A recent report by Markets and Markets states that the worldwide business size for malware analysis is expected to rise from USD 3.0 billion in 2019 to USD 11.7 billion by 2024, at a Compound annual growth rate (CAGR) of 31.0 per cent from 2019 to 2024. The reason for this burgeoning growth is attributed to the rapid rise in false notifications [2]. Evidently, software privacy and security demand is going to become even more critical and a daunting problem for developers and security experts.

Malware is deliberately programmed to disable the computer system/network, gain identity theft, steal valuable information, damage computer

systems or even portable systems and impede important activity [3, 4]. This malicious code can easily hack the secret information of business organizations as well as personal computer systems. The complexity of malware has a major impact on the software application costs. Ultimately, the cost of the invasion varies depending on the effectiveness of the attack. Modern malware has evolved to be a highly efficient method for executing cyber-attacks [5]. Prevention of cyber-attacks on the internet services has now become a key issue for system engineers and research scientists. Software security organizations are continuously adding the modern virus signatures into the data repositories, with the aim to reach the milestone of zero-day cyber-attacks over the World Wide Web [6].

As malware continues to rise in its frequency and intelligence, it becomes progressively appealing to develop solutions with enhanced simplification for analyzing the previously undiscovered malware types. Such initiatives would provide adequate technical information to address cyber-attacks. This rise in cyber-attacks is due to both the free accessibility of malware varieties on the internet and integrated options available in open source operating systems like back track, Linux, Kali, Parrot, etc. Due to the extreme complicated nature of latest generation malware, conventional security system strategies seem unable to avoid advanced cyber-attacks [7]. Machine learning provides great potential to assist throughout the identification of intervention by stealth malware. However there are significant disconnections among machine learning related malware detection "solutions" proposed by the scientific community as well as those described in IDS software in operation [8]. Several business organizations are able to uncover that the detection of cyber-attacks one after the other isn't enough. Signature based approaches for anti-malware can only recognize previously proven approaches. Thus, depending on this kind of approach can contribute to the rapid spread of previously undiscovered risks.

Malware analysis is provided by software security companies as a product, or as a process. Malware detection tool helps software security professionals to identify and evaluate instances of malware and examine whether they are harmful or not. If detected to be harmful, they can sometimes be extracted from the network and stopped from any further propagation. Such technologies may be implemented in public or private organizations to handle threat warnings and eliminate more malware incidents. There are several malware analysis methods available to evaluate differences in malware activity [10, 11]. These kinds of methods are successful in conducting manual, fine-grained analysis of deceptive malware. Furthermore, specific supplementary details are needed, such as a collection of system calls relating to malicious actions or control-flow differences selection [12]. Malware analysis system includes obtaining a prospective data set of malware by a firewall; testing the prospective data set of malware with the help of a virtual machine to decide whether the prospective data set of malicious programs is malware; and inevitably creating a signature when the prospective data set of malicious programs is malware [9].

Researchers have used several methods to determine and evaluate the impact of systems and much research has been done to rank the software security attributes [13-15]. Some other researchers have also investigated about the protection strategies including the hierarchical characterization and acceptance [16, 17]. Nevertheless, authors of the present study work have not found any research that focuses on evaluating the impact of malware analysis techniques for web applications with the help of Fuzzy based Decision-Making Process. That is why our research, in general, evaluated the impact of several malware analysis techniques by using the Fuzzy-TOPSIS method. The conclusions of this research study would certainly help the researchers and developers in choosing the most appropriate techniques for scanning the web application code, thereby enhancing the security.

The rest of this study is organized as follows: In Section 2, the paper describes the overview of malware analysis. Section 3 discusses the different related works. Section 4 defines the methodology. In Section 5, the impact of malware analysis techniques for web application has been evaluated with the help of Fuzzy ANP-TOPSIS. Sensitivity analysis and comparisons of the outcomes have been enlisted in Section 6 of this paper. Finally, discussion and conclusions are chronicled in Section 7.

2. Malware analysis

Once the presence of malware in the target system is detected, it can be analyzed through several techniques and methods. Malware analysis can be grouped into three broad categories:

- Static Malware Analysis
- Dynamic Malware Analysis
- Reverse Engineering

2.1 Static malware analysis

Static malware analysis tests a file of malware deprived of actually running the script. It is the best way to detect malware, as it may corrupt the machine

by running the script. Static malware analysis collects knowledge from malware at its most primitive form, before seeing the script. The static process is the most common technique to determine whether a file is safe from malware or not. Static malware analysis of data flow as well as other statistical features is derived without actually executing the software. To construct an intermediary description of the binary code, reverse engineering techniques such as disassembly and decompiling are used. Static methods have their own restrictions; they cannot identify malware which utilizes runtime packaging or various anti-reversal and anti-disassembly strategies like code permutation, garbage code insertion (GCI), encryption, compression, etc. [18]. Some of the common malware analysis techniques are virus scan, analysis of memory/os artifacts, PE file scanning, and disassembly of code.

Gu et al. [19] used the study of wavelets to derive features from actual data. Such features can be used to determine whether malicious code has been inserted in the compound text. Das et al. [20] introduced the frequency-centric feature development model with the help system of the patterns of recognized malware and benign samples. Then, in Field Programmable Gate Array (FPGA), they created a machine learning method by using a multilayer perceptron to train the classifier using such functions. At runtime, the qualified classifier is used with early prediction to identify the unknown samples as malware or benign. The findings indicate that to respond to new malware specimens, their approach can maintain high classification precision, quick detection, lower power consumption and versatility for simple upgrading of capabilities.

Kolosnjaji et al. [21] investigated the vulnerability of malware detection methods that use deep networks to understand the actual bytes. They proposed a gradient-based attack sufficient of evading a newly developed deep suitable network. This was done by modifying only a few different bytes at the end of each specimen of malware while maintaining its malicious function. The results indicated that their adversarial malware binaries are highly likely to escape the threatened network, yet less than 1 percent of their bytes were changed. Hashemi and Hamzeh proposed a new approach for detecting the unidentified micro-pattern based malware in executable files [22].

2.2 Dynamic malware analysis

Dynamic malware analysis approach includes executing the malware as well as analyzing its actions on the device, where the device is installed in a remote and separated setting. Dynamic malware analysis research allows one to remove the virus, make successful signatures, or do even both. For dynamic malware analysis, the binary malware is executed underneath the setting of the virtual machine (VM). The malware binary's execution time behavior is observed for example, depending upon API calls or device call indications triggered by the binary. However, there are a few restrictions in this technique. It is possible that the Dynamic techniques may not be able to examine the provided malware sample's technical capabilities because these techniques do not protect the entire actions throughout that specific operation [18]. There are some general approaches for dynamic malware analysis which are API call analysis, Malware Sandboxing and manually analysis network services. Amal et al. [23] proposed a malware analysis and classification approach named AMAL. It is an automatically generated, behavior-based malware detection and classifying program that fixes the existing systems vulnerabilities. The AMAL approach is generated by two sub-systems. These sub-systems are AutoMal and MaLabel. AutoMal offers capabilities to gather low-granularity behavioral objects that describe file system, memory, network, and registry malware use and does so by executing samples of malware in virtualized settings. On the other side, MaLabel utilizes these objects to construct symbolic features, utilize them to develop classifiers programmed by systematically screened training specimens. Thereafter these classifiers are used to categorize specimens of malware into specific behavioral classes.

Fan et al. [24] used hooking approaches to track the complex signatures which are being hidden by malware. The behavioral variations among malware and benign codes are then measured with the help of data mining methods to classify the malware. The research findings indicate that with just 80 attributes, their detection rate exceeds beyond 95 per cent. They showed that their system can obtain a low difficulty and strong detection rate.

Galal et al. [25] proposed a model-based approach of behavior-based features which defined fraudulent activity demonstrated by malware example. They first conducted dynamic malware analysis on a comparatively recent malware sample within a managed virtual setting and gathered traces of malware-invoked API calls to extract their recommended model.

Sihwail et al. [26] presented the integrated malware detection method which implements memory forensics to retrieve harmful objects from memory as well as integrates those in a complex

application of features retrieved during malware implementation. The pre-modelling techniques were also used for function engineering prior to training and evaluating the samples on the machine learning approach.

2.3 Reverse engineering

Reverse engineering for malware analysis requires disassembling a software program and also sometimes decompiling it. Throughout this mechanism, binary instructions are translated to code mnemonics so that the programmers can look at how the software is doing and what mechanisms it is affecting. By analysing the information thus obtained, the engineers can formulate strategies which can minimize the harmful effects intended for the software. Reverse engineering compiled executables is indeed a challenge with such a steeper learning curve. The function of interpreting assembly into some kind of set of abstractions, which reflect the overall flow of a system, complicates it. Almost all of the stages include identifying interesting fields of an executable and assessing its usefulness as a whole [27]. Reverse engineering malware does not have a common standard strategy, since each malware needs a special approach. Through packaging the file, advanced modern malware will avoid conventional antivirus identification. The source code for the software has also been optimized and the actual raw executable of a script was distributed to our computers. Moreover, there is no access to the source code until the executable has been decompiled. Therefore, it is possible to transfer the executable with machine code, also known as Op-Codes, to assembly instructions and to understand the disassembled assembly instructions. Reverse engineering and disassembler facilitate the malware analysis by simplifying the process for the reverse engineers [28]. Some of the common approaches for malware analysis are Binary code analysis (Disassembly), debugging and string analysis.

