FUZZY NETWORK BASED FRAMEWORK FOR SOFTWARE MAINTAINABILITY PREDICTION

XIAOWEI WANG  
Department of computer, Army Engineering University of PLA,  
Wuhan, 430075, P.R. China  
xiaowei.wang@outlook.com

ALEXANDER GEGOV  
School of computing, University of Portsmouth, Buckingham Building  
Portsmouth, PO1 3HE, United Kingdom  
alexander.gegov@port.ac.uk

ARABIKHAN FARZAD  
School of computing, University of Portsmouth, Buckingham Building  
Portsmouth, PO1 3HE, United Kingdom  
arabikhan.farzad@port.ac.uk

YUNTAO CHEN  
Department of radar, Army Engineering University of PLA,  
Wuhan, 430075, P.R. China  
yuntao.chen@163.com

QIWEI HU  
Department of Management Engineering, Army Engineering University of PLA,  
Shijiazhuang, 050003, P.R. China  
qh.w9163.com

Received (received date)  
Revised (revised date)

Software metrics based maintainability prediction is leading to development of new sophisticated techniques to construct prediction models. This paper proposes a new software maintainability prediction framework, which bases on Fuzzy Network, a novel exploratory modeling technique. The proposed framework utilizes both the metric data collected from software system and the subjective appraisals from experts. An application example of the framework is shown. In comparison to the Standard Fuzzy System based models, Fuzzy Network based models improves the transparency more than 71.3% and the accuracy more than 11.0%. It is confirmed that Fuzzy Network based framework is more appropriate for constructing SMP model.

Keywords: Software maintainability, prediction, Fuzzy Network, metrics, transparency
1. Introduction

Software maintainability, the ease with which a software system can be modified, is an important quality attribute and intrinsically associated with maintenance process. Predicting software maintainability commonly aims to estimate the cost of software project or the maintenance process. The focus of this study is on the latter one, in other words, predicting of the quality attribute of maintainability. Software metrics based software maintainability prediction (SMP) has been discussed extensively. The metrics are mainly aiming for object-oriented software\textsuperscript{1–5} and aspect-oriented software\textsuperscript{6,7}. Some systematic literature reviews have detailed the topic about software metrics of SMP\textsuperscript{8,9}. A large amount of prediction models have been built based on various techniques\textsuperscript{10–12}. However, most of the models are either problem dependent or only effective in special fields. The reasons are explained as follows.

- Different models for SMP concern different software metrics.
- Different software programming paradigms are used in different software projects to be maintained. These paradigms include Object-oriented programming (OOP), Aspect-oriented programming (AOP), Context-oriented programming (COP)\textsuperscript{8} and the earlier one, Procedural-oriented programming (POP).
- Different models are required to deal with different type of data which is collected during the prediction. The data are classified into two types: numerical information obtained from measurements, and, linguistic information obtained from human experts\textsuperscript{14}. Sometimes they are named as “crisp” and “fuzzy” information respectively.

It seems hard to build one model to fulfill all the needs of SMP.

A major challenge, is that effective prediction models should consider two types of knowledge: imprecise linguistic information from the experts and precise numerical information from historical data\textsuperscript{15}. Incorporating the experts’ knowledge poses a major constraint on the prediction model, the model should be transparent. Transparency helps the experts to directly manipulate the model structure for any necessary additions or modification. On the other hand, the experts’ knowledge is critical and beneficial for one prediction model to be effective. Historical data, or “crisps” information, provides numerical quantitative measurements from past projects regarding the internal and external quality attributes. Experts utilize their experience to provide “fuzzy” information, or qualitative descriptions of the correlation between the internal and external quality attributes. Similarly, when using the model for actual prediction, there are two sources of information: inspection data and practitioners. To make the SMP model more practical, transparency is crucial as it allows the practitioners to provide their own judgment on the predictors in linguistic terms. It is in contrast to the more theoretical black box prediction models, for example, neural network (NN)-based models, which allows neither the
incorporation of the knowledge of experts in building the model, nor the linguistic
measurements.

The primary contribution of this paper is that both numerical data from metrics
and linguistic appraisals are organized into a hierarchical structure. The proposed
framework is able to utilize both “fuzzy” and “crisp” information. The existent
SMP models seems far from being transparent enough.

Another contribution is that a Fuzzy Network (FN) based SMP framework has
been proposed to describe the relations between numerical data and linguistic ap-
praisals in a hierarchical structure. To the best of our knowledge, it is the first
time that FN is utilized for SMP. The framework is transparent enough because
FN is characterized by a white-box nature. More characters about FN will be
introduced in section 2.

