Towards A Reuse Strategic Decision Pattern Framework – from theories to practices

Victor Chang[1, 2], Mohamed Abdel-Basset[3], Muthu Ramachandran[4]

[1]International Business School Suzhou, Xi'an Jiaotong-Liverpool University, China.

[2] Research Institute of Big Data Analytics, Xi'an Jiaotong-Liverpool University, China.

Email: ic.victor.chang@gmail.com

[3]Department of Operations Research, Faculty of Computers and Informatics, Zagazig University, Sharqiyah, Egypt.

E-mail: analyst_mohamed@yahoo.com; E-mail: analyst_mohamed@zu.edu.eg

[4] Leeds Beckett University, Leeds, UK

Email: M.Ramanchandran@leedsbeckett.ac.uk

Abstract

This paper demonstrates our proposed Reuse Strategic Decision Pattern Framework (RSDPF) based on blending ANP and TOPSIS techniques, enabled by the OSM model with data analytics. The motivation, related work, theory, the use and deployment, and the service deployment of the framework have been discussed in details. In this paper, RSDPF framework is demonstrated by the data analysis and interpretations based on a financial service firm. The OSM model allows three step of processed to be performed in one go to perform statistical tests, identify linear relations, check consistency on dataset and calculate OLS regression. The aim is to identify the actual, expected and risk rates of profitability. Code and services can be reused to compute for analysis. Service integration of the RSDPF framework has been demonstrated. Results confirm that there is a high extent of reliability. In this paper, we have demonstrated the reuse and integration of the framework supported by the case study of the financial service firm with its data analysis and service to justify our research contributions – reuse and integration in statistical data mining, knowledge and heuristic discovery and finally domain transference.

Keywords: Reuse and integration; RSDPF framework; predictive analytics pattern; ANP and TOPSIS techniques; OSM case study; service integration for data science

1. Introduction

Reuse and integration are commonly used in software engineering and service computing. In software engineering, reuse of codes, processes, software development and best practice have been adopted to streamline the processes (Boehm, 2006; Cockburn, 2006; Ko et al., 2011). Explanations can be as follows. First, with regard to the reuse of codes, it can avoid developers rewriting the same or similar syntax all the times. It also allows developers to streamline the processes while working with others and to maintain a good version control. All changes can be updated and checked live. Second, with regard to software development, if similar problems or cases have been encountered, codes can be reused to improve efficiency without the need to write codes from the very beginning. Codes can be reused to make functions better and more coherent with other parts of the software or applications. When good practices are established, it can streamline the process for developers, testers and system architects together, so that each can play his/her role better. Last, all these good practices can be developed into the best practice approach.

The best practice approach may not always be replicated in all possible situations. A set of codes, such as written for specific functions or particular needs, can be developed into a software framework. It allows the best practices to be replicated, adaptable and customizable according to different needs (Cockburn, 2006). A software framework can be developed in a more structured way, so that guidelines of developed can be updated. While maintaining the guidelines of software development and getting more support from software communities, a software framework can be further developed and contributed to the establishment of standard bodies (Pressman, 2005; Bonaccorsi and Rossi, 2003; Bellifemine et al., 2008). For example, Oasis is a security standard body focused on web and system development. Their initial focus was based on XACML, an XML-based security schema for security. Another example is W3C launched by Prof Tim Berners-Lee to set up standards for the web, with their initial focus on HTML development.

However, not all software frameworks can meet demands from the users and markets. This is particularly true for smart phone applications since users are more likely to access as if like mobile personal computers. Smart phones are vulnerable to security loopholes, breaches and attacks, and thus system and software updates are more common. This can make development of software framework more challenging – either more security updates should be applied, or a better alternative is provided, such as making the software more resilient to attack (Mather et al., 2013). To make this software more resilient to attacks, a software framework can be useful to provide more checks, more functions and more robust security features.

Service Computing is another area that a software framework can be useful. Originated from Service Oriented Architecture, when all these services are available on Internet and Cloud Computing-based services, Service Computing can be very effective to deliver services online, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). When all different services are connected to each other with more users, more data can be generated. Data can be in different forms and in different types of services. Some data will be in the same or similar form regardless of the services or sectors (Russom, 2011; Witten et al., 2016). For example, people information such as users, clients and patients, are available on databases, such as SQL and SQL queries, which can be joined, moved and queried from different sources of services. Another example, images and videos with records of interviews, can be on the same file format regardless of different sectors. In the third example, text format can be in CSV file, regardless they contain user data, or weather data, or financial data, or map data. However, some data can be specific to particular disciplines with different ways to process and analyze. In this paper, we only focus on services that can process and analyze generic forms of data described earlier in this paragraph.

While there are increasing demands with 1) software engineering to get more efficient codes and practices; 2) smart phones to get more resilient frameworks; 3) service computing to process, analyze and interpret data, integration of services is an important aspect to allow developers and managers to perform and check all these three functions in one go (Pressman, 2005; Khan et al., 2013; Chen et al., 2014). This can make the software framework more robust, adaptable to changes and more easily joined to other functions (Cockburn, 2006). In order to make these three possibly work together under the same platform and circumstance, we propose a Reuse Strategic Decision Pattern Framework (RSDPF), a framework that allows good practices in software engineering. RSDPF can be supported by theoretical development, be highly adaptable, be capable of collecting, analyzing and interpreting data and be usable by the latest technologies such as smart phones. RSDPF is a framework to be validated through actual software engineering practices combining the state-of-the-art, allowing easy and complex business modeling to be checked, executed and reviewed.

The structure of this paper is as follows. Section 2 presents the literature, related work and theoretical development of the RSDPF framework. Section 3 describes the use and deployment of the RSDPF framework, including the model, features and tests to validate. Section 4 illustrates the experiments with results, analysis and discussions. Section 5 concludes with this paper with the summary of research contributions and future work.

2. Related Work

2.1 Literature Review

Reuse is defined as the ability to replicate the previous work and customize to different circumstances, cases and projects, depending on different requirements and organizational needs. Reuse is often developed in software engineering, whereby a large number of projects cannot always write code from scratch (Jennings, 2001; Engwall, 2003; Boehm, 2006). Previous successful deployment can compile them into summary of the best practices, so that similar codes can be used to some extents, so that this can reduce the completion time for project delivery (Humble, and Farley, 2010; Abrahamsson et al., 2017). Reuse can be applied to knowledge pattern and analytics pattern. In knowledge pattern, similar behaviors and similar pattern studies can be extracted, analyzed and presented. Results can be illustrated by analytics pattern, since all the trends and summary of research outputs can be presented in a way that can be easily understood (Shawe-Taylor and Cristianini, 2004; Han et al., 2011). For example, if a large and complex business systems that require problems to be identified and issues to be resolved as soon as possible, it will require the analytics patterns to show areas or units that have the most urgent or most important problems. By resolving problems or reducing its impacts to the minimum, it can bring businesses moving up again to gain more such as profitability, reputation and client satisfaction.

Framework is a useful approach to demonstrate both reuse and integration. Brunch et al. (2010) focus on high-quality documents. They use Eclipse JFace to develop and reuse their code and improve the state of an API document. They demonstrate different examples to validate their framework. Cordell et al (2011) demonstrate the concept and philosophy of a recovery and reuse framework. Steps, processes and structures are well-developed. This is similar to the "best practice" approach, since the collective wisdom from the past and the present can minimize errors and maximize outputs. Kirk et al (2007) explain how to identify and address problems in object-oriented framework reuse. They have very specific examples and guidelines in the framework. Their approach is focused on architecture, data and pattern. In each architecture, there are different kinds of patterns, in which there are data to related and connected to. Despite they present a few issues to resolve, the fundamental concept is to identify the relationship between architecture, pattern and data, so that a more appropriate resolution can be conducted. This approach is related to the preliminary form of knowledge pattern and analytics pattern. Alcalá-Fdezt et al (2011) demonstrate their KEEL data-mining software, which is a dataset repository and also an integrated framework of algorithms and experimental analysis. They show different types of code for development and explain which part of the code can be used. They perform experiments and use only mean, standard errors and p-values to validate. In other words, it is a framework they can store data, reuse code, perform experiments a validation by statistics. However, a more up-to-date approach can be developed to allow advanced statistical analysis to be performed.

2.1.1 Predictive Analytics Patterns

Existing studies has reported successful application of big data analytics patterns that can be reused while solving similar problems-solutions situations (Lee 2016 and Leung 2016). Our work on software reuse has contributed to the development of reuse of big data analytics in particular predictive analytics based on the previous solutions (Ramachandran 2008; Ramachandran and Jamnal 2014). The term data science was first coined by Peter Naur in the 60s with reference to data processing in computer science and followed by well-known statisticians such as C.F. Jeff Wu and William S. Cleveland, in the late 1990s (Barga, Fontama, and Tok, 2014). This paper defines data science as the part of computer science, which deals with making visual insights and recovering useful prediction from the large amount of existing data that is available in an organization.

However, our main aim is to evolve a framework that supports reuse of data analytics patterns and reuse them as a set of predictive analytics patterns. Predictive analytics problems such as propensity modeling, churn analysis, and product recommendation as shown in Fig. 1.

