International Journal of Engineering Research and General Science Volume 2, Issue 4, June-July, 2014
ISSN 2091-2730

Study of Different Risk Management Model and Risk Knowledge acquisition with WEKA

Kiranpreet Kaur[1], Amandeep Kaur[1], Rupinder Kaur[1]

[1]Department of Computer Science and Engineering, Guru NanakDev University, Amritsar (Pb)

E-mail- sohalkirankaur@gmail.com

ABSTRACT

Software risks can be defined as uncertainty and loss in project process. Software risk management consists of risk identification, estimation, refinement, mitigation, monitoring and maintenance steps. In this paper,the main focus is on different risk management model and the importance of automated tools in risk managementt. With the automated risk management tool, the prediction of project problem effects that can cause loss in software project in terms of their values on risk factors and rank the risk factors to observe how they can give detail about project problem effects separately. For these purpose five classification methods for prediction of problem impact and two filter feature selection methods for ranking importance of risk factors are used in this study.

Keywords— Software Risk Management Model, Multi-characters Of Risk ,WEKA tool, Risk Ranking; Risk Impact Prediction

1. INTRODUCTION

In real world, success rates of software projects are lower than expected. Software risks that occur during the software development life cycle are one of the most important reasons for this low success rates. Risk is a problem that could cause loss or threaten the success of project, but which hasn't happened yet. These potential problems might have a contrary impact on the cost, schedule, or technical success of the project, the quality of the software products, or project team collaboration. Software risk management contains preventive key steps before start of new software projects to increase success rates of software projects. These preventive key steps specify software risks, impact of these risk factors and they aim to dissipate uncertain software issues. Uncertainty can be related with time, budget, labor or any other risk factors that can appear during the software project development life cycle. Therefore risk management steps should be applied for the software project.

Risk management has the objective to reduce the harm due to risks. As with any other management, risk management employs strategies and plans to meet the objectives. Risk management benefits group under two categories: direct and indirect benefits. Direct (Primary) benefits deal with major risk, people, product and cost. Indirect (Secondary) benefits deal with optimization, pragmatic decision making, better process management and alternative approaches. The main objective of risk management is to prevent and control risks before they become corruptive so risk mitigation, monitoring and maintenance steps are applied during the risk management process. [1]

1.1Several classical mechanisms of software risk management model

A. Barry Boehm theory

80 years of the 20th century, Boehm introduced the concept of risk management software industry, Boehm software project risk management process will be divided into two basic steps: risk assessment and risk control. The first step risk assessment, including risk identification, risk analysis and risk prioritization; that is first proposed a risk list,the list of the risk assessment of the probability and impact to determine the level of risk that take into account the priority of the risk, the risk list is the basis of risk control; when determining the priority of risk factors out, the second step is risk control, including risk management plans, risk and risk control to resolve. This step, we must first develop a response plan for each major risks and risk mitigation in accordance with the practical implementation of the program's activities,and in the process to be monitored.

Boehm states the risk probability and consequences of risk occurrence attributed to two parts of "risk exposure". [2]

Boehm noticed that the most common IT risks are:
• project team members are poorly trained,
• temporary planning and project budgets are not realistic,
• wrong product features are developed,
• interfaces are not user oriented,
• testing in real life situation fails.

Not all identified risks should be treated the same. Some identified risks are more likely to occur, and some, if realized, would have a bigger impact. Risk analysis and management depends on the types of risks being considered. Within the context of the technological and business perspectives, there can be distinguished three main elements of software risk: technical, schedule/scope, cost

1. Technical risks are associated with the performance of the software product, which includes functionality, quality, reliability and timeliness issues. Even if there are no mid project changes in scope, unforeseen technical complications can also turn the project upside down. Project managers might know the technologies they are using in the project very well but still surprises are possible – this component has always been working fine but now when you integrate it with another component, it's a complete mess. The more experienced the technical people are, the lower the risk of unforeseen technical limitations is, but still this risk is always present.