Rahimian et al. [29] introduced the findings of Citadel reverse engineering and also provided new perspective into the malware's features, underlying principles, and open source components. To speed up the reverse engineering process, they further proposed a technique of clone-based analysis. In another study, Zimba et al. [30] conducted reverse engineering to decode the ransomware code. Outcome from their study shows that despite robust encryption, the ransomware utilizes the very same attack mechanism and cryptographic abstractions as with other families in the wild. Fig. 1 shows the different malware analysis approaches categories and their respective sub-categories.

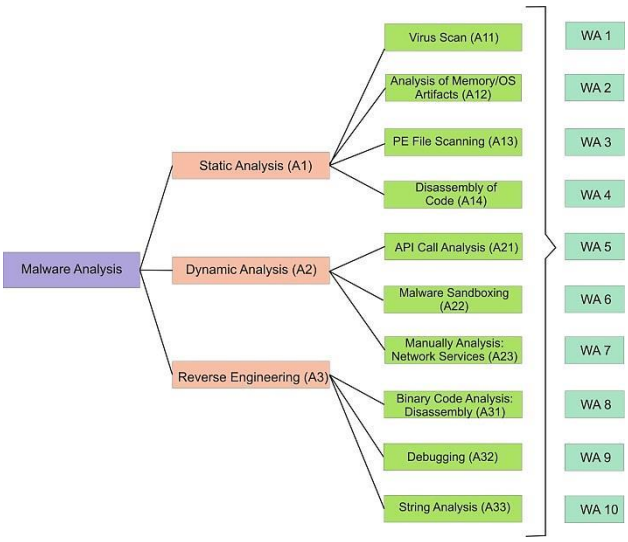

Figure. 1 Different malware analysis approaches

3. Related work

Several studies have been done on fuzzy AHP-TOPSIS Multiple-criteria decision-making (MCDM) strategies [31]. However, none of these studies focus on evaluating the impact of malware analysis techniques for web applications with the help of Fuzzy based Decision-Making Process.

Ballı and Korukoğlu [32] constructed a fuzzy decision model to pick suitable operating system (OS) for the personal computers of different organizations by keeping into account individual decision-makers' feedback. The suggested procedure is based on approaches such as Fuzzy Analytic Hierarchy Process (FAHP) and TOPSIS Technique for Order of Preference by Similarity to Ideal Solution). FAHP approach is often used by decision-makers to evaluate the weights of the parameters and then OS rankings are calculated by TOPSIS method. Demirtaş et al. [33] developed a fuzzy AHP-TOPSIS decision-making methodology to pick the most efficient ERP program for an urban passenger transportation company.

Zaidan et al. [34] measured and picked open-source electronic medical record (EMR) application packages focused on decision making by multi-criteria. An empirical analysis was carried out, and a collection of open-source EMR application packages was installed locally on different virtual machines (VM) for a closer evaluation of the systems. Further, different variables were defined as the basis for the assessment, and the systems were chosen based on a collection of metric results that use the Integrated AHP and TOPSIS methodology.

Alenezi et al. [35] developed the Fuzzy AHP-TOPSIS based hybrid approach for evaluating security architecture tactics and their attributes. They tested the efficacy of this method on a case study of real-time web application of Babasaheb Bhimrao Ambedkar University, Lucknow, India. Furthermore, various university software applications were also used to verify the findings.

Kumar et al. [36] identified usability-security as a challenge with numerous attributes referring to it. In addition, they investigated that the question needs to be evaluated for the convenience of the end consumer. In such a background, the research study also proposed the measuring methodology of Fuzzy AHP-TOPSIS to evaluate the usability-security of the web based application as well as identify the most prioritized attribute corresponding to the creation of usable-security for web application development environment.

4. Integrated fuzzy AHP-TOPSIS method

There are numerous researchers who have conducted security and mitigation analysis. A modern strategy for maximum security, social and environmental sustainability is the development of web based application protection by sustainability [37]. Therefore, Multi-Criteria Group Decision Making (MCGDM) issues are commonly reported in operation to achieve the objectives according to the requirements of the consumer and the responsiveness of the details. The literature includes many approaches which can be used to solve these issues [38]. AHP is a stronger technique than any of the other MCDA methods for determining the positive and negative qualities of the variables.

However, AHP could not overcome the inherent ambiguity and imprecision that the decision makers faced in analyzing the sensitivity of the objective data. In this paper, the researchers find that experts combined the Fuzzy concept with AHP because the modern world is extremely ambiguous in evaluating inaccurate real-world concerns [39]. Additionally, the AHP approach is focused on quite unpredictable decision level, but still the Fuzzy AHP seems to have some flaws [38, 39]. Consequently, an integrated AHP and TOPSIS Fuzzy methodology is a novel tool which can assist in the comprehensive assessment of alternatives on several parameters. The following sub sections describe the step-by-step process of Fuzzy AHP-TOPSIS method.

4.1 Fuzzy AHP

Fuzzy AHP is a powerful tool for analyzing hard decision challenges through which each complicated problem can be evaluated by specific graded goal rates. The challenge is differentiated with the help of Fuzzy-AHP to turn it into a tree like structure. Therefore, AHP has been used like a decision-making method to predict rank numbers for various alternatives, including multiple parameters expressed in such a hierarchical structure [40]. To optimize the efficacy of Fuzzy AHP method for a more feasible perspective, the Fuzzy AHP focuses on the Fuzzy Numerical interval of triangular Fuzzy Numbers. These numbers are introduced to decide the weights of interpretative components. Saaty was the first to propose the AHP process [41]. AHP process utilizes only the matrix of the pair-wise analysis to tackle the inaccuracy in challenges of decision labeling in multi-criteria [42].

The model suggested here allows the use of the triangular fuzzy figures to define the linguistic parameters and to incorporate with AHP fuzzy procedures. Because of the inaccuracy and ambiguity, Zadeh developed the fuzzy based set theory to cope with uncertainty [43]. Fig. 1 shows the tree layout for Fuzzy AHP TOPSIS. This tree layout can be designed by collating the viewpoints and responses of the domain specialists and experts through questionnaires or brainstorming. The next stage is to develop the Triangular Fuzzy Number (TFN) from the Hierarchy of the Tree. A pair-wise assessment of each category of defined goals plays a key role with the aid of one criterion's effect on other criterion.

Professionals transform this further into exact figures and TFN through linguistic values. This research paper also utilizes the TFN, which ranges from 0 to 1 [44]. The reason for using the TFN is the mathematical flexibility of the triangular fuzzy participation functions which can interact with fuzzy data [45]. In addition, linguistic factors are categorized as equally important, weakly important and so on. The precise figures are categorized as 1,2, ... 9. However, a fuzzy number M on F is named TFN, if its participation functions are specified in Eq. (1, 2):

$$\mu_a(x) = F \rightarrow [0,1] \tag{1}$$

$$\mu_a(x) = \begin{cases} \frac{x}{mi-l} - \frac{l}{mi-l} & x \in [l, mi] \\ \frac{x}{mi-u} - \frac{u}{mi-u} & x \in [mi, u] \\ 0 & Otherwise \end{cases} \tag{2}$$

Here l, mi, and u are specified consequently in the triangular membership function as a lower limit, middle limit and upper limit. The following Fig. 2 shows a TFN.

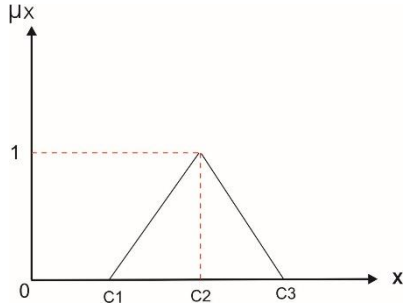

Figure. 2 Triangular fuzzy numbers

Table 1. TFN scale

Saaty Scale Definition	Fuzzy Triangle Scale	
1	Equally important	(1 ,1, 1)
3	Weakly important	(2 ,3, 4)
5	Fairly important	(4 ,5, 6)
7	Strongly important	(6 ,7, 8)
9	Absolutely important	(9 ,9, 9)
2	Intermittent values between two adjacent scales	(1 ,2, 3)
4		(3 ,4, 5)
6		(5 ,6, 7)
8		(7 ,8, 9)

A TFN could be interpreted as (l, mi, u). Domain Experts assigned points as per the scale provided in Table 1 to the variables that influence the scores in a numerical manner.