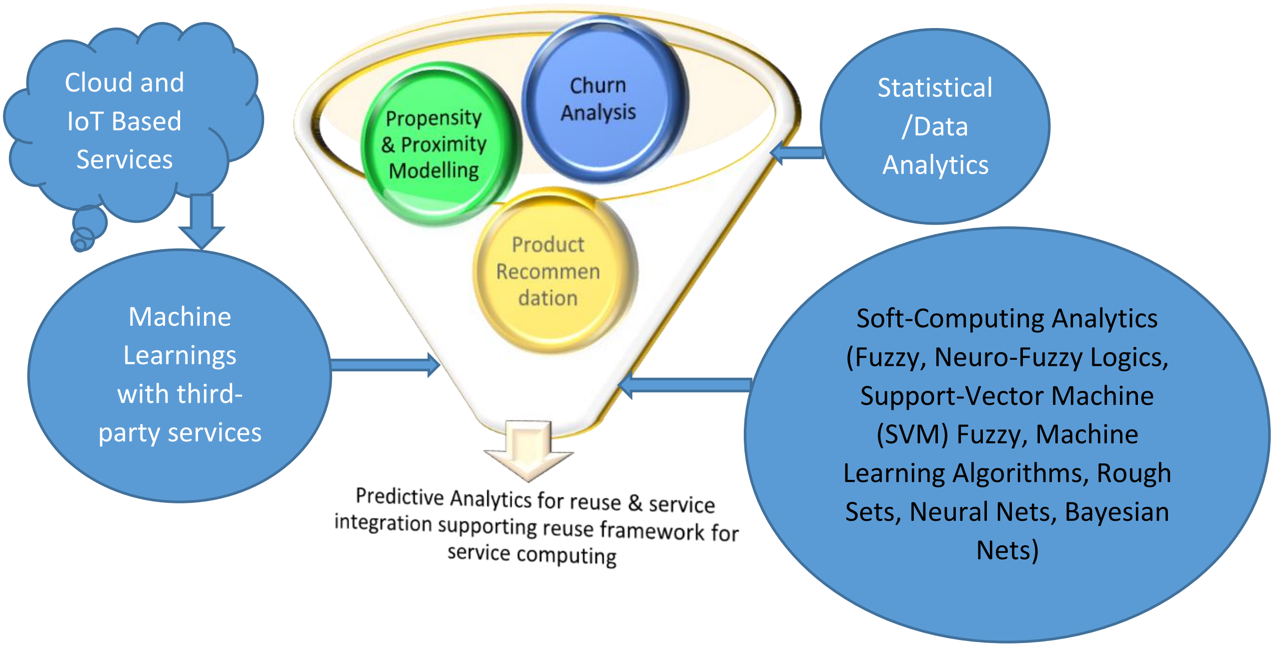


Fig. 1 Predictive Analytics Method for reuse and service computing

2.1.2 Integration

Integration is another aspect to combine different reused products and services. The aim is to streamline all processes or products involved in reuses and make them as a single solution, or a blended product or service. This can provide greater impacts to the development of software engineering, software as a service (SaaS) and applications oriented architectures. Integration can be in different forms, such as the workflow-based services (Papazoglou et al., 2007, 2008). The benefits can allow each services to be executed effectively, and the outputs of each service can become the inputs of the next services. Data can be part of the integration if the focus of research is to understand data, as well as its interpretations and lessons learned (Witten et al., 2016). Data can be directly input into the services, so all services, if presented by workflows, can be demonstrated and completed in one go. While an increasing number of services rely more on the data in order to understand the knowledge pattern and analytics pattern, this can help services to be completed faster. Status can be shown by analytics, so that monitoring of the progress and identification of problems can be easier.

Reuse and integration have been commonly developed and used in different disciplines. In order to investigate more in reuse and integration, new methods should be investigated, including the recommended practices, theories and recent development from other disciplines. For example, analytic network process and technique for order preference by similarity to ideal solution techniques to be presented in the following sections.

2.2 The Analytic Network Process (ANP) Technique

The analytic network process (ANP) is an expansion of analytic hierarchy process (AHP) based on a multi-criteria decision-making technique. Initiated by Saaty (1996), it was considered the dependency and feedback between elements of decision-making problem. The ANP is the improved version of AHP, since it models the decision-making problems as a network, but not as hierarchies. ANP is more flexible and more adaptable for software and service reuse than the original version of AHP as follows. First, the criteria do not depend on alternatives in AHP, since they do not depend on each other. Second, alternative options do not depend on each other. This provides a suitable pathway for Reuse Strategic Decision Pattern Framework (RSDPF), since decision for each minor stage do not necessarily influence each other. Each major development can be the decision of an independent process. Similar to the framework, decisions can be based on demands from specific needs or user requests. Referring to ANP in Fig. 2, each layer of components can be supportive to each other at any time. This is essential to RSDPF, allowing freedom and flexibility to add, modify and improve any code, functions, methods and services. The main steps of adopting ANP in RSDPF are as follows (Saaty et al., 2004):

1. The decision makers constructs the network of problem, which consist of, goal, criteria which can be disband to sub-criteria and finally the alternatives. Take into consideration the dependency and feedback between network elements.
2. Construct the comparisons matrices for calculating weights of criteria and alternatives by utilizing the 1-9 scale of Saaty for developers: 1 means that the two elements are with equal importance and 9 means absolutely significance of one element over another. All elements between 1 and 9 can be used. Then the consistency ratio of the comparison matrix should be checked and it must be ≤ 0.1 for each comparison matrix (Saaty, 1988). After that, calculate the eigenvector of comparison matrix by calculating the sum of column of comparison matrix and constructing a new matrix via dividing each value in column by the summation of that column. Then, take the average of new matrix rows. The comparison matrix that you may construct in ANP may be:
    - Comparisons of criteria according to goal.
    - Comparisons of sub-criteria according to criterion from the same cluster.
    - Comparisons of alternatives according to each criterion.
    - Comparisons of criteria, which belong to the same cluster according to each alternative.
3. Structure the super-matrix columns by using the eigenvectors, which can be calculated in the previous step. Then make a normalization of matrix to obtain weighted super-matrix. The weighted super-matrix then raised to a large power until the raw values will be equal to each column values of super-matrix. The result matrix named the limiting matrix.
4. Finally, select the best alternative according to weight values.

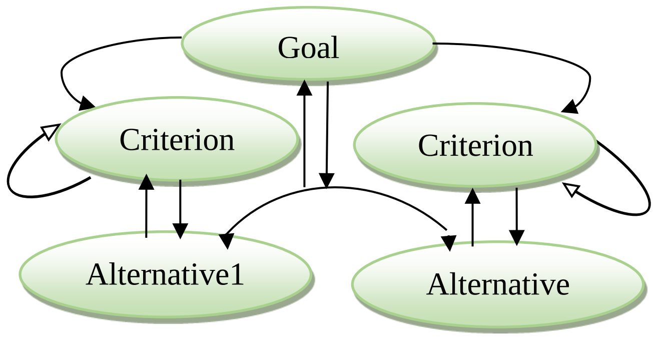


Fig.2 The ANP network approach in RSDPF

2.2.1 Preliminaries

The significant definitions of interval-valued neutrosophic sets and its operations, are presented in this section.

2.2.1.1 Interval-valued Neutrosophic Sets (INS)

The interval-valued neutrosophic set V in X is described by truth $T_V(x)$**,** indeterminacy $I_V(x)$and falsity $F_V(x)$membership degrees for each $x \in X$**.** Where $T_V(x)$= $[T_V^L(x), T_V^U(x) \subseteq [0,1]]$, $I_V(x)$= $[I_V^L(x), I_V^U(x) \subseteq [0,1]]$and $F_V(x)$=
$[F_V^L(x), F_V^U(x) \subseteq [0,1]]$. Then, we can write interval-valued neutrosophic set as
$V =< [T_V^L(x), T_V^U(x)], [I_V^L(x), I_V^U(x)], [F_V^L(x), F_V^U(x)] >$. Exactly the INS is a neutrosophic set.

2.2.1.2 Weighted Average for Interval-valued Neutrosophic Numbers (INN)

Let $y_j =< [T_j^L, T_j^U], [I_j^L, I_j^U], [F_j^L, F_j^U] >$ be a group of interval-valued neutrosophic numbers, $j = 1,2 \ldots, n$ is the number of decision makers. The weighted arithmetic average of interval-valued neutrosophic number INNWAA($y_1, y_2, \ldots, y_n$) =$\sum_{k=1}^{n} w_k y_j$ =

$$< [1 - \prod_{k=1}^{n}(1 - T_j^L)^{w_k}, 1 - \prod_{k=1}^{n}(1 - T_j^U)^{w_k}], [\prod_{k=1}^{n}(I_j^L)^{w_k}, \prod_{k=1}^{n}(I_j^U)^{w_k}], [\prod_{k=1}^{n}(F_j^L)^{w_k}, \prod_{k=1}^{n}(F_j^U)^{w_k}] > \quad (1)$$

,where w_k is the decision maker's weight vector.

2.2.1.3 INS Deneutrosophication Function

The deneutrosophication function converts each interval-valued neutrosophic number into crisp number. Let $A =< [T_A^L, T_A^U], [I_A^L, I_A^U], [F_A^L, F_A^U] >$ be interval-valued neutrosophic number , then the deneutrosophication function $D(A)$will defined by

$$D(A) = 10^{(\frac{2+(T_A^L+T_A^U)-2(I_A^L+I_A^U)-(F_A^L,F_A^U)}{4})} \quad (2)$$

2.2.1.4 Ranking Method for Interval-Valued Neutrosophic Numbers

Let A_1, A_2 are interval-valued neutrosophic numbers then,

- If $D(A_1)$ greater than $D(A_2)$, then $A_1 > A_2$
- If $D(A_1)$ less than $D(A_2)$, then $A_1 < A_2$
- If $D(A_1)$ equal $D(A_2)$, then $A_1 = A_2$.

2.3 The TOPSIS Technique

Tzeng and Hwang (2011) proposed the technique for order preference by similarity to ideal solution (TOPSIS) to aid decision makers in determining perfect positive (A^+) and negative (A^-) solution. The shortest distance from the positive ideal solution and the largest distance from the negative ideal solution is the selected alternative. The steps of TOPSIS presented as follows:

1. Construct the evaluation matrix, which consist of m alternatives and n criteria. The crossing of each alternative and criteria denoted as x_{ij}. Then, we have $(x_{ij})_{m*n}$ matrix.
2. Make a normalization process to obtain the normalized evaluation matrix using the following equation
$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}^2} \quad (3)$$
3. Multiply the weights of criteria $w = (w_1, w_2, \dots, w_n)$ by the normalized evaluation matrix to build the weighted matrix as follows:
$$v = \begin{pmatrix} v_{11} & \cdot\cdot & v_{1n} \\ \vdots & \vdots & \vdots \\ v_{m1} & \cdot\cdot & v_{nm} \end{pmatrix} = \begin{pmatrix} w_1 r_{11} & \cdot\cdot & w_n r_{1n} \\ \vdots & \vdots & \vdots \\ w_1 r_{m1} & \cdot\cdot & w_n r_{mn} \end{pmatrix} \quad (4)$$
4. Allocate the positive and negative ideal solution through the following:

$$A^+ = \{< \max(v_{ij} | i = 1,2,\dots,m) \, | j \in J^+ >, < \min(v_{ij} | i = 1,2,\dots,m) \, | j \in J^-\} \quad (5)$$

$$A^- = \{< \min(v_{ij} | i = 1,2,\dots,m) \, | j \in J^+ >, < \max(v_{ij} | i = 1,2,\dots,m) \, | j \in J^-\} \quad (6)$$

Where J^+ related to the criteria which have a positive influence and J^- related to the criteria which have a negative influence.