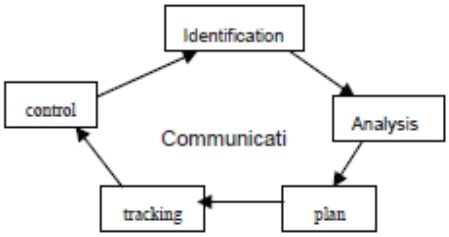
2. Schedule and scope risks are associated with the schedule and scope of the software product during development. Changes in scope are frequent in IT projects and to some extent they are quite logical – no matter how detailed your specification is, there are always suggestions that come after you have started the implementation. Often these suggestions demand radical changes and require change requests that can turn any schedule upside down. In order to address the holistic view of risks, software manager should view the risks from a different viewpoint and then get complete information. Also the scope can be affected by technical complications. If a given functionality can't be implemented because it is technically impossible, the easiest solution is to skip this functionality but when other components depend on it,
doing this isn't wise.

3. Cost risks are associated with the cost of the software product during software development, including its final delivery, which includes the following issues: budget, nonrecurring costs, recurring costs, fixed costs, variable costs,profit/loss margin, and realism.After the risks are identified they should be assessed by two dimensions - probability and impact. The project team will take these two dimensions and multiply them together to generate a risk score, so the risks can easily be ranked and ordered, allowing for the team and sponsors to dialog about how to respond to each risk. The Risk Score helps us determine a sense of priority amongst the risks. If, for example, the first risk has a score of $100K and the second of $160K, then the second risk represents a bigger threat to the project's baselines and has bigger priority.

After the risks are identified and assessed they should be mitigated with one of the response actions based on the risk type and priority. [4]

B. SEI's Continuous Risk Management (CRM) model

SEI (Software Engineering institution) as a software engineering and application of authority, based on years of software project management experience in the field, make CRM (continuous Risks Management) model . CRM model proposed in the software project life cycle should pay attention at all stages of risk identification and management, risk management, it is divided into five sections repeated cycle: identification, analysis, planning,tracking and control.



CONTINOUS RISK MANAGEMENT MODEL

SEI's CRM model has seven software risk management principles, namely: (1) global view; (2) an active strategy; (3)open communication environment; (4) Integrated management; (5) continuous process; (6) a unified perspective on the product; (7) team coordination and cooperation.[2]

In software risk management, information flow direction of information is from risk identification to risk control, and then into the risk identification, and continuously cycles and flows like this. The characteristics of this cycle will not stop until the end of the project, that would have been the project does not end risk management. First, the issue before the risk into risk assessment and then identify the impact, probability and time-consuming, risk classification and prioritization of risk; then the risk information to make decisions on the basis of action; and real-time monitoring risk indicators and risk mitigation actions; the last correction in the deviation of the risk mitigation plan. The core of risk this model is communication, which means that all parts of the project should strengthen the communication of risks, including among the various groups, such as between project phases and so on.

SEI risk management separately from the software risk identification, risk analysis, risk planning, risk tracking and risk management processes to cope with the various IDEFO (Integrated Computer-Aided Manufacturing Definition referred to as the DEFO, a standard process definition) data flow diagrams from two perspectives describes the software management process risk management; external view shows the process control, input, output and mechanism, internal view that the mechanisms used to process the input into output activities, and a clear description of the software risk management at all stages of the process of mutual effects of the interaction relationship. Software risk management process model through the control, input, output, and mechanism of the process described in the top control to decide when and how the input is a key process of change required, it must meet the entrance standards process, the output is the result of the process of change This result has already passed the process of export standard review mechanism to decide on the method used in the process.[4]

C. CMMI (Software Capability Maturity Model Integration) in the risk management process areas

In the CMM by the SEI CMMI is developed on the basis, and in the world to promote the implementation of software capability maturity assessment criteria and mainly used to guide the software development process improvement and software development capability assessment. Risk Management process area in CMMI Level III - has defined a critical stage in the process domain. The CMMI suggest three major steps in managing risks. These are prepare for risk management, identify and analyze risk and mitigate risk. It also suggest institutionalizing risk management (establish an organizational policy, planning, train people, manage configurations, relevant stakeholders, monitoring process, improvement info, higher level management etc

The core of the model is the risk library, and each activity to achieve the various targets are updated the risk library. Which activities to "develop and maintain risk management strategies" and the risk of database link is a two-way interaction, that is, work out the risk database by collecting data with the corresponding activities of the previous input.[2]