The formulas (3-6) are provided when the quantitative factors are changed to TFN [38, 39, 44] which are defined as (lij, miij, uij) where, lij is a lower value, miij is a middle value and uij is a case of the highest point. Alternatively, it acknowledges TFN [ŋij] as:

$$\eta_{ij} = \left(l_{ij}, mi_{ij}, u_{ij}\right) \tag{3}$$

where $l_{ij} \leq mi_{ij} \leq u_{ij}$

$$l_{ij} = min\left(J_{ijd}\right) \tag{4}$$

$$mi_{ij} = \left(J_{ij1}, J_{ij2}, J_{ij3}\right)^{\frac{1}{x}} \tag{5}$$

$$andu_{ij} = max\left(J_{ijd}\right) \tag{6}$$

In the Eq. (3-6), Jijk implies the comparative significance of the values among two factors that specialist d gives, while I and j denote a pair of factors that specialists determine. ŋij is calculated on the basis of the statistical mean of experts' views for a particular contrast. The statistical mean is capable of accurately integrating and indicating practitioners' agreement, and signifies the lowest and highest ratings, consequently, for the relative value of the two variables. Alternatively, Eq. (7-9) endorse composite TFN values. Consider M1, M2, M1= (l1, mi1, u1) and M2= (l2, mi2, u2). The operating rules on them are as follows:

$$\begin{aligned}&(l_1, mi_1, u_1) + (l_2, mi_2, u_2)\\ &= (l_1 + l_2, mi_1 + mi_2, u_2 + u_2)\end{aligned} \tag{7}$$

$$\begin{aligned}&(l_1, mi_1, u_1) \times (l_2, mi_2, u_2)\\ &= (l_1 \times l_2, mi_1 \times mi_2, u_1 \times u_2)\end{aligned} \tag{8}$$

$$(l_1, mi_{1,} u_1)^{-1} = \left(\frac{1}{u_1}, \frac{1}{mi_1}, \frac{1}{l_1}\right) \tag{9}$$

Using Eq. (10), a fuzzy pair-wise comparative matrix is built in the type of n x n matrix upon obtaining the TFN values for each pair of comparisons.

$$\widetilde{A^d} = \begin{bmatrix} \tilde{k}_{11}^d & \tilde{k}_{12}^d \ldots. & \tilde{k}_{1n}^d \\ \tilde{k}_{21}^d & \tilde{k}_{22}^d \ldots. & \tilde{k}_{2n}^d \\ \cdots & \cdots & \cdots \\ \tilde{k}_{n1}^d & \tilde{k}_{n2}^d \ldots & \tilde{k}_{nn}^d \end{bmatrix} \tag{10}$$

Where $\widetilde{k_{ij}^k}$ signifies the choice of the dth decision-makers for the ith criteria over most of the Jth criteria. When more than one decision-maker is available, then the average of each decision-makers' priorities is calculated by using Eq. (11).

$$\tilde{k}_{ij} = \sum_{d=1}^{d} \tilde{k}_{ij}^d \tag{11}$$

Further, the next stage is to modify the matrix for pair-wise comparison for all variables in the hierarchy using equation based on average preferences (12).

$$\tilde{A} = \begin{bmatrix} \widetilde{k_{11}} & \ldots & \widetilde{k_{1n}} \\ \cdots & \ddots & \cdots \\ \widetilde{k_{n1}} & \cdots & \tilde{k}_{nn} \end{bmatrix} \tag{12}$$

Afterwards, the researchers use the geometric mean procedure as shown in Eq. (13) to define each factor's fuzzy geometric mean and fuzzy weights.

$$\tilde{p}_i = \left(\prod_{j=1}^{n} \tilde{k}_{ij}\right)^{1/n}, i = 1,2,3 \ldots n \tag{13}$$

The next stage is to finalize the factor's fuzzy weight using Eq. (14). Where, the operator $\oplus$ represents the addition of fuzzy matrices, and the operator $\otimes$ represents the multiplication of fuzzy matrices.

$$\widetilde{w}_i = \tilde{p}_i \otimes (\tilde{p}_1 \oplus \tilde{p}_2 \oplus \tilde{p}_3 \ldots. \oplus \tilde{p}_n)^{-1} \tag{14}$$

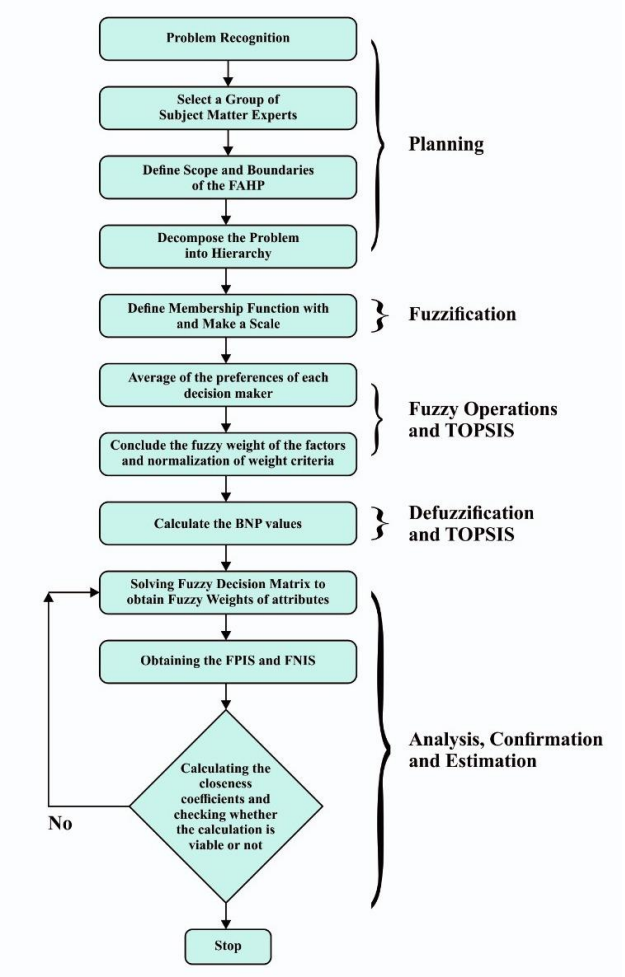

Figure. 3 Flow chart of fuzzy AHP-TOPSIS method

Therefore, with the aid of Eq. (15-16), the average and normalized weight criteria are determined.

$$M_i = \frac{\tilde{w}_1 \oplus \tilde{w}_2 \ldots\ldots \oplus \tilde{w}_n}{n} \tag{15}$$

$$Nr_i = \frac{M_i}{M_1 \oplus M_2 \oplus \ldots\ldots \oplus M_n} \tag{16}$$

In addition, by using Eq. (17), the Center of Area (COA) approach is then used to determine the Best Non-fuzzy Performance (BNP) value of the fuzzy weights of each variable.

$$BNPwD1 = \frac{[(uw1 - lw1) + (miw1 - lw1)]}{3} + lw1 \tag{17}$$

4.2 Fuzzy TOPSIS

TOPSIS considers a multi-criteria decision-making issue of *m* alternatives like a geometric structure with *m* points in the *n-dimensional* space of component. For TOPSIS, the approach used in this research paper is based on the assumption that, for higher and lower ideal solutions, a specified alternative has the shortest and the farthest range from the positive-ideal solution as well as the negative-ideal solution simultaneously [46]. Professionals find difficulty in assigning a particular output ranking to an alternative with reference to factor, as shown by Shadbegian and Gray [47]. In compatibility with the actual-world fuzzy setting, this approach applies fuzzy numbers to reflect the relative value of the factor rather than specific numbers. Furthermore, the Fuzzy AHP-TOPSIS approach is especially appropriate for finding solutions of group decision-making in fuzzy settings. Fig. 3 shows the overall weight acquisition process and the feasibility estimation of Fuzzy AHP-TOPSIS methods. Fig. 3 below demonstrates the step-by-step procedure of the Fuzzy AHP-TOPSIS system.

Table 2. Linguistic scales for the rating

Linguistic Variable	Corresponding Triangular Fuzzy Number
Very poor (VP)	(0, 1, 3)
Poor (P)	(1, 3, 5)
Fair (F)	(3, 5, 7)
Good (G)	(5, 7, 9)
Very good (VG)	(7, 9,10)

First, we determine weights of the evaluation factor. The present study applies Fuzzy AHP to determine fuzzy preference weights with the help of Eq. (1-16). Further, the researchers create the fuzzy decision matrix and choose the appropriate linguistic variables as alternatives for the criteria with the help of Eq. (18) and Table 2.

$$\tilde{K} = \begin{array}{c} \\ A_1 \\ \ldots \\ A_m \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & \ldots\ldots & C_n \end{array} \\ \begin{bmatrix} \tilde{x}_{11} & \cdots & \tilde{x}_{1n} \\ \cdots & \ddots & \cdots \\ \tilde{x}_{m1} & \cdots & \tilde{x}_{mn} \end{bmatrix} \end{array} \tag{18}$$

Where, $\tilde{x}_{ij} = \frac{1}{D}\left(\tilde{x}_{ij}^1 \cdots \oplus \tilde{x}_{ij}^d \oplus \cdots \tilde{x}_{ij}^D\right)$, and $\tilde{x}_{ij}^d$ is the performance rating of the alternative Ai with respect to factor Cj estimated by the dth practitioner and $\tilde{x}_{ij}^d = (l_{ij}^d, mi_{ij}^d, u_{ij}^d)$.

