Editor in Chief Dr. Kouroush Jenab

International Journal of Engineering (IJE)

Book: 2008 Volume 1, Issue 1
Publishing Date: 30-06-2007
Proceedings
ISSN (Online): 1985-2312

IJE Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Contents

Volume 2, Issue 1, May/June 2007.

Pages

’Genetic Information System Development and Maintenance’ Model For Effective Software Maintenance and Reuse

Sanjeev Manchanda* smanchanda@thapar.edu
School of Mathematics and Computer Applications,
Thapar University,Patiala-147004 (INDIA)
**Corresponding author*

Mayank Dave mdave67@yahoo.com
Department of Computer Engg.,
National Instt. Of Technology, Kurukshetra, INDIA

S. B. Singh sbsingh69@yahoo.com
Department of Mathematics,
Punjabi University, Patiala, INDIA

Abstract

The aim of present research work is to develop an information system development process and a model for the development of new generation information systems. New age information systems are those Information systems that are capable of fulfilling the demand of highly dynamic information requirements derived from the competitive environments of the business organizations and support controlling the complexity involved in their maintenance and software reuse. Present research work analyzes the theoretical, financial, technical and practical problems related to the information system development, maintenance and software reuse, to propose an appropriate system development process and a model for the development as well as maintenance of information systems with maximum software reuse. Proposed system development process and model provide inherent support to the business organizations, in getting total control over information system development and maintenance maximum software reuse.

Keywords: Software Maintainability, Software Reuse, Software Configuration Management, System Development Life Cycle and Software Engineering.

1. INTRODUCTION

Global expansion of business organizations has created demand for the development of very complex and globally operating information systems. Present and future generation of information systems need to be capable of fulfilling such demands of these business organizations. Information system development and maintenance processes consume a lot of organizational resources like time, money and effort of their employees as well as system developers. Rising costs of information system development and maintenance, Rapid innovations in information technology, dynamism of users' information requirements, increasing sizes of the databases, global expansion of business organizations, exponentially increasing complexity of information systems, emerging needs of information for the organizations etc. has raised many questions about previously used system development practices, reusability and maintainability of information systems. Even after using advanced technology, investing sufficient funds and placing tremendous effort in system development, organizations face the requirement of changes in existing information systems. Frequently required changes and requirement of extra resources for realizing these changes increase anxiety among management members of organization. The reason of such anxiety is that management personnel of organizations do not understand the requirement of changes and maintenance as an essential part of information systems. For example, when new machinery is purchased, wear and tear, replacement of its certain parts, up gradation of technology and scrapping of machinery after certain period, are inherent requirements of the machinery, so are the information systems, these systems need to be up graded, maintained suitably. Information systems are such parts of business organizations, which take birth with the birth of organization, grow with the growth of organization, live step by step with organization and die with the death of organization (Section 3 includes detailed Discussion about this relationship). So, there is a need to adopt the change(s) as the part of an organization and its information systems. First aim of this paper is to change this viewpoint of management personnel about the requirement of changes in their information systems. Secondly this paper stress upon controlling the system development and maintenance processes by the organization's own employees for better results. Thirdly current paper helps in selection of technology that is highly adaptable and keeps pace with future. Fourthly this paper proposes a system development process and a model that helps implementing the changes frequently and maximizes the software reuse. Lastly paper concentrate upon the issues like cost, reliability, maintainability, quality and reusability with required experimental results. Next section includes literature review of related work, followed by discussion about the relationship between organizations and information systems, proposed information system development process/model for system development and maintenance, efficiency/effectiveness of proposed methodology and model, implementation, result comparisons, conclusion and last but not the least references.

2. LITERATURE REVIEW

In this section, we examine the historical developments related to the System development models/methodologies, Process modeling, Change management, Reliability, Goal orientation in modeling and use of Internet etc. System development models/methodologies, Process modeling, Change management, Reliability etc. are clearly related to the domain of this paper, whereas goal orientation is essential for all the sections of any organization. Goal orientation is substantial for guiding organizational efforts in unique direction and information system should possibly be guided towards the organizational goals. Internet helps in developing globally operating systems. So Internet developments need to be revealed here.