The remaining part of this paper is structured as follows: Section 2 reviews
concepts from software metrics, fuzzy networks, and the difficulties about prediction.
Section 3 describes the framework of deriving SMP model. Section 4 presents an
application example of the framework. Section 5 provides conclusion and future
work.

2. Related work

This section reviews the relevant literatures on software metrics, FN, and the diffi-
culties in SMP. In particular, the use and analysis of software metrics and FN are
summarized, and this summary leads to the motivation of this research.

2.1. Software metrics

Historically speaking, to gather meaningful metrics in measuring software main-
tainability has been struggled for several decades.

Many traditional software metrics have come under academic scrutiny, includ-
ing: Halstead’s software metrics,18 McCabe’s cyclomatic metric,19 Henry and Ka-
fura’s information flow metric,20 Robillard’s statement interconnection metric,21
and Adamov’s hybrid metrics.22 These metrics are proposed primarily to measure
procedural-oriented software.

With the prevalence of object-oriented (OO) software a large amount of OO met-
rics have been devised. A tree structure of well-known maintainability metrics is
useful to evaluate OO software maintainability.23 92 metrics have been organized
into the hierarchical structure by Oman and Hagemeister. Judging from the pub-
lishing year and the software development language, they are mainly aiming for
OO paradigm. Although some of the metrics are subjective product appraisals or
only with linguistic information, the hierarchical metrics is useful for constructing
transparent framework proposed in this paper.

Six OO design metrics to predict the maintainability of OO paradigm are pro-
posed,1 and an automated data collection tool to collect an empirical sample of
these metrics is introduced in the same paper. This kind of tool is useful because
the effective use of software metrics is dependent on the statistical validation of the metrics. A similar metrics with 6 additional OO metrics is proposed and validated by the maintenance data collected from two commercial software by Li and Henry. The proposed metrics suite have been evaluated analytically by researchers, and the collected data have also been cited widely. In this paper, these data are also used to validate the proposed framework.

Most of existing OO metrics cannot be applied directly to AOP, though AOP is proposed as a supplement to OOP by supporting widespread implementation of crosscutting concern into a single unit called aspect. Suites of Aspect-oriented(AO) metrics have been suggested. The understandability metrics and cohesion metrics are suggested separately. A metrics took into account relatedness among class and aspects. These AO metrics are mostly based on OO metrics and the data sometimes collected by questionnaire are a kind of “fuzzy” information. Collaborated with the “crisp” information, all the information are useful for AO software maintainability prediction.

Recent years, large-scale OO software systems have been found to share global network characteristics such as small world and scale free, which go beyond the scope of traditional software measurement and assessment methodologies. A new hierarchical set of metrics is proposed which is in terms of coupling and cohesion to measure the complexity at various levels of granularity, namely graph, class (and object) and source code. Graph theory metrics are applied onto the transformed network with the purpose to evaluate the open-source software systems. The maintenance data during these open-source software systems’ updating are needed in order to obtain a comprehensive body of knowledge and experience. Unfortunately, the data scarcity are also a common problem for open-source software engineering research.

2.2. Fuzzy Networks

Fuzzy logic, introduced by Prof. Zadeh, has proved itself as a powerful tool for dealing with problems which are too complex to be understood quantitatively. For example, there are a large amount of fuzzy logic based SMP models for dealing with the linguistic information and “fuzzy” information of software metrics. Recent years, neural network and Bayesian associating with fuzzy logic are also utilized in SMP for better prediction performance. However, the fuzzy logic models mentioned above, with Fuzzification-Inference-Defuzzification (FID) sequence, are not with transparent structures and unable to reflect adequately any interacting modules within a modeled process. Due to the black-box nature, most fuzzy models cannot take into account explicitly any interactions among sub-processes.

The most common type of fuzzy system is the one with a single rule base. Referred as Standard Fuzzy System(SFS), it is characterized by a black-box nature whereby the inputs are mapped directly to the outputs without the consideration of any internal connections. SFS is usually quite accurate for output modeling as it
FUZZY NETWORK BASED FRAMEWORK FOR SOFTWARE MAINTAINABILITY PREDICTION

Fig. 1. Fuzzy network

reflects the simultaneous influence of all inputs on the output. But, as the number of rules increases, the transparency of SFS deteriorate and it is less clear how the model output is affected by the model inputs.

Chained Fuzzy System (CFS) cascaded multiple rule bases, are characterized by a white-box nature whereby the inputs are mapped to the outputs by means of some internal variables in the form of connections. CFS has an arbitrary structure in terms of subsystems and the connections among them. CFS is usually used as a detailed presentation of SFS for the purpose of improving transparency by explicitly taking into account all subsystems and the interactions among them. Also, efficiency is improved because of the smaller number of inputs to the individual rule bases.