Next step is to normalize the fuzzy decision matrix with the assistance of Eq. (19). The normalized fuzzy decision matrix is represented by $\tilde{P}$ and is depicted as follows.

$$\tilde{P} = \left[\tilde{p}_{ij}\right]_{m \times n} \tag{19}$$

Thereafter, the normalization process can be achieved with the help of Eq. (20).

$$\tilde{p}_{ij} = \left(\frac{l_{ij}}{u_j^+}, \frac{mi_{ij}}{u_j^+}, \frac{u_{ij}}{u_j^+}\right),$$
$$u_j^+ = max\{u_{ij}, i = 1,2,3..n\} \quad (20)$$

Alternatively, we can set the best desired level u_j^+ and j = 1, 2, . . . , n is equal to 1; otherwise, the worst is 0. The normalized $\tilde{p}_{ij}$ continues to be triangular fuzzy numbers(TFNs). For trapezoidal fuzzy numbers, the normalization process can be performed in the similar manner. The weighted fuzzy normalized decision matrix ($\tilde{Q}$) is quantified with the help of Eq. (21).

$$\tilde{Q} = \left[\tilde{q}_{ij}\right]_{m\times n} i = 1,2,..m;\ j = 1,2,3 \ldots n \quad (21)$$

Where, $\tilde{q}_{ij} = \tilde{p}_{ij} \otimes \tilde{w}_{ij}$ and then, define the Fuzzy Positive-Ideal Solution (FPIS) and Fuzzy Negative-Ideal Solution (FNIS). The weighted normalized fuzzy decision matrix indicates that the elements $\tilde{q}_{ij}$ are normalized positive TFN and their ranges belong to the closed interval [0, 1]. Thereafter, we can describe the FPIS A+ (aspiration levels) and FNIS A− (the worst levels) as shown in Eqs. (22) and (23).

$$A^+ = \left(\tilde{q}_{1,\cdots\ldots}^*\tilde{q}_{j,\cdots\ldots}^*\tilde{q}_{n,}^*\right) \quad (22)$$

$$A^- = \left(\tilde{q}_{1,\cdots\ldots}^*\tilde{q}_{j,\cdots\ldots}^*\tilde{q}_{n,}^*\right) \quad (23)$$

Where, $\tilde{q}_1^* = (1,1,1)\otimes\tilde{w}_{ij} = \left(Lw_j, Mw_j, Hw_j\right) and \tilde{q}_{ij}^- = (0,0,0)$, $j = 1,2,3 \ldots n$. Thereafter, the distance of each alternative is calculated. The distances ($\tilde{d}_i^+ and\, \tilde{d}_i^-$) of each alternative from A+ and A- can be estimated by using the area compensation technique as shown in Eqs. (24) and (25).

$$\tilde{d}_i^+ = \sum_{j=1}^{n} d\left(\tilde{q}_{ij}, \tilde{q}_{ij}^*\right) i = 1,2,..m;\ j = 1,2,3 \ldots n \quad (24)$$

$$\tilde{d}_i^- = \sum_{j=1}^{n} d\left(\tilde{q}_{ij}, \tilde{q}_{ij}^*\right) i = 1,2,..m;\ j = 1,2,3 \ldots n \quad (25)$$

In the ensuing step, we find the closeness coefficients (relative gaps–degree) and develop the alternatives to achieve the aspiration levels in each factor. Chou et al. proposed that $C\tilde{C}_i$ is accurate for evaluating the fuzzy gaps–degree on the basis of the fuzzy closeness coefficients to improve the alternatives [48]. Once $\tilde{d}_i^+$ and $\tilde{d}_i^-$ of each alternative have been evaluated, the similarities to the ideal solution are calculated. This step solves the similarities to an ideal solution as shown in Eq. (26).

$$C\tilde{C}_i = \frac{\tilde{k}_i^-}{\tilde{k}_i^+ + \tilde{k}_i^-} = 1 - \frac{\tilde{k}_i^+}{\tilde{k}_i^+ + k_i^-},\ i = 1,2,\ldots.,m \quad (26)$$

Where, $\frac{\tilde{k}_i^-}{\tilde{k}_i^+ + \tilde{k}_i^-}$ is defined as fuzzy satisfaction degree in the ith alternative and $\frac{\tilde{k}_i^+}{\tilde{k}_i^+ + \tilde{k}_i^-}$ is the fuzzy gap–degree in the ith alternative. On the basis of these, the ranks of the alternatives are achieved.

This sub-section discusses different statistical findings of integrated fuzzy AHP-TOPSIS model implementation. Security experts usually do a behavior-based research of malware to analyze about previously identified examples of malware or family of malware. To achieve this, it is important to identify and characterize questionable behaviors from large sets of signs of implementation [49]. IT security experts and academicians face a complicated task of assessing the impact of malware analysis techniques numerically in current cyber-attack setting. The identification of malware analysis techniques is the most essential step to preserve the infrastructure of information and communication technology. An effective impact analysis of malware techniques from the security practitioners will provide reliable and efficient problem solving strategy. To accomplish this objective, in our research paper, we have used an emphatically established and validated decision-making strategy, the integrated fuzzy AHP-TOPSIS. This technique is conversant for prioritizing the malware analysis techniques based on their impact evaluation in current cyber security setting. For eliciting a more convincing outcome, we took suggestions from 70 IT security experts who come from different software industries and educational backgrounds. The information outsourced from these specialists was collected for our empirical investigations. Eqs. (1)–(21) are used according to Fig. 3 to determine the impact of the mentioned malware analysis techniques.

We enlisted the tabulations of Table 1 as well as Eqs. (1)–(9) to transform linguistic factors into quantitative values and TFN figures. This was done to determine the variables and calculate the findings. Similarly, the pair-wise comparative matrix of the attributes at level 1 is developed by using Eq. (10) as shown in Table 3. Likewise, the composite pair-wise comparative matrix for the level 2 and level 3 hierarchies has been collated in Tables 4–13.

We compute the measurement of the weights with the help of Eqs. (11)–(13). Likewise, the remaining p and I can be collected. Eqs. (14)–(16) are used here for weight measurement of each factor. However, we determined the BNP significance of factors via Eq. (17). The global weights have always been determined with every second-layer element as shown in Table 8.

In Table 8, several factors are replicated; however, the effect they provide to their significantly high layer component is complex. To be more comprehensive, an integration to measure the weights of the factor of each point is performed. Furthermore, Table 9 demonstrates the final dependent weights with the help of the hierarchy.

5. Statistical data analysis

This sub-section discusses different statistical findings of integrated fuzzy AHP-TOPSIS model implementation. Security experts usually do a behavior-based research of malware to analyze about previously identified examples of malware or family of malware. To achieve this, it is important to identify and characterize questionable behaviors from large sets of signs of implementation [49]. IT security experts and academicians face a complicated task of assessing the impact of malware analysis techniques numerically in current cyber-attack setting. The identification of malware analysis techniques is the most essential step to preserve the infrastructure of information and communication technology. An effective impact analysis of malware techniques from the security practitioners will provide reliable and efficient problem solving strategy. To accomplish this objective, in our research paper, we have used an emphatically established and validated decision-making strategy, the integrated fuzzy AHP-TOPSIS. This technique is conversant for prioritizing the malware analysis techniques based on their impact evaluation in current cyber security setting. For eliciting a more convincing outcome, we took suggestions from 70 IT security experts who come from different software industries and educational backgrounds. The information outsourced from these specialists was collected for our empirical investigations. Eqs. (1)–(21) are used according to Fig. 3 to determine the impact of the mentioned malware analysis techniques.

We enlisted the tabulations of Table 1 as well as Eqs. (1)–(9) to transform linguistic factors into quantitative values and TFN figures. This was done to determine the variables and calculate the findings. Similarly, the pair-wise comparative matrix of the attributes at level 1 is developed by using Eq. (10) as shown in Table 3. Likewise, the composite pair-wise comparative matrix for the level 2 and level 3 hierarchies has been collated in Tables 4–13.

We compute the measurement of the weights with the help of Eqs. (11)–(13). Likewise, the remaining p and I can be collected. Eqs. (14)–(16) are used here for weight measurement of each factor. However, we determined the BNP significance of factors via Eq. (17). The global weights have always been determined with every second-layer element as shown in Table 8.