The documented collection of policies, processes and procedures used by a development team or organization to practice software engineering, is known as software development methodology (SDM) or system development life cycle (SDLC). Development of system development methodologies originated from the waterfall model (Royce [47]), in which development was supposed to proceed linearly through the phases of the requirement analysis, design, implementation, testing (validation), integration and maintenance. Researchers criticized Waterfall model for its excessive separation of different phases. In an attempt to overcome the shortcomings of the waterfall model many new software development approaches such as spiral model (Boehm [8]), iterative enhancement (Basili et al. [5]), rapid prototyping (Gomaa [25]), evolutionary prototyping and incremental development (Floyd [22]) had been suggested. In spiral Development process, a desired capability is identified, but the end-state requirements are not known at program initiation. Those requirements are refined through demonstration and risk management, there is continuous user feedback and each increment provides the user the best possible capability. The requirements for future increments depend on feedback from users and technology maturation, whereas the basic idea behind the iterative enhancement was to develop a software system incrementally, allowing the developer to take advantage of what was being learnt during the development of earlier, incremental, deliverable versions of the system. Software development approaches incorporating prototyping have gained respectability as they have proved to be able to dynamically respond to changes in user requirements, reduce the amount of rework required and help control the risk of incomplete requirements. Currently reuse model has achieved tremendous success in information system development (Frakes [24]). The aim of Component-based software development (CBSD) (Aoyama [2]) is to develop new software by widely reusing pre-fabricated software components. Many other researches (Kaushaar et al. [34], Boehm et al. [9], Gordon [27], Alavi [1], Naumann et al. [42], Tate et al. [49], Palvia et al. [43] etc.) contributed in development and growth of these methodologies.

Many researchers have contributed to business process development. A large number of process models were developed (Armenise et al. [3], Bandinelli et al. [4], Bubenko [11], Decker et al. [18], Jarzabek et al. [33], Jacobson et al. [32], Rumbaugh et al. [48] and Marca et al. [40]).

Software configuration (Change) management (SCM) is the discipline (Tichy [50]) that enables us to keep evolving software products under control and thus contributes to satisfy quality and delay constraints. The purpose of SCM is to manage change throughout the software development process (Bersoff et al. [6]). Change is a very natural and intrinsic aspect of software development process. SCM has been the focus of software engineering research and a great amount of research has been carried out on SCM. In the previous research, eight areas of functionality of SCM systems were found: version control, configuration support, team support, change control, build support, process control, status reporting, and audit control (Burrows et al. [12] and Dart [15, 16]). These functional areas mainly cover the management issues of software development. To provide these eight areas of functionality, different SCM systems use different models, such as the checkout/check in model, the composition model, the long transaction model, and the change set model (Feller [21]). In recent years, the focus of SCM research is on software process support in SCM systems (Estublier et al. [20], Leblang [38]), distributed configuration management systems (Hunt et al. [29] and Milewski [41]) and unified version models (Conradi et al. [14]) etc.

Goals are essence of management. Management by objectives (Drucker [19]) is one of the most important motivation factors for the success of any organization. Attempts have been made to incorporate goals into process modeling (Kueng et al. [37]), that suggested an informal approach in which goals provide a basis for process definition. Other model (Khomyakov et al. [35]) based on mathematical systems theory was proposed. This set of concepts extended (Bider et al. [7]) and used for defining a process pattern, allowing the design of generic processes that can be specialized for specific situations. The goals addressed by this approach are operational goals only, termed “functional goals”. Goals and soft-goals are applied for requirements elicitation in combination with scenarios (Rolland [45, 46]) and others contributions to relate goals with process models and its impact on strategic success.