Hierarchical Fuzzy System (HFS) is a special type of CFS that has a specific structure. Each subsystem in HFS has two inputs and one output.

Fuzzy Network, a novel concept introduced by Dr. Alexander Gegov,\(^{17}\) is the one with networked rule bases. It is characterized by a white-box nature whereby the inputs are mapped to the outputs by means of connections.\(^{16}\) In this paper, FN can be seen as a hybrid between SFS and HFS. On one hand, the structure of FN is similar to the structure of HFS due to the explicit presentation of subsystems and the interactions among them. On the other hand, the operation of FN resembles the operation of SFS as the multiple rule bases are simplified to a linguistically equivalent single rule base. This simplification is based on the linguistic composition approach. Based on the horizontal merging and vertical merging of rule bases, a multiple rule base system such as HFS can be first converted into FN and then the latter is composed into a single rule base system such as SFS. The Theorem 1 is the important theoretical base for the transparent SMP model proposed in this paper.

Theorem 1 A HFS with set of m inputs\(\{x_1, x_2, \ldots, x_m\}\), a set of m-1 network nodes \(\{N_{11}, N_{12}, \ldots, N_{1,m-1}\}\), a set of m-2 connections \(\{z_1, z_2, \ldots, z_{m-2}\}\) and a single output \(y\), as described by the topological expression in Eq.(1)

\[
[N_{11}](x_1, x_2|z_1) \ast [N_{12}](z_1, x_3|z_2) \ast \ldots \ast [N_{1,m-1}](z_{m-2}, x_m|y)
\] (1)

as described by the block-schemes in Fig.1, can be represented as a SFS with the same set of m inputs, a single network node N, no connections and the same single
output, and the topological expression in Eq.(2)
\[
m - 1 \sum_{p=1}^{m-1} \left( N_{1p} + \sum_{q=p+1}^{m-1} I_{qp} \right) (x_1, x_2, \ldots, x_m | y)
\] (2)
where \( N = \prod_{p=1}^{m-1} (N_{1p} + \sum_{q=p+1}^{m-1} I_{qp}) \), node \( I_{qp} \) is identity node in level \( p \) and layer \( q \).

2.3. Modeling problems

The number of rules in SFS increases exponentially as the number of the system variables based on fuzzy rules increased. In HFS, the number of rules increases linearly with the number of system variables. The best application of HFS would be systems with a natural hierarchical structure. In such a case, \( z_1, z_2 \ldots \) correspond to the physical variables of the system. If this is not the case, the \( z_1, z_2 \ldots \) can still be interpreted as the “internal state variables” of the system. This is analogous to the states in the state-space representation of systems, where a state characterizes some key feature of the system but does not necessarily correspond to any physical variable.

Our critical evaluation reveals that no single soft computing-based SMP system exists that incorporates the following:

- Tolerance of imprecision;
- Incorporating expert’s knowledge in a well-defined manner;
- Total transparency in the prediction system;
- Ability to explain prediction results through rules or other means;
- Adaptability to cope with continuous change in software development technologies and environments.

In the real maintaining project, there are two kinds of data to be dealt with: the numerical data and the expert’s linguistic information. After the software been maintained, there are data tables with numerous attributes of this software and a large number of records during the whole developing process. It’s extremely difficult for humans to understand the relations between the independent variables and the dependent. What’s more, some real data can be very bad, which is always taken as “wild data”. That means not all the real numerical data have same reliabilities.

There are two approaches to generate rule base. In the knowledge driven fuzzy models, the requisite rule base is provided by domain experts and knowledge engineers. In the data driven models, the rule base is generated from the available numerical data. Because it is difficult to find the domain experts and to extract the requisite knowledge from the experts, the data driven modeling assumes significance. Under the circumstances of highly complex and highly nonlinear systems, the rule base extraction problem becomes NP hard problem. When the problem is very complex, application of classical methods turns out to be computationally expensive. At this time, Fuzzy Network is particularly useful for SMP problems which
cannot be easily represented by mathematical modeling because the relationship between software metrics and maintainability is complex and non-linear. A suit of methods are needed in the process of building the prediction model.

3. Prediction model design

As shown in Fig.2, the SMP model based on the proposed framework is a multi-input-single-output (MISO) fuzzy system, and also consists of four major modules: fuzzification, inference engine, knowledge base and defuzzification module. The fuzzification module transforms the crisp inputs of independent variables into linguistic terms. These linguistic terms are then processed in fuzzy domain by inference engine based on the knowledge base from software experts and engineers. The knowledge base is composed of the Rule Base, characterizing the control goals and control policy by a set of linguistic control rules; and of the data base, containing the linguistic term sets and the membership functions defining their semantics. Finally, the processed output (CHANGE) is transformed from fuzzy domain (linguistic terms) to crisp domain (numerical number) by defuzzification module.