In Table 8, several factors are replicated; however, the effect they provide to their significantly high layer component is complex. To be more comprehensive, an integration to measure the weights of the factor of each point is performed. Furthermore, Table 9

Table 3. Fuzzy-aggregated pair-wise comparison matrix at level 1

	A1	**A2**	**A3**
A1	1.0000,1.0000 ,1.0000	0.2300, 0.2800, 0.3600	0.3000, 0.4400, 0.8000
A2	-	1.0000,1.0000 ,1.0000	0.6600, 1.1700, 1.6900
A3	-	-	1.0000,1.0000 ,1.0000

Table 4. Fuzzy aggregated pair-wise comparison matrix at level 2 for static analysis

	A11	**A12**	**A13**	**A14**
A11	1.0000,1. 0000,1.00 00	0.6900, 0.8900, 1.1000	0.2300, 0.2800, 0.3600	0.7000, 0.9500, 1.3500
A12	-	1.0000,1. 0000,1.00 00	0.4900, 0.6400, 1.0000	0.2700, 0.3500, 0.5200
A13	-	-	1.0000,1. 0000,1.00 00	1.0000, 1.3200, 1.5500
A14	-	-	-	1.0000,1. 0000,1.00 00

Table 5. Fuzzy aggregated pair-wise comparison matrix at level 2 for dynamic analysis

	A21	**A22**	**A23**
A21	1.0000, 1.0000, 1.0000	0.6600, 1.1700, 1.6900	1.1500, 1.4400, 1.7000
A22	-	1.0000, 1.0000, 1.0000	1.0000, 1.5200, 1.9300
A23	-	-	1.0000,1.0000 ,1.0000

Table 6. Fuzzy aggregated pair-wise comparison matrix at level 2 for reverse engineering

	A31	**A32**	**A33**
A31	1.0000,1.0000 ,1.0000	1.1900, 1.5800, 2.1500	0.4900, 0.6400, 1.0000
A32	-	1.0000,1.0000 ,1.0000	0.2200, 0.2900, 0.4200
A33	-	-	1.0000, 1.0000, 1.0000

Table 7. Combined pairwise comparison matrix at level 1

	A1	**A2**	**A3**	**Weights**
A1	1.0000	1.1730	0.4940	0.2749
A2	0.8525	1.0000	1.1720	0.3296
A3	2.0243	0.8532	1.0000	0.3955
				C.R.=0.0488

Table 8. Aggregated pair-wise comparison matrix at level 2 for static analysis

	A11	**A12**	**A13**	**A14**	**Weights**
A11	1.0000	0.8920	1.1730	0.9940	0.2463
A12	1.1211	1.0000	0.6910	0.3720	0.1820
A13	0.8525	1.4472	1.0000	1.2980	0.2724
A14	1.0061	2.6882	0.7704	1.0000	0.2993
					CR= 0.0349

Table 9. Aggregated pair-wise comparison matrix at level 2 for dynamic analysis

	A21	**A22**	**A23**	**Weights**
A21	1.0000	1.1720	1.3630	0.3843
A22	0.8533	1.0000	1.4910	0.3562
A23	0.7337	0.6707	1.0000	0.2595
				CR= 0.0025

Table 10. Aggregated pair-wise comparison matrix at level 2 for reverse engineering

	A31	**A32**	**A33**	**Weights**
A31	1.0000	1.6330	0.6910	0.3159
A32	0.6124	1.0000	0.3030	0.1731
A33	1.4472	3.3003	1.0000	0.5110
				CR= 0.0052

demonstrates the final dependent weights with the help of the hierarchy.

Therefore, after a scrutiny of different defined criteria, we determined the importance of several malware analysis techniques in alternative options. Ten consecutive web application projects were obtained from a local University in Lucknow, India, to estimate the security. Alternatives WA1, WA2, WA3 ... WA10, which describe the institutional services initiative, are all very responsive programs. With the aid of Table 2 and Eq. (4–9), we took the

Table 11. Summary of the results

Level 1 Methods	**Local Weights of Level 1**	**Level 2 Methods**	**Local Weights of Level 2**	**Overall Weights**	**Overall Ranks**
A1	0.2749	A11	0.2463	0.0677	9
		A12	0.1820	0.0500	10
		A13	0.2724	0.0749	7
		A14	0.2993	0.0823	6
A2	0.3296	A21	0.3843	0.1267	2
		A22	0.3562	0.1174	4
		A23	0.2595	0.0855	5
A3	0.3955	A31	0.3159	0.1250	3
		A32	0.1731	0.0685	8
		A33	0.5110	0.2021	1

inputs from the six projects' technical results, as shown in Table 11. By using the Eqs. (18)–(20), we calculated the regularized Fuzzy decision matrix as provided in Table 12. Then, we calculated the weighted normalized Fuzzy decision matrix with the help of Eq. (21) as shown in Table 13. By using Eqs. (22)–(26), we evaluated the fuzzy degree of satisfaction and fuzzy gap, which can be seen in Table 14.

6. Validation of findings

6.1 Sensitivity analysis

An analysis of the obtained outcome from different perspectives is imperative for any scientific research paper. Sensitivity analysis process is one of the most effective and efficient methods in order to authenticate the validity of findings [15]. Moreover if the factors are changed, sensitivity analysis offers a mechanism for research scientists to examine their collected outcomes. In this research paper, the proposed analysis used twenty experiments to analyze sensitivity, since the last hierarchy level has six factors. The sensitivity weights of each factor are different at the time of analysis, and the other factors weights and degree of satisfaction are constant at the very same time. The estimated effects of the sensitivity analysis are shown in Table 16.

6.2 Comparative analysis

Comparative analysis is an integral process for corroborating the efficacy of the techniques employed by a research scientist [15]. We also performed a comparative analysis of the findings with another related technique called the classical AHP-TOPSIS. For gauging the findings through the classical AHP-TOPSIS method, we used the same data for estimation. Fig. 4 demonstrates the radar

Table 12. Subjective cognition results of evaluators in linguistic terms

	WA1	WA2	WA3	WA4	WA5	WA6	WA7	WA8	WA9	WA10
A11	5.7320, 7.7320, 9.2710	4.2700, 6.2700, 8.1800	4.0900, 6.0900, 8.0900	1.1800, 3.0000, 5.0000	5.1800, 7.1800, 8.8200	2.0900, 4.0900, 6.0900	1.7300, 3.5500, 5.5500	1.1800, 3.0000, 5.0000	5.1800, 7.1800, 8.8200	2.8200, 4.8200, 6.8200
A12	5.0000, 7.0000, 8.4500	5.7300, 7.7300, 9.0000	4.2700, 6.2700, 7.9100	1.1800, 3.0000, 5.0000	4.1800, 6.0900, 7.6400	3.5500, 5.5500, 7.2700	0.8200, 2.4500, 4.4500	1.1800, 3.0000, 5.0000	4.1800, 6.0900, 7.6400	2.8200, 4.8200, 6.6400
A13	5.1800, 7.1800, 8.6400	5.3600, 7.3600, 8.7300	5.3600, 7.3600, 8.7300	0.7300, 2.4500, 4.4500	5.0000, 7.0000, 8.4500	4.8200, 6.8200, 8.2700	1.0000, 2.6400, 4.6400	0.7300, 2.4500, 4.4500	5.0000, 7.0000, 8.4500	2.8200, 4.8200, 6.7300
A14	5.7300, 7.7300, 9.0900	5.3600, 7.3006, 8.7300	5.5500, 7.5500, 8.9100	0.6400, 2.2700, 4.2700	5.3600, 7.3600, 8.7300	4.0900, 6.0900, 7.7300	0.7300, 2.2700, 4.2700	0.6400, 2.2700, 4.2700	5.3600, 7.3600, 8.7300	2.0900, 3.9100, 5.8200
A21	6.2700, 8.2700, 9.4500	5.7300, 7.7300, 9.0000	5.3600, 7.3600, 8.7300	0.3600, 1.7300, 3.7300	6.2700, 8.2700, 9.4500	3.1800, 5.1800, 7.0000	1.6400, 3.3600, 5.3600	0.3600, 1.7300, 3.7300	6.2700, 8.2700, 9.4500	3.9100, 5.9100, 7.5500
A22	5.7300, 7.7300, 9.0900	5.3600, 7.3006, 8.7300	5.5500, 7.5500, 8.9100	1.6400, 3.5500, 5.5500	5.3600, 7.3600, 8.7300	4.0900, 6.0900, 7.7300	0.7300, 2.2700, 4.2700	1.6400, 3.5500, 5.5500	5.3600, 7.3600, 8.7300	2.0900, 3.9100, 5.8200
A23	5.0000, 7.0000, 8.4500	5.7300, 7.7300, 9.0000	4.2700, 6.2700, 7.9100	1.1800, 3.0000, 5.0000	4.1800, 6.0900, 7.6400	3.5500, 5.5500, 7.2700	0.8200, 2.4500, 4.4500	1.1800, 3.0000, 5.0000	4.1800, 6.0900, 7.6400	2.8200, 4.8200, 6.6400
A31	5.1800, 7.1800, 8.6400	5.3600, 7.3600, 8.7300	5.3600, 7.3600, 8.7300	0.7300, 2.4500, 4.4500	5.0000, 7.0000, 8.4500	4.8200, 6.8200, 8.2700	1.0000, 2.6400, 4.6400	0.7300, 2.4500, 4.4500	5.0000, 7.0000, 8.4500	2.8200, 4.8200, 6.7300
A32	5.7300, 7.7300, 9.0900	5.3600, 7.3006, 8.7300	5.5500, 7.5500, 8.9100	0.6400, 2.2700, 4.2700	5.3600, 7.3600, 8.7300	4.0900, 6.0900, 7.7300	0.7300, 2.2700, 4.2700	0.6400, 2.2700, 4.2700	5.3600, 7.3600, 8.7300	2.0900, 3.9100, 5.8200
A33	6.2700, 8.2700, 9.4500	5.7300, 7.7300, 9.0000	5.3600, 7.3600, 8.7300	0.3600, 1.7300, 3.7300	6.2700, 8.2700, 9.4500	3.1800, 5.1800, 7.0000	1.6400, 3.3600, 5.3600	0.3600, 1.7300, 3.7300	6.2700, 8.2700, 9.4500	3.9100, 5.9100, 7.5500

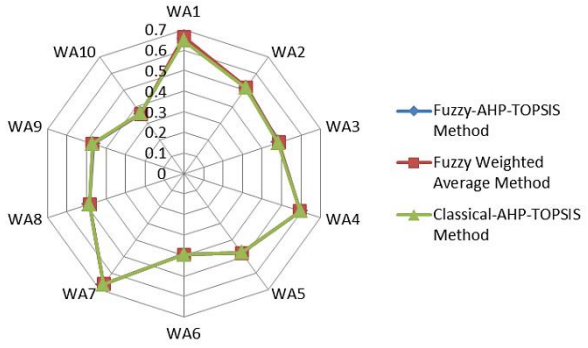

Figure 4. Radar chart representation of comparing the results from different methods

chart representation of the findings obtained from three different techniques. The findings presented in Fig. 4 indicate a strong correlation between the findings obtained from both techniques [15]. It clearly demonstrates that the Fuzzy-based approach offers better research findings over the classical approach.