5. Measure the Euclidean distance among positive (d_i^+) and negative ideal solution (d_i^-) for all alternatives as follows:
$$d_i^+ = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^+)^2}\, , i = 1,2,\dots,m. \quad (7)$$
$$d_i^- = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^-)^2}\, , i = 1,2,\dots,m. \quad (8)$$
6. Calculate the closeness coefficient for the alternatives according to A^+ using the following equation
$$c_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad \text{for } i = 1,2,\dots,m. \quad (9)$$
7. Rank alternatives with respect to the largest value of c_i.

3. The Reuse Strategic Decision Pattern Framework (RSDPF)

This section describes the proposal of RSDPF based on the integration of ANP and TOPSIS techniques. Details are described in four phases. Each phase has its own steps as follows.

Phase 1: Breakdown the complex problem for understanding it better.

Step 1.1. Establish a panel of experts for sharing in decision making process. If we establish the panel with n member then, the panel=[e_1, e_2,..., e_n].

Step 1.2. Determine the criteria of the problem from the literature review and make a vote for the experts to confirm these criteria.

Step 1.3. Identify the alternatives of the problem.

Step 1.4. Construct the problem hierarchy.

We used ANP for making network model of the problem. A sample of ANP network presented in Fig.3.

Phase 2: The weights of problem elements must calculate through the following

Step 2.1. Structure the interval-valued comparison matrices relevant to each expert. Then, aggregate expert matrices which are on the same problem element using Equation (1).

Experts in this step compares criteria relevant to overall objective. Similarly sub-criteria relevant to criteria. Also the alternatives relevant to criteria also compared. The interdependencies between problem elements should also be compared pair-wisely. In traditional ANP a 9-point scale of Saaty (Adalı and Işık, 2017) was used to represent comparisons. In this paper, the interval-valued RSDPF numbers are used to clarify pair-wise comparisons. The interval-valued RSDPF scale for representing pair-wise comparisons are given in Table 1. The values in Table 1 returned to authors opinions for making comparison matrices. Since in case of comparing alternative 1 with alternative 2, and the first alternative was "Very strongly important" than second one. Then the truth degree is high and indeterminacy degree are very small, because the term" Very strongly important " means that, the decision makers are very confident of comparison result with a large percentage. So we represented this linguistic term using interval-neutrosophic number equals ([0.8,0.9],[0.0,0.1],[0.0,0.1]) as appears in Table 1. All other values in Table 1 were scaled with the same approach. If decision maker does not use any of these values: Evenly important, Low important, Basically important, Very strongly important and Absolutely important, then he/she can use any indeterminate values as illustrated in Table 1. For example, this interval-valued neutrosophic number ([0.3,0.4],[0.1,0.2],[0.6,0.7]) means that, decision maker's judgment is as follows: The truth degree about his/her judgment is between 30%-40%, his/her indeterminate degree about given judgment is between 10%-20% , and his/her falsity degree is between 60%-70%. So the increasing value of falsity degree in this interval-valued neutrosophic number and the minimum value of truth degree besides existing indeterminate information made us use this scale for representing very low important criteria.

Table 1. The interval-valued RSDPF scale for comparison matrix

Variables for code reuse	Interval-valued RSDPF numbers for relative importance <T,I,F>
Evenly important	([0.5,0.5],[0.5,0.5],[0.5,0.5])
Low important	([0.4,0.5],[0.1,0.2],[0.2,0.3])
Basically important	([0.6,0.7],[0.0,0.1],[0.0,0.1])
Very strongly important	([0.8,0.9],[0.0,0.1],[0.0,0.1])
Absolutely important	([1,1],[0.0,0.1],[0.0,0.0])
Intermediate values	([0.3,0.4],[0.1,0.2],[0.6,0.7]), ([0.6,0.7],[0.1,0.2],[0.0,0.1]), ([0.7,0.8],[0.0,0.1],[0.0,0.1]), ([0.9,1],[0.0,0.1],[0.0,0.1]).

Step 2.2. Apply the de-RSDPF process to transform the interval-valued RSDPF numbers back to crisp numbers by using Equation (2).

Step 2.3.Check the consistency of comparison matrices through using the super decision software.

Step 2.4. Calculate the eigenvector of matrices to determine weight, which will be used in constructing super-matrix.

Step 2.5. Construct the super-matrix of interdependencies.

Step 2.6. The weights of criteria are calculated by multiplying the local weight, which obtained from experts' comparison matrices of criteria relevant to goal, by the weight of interdependence matrix of criteria. Additionally, the sub-criteria global weights can be calculated by using the inner interdependent weights of the criteria and local weight of the sub-criteria. For each sub-criteria calculate the global weights via multiplying its local weight by the inner interdependent weight of the criterion to which it belongs.

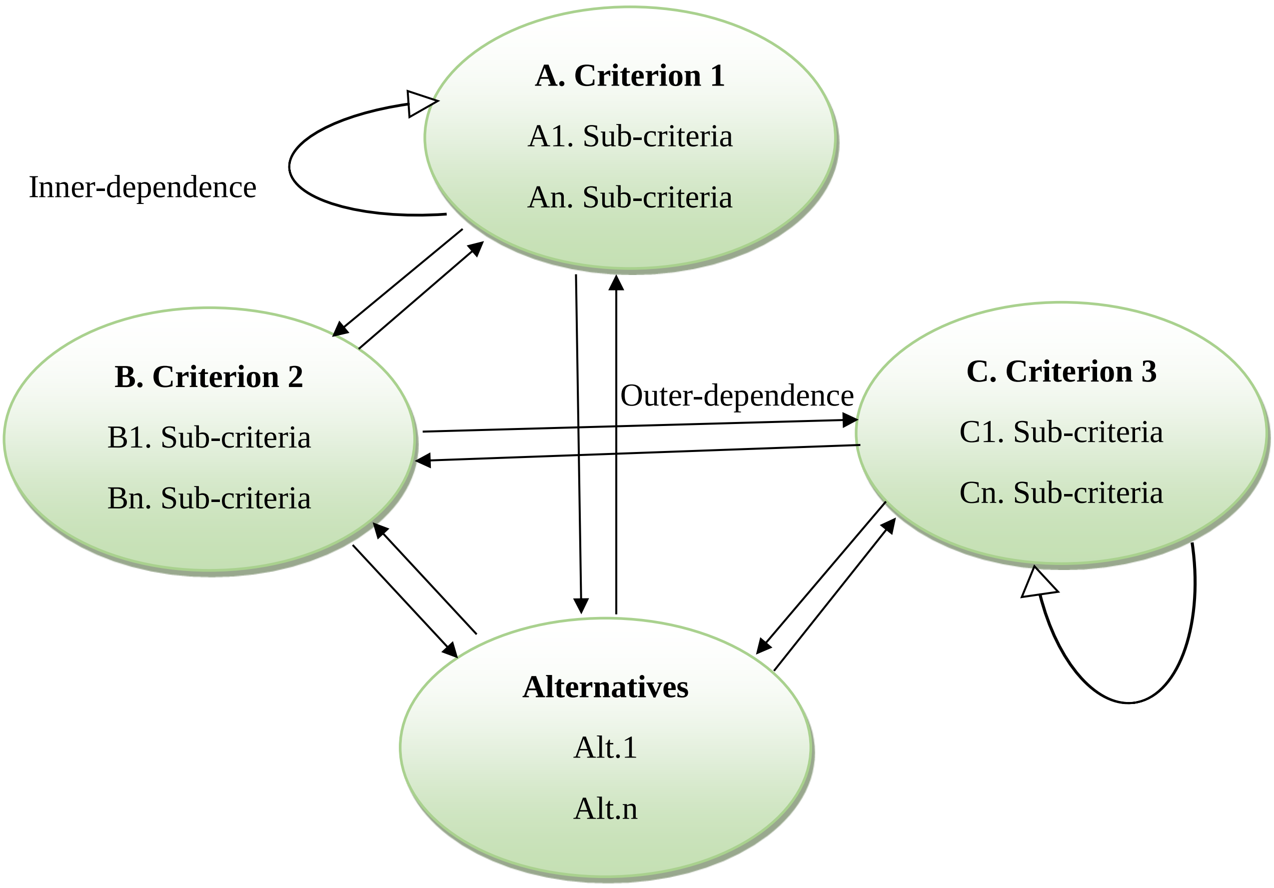


Fig.3 A sample of ANP model interdependencies.

Phase 3: Rank alternatives of problems.

Step 3.1. Construct the evaluation matrix, which consist of m alternatives and n criteria. Then make a normalization process to obtain the normalized evaluation matrix using Equation (3).

Step 3.2. Structure the weighted matrix through multiplying criteria's weights, which obtained from ANP by the normalized evaluation matrix as in Equation (4).

Step 3.4. Identify the positive and negative ideal solution using Equations (5), (6).

Step 3.5. Measure the Euclidean distance between positive (d_i^+) and negative ideal solution (d_i^-) using Equations (7), (8).

Step 3.6. Calculate the closeness coefficient and make the final ranking of alternatives.

Phase 4: Validate the model and make comparisons with other existing methods. The graphical illustration of the suggested framework presented in Fig.4.