D. MSF Risk Management Model

MSF (Microsoft Solutions Framework) is the concept of risk management: risk management must be active, it is a formal system process, also risk should be continuous assessment, monitoring, management, until it is resolved or the issue is handled. The greatest feature of this model is the integration of learning activities, risk management, stressing the importance of learning experience from previous projects. Microsoft's research stated that an investment of merely 5% of the total budget into risk management could result in a probability of 50-70% to complete the project on time.[2]

E. IEEE risk management standards

It defines the process of software development life cycle risk management process for software companies in the software development projects also apply to individual risks emerging in the software development. It defines the risk management process is a continuous process, which systematically describe and manage the software development life cycle, including the following activities: planning and implementation of risk management, managing project risk list, risk analysis, monitoring risk, address risk, assessing risk management process.[2]

Institute of Risk Management (IRM), The Association of Insurance and Risk Managers (AIRMIC), and The National Forum for Risk Management in the Public Sector (ALARM) have a generic and valuable standard on risk management. Thereupon the standard contains these elements: risk definition, risk management, risk assessment, risk analysis, risk evaluation, risk reporting and communication, risk treatment, monitoring and review of the risk management process. [3]

F. Collaborative Risk Management

Collaborative risk identification

One of the first activities in a project is defining the project goals and description. This information is very important to understand the range and complexity of the project. Usually this process is developed by the professionals who are closed to the clients like, for

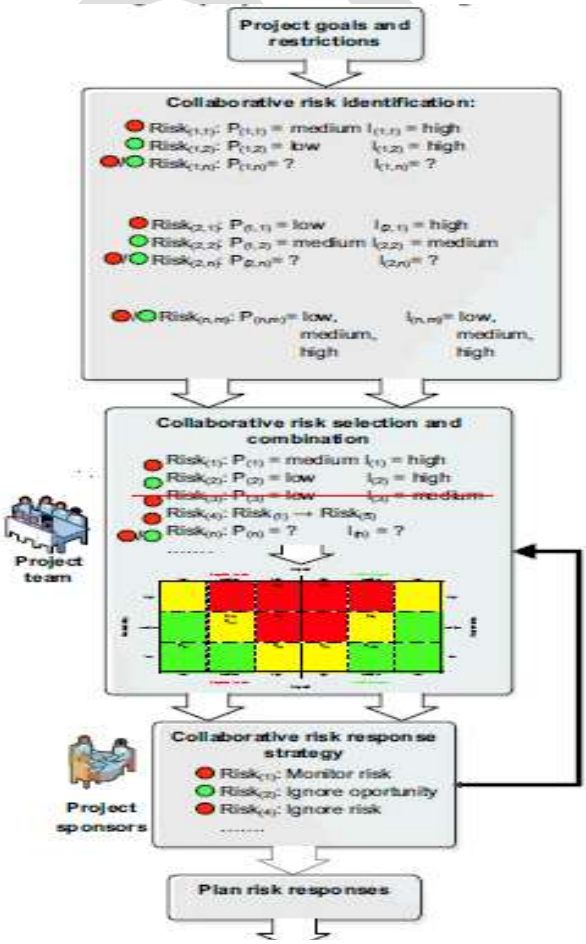
instance, project leader and consultants (or, in some cases, the entire project team).With the goals defined, team members, according to their skills and experience, can start identifying risks that can affect the project goals, including the risks which have positive and negative impact. For each identified risk they will categorize the risk impact and the probability in a scale: low, medium and high. In this process, project members perform the risk identification alone. This approach may be useful to determine the risk attitude and risk tolerance of each member or group area, which will allow identifying the organization global risk tolerance. This will also allow understanding future decisions and monitoring the risk tolerance evolution of the organisation.This stage ends with a first draft of the risk register of each project member, describing the probability and impact.

Collaborative risk selection and combination

After generating the preliminary risk records, the project leader analyzes all risks and may change, filter or merge some risks. Then him, with the project team, can analyze and identify the risk dependencies (identifying the risks that may be influenced by other risks). The probability and impact assessment of the risk will follow risk dependency theory, used to compute the final combined risk probability and impact. By this way, the project team will be able to identify and analyze the risks and evaluate if its combination can lead to disproportioned project failure.After the selection and combination, the project team will generate the risk probability matrix according to the scale (low, medium or high). This matrix gives a visual representation of the risks rank and helps risk prioritization. The output of this stage is the risk register with the filtered risks sort by priority.