7. Conclusion

Software security breaches and advanced malware exploits are now widespread, infiltrating every sector. Various methods for targeting pose serious challenges for IT security experts who are actively working on tactics to contain and reduce these infringements. In this kind of scenario, the different organizations need to work on conclusive strategy that effectively neutralizes the impact of the attacks. There are several malware analysis techniques available today which allow the researchers and security experts to realize the danger and purpose of a particular sample easily and in detail. This expertise enables the researcher to deal with future malware production patterns or to optimize established identification procedures to minimize the risk from such an application [50]. Our research paper explores the different malware analysis techniques available to understand a given sample's behavior. Furthermore, our study also highlights the perils of competing motivation among the malware writers to conceal the malicious nature of their

Table 13. The normalized fuzzy-decision matrix

	WA1	WA2	WA3	WA4	WA5	WA6	WA7	WA8	WA9	WA10
A11	0.5000, 0.7100, 0.8900	0.5200, 0.7400, 0.9400	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9200	0.2000, 0.4700, 0.7700	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9200	0.2000, 0.4700, 0.7700	0.2000, 0.5000, 0.8000	0.4700, 0.6800, 0.8800
A12	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9300	0.5900, 0.8000, 0.9700	0.6000, 0.8100, 1.0000	0.4200, 0.6900, 0.9900	0.5900, 0.8000, 0.9700	0.6000, 0.8100, 1.0000	0.4200, 0.6900, 0.9900	0.3800, 0.6600, 0.9600	0.6100, 0.8200, 0.9800
A13	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9200	0.5900, 0.8000, 0.9600	0.4600, 0.6800, 0.8800	0.2000, 0.4700, 0.7700	0.5900, 0.8000, 0.9600	0.4600, 0.6800, 0.8800	0.2000, 0.4700, 0.7700	0.2000, 0.5000, 0.8000	0.5500, 0.7600, 0.9300
A14	0.5900, 0.8000, 0.9700	0.6000, 0.8100, 1.0000	0.5400, 0.7500, 0.9300	0.4600, 0.6800, 0.8700	0.1800, 0.4500, 0.7400	0.5400, 0.7500, 0.9300	0.4600, 0.6800, 0.8700	0.1800, 0.4500, 0.7400	0.2000, 0.5000, 0.8000	0.4700, 0.6800, 0.8700
A21	0.4600, 0.6700, 0.8600	0.3800, 0.6000, 0.8000	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9300	0.4200, 0.6900, 0.9900	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9300	0.4200, 0.6900, 0.9900	0.3800, 0.6600, 0.9600	0.4700, 0.6800, 0.8700
A22	0.5000, 0.7100, 0.8900	0.5200, 0.7400, 0.9400	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9200	0.2000, 0.4700, 0.7700	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9200	0.2000, 0.4700, 0.7700	0.2000, 0.5000, 0.8000	0.4700, 0.6800, 0.8800
A23	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9300	0.5900, 0.8000, 0.9700	0.6000, 0.8100, 1.0000	0.4200, 0.6900, 0.9900	0.5900, 0.8000, 0.9700	0.6000, 0.8100, 1.0000	0.4200, 0.6900, 0.9900	0.3800, 0.6600, 0.9600	0.6100, 0.8200, 0.9800
A31	0.5400, 0.7500, 0.9200	0.5200, 0.7400, 0.9200	0.5900, 0.8000, 0.9600	0.4600, 0.6800, 0.8800	0.2000, 0.4700, 0.7700	0.5900, 0.8000, 0.9600	0.4600, 0.6800, 0.8800	0.2000, 0.4700, 0.7700	0.2000, 0.5000, 0.8000	0.5500, 0.7600, 0.9300
A32	0.5900, 0.8000, 0.9700	0.6000, 0.8100, 1.0000	0.5400, 0.7500, 0.9300	0.4600, 0.6800, 0.8700	0.1800, 0.4500, 0.7400	0.5400, 0.7500, 0.9300	0.4600, 0.6800, 0.8700	0.1800, 0.4500, 0.7400	0.2000, 0.5000, 0.8000	0.4700, 0.6800, 0.8700
A33	0.5900, 0.8000, 0.9600	0.4600, 0.6800, 0.8800	0.3000, 0.5700, 0.8300	0.2600, 0.5300, 0.8200	0.1600, 0.4200, 0.7200	0.3000, 0.5700, 0.8300	0.2600, 0.5300, 0.8200	0.1600, 0.4200, 0.7200	0.1200, 0.3900, 0.6900	0.4300, 0.6400, 0.8600

production for commercial benefits. The hierarchical representation of different malware analysis techniques is shown in Fig. 1 that tabulates the categories and the sub-categories. We have used the widely reliable and accepted integrated fuzzy AHP-TOPSIS method to provide a prioritized ranking outcome to the created hierarchy in Fig. 1, while defining all of these important findings. The findings obtained with the help of integrated fuzzy AHP-TOPSIS method which assist IT security experts to understand the process of evaluating the impact of malware analysis techniques. The main outcomes of the present research study can be defined as:

- Findings of the present study show that amongst the Static Analysis (A1), Dynamic Analysis (A2) and Reverse Engineering (A3), the most powerful malware attack analysis approach is the Reverse Engineering. Our analysis also proves that String Analysis (A33) is the best technique for the Reverse Engineering malware analysis approach.
- The most prioritized sub-factor in the analyzed outcomes is the String Analysis. Strings in software are values which are installed from the sample of malware when performed. To accomplish strong indication from the malware sample, the reverse engineering process has to be done for string analysis.
- Findings of this research study are conclusive and the outcomes prove that the proposed methodology, if enlisted by the IT security experts and researchers, would be highly effective for malware analysis.
- This study will assist the IT security experts in improving the existing malware analysis mechanism and market by presenting an empirically evaluated and validated malware analysis technique. Researchers and academicians from the software security field may use the findings and improve the mechanism of malware analysis techniques.
- The study has discovered the most appropriate malware analysis technique by executing it on different versions of university's web application on different levels. By following this measure,