The potential of the Internet to reach a large and growing body of customers, coupled with low communication costs, makes it a very attractive business medium to many organizations. Although there is significant interest in the use of the Internet for business purposes, studies articulating issues that can guide business managers in its use are lacking. Use of Internet and related issues are the hottest research areas. Different studies conducted and contributed (Brandtweiner et al. [10], Cho [13], Gonsalves et al. [26], Hamill [28], Weill [51] and Lee [39] etc.) to the aspects related to the use of Internet for business, marketing, performance and modeling of systems etc.

Therefore, a great amount of active research has been carried out for developing system development methodologies, business process development, change management, goal orientation and its organizational implementation in modeling and Internet usage. Still we are facing many problems in development and maintenance of information systems. This paper is an effort to identify those problems and finding practical solutions for them.

3. RELATIONSHIP BETWEEN ORGANIZATION AND INFORMATION SYSTEM

Information systems are cores of business organizations. Flow of Information in an organization is as if blood floats in veins of human body. Information system born with the birth of an organization, it grows as the organization grows and stays alive forever with the organization or up to the death of the organization. Growth of organization directly affects the growth of information systems; changes in organizations create need for changes in information systems as well, e.g. When a firm is originated, its operations are limited up to small geographical areas and a small number of people. Simultaneously, information system is also small and can be handled manually or with little automation by small number of people. Once a company starts expanding its operations from one city to many cities and from one country to many countries, information system is also expanded concurrently to handle the organizational information needs. Even after many developments, software development and maintenance are very tedious tasks. Many problems generated due to information, systems faced by the organizations can be listed as budget of information system, selection of technology, selection of system development companies, duration of system development project, change identification, vision for future changes, flexibility of information system to incorporate future changes, level of changes, maintenance terms of the systems, system improvement policies, control over system development and maintenance, Data security, Fulfillment of right kind of user needs, Connectivity of different users etc. Current paper is an effort to find the solution of many of the problems listed above. First of all, there is a need to change the view about requirement of change(s) in information systems. Business organizations view change requirements of information system as burden for their organization. Organizational members especially need to adopt change requirements of information systems as inherent necessity of their information system. Secondly vision of organizational members should be global and the initial selections of technology need to support global expansion of information systems. Nevertheless to remark that Internet based technologies are the future of information systems. All organizations today need to adopt online technologies that support information exchange worldwide. Thirdly the selection of system development model and processes should support frequent change requirements, software reuse and controlled maintainability of information systems. An information system survives with organization, changes occurring in organization need to change information systems. If information systems reflect the organization, then how information needs of an information system may be understood and implemented in a short span of system development project. There is a fundamental need to develop information systems continuously throughout the life of organization, so that required changes can be implemented effectively and system architecture must support such requirement. In following sections we propose a system development process and a model to develop as well as maintain the information systems continuously throughout the life of an organization.

4. GENETIC INFORMATION SYSTEM DEVELOPMENT AND MAINTENANCE PROCESS

System development and maintainability can be achieved efficiently, if it is under direct control of organization's management. Management of organization can control their own employees far better than the employees of other organizations. Team building of organization's employees and flexible system development process are the ingredients of proposed Genetic information system development and maintenance process as explained below.

4.1 Team Organization for System Development and Maintenance

First of all, there is a need to have a group of personnel (optimum number of people) within organization to analyze, develop, test, implement and maintain the information system for their own organization throughout the life of an organization. This group of people will continuously look forward for the information requirements of the management personnel continuously and build the system or incorporate the changes continuously, without affecting the earlier implemented system at large, but only affected parts will be required to be updated, implemented, documented and only concerned people will be informed about the changes. This group of people will not only look after the information requirements of the management personnel or the public concerned, but also look forward for the technological developments worldwide, so that their organization can be benefited through the involvement of latest technological developments. More people, expertise and organizations may be hired, contracted or outsourced to help the development, whenever it is necessary.