There are two kinds of Fuzzy Inference System Modeling: Mamdani and Sugeno fuzzy inference systems. Because it is a more compact and computationally efficient representation than a Mamdani system, a Sugeno system lends itself to the use of adaptive techniques for constructing fuzzy models. These adaptive techniques can be used to customize the membership functions so that the fuzzy system best models the data. And by comparing the simulation results shown in Section 4, the Sugeno FN based SMP model is superior to the Mamdani one in terms of accuracy. Therefore, our prediction model is constructed as a kind of Sugeno system.

Construction a SMP model based on the framework includes 6 steps as shown below.
3.1. Selection of input variables

The linguistic information can be organized into hierarchical structure. It is confirmed the known linear relationship between software metrics and change rates of code lines. To better understand these results, maintainability can be described in terms of a hierarchy.

In order to reduce the complexity, only those metrics and attributes with significant contribution to the maintainability are selected. In practical software metrics, we may have the knowledge that some metrics are more important than others. Sometimes, we can determine a ranking of importance of the metrics involved.

In this step, the input variables $x_i (i = 1, 2, \ldots, m + n)$ are divided into two sets. As showed in Fig.2, there are $m+n$ inputs. SET1 is composed of the variables $x_i (i = 1, 2, \ldots, m)$ with numerical data, and SET2 is composed of the variables $x_i (i = m + 1, m + 2, \ldots, m + n)$ only with the expert’s linguistic information. Combined with the numerical data of output $z_1$ (here $z_1$ is CHANGE), the numerical data are transformed into a set of input-output data pairs.

3.2. Presentation of inputs by linguistic terms

Each input space is divided into $2^K+1$ sections. $K$ can be different for different input variables, and the lengths of these sections can be equal or unequal. NN, Genetic algorithms (GA) and Fuzzy C-means clustering (FCM) etc. can be utilized to partition input and output variable ranges. FCM, which is far more widely applied, only requires the expected number of clusters ($2^K+1$ sections) as an input, does not require cluster characteristics to be specified a priori. The power of a clustering algorithm is that it does not rely on predefined classes in the way that NN and other supervised learning approaches do. Instead, clustering algorithms search for the structures naturally present in the data. Therefore, the input spaces in SET1 are decided by FCM. It means all the inputs of node $N_{11}$ in Fig.2 are divided into $2^K+1$ sections by FCM.

3.3. Development of membership functions

The process of assignment membership values or functions can be intuitive or it can be based on some algorithmic or logical operations. Six straightforward methods are described in the literature to assign membership values or functions to fuzzy variables. Based on the judgment about the probability density functions from researchers and ourselves, the shape of membership function of input variables of node $N_{11}$ is Gaussian curve. The parameters of Gaussian functions for these variables are determined by machine learning method from numerical data.

For non-identity nodes, a natural membership function type that readily comes to mind is the triangular membership functions. Two main reasons motivate the choice: one is their optimal interface design and the other is its semantic integrity. The triangular membership functions are shown in Fig.3, where $x_{ij}$
is the jth value of input variable $x_i$, $f_i$ is its membership ($m < i \leq m+n$), and $f_{ij}$ of $x_{ij}$ is decided by the section $R^h_a (0 < h \leq 2K+1, R^h_a \in R_a)$. In other words, $f_{ij}$ is the max one in the memberships of $x_{ij}$ in all sections $R_a$. Then, $x_{ij}$ is considered to be $R^h_a$ with the membership $f_{ij}$.

$$f_{ij} = 0, \text{ if } x_{ij} < t^{-1}_{mid} \text{ or } x_{ij} > t^{h+1}_{mid}$$

$$f_{ij} = (x_{ij} - t^{-1}_{mid})/(t^h_{mid} - t^{-1}_{mid}), \text{ if } t^{-1}_{mid} \leq x_{ij} < t^h_{mid}$$

$$f_{ij} = (t^{h+1}_{mid} - x_{ij})/(t^h_{mid} - t^{h+1}_{mid}), \text{ if } t^h_{mid} < x_{ij} \leq t^{h+1}_{mid}$$

Where $t^h_{mid}$ is the center of section $R^h_a$, $t^{-1}_{mid}$ is the center of left neighboring section $R^{h-1}_a$, $t^h_{mid}$ is the center of right neighboring section $R^{h+1}_a$. The symbol '/' denotes arithmetic division. Specially, when $x_{ij} = t^h_{mid}$, $f_{ij}$ reaches its maximum at 1. Similarly to the inputs, the membership $f^j_{z_1}$ of the output $z_1$ is decided by the section $R^h_b (0 < h \leq 2L+1)$ with maximum degree and $z_{1j}$ is considered to be $R^h_{b_j}$.