Collaborative risk response strategy

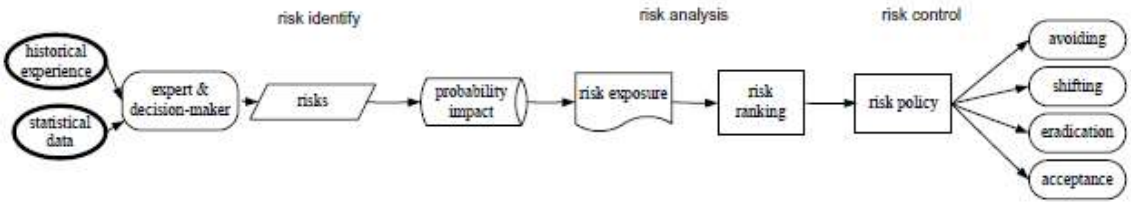
Regarding the organization's risk tolerance and appetite, the project sponsors may analyze and decide what risk or opportunities they want to explore or ignore. Also they can add new risks, delete or combine the existing ones, which may require new risk analysis by the project team. According to the risk matrix, project sponsors may want to monitor risk/opportunity, reduce the impact of the risk by taking some previous actions or enhance the probability/impact of the opportunities. With the project sponsors decisions about the identified risks, it would be possible to analyze some risk relevant issues. The decisions in this stage will guide the rest of the organization in terms of RM activities.[6]



Collaborative Risk Management

1.2 MULT-CHARACTERS OF RISKS

Risks are challenges that can have negative influences on a project unless they are handled properly. The efficiency of risk management depends upon the cognition to risks. In this section, we cognize characters of software risks, including multi-stages, multi-roles, multi-situations, uncertainty, multi-methods, multi-dimensions, multi-attributes and multiobjects. We call these characters as multi-characters of the software risks.



Flow Of Risk Management

Multi-stages

According to the software life cycle, software risks may exist or derive from different stages, i.e., bidding stage, requirement analysis stage, source-code writing stage, product delivery stage and maintenance stage. Because software risks exist during the software life cycle, risk management exists during the software life cycle too. Potential key risks should especially be identified and prohibited in time, which averts more potential losses. Less losses mean more profits. It is necessary for managers to attach importance to risk management during the
development processes and deal with the risks properly.

Multi-roles

Actually, roles in a software project include varied roles relative to software risks from the bidding to the delivery and maintenance of the software product. In the bidding stage, roles include tender, bidder, and supervisor. In the project approval stage, roles may include investor, developer, and uncertain market with risks. In the normal development stage, the development team may be private, joint-ventured or transnational enterprises, and then roles include mainly investors (stockholders), managers, developers, and customers (market).In the delivery and maintenance stages, roles include investors, the market branch, the development branch, maintainers, customers, etc. Different roles may bring different risks; different risks should be dealt by different people.

Multi-situations

Different development teams have different development models and different management models. Development environment are varied. For example, there are varied development teams or contractors, such as private enterprise, joint-ventured or foreign-funded enterprises.
According to their practical situation, they may adopt development models such as waterfall, spiral, prototype, or different development methods such as structured programming, object-oriented design. Different environment need different kinds of management to staff members, enterprise impression, supervision, etc. Different kinds of development teams may have different kinds of risks with different development models and management models.Risks exist in different domains, such as in flood risk, grassland fire, medical science, geo-field.

Uncertainty

Risk may occur, or may not occur. Risk occurs with different probabilities at different time and in different environment. Risks are uncertain. If managers deal with risks correctly, risks may be prevented; if managers do not attach attention to risks or do not deal with risks incorrectly, the risks may bring losses (or bring fewer profits sometimes). Managers should control important risks properly and prevent their occurrence or reduce their adverse impact during the risk control stage.

Multi-methods

There are many identifying method, ranking methods, such as Delphi and AHP methods, and ranking methods, such as risk exposure or risk matrices.