Table 14. The weighted normalized fuzzy-decision matrix

	WA1	WA2	WA3	WA4	WA5	WA6	WA7	WA8	WA9	WA10
A11	0.0040, 0.0140, 0.0440	0.0030, 0.0120, 0.0410	0.0030, 0.0120, 0.0410	0.0050, 0.0160, 0.0480	0.0050, 0.0160, 0.0490	0.0030, 0.0130, 0.0450	0.0030, 0.0120, 0.0410	0.0050, 0.0160, 0.0480	0.0050, 0.0160, 0.0490	0.0030, 0.0130, 0.0450
A12	0.0040, 0.0140, 0.0440	0.0030, 0.0120, 0.0420	0.0030, 0.0120, 0.0420	0.0020, 0.0100, 0.0370	0.0020, 0.0090, 0.0380	0.0020, 0.0100, 0.0390	0.0030, 0.0120, 0.0420	0.0020, 0.0100, 0.0370	0.0020, 0.0090, 0.0380	0.0020, 0.0100, 0.0390
A13	0.0010, 0.0060, 0.0190	0.0020, 0.0060, 0.0200	0.0020, 0.0060, 0.0200	0.0010, 0.0050, 0.0190	0.0010, 0.0050, 0.0180	0.0010, 0.0040, 0.0170	0.0020, 0.0060, 0.0200	0.0010, 0.0050, 0.0190	0.0010, 0.0050, 0.0180	0.0010, 0.0040, 0.0170
A14	0.0020, 0.0080, 0.0270	0.0020, 0.0080, 0.0250	0.0020, 0.0080, 0.0250	0.0020, 0.0070, 0.0270	0.0020, 0.0070, 0.0250	0.0000, 0.0040, 0.0170	0.0020, 0.0080, 0.0250	0.0020, 0.0070, 0.0270	0.0020, 0.0070, 0.0250	0.0000, 0.0040, 0.0170
A21	0.0010, 0.0050, 0.0180	0.0020, 0.0070, 0.0220	0.0020, 0.0070, 0.0220	0.0020, 0.0070, 0.0240	0.0010, 0.0050, 0.0180	0.0000, 0.0020, 0.0090	0.0020, 0.0070, 0.0220	0.0020, 0.0070, 0.0240	0.0010, 0.0050, 0.0180	0.0000, 0.0020, 0.0090
A22	0.0040, 0.0140, 0.0440	0.0030, 0.0120, 0.0420	0.0030, 0.0120, 0.0420	0.0020, 0.0100, 0.0370	0.0020, 0.0090, 0.0380	0.0020, 0.0100, 0.0390	0.0030, 0.0120, 0.0420	0.0020, 0.0100, 0.0370	0.0020, 0.0090, 0.0380	0.0020, 0.0100, 0.0390
A23	0.0010, 0.0060, 0.0190	0.0020, 0.0060, 0.0200	0.0020, 0.0060, 0.0200	0.0010, 0.0050, 0.0190	0.0010, 0.0050, 0.0180	0.0010, 0.0040, 0.0170	0.0020, 0.0060, 0.0200	0.0010, 0.0050, 0.0190	0.0010, 0.0050, 0.0180	0.0010, 0.0040, 0.0170
A31	0.0020, 0.0080, 0.0270	0.0020, 0.0080, 0.0250	0.0020, 0.0080, 0.0250	0.0020, 0.0070, 0.0270	0.0020, 0.0070, 0.0250	0.0000, 0.0040, 0.0170	0.0020, 0.0080, 0.0250	0.0020, 0.0070, 0.0270	0.0020, 0.0070, 0.0250	0.0000, 0.0040, 0.0170
A32	0.0010, 0.0050, 0.0180	0.0020, 0.0070, 0.0220	0.0020, 0.0070, 0.0220	0.0020, 0.0070, 0.0240	0.0010, 0.0050, 0.0180	0.0000, 0.0020, 0.0090	0.0020, 0.0070, 0.0220	0.0020, 0.0070, 0.0240	0.0010, 0.0050, 0.0180	0.0000, 0.0020, 0.0090
A33	0.0030, 0.0110, 0.0360	0.0020, 0.0090, 0.0300	0.0020, 0.0090, 0.0300	0.0020, 0.0100, 0.0350	0.0030, 0.0110, 0.0360	0.0020, 0.0090, 0.0340	0.0020, 0.0090, 0.0300	0.0020, 0.0100, 0.0350	0.0030, 0.0110, 0.0360	0.0020, 0.0090, 0.0340

Table 15. Closeness coefficients to the aspired level among the different alternatives

Alternatives		d+i	d-i	Gap Degree of CC+i	Satisfaction Degree of CC-i
Alternative 1	WA1	0.0450	0.0270	0.3780	0.6548
Alternative 2	WA2	0.0390	0.0360	0.4970	0.5165
Alternative 3	WA3	0.0380	0.0410	0.5390	0.4854
Alternative 4	WA4	0.0370	0.0270	0.4340	0.5965
Alternative 5	WA5	0.0390	0.0460	0.5450	0.4785
Alternative 6	WA6	0.0350	0.0480	0.6280	0.3965
Alternative 7	WA7	0.0480	0.0260	0.3670	0.6685
Alternative 8	WA8	0.0330	0.0420	0.5660	0.4854
Alternative 9	WA9	0.0370	0.0480	0.5490	0.4658
Alternative 10	WA10	0.0300	0.0470	0.6060	0.3582

prospective research scientists will be able to determine the factors influencing malware attack analysis along with their corresponding weights.

- Malware analysis market and opportunities that this domain affords are a plenty. There are numerous aspects to study in this filed and each aspect is imbued with its set of complexities. Hence our research study also has limitations, especially when compared with the other sectors. Although extensive research must dwell upon all the technological, ethical and organizational consequences in a single manuscript, the emphasis of this research is only on the hypothetical situation and its consequences for cyber-security.

Conflicts of Interest

The authors declare no conflict of interest.

Table 16. Sensitivity analysis

Scenario	Weights/ Alternatives		WA1	WA2	WA3	WA4	WA5	WA6	WA7	WA8	WA9	WA10
Exp-0	Original Weights	Satisfaction Degree (CC-i)	0.6548	0.5165	0.4854	0.5965	0.4785	0.3965	0.6685	0.4854	0.4658	0.3582
Exp-1	A11		0.6698	0.5298	0.4989	0.6066	0.491	0.394	0.673	0.5053	0.4941	0.3489
Exp-2	A12		0.6398	0.5032	0.4719	0.5864	0.466	0.399	0.664	0.4655	0.4375	0.3675
Exp-3	A13		0.6681	0.4357	0.4096	0.6305	0.5003	0.3529	0.6096	0.4909	0.4544	0.3765
Exp-4	A14		0.6415	0.5973	0.5612	0.5625	0.4567	0.4401	0.7274	0.4799	0.4772	0.3399
Exp-5	A21		0.5633	0.5177	0.4782	0.5819	0.4737	0.35	0.6423	0.4015	0.4045	0.3473
Exp-6	A22		0.7463	0.5153	0.4926	0.6111	0.4833	0.443	0.6947	0.5693	0.5271	0.3691
Exp-7	A23		0.7431	0.6106	0.5862	0.599	0.4789	0.5297	0.7842	0.5685	0.5295	0.322
Exp-8	A31		0.5665	0.4224	0.3846	0.594	0.4781	0.2633	0.5528	0.4023	0.4021	0.3944
Exp-9	A32		0.5917	0.4627	0.4463	0.5632	0.5166	0.3289	0.5969	0.4498	0.4413	0.3882
Exp-10	A33		0.5286	0.4089	0.4072	0.5299	0.5547	0.2613	0.5253	0.4142	0.4168	0.4134

Author Contributions

Conceptualization, Mamdouh Alenezi, Bineet Kumar Gupta, and Md Tarique Jamal Ansari; Methodology, Md Tarique Jamal Ansari; Software, Md Tarique Jamal Ansari; Validation, Md Tarique Jamal Ansari, Bineet Kumar Gupta, Alka Agrawal, Rajeev Kumar, and Raees Ahmad Khan; Formal Analysis, Alka Agrawal, MdTarique Jamal Ansari, Bineet Kumar Gupta, and Rajeev Kumar; Investigation, MamdouhAlenezi; Resources, MdTarique Jamal Ansari; Data Curation, Alka Agrawal; Writing—Original Draft preparation, MdTarique Jamal Ansari, and Rajeev Kumar; writing—Review and Editing, MdTarique Jamal Ansari, Rajeev Kumar, Bineet Kumar Gupta, and Alka Agrawal; Visualization, MdTarique Jamal Ansari; Supervision, Raees Ahmad Khan; Project Administration, Alka Agrawal, and Raees AhmadKhan; Funding Acquisition, Mamdouh Alenezi.

Acknowledgments

Authors are grateful to the Prince Sultan University, Saudi Arabia, for sponsoring this research quest.

References

[1] M. T. J. Ansari, D. Pandey, and M. Alenezi, "STORE: Security threat oriented requirements engineering methodology", Journal of King Saud University-Computer and Information Sciences, 2018. (In press)

[2] Malware Analysis Market. (2019, November). Retrieved April 2020, from https://www.marketsandmarkets.com/Market-Reports/malware-analysis-market-108766513.html

[3] A. Makandar and A. Patrot, "Malware class recognition using image processing techniques", In: *Proc. of 2017 International Conf. on Data Management, Analytics and Innovation (ICDMAI)* pp. 76-80, 2017.

[4] M. T. J. Ansari and D. Pandey, "Risks, security, and privacy for HIV/AIDS data: Big Data perspective", In *Big Data Analytics in HIV/AIDS Research* pp. 117-139, 2018.

[5] Ş. Bahtiyar, M. B. Yaman, and C. Y. Altıniğne, "A multi-dimensional machine learning approach to predict advanced malware", *Computer Networks*, Vol. 160, pp. 118-129, 2019.

[6] W. Yan, "CAS: A framework of online detecting advance malware families for cloud-based security", In: *Proc. of 2012 1st IEEE International Conf. on Communications in China (ICCC)* pp. 220-225, 2012.

[7] Rosenberg, G. Sicard, and E. O. David, "End-to-end deep neural networks and transfer learning for automatic analysis of nation-state malware", *Entropy*, Vol. 20, No. 5, p. 390, 2018.

[8] E. M. Rudd, A. Rozsa, M. Günther, and T. E. Boult, "A survey of stealth malware attacks, mitigation measures, and steps toward autonomous open world solutions", *IEEE Communications Surveys & Tutorials*, Vol. 19, No. 2, pp. 1145-1172, 2016.

[9] H. Xie, X. Wang, and J. Liu, *U.S. Patent No. 9,047,441*. Washington, DC: U.S. Patent and Trademark Office, 2015

[10] P. M. Comparetti, G. Salvaneschi, E. Kirda, C. Kolbitsch, C. Kruegel, and S. Zanero, "Identifying dormant functionality in malware programs", In *2010 IEEE Symposium on Security and Privacy*, pp. 61-76, 2010.