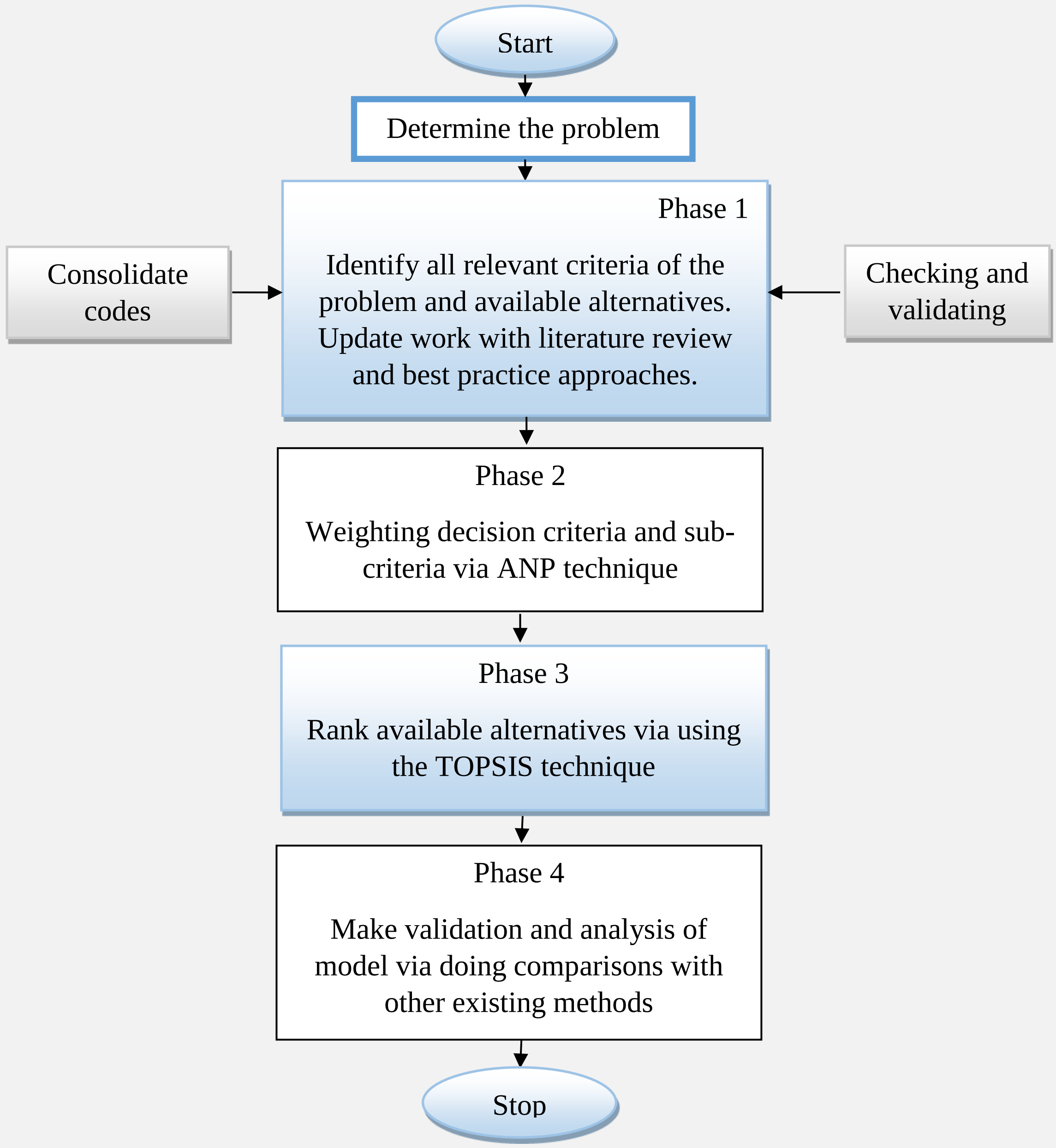


Fig.4 The framework proposed phases.

Section 4 will describe how to make those recommended steps into actionable work. A useful and dynamic framework allow multi-functions and multi-purpose characteristic (Xin and Yang, 2017). In other words, a framework is dynamic, flexible but structured and organized. To enable this, Organizational Sustainability Modeling (OSM) has been used. It is acting like an engine behind the integration of ANP and TOPSIS techniques. All complex data processing, analysis and interpretation can be presented in analytics, including outputs in graphics and visualization to be presented in Section 4. OSM is a pioneering model developed by Chang (2014), which can analyze return and risk of data analysis for financial services, retailed industry and research institutes in particular. Additionally, another important aspect is to make the theories of ANP and TOPSIS techniques into practices – calculating risk and return from the large amount of financial data provided in Section 4.

To move this forward, a challenge is to understand how a business perform and its detailed business performance analysis. In this paper, we can demonstrate how to analyze data and explain its interpretations for the business. Additionally, the concept of service integration will be illustrated in Section 5.

4. The use and deployment of RSDPF Framework

4.1 Organizational Sustainability Modeling (OSM) and the three step tests

This section describes the use and deployment of the RSDPF Framework, including the "outlook" of the framework, the model it deploys, the code it can be reused and results of analysis. OSM is used to make RSDPF dynamic, structured and visual. It can be adopted in other domains. For example, a paper investigating on improving manufacturing sector's business performance was lead and demonstrated by the effectiveness of the OSM model (Chang, 2017). In another example, OSM is used to evaluate performance between Cloud and non-Cloud services in healthcare industry (Chang and Wills, 2016). Resolving challenges and issues in different sectors are goals of developing OSM. In this paper, OSM is the model directly translated into code, user interface and results in analytics. Different parts of the code will be used to present how to replicate. Formula 10 shows the OSM model to calculate beta, the uncontrolled risk for the markets and projects.

$$\beta = \frac{\mathrm{e} - \mathrm{r}_c}{\mathrm{a} - \mathrm{r}_c} \qquad (10)$$

The variable 'a' is the actual rate of return and the variable 'e' is the expected rate of return. Return can be in three major categories: technical such as efficiency and productivity; financial such as profitability and people such as improvement in user satisfaction. The variable r_c is the risk-control rate, or the rate of manageable risk. There are two types of risk – one can be managed such as the management of personal time, completion task and progress; and the other one cannot be controlled, which include the change of weathers, the change of market trends and anything unpredictable. The essence of applying the RSDPF Framework is to keep track of actual rates of return and calculate the unpredictable and underlying risk. In order to achieve this, research work with a financial service can be used to support the validity of the RSDPF Framework.

As shown in Fig. 5, there are three major tests for this data analysis: (1) tests for Normality; (2) tests for linear relationship and 3) Ordinary Least Square (OLS), as part of OSM method.

The first normality test is to perform statistical tests to check if data is consistent with each other; any missing values and outliers; and results are close to each other.

The second test is to check whether all datapoints follow the linear relations and evidence of having the "best fit" for linear regressions. If so, the linear regression will be performed.

OLS is analyzed in the third test to see if the values of the regression are close to the ideal values and check consistency between the ideal situations (theory) and datasets and its analysis (practice). Some of the detailed steps were described in Chang et al. (2016) and Chang (2017). The emphasis of our approach is not to re-introduce the details of setting but how to use them all at once, similar to what Fig. 5 has demonstrated. Instead of going through each process manually, RSDPF framework allow the execution of these three steps all at once. Results for each test can be presented by report 1 to 3 respectively. Upon clicking report, the summary of all analysis can be presented, which will be shown in the latter part of this section. Upon completing all tests, "NOR" will present the final analysis and its output dataset.

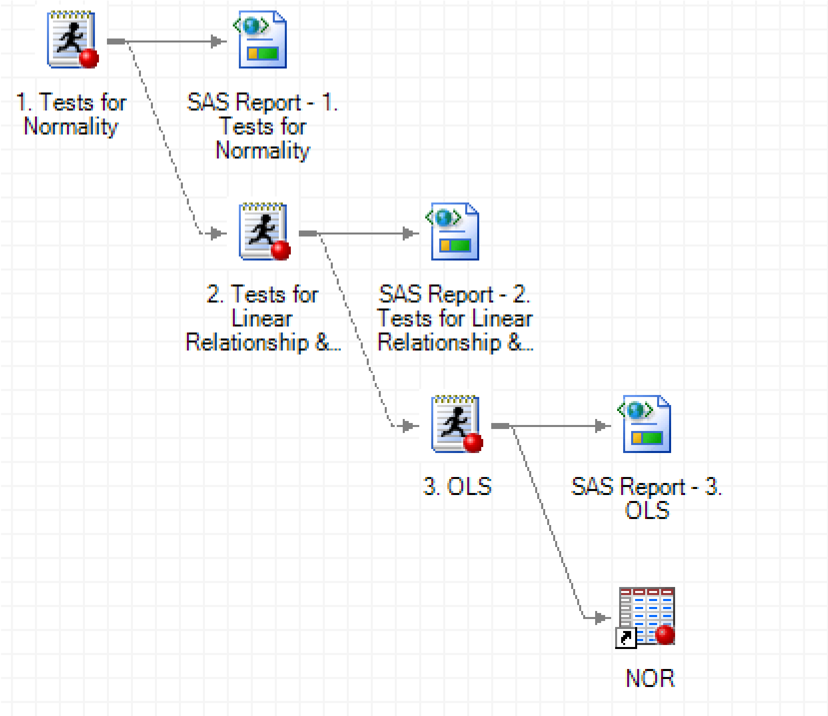


Fig 5. Performing three steps of tests for our RSDPF Framework

4.2 Code behind the scene

This section describes the code behind the scene between Table 2 and 5. Table 2 shows the Code reuse case 1 of our RSDPF Framework. It has the SQL procedure to extract the expected string. Then it can sort the data. Followed by transpose data to make the data into the desired positions like in Table 1, so that data-processing can take place.