Multi-dimensions

Software risks are normally identified into different dimensions (categories). For example, software risks have six dimensions: user, requirement, project complexity, planning & control, team and organization environment; or three dimensions: project size,

technological experience and project structure; five dimensions: technological newness, application size, expertise, application complexity and organizational environment.
During the risk identification phase, people can identify all the possible risks into a list, or sort these risks into several dimensions according to their experience and comprehension to the project and respective risks.

Multi-attributes

Risk management uses probability and loss to rank risks.Probability of a risk is sometimes referred as "occurrence probability", "frequency" or "likelihood". Impact of a risk is sometimes referred as "magnitude", "loss", "severity", etc. Changing the names does not affect the logic of risk assessment. During the risk identification process, decision-makers identify risks with large losses or great probabilities. During the risk assessment process, decision-makers evaluate risks according to attributes of the risks or combination of the attributes. For example, risk exposure is the product of probability and loss of a risk, and the exposure value can be used to rank risks.

Multi-objects

There are many risks in software development. The target of risk management is to deal with most risks, or all the major risks with limited project resources. Each risk is an object to management. We say there are multi-objects for the risk management. There may be a lot of risks in a project. Managers cannot deal with all the risks identically for limited human and material resources. It is necessary to assess risks and get most important risks to deal with first.
Since there are a lot of methods, frames or ideas to identify, evaluate, control risks, managers or decision makers should choose the most suitable method forthemselves

Risk Identification is an iterative process that seeks to identify risks that may affect the project and documenting their characteristics. Currently there are different techniques to make the identification of risks such as: brainstorming, Delphi, interviews, SWOT analysis, checklist, cause-effect diagram, flowchart, diagram of influence. The output of this work would be the Risk register. [5]

3. Use of Automated Risk Management Tools

In order to offer high-quality software products to market in time and under market requirements, it is important to find computer-based tools with high accuracy probability to help managers to make decision. Software risk analysis and management can be partially transferred into data analysis or data mining. Automated tools are designed to assist project managers in planning and setting up projects, assigning resources to tasks, tracking progress, managing budgets, requirements, changes and risks as well as analyzing workloads.

Risk analysis and management are usually based on information collected from traditional knowledge, or analogy to well-known cases, common sense assessment, results of experiments or tests, reviewing of inadvertent exposure. The first thing for the automated tools is to collect historical data to build up a database. Once the database exists, it will process the data and mine some useful information to help manager to analyze risks and make decisions. There are lots of methods in Machine Learning study. For example, clustering skills are used to assign risk label to different risks. In each cluster, risks may have similar attributes. Association rule method is used to analyze each cluster to find the relationship of risks and risks factors. Some other artificial intelligence methods (9K-near neighbor approach, ID3 decision tree, Neural Network, etc) are used to build risk assessment models and to predict risks of software development. In the market, there are many popular software for decision making that is also applicable for risk management in software risks analysis.[2]

According to Hu and Huang, they randomly divide their software risk dataset into two subsets, 100 samples for training, and 20 for testing. They start by predicting the risks with standard neural network. Then predictions were made using standard multilayer neural network, support vector machines, the combination of genetic algorithm and neural network. They compared results of three classifiers. The standard neural networks can predict the outcome of software projects with 70% in accuracy. SVM on the other hand achieved higher accuracy of 80%. The highest correct prediction results are obtained from the combination of genetic algorithm and neural network as 85% [7].

According to Amanjot Singh Klair and Raminder Preet Kaur, SVM and kNN based approach could serve as an economical, automatic tool to generate ranking of software by formulating the relationship based on its training. They have gone through the survey of the SVM and kNN models for various applications and they conclude that most of the software quality evaluation problems the performance of SVM model is better than the kNN approach [8].

Hu and Zhang published an article about an intelligent model for software project risk prediction. They compared ANN and SVM methods. For ANN method, the probabilities for two categories of prediction errors are 10% and 15% respectively, and for SVM method, 5% and 10% respectively, which shows that the proposed SVM-based risk prediction model achieves better performance [9].