4.2 Development and Maintenance of the system

Secondly there is need to develop the system by iterating the following system development steps throughout the life of an organization continuously:

STEP I: Genetic Creation of Processes and Sub Processes-First of all there is a need to identify information requirements from scratch or from previously implemented major processes and sub processes of the organization and to program them to meet the requirements by formulating a library (L) of processes and sub processes i.e.

$$L = \{P_1, P_2, \ldots\ldots., P_n\} \qquad (1)$$

where $P_k = \{SP_{k1}, SP_{k2}, \ldots\ldots, SP_{kl}\}$, $\forall k = 1 \ to \ n \ , \ l \geq 1, \ n \ and \ l \ are \ \mathrm{var}iables.$

Number of processes and sub processes will be finite and almost different for all the processes, whereas maximum value of l and n for the number of sub processes and processes respectively will be dependent upon the decision of system analyst. Processes and sub processes will work as genes for the genetic development process. The goal of design (from a low-level perspective) is to minimize coupling and maximize cohesion (Kramer [36]), can only be achieved from the division of different information needs in independent categories of processes. A process is defined as a set of sub processes and will fulfill information requirements through execution of proper logic and will provide access to the authenticated database(s). Genetic development means the primary set or sequence of processes that lead to the formation and subsequent development of processes/sub processes. Development of processes and carried out at three different levels as follows:

(i) User Level: Information requirements are identified at this level, so system development process actually begins at this level. First of all, users' information requirements need to be identified. Once users' information requirements are identified, then further level can be sequentially explored. List of identified information requirements needs to be reviewed successively.

(ii) Database level: Logical designs of schema/subschema need to be developed and implemented according to the user requirements and the organizational considerations. Schema / Subschema need to be updated according to the initially defined/emerging/modified information requirements. Detailed design considerations need to be defined so that less frequent changes are required later. Technology used for System development needs to be adaptive for any kinds of changes from logical to physical memory levels.

(iii) Programming Level: Process development is completed by converting user requirements into programming code and connecting databases of significant concern by developing programs and logic to find practical solutions of user requirements/problems. Genetic development effort is constantly required to fulfill the user requirements from the databases through programming.

Library or Set of processes during all iterations of genetic creation of processes will require different processes to undergo fitness screening. Fitness screening of processes/sub processes need to be implemented during every review of processes/sub processes.

Function **f** for fitness of processes/sub processes (based on mutation, selection and reproduction etc.) (Darwin [17] and Forrest [23]) need to be applied on set L$^{(i)}$ to get L$^{(i+1)}$ i.e. Set of Processes after qualifying fitness criteria.

$$\mathbf{f}: L^{(i)} \rightarrow L^{(i+1)} \qquad (2)$$

Where $i \geq 0$ represents (i+1)th generation/iteration/review of process/sub process development and L$^{(i+1)}$ will become a set

$$L^{(i+1)} = \{P_1, P_2, ..., P_{n'}\}^{(i+1)} \qquad (3)$$

This is having n' number of processes with their sub processes after review in current generation. At the time when new system is developed from scratch library L will be developed by newly developed processes, whereas in next generations this library will undergo development and fitness processing by revising earlier processes or by including new processes. After screening of the processes, n is assigned the value of n' and procedure enters to next step.

STEP II: Implementation of Quality Parameters- Then each process and sub process needs to undergo quality screening based on parameters set for quality assurance and implementation as follows:

Function **q** Quality parameters need to be applied on the set L to get QL i.e. Quality Library.

$$\mathbf{q}: L \rightarrow QL \qquad (4)$$

Where QL will become a set $QL = \{QP_1, QP_2, ..., QP_n\}$ of Quality processes and sub processes. Parameters of quality may be based on quality standards, syntactical, logical, environmental and organizational factors etc. Quality parameters are usually dependant upon system analysts' vision about the quality and differ from system to system and place to place.

STEP III: Customization- Now Quality Library needs to be customized according to the needs of individual users as follows.