Table 2: Code reuse case 1 of our RSDPF Framework

```
PROC SQL NOPRINT;     /* Modify the name to allow tabulation */
  UPDATE WORK.CTD_T
  SET _NAME_ = SUBSTR(_NAME_,1,FIND(_NAME_,'_',LENGTH(_NAME_)*-1)-1);
QUIT;

PROC SORT DATA=WORK.CTD_T OUT=WORK.CTD_S; /* Sort by Name to allow
tabulation */
      BY _NAME_;
RUN;
%_eg_conditional_dropds(WORK.CTD_T); /* Drops the temporary dataset */

PROC TRANSPOSE DATA=WORK.CTD_S /* Create a final table of variables and
measures */
OUT=WORK.CTD(RENAME=(COL1=Mean COL2=Median      COL3=Mode COL4=Std_Dev
COL5=Minimum COL6=Maximum COL7=Range COL8=Lower_Quartile
      COL9=Quartile_Range     COL10=Upper_Quartile COL11=Count));
      BY _NAME_;
RUN;
```

Table 3 shows code reuse case 2 of our RSDPF Framework, which is focused on how to make datapoints into report. All the statistical key values and starting vales are defined, recorded and presented in the software report.

Table 3: Code reuse case 2 of our RSDPF Framework

```
PROC REPORT DATA=WORK.CTD; /* Display a formatted report for export using
labels not variable names */
      COLUMNS _NAME_
                   ('Measures of Central Tendency' Mean Median Mode)
                   ('Measures of Dispersion' Minimum Maximum Range Std_Dev
Lower_Quartile Upper_Quartile Quartile_Range);
      DEFINE _NAME_ / DISPLAY 'Variable';
      DEFINE Mean / DISPLAY 'Mean' FORMAT=8.2;
      DEFINE Median / DISPLAY 'Median' FORMAT=8.2;
      DEFINE Mode / DISPLAY 'Mode' FORMAT=8.2;
      DEFINE Minimum / DISPLAY 'Minimum' FORMAT=8.2;
      DEFINE Maximum / DISPLAY 'Maximum' FORMAT=8.2;
      DEFINE Range / DISPLAY 'Range' FORMAT=8.2;
      DEFINE Std_Dev / DISPLAY 'Std Dev' FORMAT=8.2;
      DEFINE Lower_Quartile / DISPLAY 'Lower Quartile' FORMAT=8.2;
      DEFINE Quartile_Range / DISPLAY 'Quartile Range' FORMAT=8.2;
      DEFINE Upper_Quartile / DISPLAY 'Upper Quartile' FORMAT=8.2;
      TITLE2 "Measures of Central Tendency & Dispersion";
RUN;
```

Table 4 shows how to analyze data. In OSM, rc is the risk value, or the rate in which risk can be managed. The variable 'a' is indicated as the actual value, or actual rate of return. The variable 'e' is indicated as the expected value, or expected rate of return. It is a model to test the rate of return and risk. Transpose function is to make the final results in a list, so that it can read and further analyzed more easily.

Table 4: Code reuse case 3 of our RSDPF Framework

```
PROC MEANS DATA=WORK.RiskReturn /* Generate a table of measures */
           NMISS
           N NOPRINT; /* Hide the output, create a dataset for reuse */
      VAR rc a e;
      OUTPUT OUT=WORK.MO
           NMISS=
           N= /AUTONAME; /* Generate field names automatically to SAS
standards for these functions */
RUN;

PROC TRANSPOSE DATA=WORK.MO(DROP=_TYPE_ _FREQ_) OUT=WORK.MO_T; /* Pivot
the results to create a list */
RUN;
```

Table 5 shows the reuse code to calculate key outputs of our analysis. The aim is to compute all key outputs: mean square errors, Durban-Watson test, the regressed R-squared values and the total R squared values. The last two may not always be the same if there are multiple linear regression experienced in the datasets. Additionally, residual plots will indicate all the sums of uncertainties are within the acceptable range. This is important since excessive residuals can become "noises", which may interfere the quality of data analysis.

Table 5: Code reuse case 4 of our RSDPF Framework

```
PROC SQL NOPRINT;
      CREATE TABLE WORK.FS AS
      SELECT Model, Label2 AS Label LABEL='Test', cValue2 AS cValue,
nValue2 AS nValue LABEL='Normalised Value' FORMAT 8.2
      FROM WORK.FS_T
      WHERE Label2 IN ('MSE', 'Durbin-Watson', 'Regress R-Square',
'SSE','Total R-Square');

      INSERT INTO WORK.FS(Model, Label, cValue, nValue)
      SELECT Model, Label1, cValue1, nValue1
      FROM WORK.FS_T
      WHERE Label1 IN ('MSE', 'Durbin-Watson', 'Regress R-Square',
'SSE','Total R-Square');
QUIT;

PROC REPORT DATA=WORK.FS;
      COLUMNS ('Independence of Observations' Label nValue);
      DEFINE Label / DISPLAY ORDER ORDER=INTERNAL;
      DEFINE nValue / DISPLAY;
      TITLE "Independence of Observations";
RUN;
TITLE;

/* Constant Variance of Errors (Heteroscedasticity) */
ODS SELECT ResidualPlot;  /* Only output the residual scatterplot */
```

One major benefit of using the RSDPF framework is to allow the execution of these three tests altogether at once. There is an execution command to click. After clicking it, results can be computed within seconds.

4.3 Results and analysis of the three-test and financial analysis

This section describes the results and analysis of three-test and financial analysis of the case study. Fig. 6 shows the part 1 of report result. It tests the variables in OSM model, understand the pattern of each key variables and the range in different quartile of their datapoints. It checks any missing values for outliers. It is checking and analyzing all datapoints have consistency with each other. All the key outputs of variable ‘a’ are higher than ‘’e, which means the actual results or rates of actual return are higher than the expected return. The linear regression line of all datapoints can validate the followings: First, all datapoints follow linear regression. Second, risk control rate can be calculated based on the gradient of the linear regression. It indicates that risk has been managed in a good shape since the datapoints plot follows a simple linear regression.

Descriptive Statistics for Dataset 1

Measures of Central Tendency & Dispersion

	Measures of Central Tendency			Measures of Dispersion						
Variable	Mean	Median	Mode	Minimum	Maximum	Range	Std Dev	Lower Quartile	Upper Quartile	Quartile Range
a	5.71	5.71	5.63	5.49	5.94	0.45	0.13	5.60	5.80	0.21
e	3.85	3.85	3.86	3.63	4.06	0.43	0.12	3.75	3.95	0.20
rc	1.86	1.86	1.86	1.74	1.96	0.22	0.06	1.81	1.91	0.10

Page Break

Missing Values and Outliers

	Missing Values		Outlier Detection			
Variable	Count	Missing	Lower Quartile Threshold	Upper Quartile Threshold	Below Lower Quartile Threshold	Above Upper Quartile Threshold
a	40	0	5.28	6.11	0	0
e	40	0	3.46	4.24	0	0
rc	40	0	1.66	2.05	0	0

Page Break

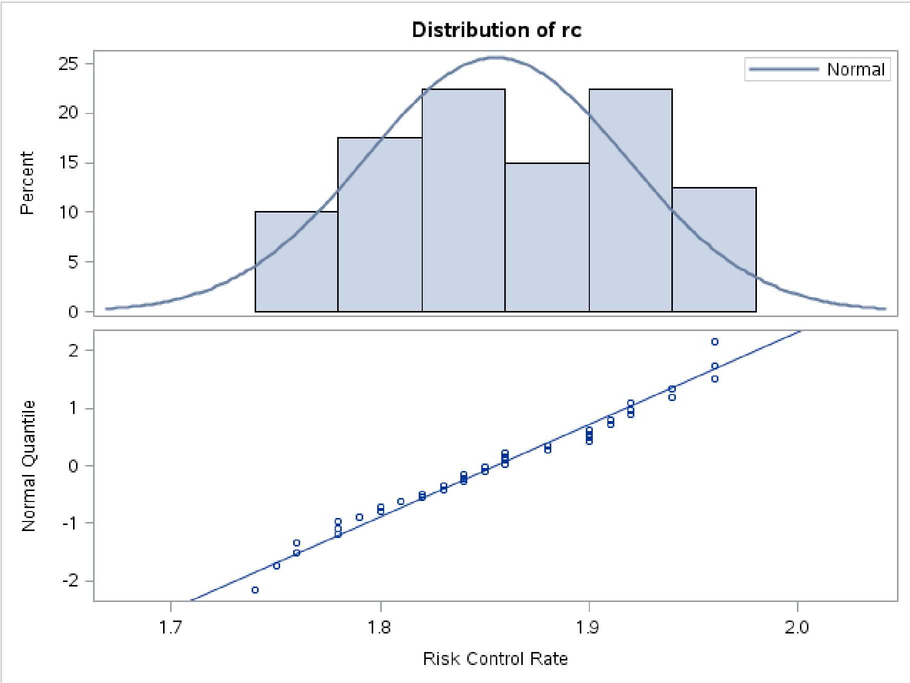


Fig 6. Report result part 1

Fig. 7 shows the skewness and kurtosis of the normality test. Both skewness and kurtosis can test the behavioral trends and patterns of the datasets, and see if they are more towards positive or negative skews, and whether the "normalized curve" is more or more flat, or more focused towards the middle. In Fig. 6, distribution of 'rc' is well-balanced between the two "twin-centers".

Shapiro-Wilk test can determine whether the three variables are either close to the "goodness of the fit", which has the maximum value as 1. The higher the value, the closer the output is close to the goodness of the fit. Results in Fig 6 show they can follow linear regression very closely. However, p-values are high because their output values are not in the "normalized curve" in Fig. 6. This is a healthy sign since actual rate of return, expected rate of return and risk-control rates are not "fixed values" prescribed by the normalized curve. However, another test will be conducted to justify the validity of the tests and datasets in Fig. 7.