Tang and Wang published an article about software project risk assessment model based on fuzzy theory. They created a model based on fuzzy theory about software project risk assessment. That model can measure a combination of impact risk and it resolved the uncertainty. They calculate quantitative data of risk-equivalent and semantic distance between fuzzy numbers. They combined demand, technology and software performance risk with progress, costs and software quality [10].

3.1 WEKA –data mining tool

Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at theUniversity of Waikato, New Zealand. Weka is free software available under the GNU General Public License.It is a collection of machine learning algorithms for data mining tasks. The algorithms are applied directly to a dataset. WEKA implements algorithms for data preprocessing,classification, regression, clustering and association rules; It also includes visualization tools. The new machine learning schemes can also be developed with this package.Weka supports several standard data mining tasks, more specifically,data preprocessing, clustering, classification,regression,visualization,and feature selection.

Main features of WEKA include:

- 49 data preprocessing tools
- 76 classification/regression algorithms
- 8 clustering algorithms
- 15 attribute/subset evaluator + 10 search algorithm for feature selection
- 3 algorithms for finding association rules
- 3 graphical user interfaces :
  The Explorer
  The Experimentor
  The Knowlegde flow

Fig. 1: Weka GUI

The data file normally used by Weka is in ARFF file format, which consists of special tags to indicate different things in the data file (foremost: attribute names, attribute types, attribute values and the data). The main interface in Weka is the Explorer. It has a set of panels, each of which can be used to perform a certain task. Once a dataset has been loaded, one of the other panels in the Explorer can be used to perform further analysis. [11]

3.1 RESULTS AND DISCUSSION

The first aim is to measure the importance of risk factors with using 384 problems and six risk factors. Correlation between "severity" and each six risk factors are calculated separately by Chi-Squared Statistics and Information Gain approaches to find out the importance level of the risk factors.

The second aim is to predict impact of the problem with using the model that is formed by Turkcell ICT data set. Each problem has different values on risk factors so we can estimate the impact of the problem with building a risk model. 384 problems are used as tuple and 6 risk factors are used as feature in our study. Severity value, which is Low, Medium or High, is used as a class label.

384 problems have class label so we used classification methods in our study. Support Vector Machines (SVMs), Naive Bayes, Decision Tree (J48), k-Nearest Neighbor (kNN) and Multilayer Perceptron Neural Networks (MLPs) classifiers are used in this work.

Importance Ranking of Risk Factors

In this, the importance ranking of risk factors is obtained by feature selection methods that are information gain and chi square statistics by using WEKA tool. Importance of risk factors emphasis the most significant risk factors that determine impact of problems. Problem severity can also be predicted with using classifiers in classification phase. We obtain correlation values of risk factor and severity of problem. Ranking order of risk factors according to impact power is also given in table below . "Regulation Effect" is the most distinctive and important risk factor to determine problem severity. IG and X2 approaches get the same results for the two most distinctive risk factors. This shows that problem values on "Regulation Effect" and "Financial Effect" are more distinctive than other risk factors to predict the problem severity. With the same logic "Employee Effect" and "Brand Effect" are less distinctive than other risk factors to predict the problem severity. IG and X2 approaches get the same results for the two lest distinctive risk factors. To sum up, if a problem in the project affects regulations in the company or financial values of the company, this makes severity of this problem high. If our data set consisted of hundreds of risk factors, determination of ranking of risk factors would enable to reduce unnecessary risks for risk evaluation phase

Correlation values of risk factors

Ranking Order of Risk Factors According to Impact Power	Impact Values of Risk Factors (with using Information Gain)
1	REGULATION EFFECT
2	FINANCIAL EFFECT
3	SUBSCRIBERS EFFECT
4	SOX EFFECT
5	EMPLOYEE EFFECT
6	BRAND EFFECT
Ranking Order of Risk Factors According to Impact Power	**Impact Values of Risk Factors (with using Chi-Squared Statistics)**
1	REGULATION EFFECT
2	FINANCIAL EFFECT
3	SOX EFFECT
4	SUBSCRIBERS EFFECT
5	EMPLOYEE EFFECT
6	BRAND EFFECT

Problem Impact Prediction

Turkcell data set supply problem severity values so prediction of problem impact become a classification problem. Data set has six features (six risk factors) and each problem has a class label (severity value) so forming a training model then test this model with same data gives an idea about prediction of problem impact.
10 fold cross validation evaluation technique is used to get accuracy values in classification phase. 10 fold cross validation evaluation technique splits data set into ten parts randomly then it uses nine part to build training model and one part is used as test data. It is repeated ten times to get all classification test results. Classification performances of all five classifiers are measured by using Precision, Recall and F-measure values.