### 3.4. Generation of rule base

The rule base of $N_{11}$ is obtained from the pairs of desired input-output data by machine learning. $C_{mj}$ is Cluster mj, and the rule base of $N_{11}$ includes $2K+1$ rules, which are shown below:

**Rule $j$:**

If $x_1$ is $C_{1j}$ and $x_2$ is $C_{2j}$ and ... and $x_m$ is $C_{mj}$ then $z_{1j} = f(x_1, x_2, \ldots, x_m)$

... Where $0 < j \leq 2K+1$.

It’s easy to decide the rules generated in this way are “and” rules. All the input variables simultaneously affect the output, and rules in antecedent must be met simultaneously in order for consequent to occur. For the SMP problems, only “and” rules are required since the antecedents are different components of a single input vector.
3.5. Generation of hierarchical structure

In SET2, the inputs are Subjective Product Appraisals (SPA) and evaluated by 2K+1 points. For example, if K=1, the positive integers 1-3 present the choice scale (Low, Nominal, High). Fig. 4 is a simple example to illustrate the generation progress.

The rule bases are derived from the knowledge of experts and engineers. It’s worth noting that the quality of the rule bases affect the output in some degree. The methods to improve the quality are not discussed in this paper.

3.6. Derivation of crisp value for maintainability

In this step, the rule bases of all nodes are merged vertically and then horizontally. That’s to say, the rule bases of $N_{11}, N_{21}, I_{31}$ and $I_{n,1}$ are merged into rule base $V_1$, and then rule base $V_2$ from $I_{12}, N_{22},...$ and $I_{n,2}$. Rule base $V_{n-1}$, from the last vertical merging the rule bases of $I_{1,n-1}$ and $N_{2,n-1}$. All the rule base of $V_i (0 < i \leq n-1)$ are merged horizontally. By this way, the single rule base RB of the whole SMP model is achieved, as shown by topological expression in Eq.(3) where ‘+’ means vertical merging operation and ‘*’ means horizontal merging operation.

\[
[(N_{11} + N_{21} + I_{3,1} + ... + I_{n,1}) * (I_{12} + N_{n-1,2} + ... + I_{n,2}) * ... * (I_{1,n-1} + N_{2,n-1}) * N_{1,n}](x_1, ..., x_{m+n}|y)
\]

Both vertical and horizontal merging can be operated automatically with the help of functions which have been developed by PhD Nedyalko Petrov in University of Portsmouth.

After the multiple rule bases are merged into a linguistically equivalent single rule base, the FN model of SMP is a kind of FS model with m+n inputs, one output (maintainability) and rule base RB.

And then, the SPAs for each input in SET2 are transformed to numerical data, as shown in equation(4).

\[
SPA_i = \frac{100*2^{(2i-1)}}{2^{2K+3}}
\]

Where $0 < i \leq 2K + 1$, i is the positive integers. $SPA_i$ for $x_{m+1}$ to $x_{m+n}$ are added into data pairs of $x_1$ to $x_m$.
The crisp value for maintainability is achieved after the data pairs have been inputted.

4. Framework application example

In this section, the framework to construct a SMP model is applied. And, the dataset is from object-oriented maintainability datasets. Measurements in the datasets are collected from a total of 110 classes in two software systems. The UIMS and QUES datasets contain 39 classes and 71 classes respectively. Maintainability is measured by CHANGE which recorded the number of changed lines in the code during a three-year maintenance period. In this research, QUES is selected because more samples (classes) it has.

4.1. Framework application process

As shown in Fig.5, an example model of SMP framework is built up. The process to build up the model shows in the following 6 steps.