Skewness and Kurtosis

		Skewness			Kurtosis		
Variable	Count	Skewness	Standard Error	Z-Score	Kurtosis	Standard Error	Z-Score
a	40	0.19	0.37	0.51	-0.98	0.73	-1.34
e	40	-0.04	0.37	-0.11	-0.95	0.73	-1.30
rc	40	-0.04	0.37	-0.10	-0.96	0.73	-1.30

Page Break

Skewness and Kurtosis

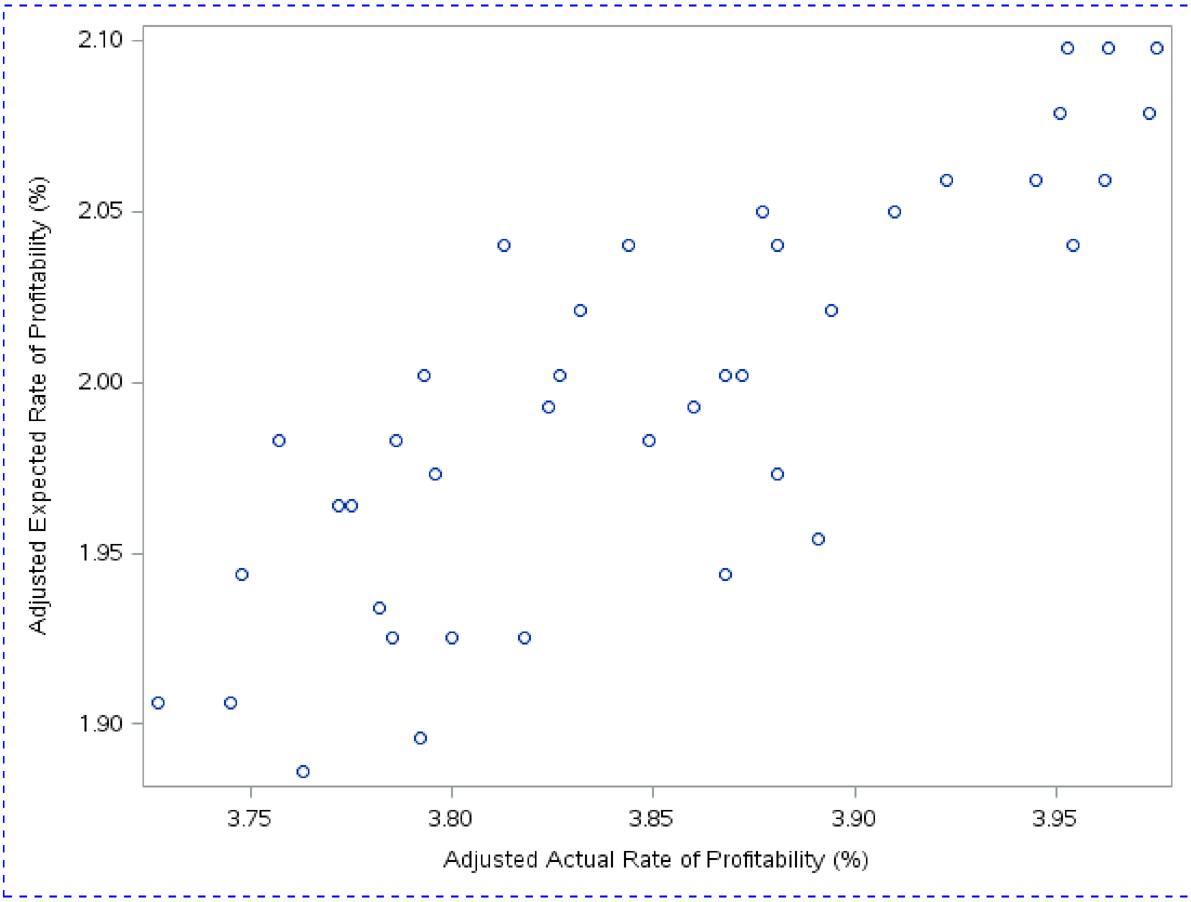
	Shapiro-Wilk Test	
Variable	Statistic	P-Value
rc	0.97	0.27
a	0.96	0.14
e	0.97	0.28

Page Break

Fig. 7 Report result part 2

Fig. 8 shows the scatterplot to identify whether datasets have linear relationship before performing regression. Similarly, outliers can be identified, so that the scientists to decide to include or exclude outliers before performing regressions, since different research can deal with outliers differently. Pearson's coefficient test has been used to determine the coefficient values of variables 'a' and 'e' during linear regression. The magnitude of the coefficients can determine the extent of the regression. This is a more relevant test to justify the p-value for 'a' and 'e'. If all these databsets have patterns similar to the "goodness of the fit", then p-values are small close to 0.

Scatterplot - Test for linear relationship & outliers



Page Break

Pearson's Coefficient

2 Variables:	a_rc e_rc

Simple Statistics							
Variable	**N**	**Mean**	**Std Dev**	**Sum**	**Minimum**	**Maximum**	**Label**
a_rc	40	3.85073	0.07277	154.02900	3.72700	3.97500	Adjusted Actual Rate of Profitability
e_rc	40	1.99733	0.05989	79.89300	1.88600	2.09800	Adjusted Expected Rate of Profitability

Pearson Correlation Coefficients, N = 40 Prob > |r| under H0: Rho=0		
	a_rc	**e_rc**
a_rc	1.00000	0.82868
Adjusted Actual Rate of Profitability		<.0001
e_rc	0.82868	1.00000
Adjusted Expected Rate of Profitability	<.0001	

Fig 8. Report result part 3

Fig. 9 shows key results in OLS, which contains summary of key statistical values. Most important ones include R-squared values. While it is above 0.5 of suggested minimum value, it shows the regression results are fairly consistent. Root of Mean-Squared Error is very low, suggesting range of errors is small. Similarly, standard errors are low and t-values are high for “a_rc”, which stands for adjusted rate of profitability. Pr > |t| stands for p-values for such tests and they have low values under 0.05. All these regression results show the values are acceptable. Durbin-Watson should have an expected value between 1 and 4.

Ordinary Least Squares

Ordinary Least Squares Estimates			
SSE	0.04381924	DFE	38
MSE	0.00115	Root MSE	0.03396
SBC	-151.76963	AIC	-155.14739
MAE	0.0272194	AICC	-154.82306
MAPE	1.37295634	HQC	-153.92609
Durbin-Watson	1.7897	Regress R-Square	0.6867
		Total R-Square	0.6867

Parameter Estimates						
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Variable Label
Intercept	1	-0.6286	0.2878	-2.18	0.0352	
a_rc	1	0.6819	0.0747	9.13	<.0001	Adjusted Actual Rate of Profitability

Fig. 9 Key results of OLS

Fig. 10 shows the adjusted actual an expected rate of profitability of financial service firm, since targets and actual outcomes for each month have been adjusted based on the firm's strategy and business performance. Each datapoint represents actual and expected rate of profitability each month. Over the periods of 40 months, Fig 9 has 40 datapoints. Beta represents the market risk, which can be calculated by finding the gradient of "the best fit". Beta is equal to 0.6819. All datapoints are within 95% confidence interval demonstrating the good quality of datapoints.

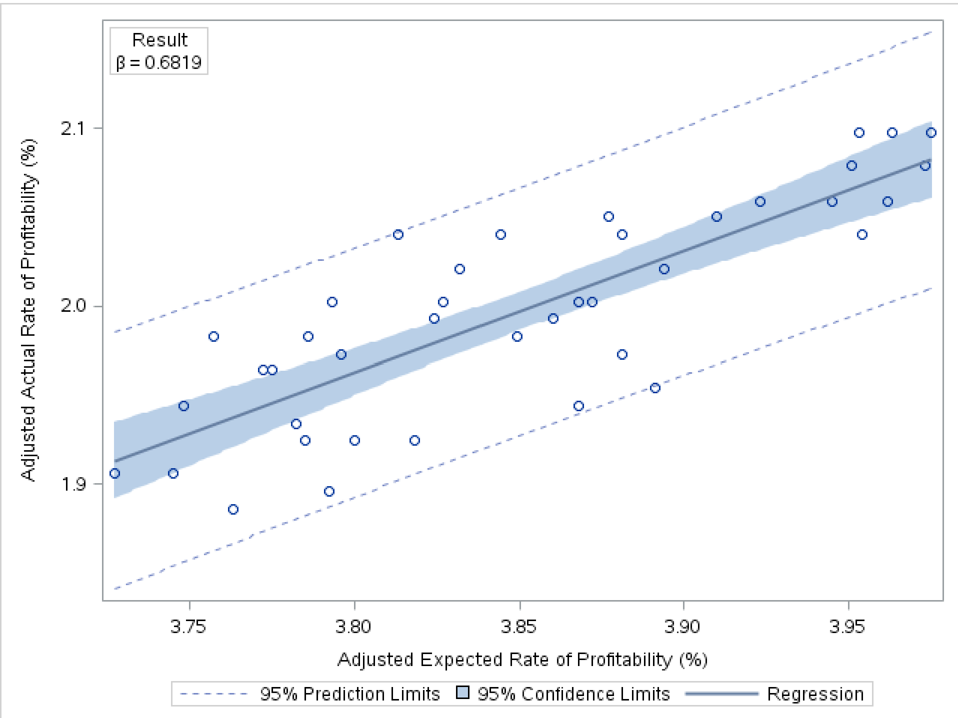


Fig. 10 Adjusted actual and expected rate of profitability

5. Service integration of the RSDPF Framework

5.1 The RSDPF framework for data services

Integration is an important aspect of a framework and this section presents the service integration of the RSDPF framework. Fig. 11 shows the service integration enabled by our framework focusing on big data processing, analysis, storage and integration. It has four layers. The top first layer begins the process with three form of data: analytics, web service and real-time. It first goes to data streaming service and then data security checks, to ensure all data can be safe, clean and trusted to be use. If not, data will be destroyed. If it passes quarantine test, it can proceed to the second top layer following a sequence of events and test. Visualized Services will be the first, to allow data to be visualized, so that tracking and monitoring can be more conveniently used. It then goes to the Service Discovery to identify which service track to follow. Similarly, there is another service wild card to allow new participants on the second layer. Service Mediation is the next sequence to ensure service can be ready. It can also accept requests from SLA service providers. In the next sequence, it is “Message Queue management” to streamline all service sin queue. It then comes to “Service Interface Connections” to connect to different services, followed by “Service Integration of different service buses.