Function **c** of Customization needs to be applied on QL to give a set CQL.

$$\mathbf{c}: QL \rightarrow CQL \qquad (5)$$

Where CQL will become a set $CQL = \{CQP_1, CQP_2, ..., CQP_n\}$ of Customized Quality processes and sub processes. Same module may be customized differently for different users, e.g. finance related data may be more detailed for financial experts, whereas summarized for the marketing analyst. Customization of processes depend upon type of user, user's level in organizational hierarchy, authority etc.

STEP IV: Security- Now security criteria need to be applied on Customized Quality Processes for the secure access of the organizational database(s).

Function **s** of Security criteria need to be applied on CQL to generate a set CSQL.

$$\mathbf{s}: CQL \rightarrow SCQL \qquad (6)$$

Where SCQL will become a set $SCQL = \{SCQP_1, SCQP_2, ..., SCQP_n\}$ of Secure Customized Quality processes and sub processes, which may be assigned to the authorized users through their login accounts. Security requirements of different processes may be different, e.g. confidential organizational resources demand more secure access, rather public information resources.

STEP V: Total Quality Management (TQM): It is a philosophy toward continually improving your business and products (Ishikawa [31] and Hyde [30]). Information systems need to be improved continuously. Organizational team of system development personnel needs to search for better options for the organizational information system and then to implement them in the system. These options include the comprehensive search for better technology, better processes, enhancement of earlier processes, search for new information needs, other problems related to earlier implementation.