4.1.1. Selection of input variables

There are 10 metrics in the QUES dataset as shown in Table1. In order to select 3 input variables, Pearson’s and Spearman’s Correlation Coefficients are used to get top 3 relevant independent variables to the dependent variable, and to indicate the strength and direction of a linear relationship between two variables. As shown in Table1, SIZE1, MPC, RFC and WMC are the top 4 ones which have the significant Pearson’s and Spearman’s correlation with CHANGE. Because the Spearsan’s correlation coefficient is insignificant (0.084) between WMC and CHANGE, WMC is deleted from the input variables. Therefore, MPC, RFC and SIZE1 are the three input variables, named as \( x_1, x_2 \) and \( x_3 \). The definitions of inputs in SET2 depend on the needs of prediction. Derived by the knowledge of experts and engineers, the inputs of SET2 are \( x_4 \) (Extent of the supporting document set), \( x_5 \) (Comprehensibility of the supporting document set) and \( x_6 \) (Programming language complexity).
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
<th>Pearson</th>
<th>Spearman</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIT</td>
<td>Depth of the inheritance tree</td>
<td>-0.090</td>
<td>-0.041</td>
</tr>
<tr>
<td></td>
<td>Inheritance level number of the class, 0 for the root class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOC</td>
<td>Number of children</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>Number of direct sub-classes that the class has</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPC</td>
<td>Message-passing coupling</td>
<td>0.461**</td>
<td>0.547**</td>
</tr>
<tr>
<td></td>
<td>Number of send statements defined in the class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RFC</td>
<td>Response for class</td>
<td>0.388**</td>
<td>0.379**</td>
</tr>
<tr>
<td></td>
<td>Total of the number of local methods and methods called by local methods in the class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCOM</td>
<td>Lack of cohesion of methods</td>
<td>0.050</td>
<td>-0.048</td>
</tr>
<tr>
<td></td>
<td>Number of disjoint sets of local methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAC</td>
<td>Data abstraction coupling</td>
<td>0.083</td>
<td>-0.191</td>
</tr>
<tr>
<td></td>
<td>Number of abstract data types defined in class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WMC</td>
<td>Weighted method per class</td>
<td>0.425**</td>
<td>0.084</td>
</tr>
<tr>
<td></td>
<td>Sum of McCabe's cyclomatic complexity of all local methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOM</td>
<td>Number of methods</td>
<td>0.142</td>
<td>0.049</td>
</tr>
<tr>
<td></td>
<td>Number of local methods in the class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIZE2</td>
<td>Number of properties</td>
<td>0.149</td>
<td>-0.001</td>
</tr>
<tr>
<td></td>
<td>Total number of attributes and the number of local methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIZE1</td>
<td>Lines of code</td>
<td>0.635**</td>
<td>0.621**</td>
</tr>
<tr>
<td></td>
<td>Number of semicolons in the class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANGE</td>
<td>Number of lines changed in the class</td>
<td>**</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Insertion or deletion is counted as 1, change of the contents is counted as 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

** Correlation is significant at the 0.01 level (1-tailed)

4.1.2. Presentation of inputs and outputs by linguistic terms

Each input space is divided into 2K+1 sections. In Fig.5, the spaces of 3 inputs ($x_1$, $x_2$, and $x_3$) and 1 output ($z_1$) are divided by FCM algorithm. The clustering process stop when the maximum number of iterations reach or when the objective function improvement between two consecutive iterations is less than the minimum amount of improvement specified.

The number of input space sections should be compatible with the number of “quantifiers” that a human being can handle. This number should not exceed the limit of 3-7 distinct terms. Here, 3 is specified as the number of cluster. And then each sample can be presented by the linguistic terms 1, 2 or 3. The process of clustering is done with the help of MATLAB®-Fuzzy Logic Tool Box(genfis), and
4.1.3. Development of membership functions

For node $N_{11}$, the input and output membership functions are derived from the samples in dataset QUES. Table 2 is parameters of membership functions after all 71 sample are inputted as training data, and $\mu$ is the mean or expectation of the distribution (and also its median and mode) and $\delta$ is its standard deviation.

<table>
<thead>
<tr>
<th>variable</th>
<th>cluster 1</th>
<th>cluster 2</th>
<th>cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mu$</td>
<td>$\delta$</td>
<td>$\mu$</td>
</tr>
<tr>
<td>MPC</td>
<td>15.97</td>
<td>13.7</td>
<td>18.82</td>
</tr>
<tr>
<td>RFC</td>
<td>37.64</td>
<td>21.45</td>
<td>65.33</td>
</tr>
<tr>
<td>SIZE1</td>
<td>179.2</td>
<td>93.11</td>
<td>346.5</td>
</tr>
<tr>
<td>$z_1$</td>
<td>46.14</td>
<td>47.61</td>
<td>80.34</td>
</tr>
</tbody>
</table>

4.1.4. Generation of rule base

The rule base of $N_{11}$ is obtained from the pairs of desired input-output data of QUES. The number of clusters for each variable is 3. For example, the training data of $x_1$ is clustered as $C_{11}, C_{21}$ and $C_{31}$.