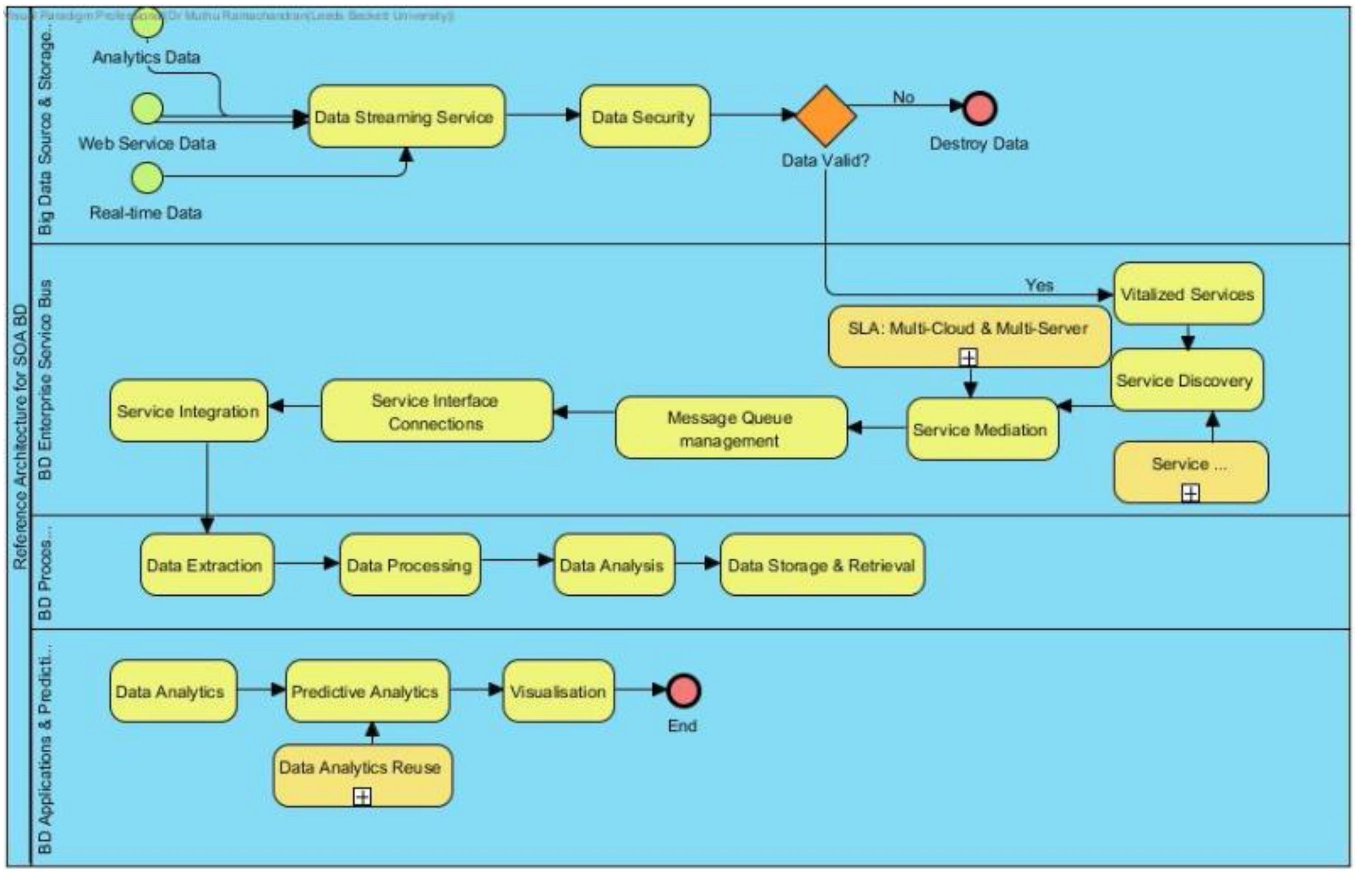


Fig. 11 The service integration enabled by the RSDPF framework

When the Service Integration is completed, it goes to the third top layer, in which big data process is the focus. Data Extraction is used to extract important data from all services. It then follows by Data Processing, Data Analysis and finally Data Storage and Retrieval, so that results of previous tests can be reused. The bottom layer is independent of the top three layers but outputs of these three layers can be presented in the form of Data analytics to show the outputs in graphical formats. Predictive Analysis is the next to forecast the likely outcome and compared with the actual outcome. Visualization is to present results in high-quality images, graphs, videos and multimedia. Examples demonstrated in Section 3 fall into this category. After collecting data related to the return and risk of financial or retailed services, a series of data extraction, processing, analysis, storage and retrieval can be conducted. Results can be presented in analytics which can combine with specific models to predict the risk and return. Advanced techniques can be used to develop features and services for visualization. The difference between the integration by this approach versus data fusion by Sun et al. (2018) is that all integration can be achieved by the workflow, which defines the steps and sequences already. Data is in the uniform format and size for the input. This can streamline the process to analyze ad interpret data.

5.2 Results and analysis for using the data service and workflow

This section shows results and analysis of using the RSDPF framework, including the accuracy and F-measure, as well as execution time. Precision and recall have been commonly used for performance evaluation of research. The third top layer of the RSDPF framework is the benchmark of performance evaluation. If there are 10,000 data altogether to be tested in experiments, then the goal is to identify how many data can be successfully extracted, processed, analyzed and stored. The completion of all four steps in this layer is considered as a successful data service process. In this case, precision is rate of correctly processed all data service requests to the number of all data service requests. Recall is the rate of correctly completed all data service requests to the number of all data service requests. The reason is precision can identify how many data to be done and recall can finalize how many data have their services completed. F-measure is related to the extent of reliability of the RSDPF framework. It can be presented in terms of precision and recall as follows.

$$\text{F-measure} = \frac{2 \times precision \times recall}{precision + recall} \quad (11)$$

The higher the value of F-measure, it has a higher reliability since all the data requests can be identified and then completed with their service requests.

The next step is to perform experiment to identify F-measure values. 1,000 data can be used each time to test the capacity and reliability of the RSDPF framework. Five experiments are then conducted to get the mean values. Each time 1,000 data are added up, until it reaches to 10,000 data. F-measure values in percentage are then recorded. As shown in Fig. 12, F-measure has high percentages throughout the experiments. It has started from 99.8% with 1,000 data as the inputs and the F-measure values go down to 97.3% eventually for 10,000 data as the input. Results show that F-measures are highly consistent. During the experiments, execution time for inputting service data and service completion are measured, with the mean values for five experiments recorded. Fig. 13 shows the execution time for inputting between 1,000 and 10,000 data for the RSDPF framework. Results also follow the linear relationship, meaning the service completion can be within the expected range of completion time. The lowest execution time is 228 seconds and the highest is 3004 seconds. In other words, all services can be completed in 51 minutes while serving a large quantity of data requests.

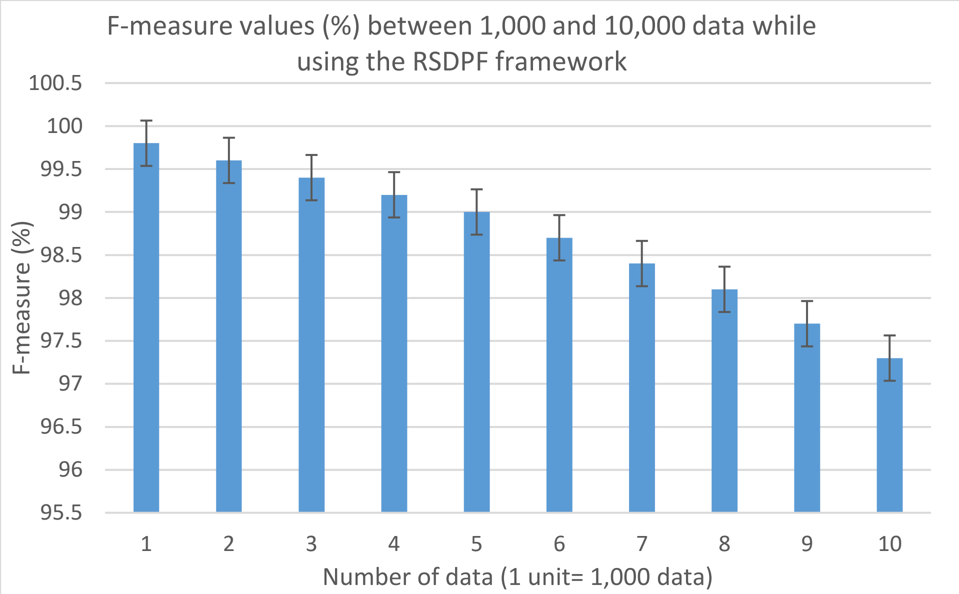


Fig. 12 F-measure values (%) between 1,000 and 10,000 data while using the RSDPF framework

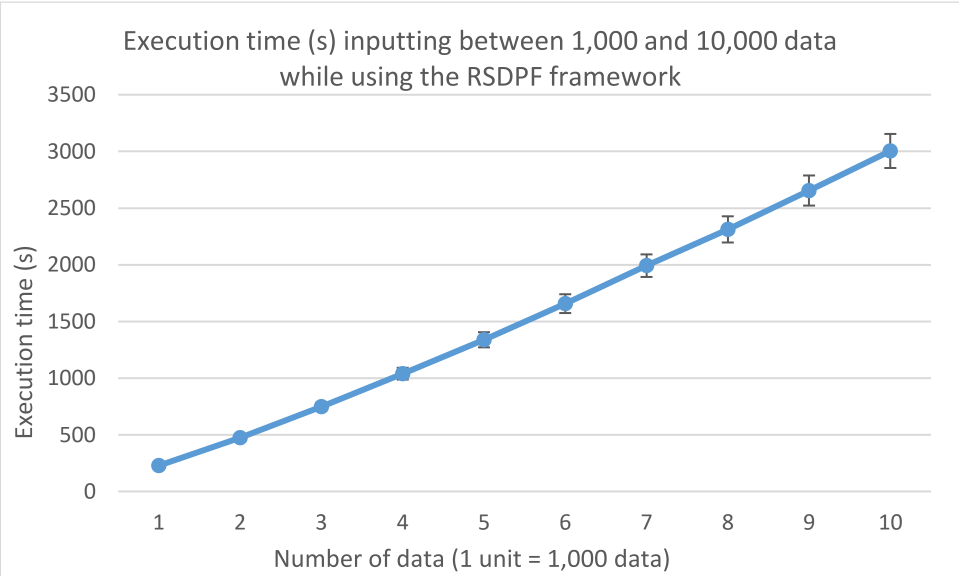


Fig. 13 The execution time for inputting between 1,000 and 10,000 data for the RSDPF framework