[11] N. M. Johnson, J. Caballero, K. Z. Chen, S. McCamant, P. Poosankam, D. Reynaud, and D. Song, "Differential slicing: Identifying causal execution differences for security applications". In *2011 IEEE Symposium on Security and Privacy*, pp. 347-362, 2011.

[12] D. Kirat and G. Vigna, "Malgene: Automatic extraction of malware analysis evasion signature", In: *Proc. of* the *22nd ACM SIGSAC Conf. on Computer and Communications Security*, pp. 769-780, 2015.

[13] Ö. Uygun and A. Dede, "Performance evaluation of green supply chain management using integrated fuzzy multi-criteria decision making techniques", *Computers & Industrial Engineering*, Vol. 102, pp. 502-511, 2016.

[14] S. Önüt, S. S. Kara, and E. Işik, "Long term supplier selection using a combined fuzzy MCDM approach: A case study for a telecommunication company". *Expert systems with applications*, Vol. 36, No. 2, pp. 3887-3895, 2009.

[15] R. Kumar, M. Zarour, M. Alenezi, A. Agrawal, and R. A. Khan, "Measuring security durability of software through fuzzy-based decision-making process", *International Journal of Computational Intelligence Systems*, Vol. 12 No. 2, pp. 627-642, 2019.

[16] Ryoo, B. Malone, P. A. Laplante, and P. Anand, "The use of security tactics in open source software projects" *IEEE Transactions on Reliability*, Vol. 65, No. 3, pp. 1195-1204, 2015.

[17] R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.

[18] Z. Salehi, A. Sami, and M. Ghiasi, "Using feature generation from API calls for malware detection", *Computer Fraud & Security*, Vol. 9, pp. 9-18, 2014.

[19] B. Gu, Y. Fang, P. Jia, L. Liu, L. Zhang, and M. Wang, "A new static detection method of malicious document based on wavelet package analysis", In: *Proc. of 2015 International Conf. on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pp. 333-336, 2015.

[20] S. Das, Y. Liu, W. Zhang, and M. Chandramohan, "Semantics-based online malware detection: Towards efficient real-time protection against malware", *IEEE Transactions on Information Forensics and Security*, Vol. 11, No. 2, pp. 289-302, 2015.

[21] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, and F. Roli, "Adversarial malware binaries: Evading deep learning for malware detection in executables", In: *Proc. of 2018 26th European Signal Processing Conf. (EUSIPCO)*, pp. 533-537, 2018.

[22] H. Hashemi and A. Hamzeh, "Visual malware detection using local malicious pattern", *Journal of Computer Virology and Hacking Techniques*, Vol. 15, No. 1, pp. 1-14, 2019.

[23] A. Mohaisen, O. Alrawi, and M. Mohaisen, "Amal: High-fidelity, behavior-based automated malware analysis and classification" C*omputers & security*, Vol. 52, pp. 251-266, 2015.

[24] C. I. Fan, H. W. Hsiao, C. H. Chou, and Y. F. Tseng, "Malware detection systems based on API log data mining", In: *Proc. of 2015 IEEE 39th Annual Computer Software and Applications Conf.* Vol. 3, pp. 255-260, 2015.

[25] H. S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection", *Journal of Computer Virology and Hacking Techniques*, Vol. 12, No. 2, pp. 59-67, 2016.

[26] R. Sihwail, K. Omar, K. A. Zainol Ariffin, and S. Al Afghani, "Malware detection approach based on artifacts in memory image and dynamic analysis", *Applied Sciences*, Vol. 9, No. 18, p. 3680, 2019.

[27] D. A. Quist and L. M. Liebrock, "Visualizing compiled executables for malware analysis", *In 2009 6th International Workshop on Visualization for Cyber Security*, pp. 27-32, 2009.

[28] S. Naveen and T. G. Kumar, "Ransomware Analysis Using Reverse Engineering", In: *Proc. of International Conf. on Advances in Computing and Data Sciences*, pp. 185-194, 2019.

[29] A. Rahimian, R. Ziarati, S. Preda, and M. Debbabi, "On the reverse engineering of the citadel botnet", In *International Symposium on Foundations and Practice of Security*, pp. 408-425, Cham, 2013.

[30] Zimba, L. Simukonda, and M. Chishimba, "Demystifying ransomware attacks: reverse engineering and dynamic malware analysis of wannacry for network and information security", *ZAMBIA INFORMATION COMMUNICATION TECHNOLOGY (ICT) JOURNAL*, Vol. 1, No. 1, pp. 35-40, 2017.

[31] E. Triantaphyllou, B. Shu, S. N. Sanchez, and T. Ray, "Multi-criteria decision making: an operations research approach", *Encyclopedia of Electrical and Electronics Engineering*, Vol. 15, pp. 175-186, 1998.

[32] S. Ballı and S. Korukoğlu, "Operating system selection using fuzzy AHP and TOPSIS methods", *Mathematical and Computational Applications*, Vol. 14, No. 2, pp. 119-130, 2009.

[33] N. Demirtaş, Ö. N. Alp, U. R. Tuzkaya, and H. Baraçli, "Fuzzy AHP–TOPSIS two stages methodology for ERP software selection: an application in passenger transport sector", In: *Proc. of 15th International Research/Expert Conf. Trends in the Development of Machinery and Associated Technology*, 2011. (In press)

[34] A. A. Zaidan, B. B. Zaidan, A. Al-Haiqi, M. L. M. Kiah, M. Hussain, and M. Abdulnabi, "Evaluation and selection of open-source EMR software packages based on integrated AHP and TOPSIS", *Journal of Biomedical Informatics*, Vol. 53, pp. 390-404, 2015.

[35] M. Alenezi, A. Agrawal, R. Kumar, and R. A. Khan, "Evaluating performance of web application security through a fuzzy based hybrid multi-criteria decision-making approach: Design tactics perspective", *IEEE Access*, Vol. 8, pp. 25543-25556, 2020.

[36] R. Kumar, A. I. Khan, Y. B. Abushark, M. M. Alam, A. Agrawal, and R. A. Khan, "An Integrated Approach of Fuzzy Logic, AHP and TOPSIS for Estimating Usable-Security of Web Applications", *IEEE Access*, Vol. 8, pp. 50944-50957, 2020.

[37] H. J. Schellnhuber and V. Wenzel (Eds.) "*Earth System Analysis: Integrating Science for Sustainability*", *Springer Science & Business Media*, 2012.

[38] A. Ishizaka and P. Nemery, "Multi-criteria decision analysis: methods and software", 2013.

[39] C. W. Chang, C. R. Wu, and H. L. Lin, "Integrating fuzzy theory and hierarchy concepts to evaluate software quality", *Software Quality Journal*, Vol. 16, No. 2, pp. 263-276, 2008.

[40] R. Paradis and B. Tran, "Balancing Security/Safety and Sustainability Objectives", *National Institute of Building Sciences*, 2010.

[41] T. L. Saaty, "How to make a decision: the analytic hierarchy process", *European journal of operational research*, Vol. 48, No. 1, pp. 9-26. 1990.

[42] A. Dawood, K. Y. Sharif, A. A. Zaidan, A. A. A. Ghani, H. B. Zulzalil, and B. B. Zaidan, "Mapping and Analysis of Open Source Software (OSS) usability for sustainable OSS product", *IEEE Access*, Vol. 7, pp. 65913-65933, 2019.

[43] A. Zadeh, "Fuzzy sets", *Information and control*, Vol. 8, No. 3, pp. 338-353, 1965.

[44] J. F. Chen, H. N. Hsieh, and Q. H. Do, "Evaluating teaching performance based on fuzzy AHP and comprehensive evaluation approach", *Applied Soft Computing*, Vol. 28, pp. 100-108, 2015.

[45] T. Frick, "Designing for sustainability: a guide to building greener digital products and services", 2016. (*In press*)

[46] Y. C. Chou, H. Y. Yen, V. T. Dang, and C. C. Sun, "Assessing the human resource in science and technology for Asian countries: Application of fuzzy AHP and fuzzy TOPSIS", *Symmetry*, Vol. 11, No. 2, pp. 251, 2019.

[47] W. B. Gray and R. J. Shadbegian, "The environmental performance of polluting plants: A spatial analysis", *Journal of Regional Science*, Vol. 47, No. 1, pp. 63-84, 2007.

[48] Y. C. Chou, H. Y. Yen, V. T. Dang, and C. C. Sun, "Assessing the human resource in science and technology for Asian countries: Application of fuzzy AHP and fuzzy TOPSIS", *Symmetry*, Vol. 11, No. 2, pp. 251, 2019.

[49] M. Wagner, A. Rind, N. Thür, and W. Aigner, "A knowledge-assisted visual malware analysis system: Design, validation, and reflection of KAMAS", *Computers & security*, Vol. 67, pp. 1-15, 2017.

[50] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools", *ACM Computing Surveys (CSUR)*, Vol. 44, No. 2, pp. 1-42, 2008.