4.1.5. Derivation of hierarchical structure and rules

The three inputs of SET2 are organized into hierarchical structure, as shown in Fig.5. The structure is not easy to make sense because $z_2$ and $z_3$ are with exact connotation. Here, $z_2$ is the subjective complexity appraisal, and $z_3$ is subjective supporting environment appraisal. As shown in Table 3, the rule bases of $N_{21}$, $N_{22}$ and $N_{13}$ are presented by Boolean matrices, which are derived by the knowledge of experts.

4.1.6. Derivation of a crisp value for maintainability

The single rule base RB of the whole SMP model is shown by topological expression in Eq. (5).

$$[\((N_{11} + N_{21} + I_{31}) \ast (I_{12} + N_{22}) \ast N_{13}\)](x_1, \ldots, x_6|y)$$  \hspace{1cm} (5)

Now, the FN model of SMP is a kind of FS model with 6 inputs, one output (maintainability) and rule base RB with 81 rules.
4.2. Simulation and results

Given the case study of the maintainability prediction problem, a set of experiments using the SMP models are conducted. The dataset QUES is used as training set and validation set. The training datasets are chosen from the 71 samples of QUES and then the performance are calculated.

4.2.1. Performance indicators

As opposed to most existing approaches where the focus is to improve accuracy, the focus of the FN approach is to maintain both accuracy and transparency by means of the modular rule bases that reflect the subsystems of the modeled system. When SFS and FN are used for modeling, the quality of the associated models can be quantified using performance indicators. In particular, two model performance indicators are introduced further below. They are called Accuracy Index (AI) and Transparency Index (TI).\(^{16}\) The first performance indicator AI reflects the accuracy of the model by means of Mean Magnitude of Relative (MMRE), as shown by Eq. (6).

MMRE measures the average relative discrepancy. It is equivalent to the average error relative to the actual effort in the prediction. In our study we express MMRE as actual values however in some studies it is expressed in %. MMRE has been regarded as a versatile assessment criterion and has number of advantages such as it can be used to make comparisons across datasets and all kinds of prediction model types and it is independent of measuring unit and scale independent.\(^{46-48}\)

\[
MMRE = \sum_{i=1}^{n} \frac{|\text{prediction} - \text{actual}|}{\text{actual}} / n
\]  \hspace{1cm} (6)
Where \( n \) is the number of data pairs, \( \text{actual} \) is the value of \( \text{CHANGE} \), and \( \text{prediction} \) is the FN output for each data pair. A lower AI implies better accuracy.

The second performance indicator TI reflects the transparency of the model by means of the extent of its opaqueness from the inside, as shown by Eq.(7)

\[
TI = \frac{(p + q)}{(n + m)}
\]  

(7)

Where \( p \) is the overall number of inputs, \( q \) is the number of outputs, \( n \) is the number of non-identity nodes and \( m \) is the number of non-identity connections. A lower TI implies better transparency.

4.2.2. Experimental results

In this subsection, the experimental results is shown which are obtained by applying the FN based SMP model. The results are compared with SFS model, which is with 3 inputs (MPC, RFC and SIZE1) and 1 output (CHANGE). The 71 samples of dataset QUES is utilized to achieve AI results of both FN model and SFS model. And TI is achieved based on Eq.(7). The results of AI and TI have been shown in Table 4, in which No.3 means \( N_{11} \) with 3 inputs.

The simulation results for the FN and the SFS are shown in Figs. 7, where the original data and the simulation results are presented together. And the FN model and the SFS model are with the same \( N_{11} \), which is with three inputs and one output.

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Accuracy</th>
<th>Transparency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FN</td>
<td>SFS</td>
</tr>
<tr>
<td>No.2</td>
<td>0.513</td>
<td>0.576</td>
</tr>
<tr>
<td>No.3</td>
<td>0.443</td>
<td>0.539</td>
</tr>
<tr>
<td>No.4</td>
<td>0.440</td>
<td>0.494</td>
</tr>
<tr>
<td>No.5</td>
<td>0.332</td>
<td>0.449</td>
</tr>
<tr>
<td>No.6</td>
<td>0.309</td>
<td>0.390</td>
</tr>
</tbody>
</table>

The results shows that, in terms of accuracy, the No.3 FN model(0.443) is superior to the SFS model(0.539); and in terms of transparency, the No.3 FN model(1) is significantly higher than the SFS model(4). This means that both accuracy and transparency are higher comparing the FN model with the SFS model. The possibility to improve the accuracy of FN model by adjusting the number of SET1 inputs, is discussed in the next subsection.
4.3. Discussions

As shown in Fig. 5, the FN based model is a basic model with 3 SET1 inputs, 3 SET2 inputs, and 3 linguistic terms for each input. In this subsection, we discuss the influence of different number of SET1 inputs. The SET2 remains unchanged.