5.3 Comparison with other frameworks

Damschroder et al. (2009) have developed a conceptual framework for health services. The focus on the research issues and concerns raised in the past. They ensure that their implementation of their framework can make recommended practices. By following policies and recommendations, they allow good practices in heal services to be validated. Patton and McMahon (2006) propose a theory framework of the career development and counselling. They describe the elements and processes for each of career development and counselling. By following the recommended processes, each individual can improve and get closer to their goals and eventually can make all the theories into practices. Brunch et al. (2010) use Eclipse JFace to develop and reuse their code and improve the state of an API document. Without those environment, it cannot function well. Reuse and integration should be independent of the package like our approach. Xin and Yang (2017) define their frameworks with multi-functional and multi-purpose characteristics. They have the workflow to design the processes and software engineering approach to manage the quality of their processes. Although they have positive influences to our work, they do not have the real life examples to consolidate like our framework does.

One of our objectives is to transform the abstract and theoretical concepts into data and visual analytics, so that we can perform financial and risk analysis, as demonstrated between Sections 4 and 5.2. We use OSM to integrate ANT and TOPSIS techniques and acts as the "engine behind the framework". We allow data processing, analysis and visualization of our outputs, with the particular focus to the risk and return of our financial analysis. The RSDPF framework can be used to demonstrate reuse and integration, as well as to make theoretical work to a service that can compute risk and return for financial.

5.4 Justification of research contributions

In this paper, we have demonstrated the theory, use and deployment of the RSDPF framework. Theoretical development is based on the integration of ANP and TOPSIS techniques. In section 4, data was from a financial service firm. We cannot enclose the identity due to the agreement. However, the workflows, results, analysis and interpretations were already described in Section 4 to show the existence of the case. Our approaches and analysis were presented to demonstrate the effectiveness of the RSDPF framework with results explained and justified for reuse and integration. In order to justify our research contributions and validity of our framework, we explain as follows.

- Reuse and integration in statistical data mining: Section 4 has demonstrated the full use case for statistical data mining. Implicit meanings of data analysis have bene fully extracted and explained in full details with the support of the results and analysis.
- Execution of three tests in one go: The developed service shown in Fig. 5 allows all three tests to be executed and analyzed at once rather than performing them individually.
- Reuse and integration in knowledge and heuristic discovery: Before deploying the RSDPF framework, the financial services firm did not know implications to them about risk and return. Both the rates of actual and expected return of profitability were computed, as well as the underlying rate of risk (beta) was calculated through the linear regression by our model, OSM.
- Reuse and integration in domain transference: The demonstrated work can be applied in different domains since it can be used for data-driven or data-oriented services. In many sectors, service-based data can be generated, transferred and analyzed by different service providers and platforms at different periods of time.

Our RSDPF framework can be used in such a way to process, analyze, present and interpret data, allow the execution of tests at once, and enable service integration to take place. Our research work can be applied in other areas, as long as the data can be provided. It is the data that we analyze. This can break away from certain restrictions imposed by domain specific issues.

6. Conclusion

Reuse and integration play important roles for software engineering and service computing. Demands for data mean innovative ways should be developed. In this paper, we demonstrated our proposed RSDPF framework based on blending ANP and TOPSIS techniques, and predictive analytics patterns. A real financial service firm's case was used to demonstrate a successful use case. The RSDPF framework allows easy use of code reuse, with three step tests and financial analysis performed. Therefore, the actual, expected and risk rates of profitability could be calculated. Results and analysis can provide real insights to the firm. Additionally, service integration of the RSDPF framework was illustrated. Four layers of services were explained. Large-scale data tests on service integrations were performed and the framework was confirmed with a high extent of reliability. Reuse and integration can play crucial roles for different sectors and projects. In this paper, all these examples can be fully transferred in other domains. We also justified our research contributions in reuse and integration. Our future work will also expand the RSDPF framework in cybersecurity.

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439.

Adalı, E. A., & Işık, A. T. (2017). The multi-objective decision making methods based on MULTIMOORA and MOOSRA for the laptop selection problem. Journal of Industrial Engineering International, 13(2), 229-237.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing, 17.

Barga, R., Fontama, V, & Tok, W. Y (2014) Predictive Analytics with Microsoft Azure Machine Learning: Build and Deploy Actionable Solutions in Minutes, Apress/Springer, ISBN 978-1-4842-0446-7.

Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2008). JADE: A software framework for developing multi-agent applications. Lessons learned. Information and Software Technology, 50(1), 10-21.

Boehm, B. (2006, May). A view of 20th and 21st century software engineering. In Proceedings of the 28th international conference on Software engineering (pp. 12-29). ACM.

Bonaccorsi, A., & Rossi, C. (2003). Why open source software can succeed. Research policy, 32(7), 1243-1258.

Bruch, M., Mezini, M., & Monperrus, M. (2010, May). Mining subclassing directives to improve framework reuse. In Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on (pp. 141-150). IEEE.

Chang, V. (2014). A proposed model to analyse risk and return for Cloud adoption. Lambert Academic Publishing, ISBN: 978-3-659-58769-6.

Chang, V., & Wills, G. (2016). A model to compare cloud and non-cloud storage of Big Data. Future Generation Computer Systems, 57, 56-76.

Chang, V. (2017). Presenting Cloud Business Performance for Manufacturing Organizations. Information Systems Frontiers, 1-17.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19(2), 171-209.

Cockburn, A. (2006). Agile software development: the cooperative game. Pearson Education,2nd Edtion, ISBN 0321482751.

Cordell, D., Rosemarin, A., Schröder, J. J., & Smit, A. L. (2011). Towards global phosphorus security: A systems framework for phosphorus recovery and reuse options. Chemosphere, 84(6), 747-758.

Damschroder, L. J., Aron, D. C., Keith, R. E., Kirsh, S. R., Alexander, J. A., & Lowery, J. C. (2009). Fostering implementation of health services research findings into practice: a consolidated framework for advancing implementation science. Implementation science, 4(1), 50.

Engwall, M. (2003). No project is an island: linking projects to history and context. Research policy, 32(5), 789-808.

Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Adobe Reader). Pearson Education.

Jennings, N. R. (2001). An agent-based approach for building complex software systems. Communications of the ACM, 44(4), 35-41.

Khan, W. Z., Xiang, Y., Aalsalem, M. Y., & Arshad, Q. (2013). Mobile phone sensing systems: A survey. IEEE Communications Surveys & Tutorials, 15(1), 402-427.

Kirk, D., Roper, M., & Wood, M. (2007). Identifying and addressing problems in object-oriented framework reuse. Empirical Software Engineering, 12(3), 243-274. Kirk, D., Roper, M., & Wood, M. (2007). Identifying and addressing problems in object-oriented framework reuse. Empirical Software Engineering, 12(3), 243-274.

Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., ... & Rosson, M. B. (2011). The state of the art in end-user software engineering. ACM Computing Surveys (CSUR), 43(3), 21.

Lee, S., Kang, Y., Ialongo, N. S., & Prabhu, V. V. (2016). Predictive analytics for delivering prevention services. Expert Systems with Applications, 55, 469-479.

Leung, C. K., Jiang, F., Zhang, H., & Pazdor, A. G. (2016, August). A Data Science Model for Big Data Analytics of Frequent Patterns. In Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2016 IEEE 14th Intl C (pp. 866-873). IEEE.

Mather, T., Kumaraswamy, S., & Latif, S. (2009). Cloud security and privacy: an enterprise perspective on risks and compliance. " O'Reilly Media, Inc.".

Patton, W., & McMahon, M. (2006). The systems theory framework of career development and counseling: Connecting theory and practice. International Journal for the Advancement of Counselling, 28(2), 153-166.

Papazoglou, M. P., & Heuvel, W. J. (2007). Service oriented architectures: approaches, technologies and research issues. The VLDB Journal—The International Journal on Very Large Data Bases, 16(3), 389-415.

Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F. (2008). Service-oriented computing: a research roadmap. International Journal of Cooperative Information Systems, 17(02), 223-255.

Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave Macmillan, 6th Edition, ISBN 0-07-285318-2.

Ramachandran, M (2008) Software components: guidelines and applications, Nova science, NY

Ramachandran, M and Jamnal, G (2014) Developing reusable .NET software components, Science and Information Conference (SAI), 2014

Russom, P. (2011). Big data analytics. TDWI best practices report, fourth quarter, 19, 40.

Saaty T. L. (1998), What is the analytic hierarchy process?, in Mathematical models for decision support, ed: Springer, 1988, 109-121.

Saaty, T. L. (1996). Decision making with dependence and feedback: The analytic network process (Vol. 4922). Pittsburgh: RWS publications.

Saaty T. L., Decision making—the analytic hierarchy and network processes (AHP/ANP), Journal of systems science and systems engineering, 13, 1-35, 2004.

Shawe-Taylor, J., & Cristianini, N. (2004). Kernel methods for pattern analysis. Cambridge university press.

Sun, G., Chang, V., Yang, G., & Liao, D. (2018). The cost-efficient deployment of replica servers in virtual content distribution networks for data fusion. Information Sciences, 432, 495-515.

Tzeng, G. H., & Huang, J. J. (2011). Multiple attribute decision making: methods and applications. CRC press.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Xin, T., & Yang, L. (2017, June). A framework of software reusing engineering management. In Software Engineering Research, Management and Applications (SERA), 2017 IEEE 15th International Conference on (pp. 277-282). IEEE.