Classification Performance Values Of Classifiers

	NB	SVMs	J48	MLPs	kNN
Precision	0.87	0.955	0.898	0.972	0.87
Recall	0.917	0.966	0.911	0.979	0.917
F Measure	0.885	0.96	0.904	0.975	0.885
Kappa Statistic	0.1795	0.7699	0.3907	0.875	0.1795
# of Correctly Classified Instances	352	371	350	376	352

The highest F-measure value, 97.5 percent, is obtained from MLPs classifier. MLPs also give highest Kappa statistic than other classifiers. It classified 376 problem severity values correctly. The result of SVMs follows results of MLPs. The second highest F-measure value is obtained from SVMs classifier and also second highest Kappa statistic is taken from this classifier. NB and kNN give the lowest F-measure values and Kappa statistic values. There is an important point that number of correctly classified instances by J48 classifier is less than number of correctly classified instances by kNN and NB classifiers but Kappa statistic and F-measure value of J48 is higher than Kappa statistic and F-measure values of kNN and NB. It proves that number of correctly classified instances is not capable for evaluating classification performance. F-measure and Kappa statistic are more reliable for non-homogenous data sets in classification.

4. CONCLUSION:

It is concluded that Software risks that occur during the software development life cycle are one of the most important reasons for this low success rates so it is important to deal with the risk before they become corruptive. Hence Software risk management contains preventive key steps before start of new software projects to increase success rates of software projects. These preventive key steps specify software risks, impact of these risk factors and they aim to dissipate uncertain software issues. In order to offer high-quality software products to market in time and under market requirements, it is important to find computer-based tools with high accuracy probability to help managers to make decision. The proposed risk management tools and methods help the project managers deal with risk management programs in a most effective and efficient manner.

Acknowledgement

I wish to thanks who directly and indirectly contribute in paper, First and foremost, I would like to thank Mrs.Amandeep kaur for his most support and encouragement. She kindly read my paper and offered valuable details and provide guidelines.Second,I would like to thanks all the authors whose paper i refer for their direct and indirect support to complete my work.

REFERENCES:

1. M. Özgür Cingiz, Ahmet Unudulmaz, Oya Kalıpsız ,Computer Engineering Department ,Yıldız Technical University, Prediction of Project Problem Effects on Software Risk Factors, 12th IEEE International Conference on Intelligent Software Methodologies, Tools and Techniques ,September 22-24, 2013.

2. Pu Tianyin, Development of software project risk management model review, IEEE,2011

3. IRM,A Risk Management Standard Published by AIRMIC, ALARM, 2002.

4. Software risk management, Sergey M. Avdoshin, Elena Y. Pesotskaya,IEEE, 2011

5. Yu Wang,Shun Fu ,A General Cognition to the Multi-characters of Software Risks, International Conference on Computational and Information Sciences,2011

6. Pedro Sá Silva, António Trigo, João Varajão, Collaborative Risk Management in Software Projects, Eighth International Conference on the Quality of Information and Communications Technology,2012

7. Y. Hu, J. Huang, J. Chen, M. Liu, K. Xie, "Software Project Risk Management Modeling with Neural Network and Support Vector Machine Approaches", International Conference on Natural Computation, 2007

8. A.S.Klair, R.P.Kaur, "Software Effort Estimation using SVM and kNN" International Conference on Computer Graphics, Simulation and Modeling, 2012, Pattaya (Thailand)

9. Y. Hu, X. Zhang, X. Sun, M. Liu, J. Du "An Intelligent Model for Software Project Risk Prediction", International Conference on Information Management, 2009

10. A.Tang,R.Wang, "Software Project Risk Assesment Model Based on Fuzzy Theory", International Conference on Computer and Communication Technologies in Agriculture Engineering, 2010

11. http://en.wikipedia.org/wiki/Weka_(machine_learning)