In order to explore the relation between the accuracy and the number of SET1 inputs, 4 models are set up with 2, 4, 5 and 6 inputs of SET1 respectively.

Firstly, the top 6 Pearson’s Correlation Coefficient metrics are chosen. They are SIZE1, MPC, RFC, WMC, NOM and DAC, which are ranked from largest to smallest coefficient.

Secondly, 4 SFS models with the metrics above are set up, and they have 2, 4, 5
and 6 inputs, respectively. Correspondingly, there are 4 FN models with the same $N_{11}$ node to SFS models.

At last, The 71 samples of dataset QUES are utilized to simulate the maintainability. As shown in Table 4, the AI results are compared between FN models and SFS models. In the table, No.2 means $N_{11}$ with 2 inputs, No.4 means $N_{11}$ with 4 inputs, etc.

The simulation results for the FN and the SFS models are shown in Figs. 6-10, where the original data and the simulation results are presented together. In each figure, the FN model and the SFS model are with the same $N_{11}$, which is with 2, 4, 5 or 6 inputs separately.
The comparative results of accuracy and transparency for different models are shown in Table 4. It shows that the accuracy of both the FN and SFS models increases gradually with the number of SET1 inputs. Interestingly, each FN model shows significantly higher accuracy than SFS model, ranging from 11% to 26%. It shows that the accuracy of FN model is superior to SFS model significantly.

As far as transparency is concerned, the FN model is also significantly better to SFS model, no matter how many inputs the SET1 have. And, the transparency decreases with increasing the number of SET1 inputs for both FN and SFS models.

Over all, compared to SFS model, the average accuracy of the FN models increases 17.3%, and the average transparency increases 76.3%.

Therefore, the main advantage of our FN model is that it offers higher transparency than the other models which are more or less black boxes. And it is worth noting that, the differences in accuracy between our models and others’ are also significant.

Based on the discussion above, we can draw the conclusion that FN model is superior than SFS model in terms of both accuracy and transparency. Based on FN, the SMP framework are able to estimate maintainability.

5. Threats to validity

Data scarcity poses a major threat to the validity of the results of the framework proposed in this study. Therefore, we cannot claim that any of the inputs we selected is generic enough to be used in other models. However, data scarcity is a common problem for software engineering research. More calibration from the specific practitioner environment would be needed. Future effort will try to make contact with some potential software houses to share the maintenance data during their software systems’ updating. It is worth emphasizing, however, that the framework presented
in this paper can be used by the practitioners to develop their own models using their maintenance data and repositories.

6. Conclusion and future work

In this paper, it is the first time FN is utilized for SMP. One of the main motivations for using FN in SMP research is that transparency should be taken as an important aspect of SMP models in order to understand and reuse the existing models. As noted earlier, FN based SMP model has been characterized by a white-box nature whereby the software metrics are mapped to the CHANGE by means of connections. It means the whole implied information in the model has been shown. Based on FN, the SMP models are with more understandability and reusability than the earlier models, at the same time, with more accuracy.

Furthermore, FN model is a special type of SFS, by which the numerical data from metrics and linguistic appraisals can be combined into the SMP model at the same time. Although this strategy is not perfect, but it tended to catch more subjective appraisals which have shown their powerful efficacy in other software engineering models such as CMM. Only 3 subjective appraisal metrics are chosen as example in this paper, but the results indicate that finding a qualitative rule is possible. The authors do not claim that the rules between the 3 subjective appraisal metrics are the most accurate ones. The authors also do not mean to suggest that the FN based models are superior than the earlier models based on other methods in terms of performances other than accuracy (MMRE) and transparency. This study have suggest that FN based models improves the transparency and accuracy significantly. Notwithstanding its limitation, the accuracy can be improved if more accurate subjective appraisals and more linguistic terms in SMP models are considered. Despite its preliminary character, this study clearly indicate the SMP framework is effective to predict software maintainability.

Ongoing work about this framework involves the use of changing the number of inputs and linguistic terms to explore the influence in terms of accuracy. Another perspective would be to extend the proposed work to provide “estimation” as a service using the cloud concept to replace or enforce the subjective appraisals from experts.

Acknowledgments

The first author would like to thank the School of Computing at University of Portsmouth for offering facilities in carrying out this research. This work has been partially supported by China Scholarship Council (CSC) under grant No. 201307820040. The thanks should be given to PhD Han Liu and Abdul Yaakob’s contributions to the early version, to PhD Ting Yang and Qing Liu’s advices to the latest version of this paper. The thanks should also be given to anonymous reviewers for their valuable and constructive comments that help us to improve the quality of our paper.
References