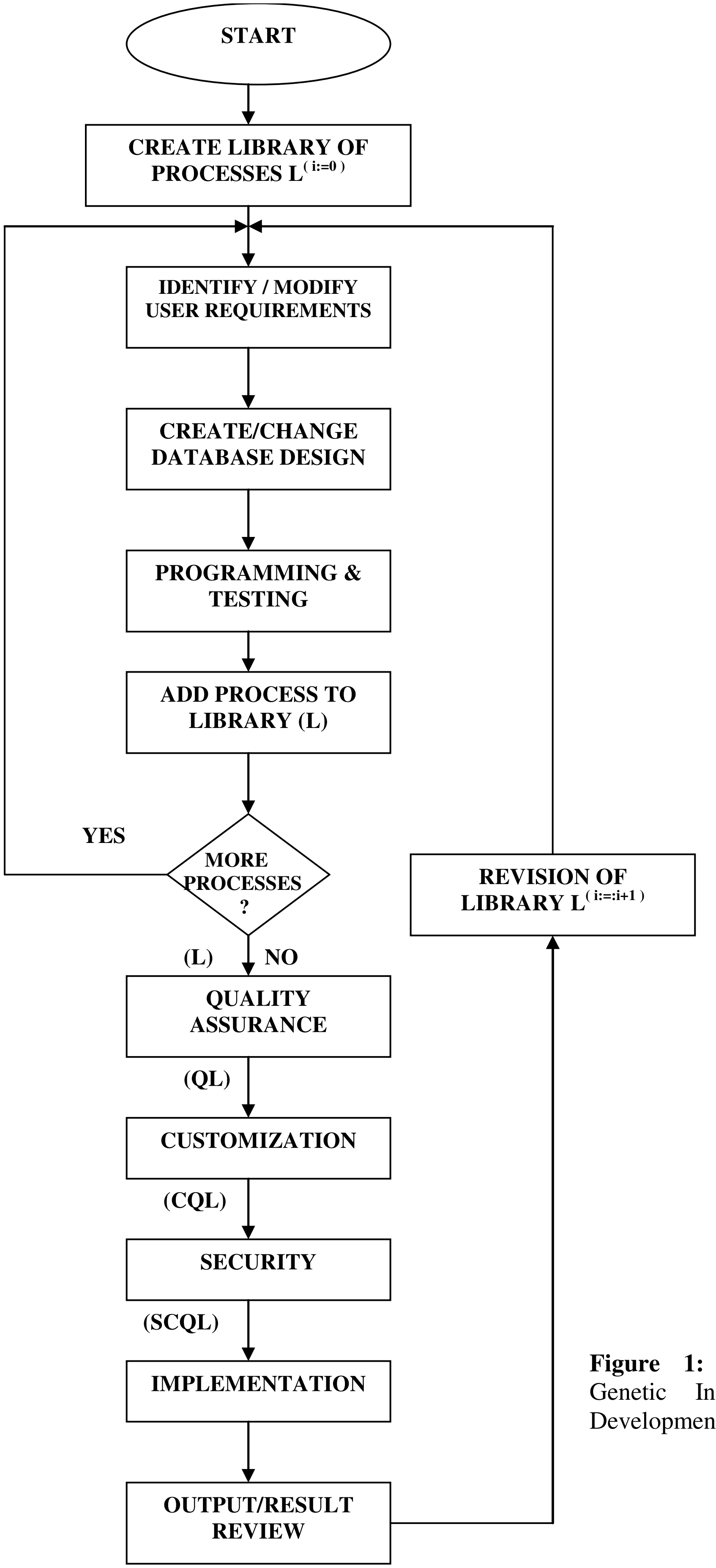

START
CREATE LIBRARY OF PROCESSES L(i:=0)
IDENTIFY / MODIFY USER REQUIREMENTS
CREATE/CHANGE DATABASE DESIGN
PROGRAMMING & TESTING
ADD PROCESS TO LIBRARY (L)
YES
MORE PROCESSES ?
(L)
NO
REVISION OF LIBRARY L(i:=:i+1)
QUALITY ASSURANCE
(QL)
CUSTOMIZATION
(CQL)
SECURITY
(SCQL)
IMPLEMENTATION
OUTPUT/RESULT REVIEW

Figure 1: Flow Chart For Genetic Information System Development And Maintenance.

All five steps need to be iterating continuously. Figure 1 illustrates the flowchart that includes the above discussed system development and maintenance procedure with a start and continuous development, but without stopping criteria.

Proposed system development will be a combined and continuous effort of organization personnel (users), system analysts, programmers and Database Administrator (DBA) to develop, implement and maintain information system and business processes. System analyst need to identify organizational structure, personnel, subordinates and their information requirements etc. continuously, DBA need to be involved simultaneously for authorizing required database access to different personnel. Management is involved for identifying their personalized information requirements and its implementation in customized way.

5. PROPOSED MODEL FOR SYSTEM DEVELOPMENT

Different users have different type of information requirements. Type of user, user's level in organizational hierarchy, his/her authority, technical ability of system to fulfil users' information requirements etc. are the factors that play an important role in differentiating different users and their information needs. Categorizing users based on their common information requirements is always a better option, e.g. grouping users on departmental level, common work level etc. Users with common information needs may be grouped together to fulfil their information needs, whereas their authority in organizational hierarchy may help to distinguish their access to the information resources, e.g. marketing department may be grouped together to fulfil their information needs related to marketing aspects. Nevertheless, higher-level managers will have higher as well as summarized access to the databases as compared to the low level executives, who usually have lesser but detailed access to the databases. Processing of information may be distinguished by different type of processing set-ups like one-dimensional data processing, multi-dimensional data processing and knowledge discovery in databases (i.e. data mining or vaguely defined information requirements) etc. Different sets of information requirements demand different kind of processing set-ups and different databases to be connected, e.g. marketing executives may require many databases to be accessed worldwide to get latest information and different type of processing set-ups, so that right kind of information is accessed. On the other hand, financial people require information related to fund flow, sales and expenditures only. Such information requirements, processing set-ups and database connectivity necessitate the development of processes/sub processes for the extraction of required information. Each process is developed to fulfil a homogeneous/related set of users' requirements. Domain of Individual process need to be defined very precisely, so that each process is connected to the databases concerned with its own domain and client/server side logic, security guidelines, quality standards etc. may be settled according to these homogeneous information requirements. Different processes (i.e. heterogeneous set of processes) are combined parallel to each other, with pre-defined accessibility. Implementation of whole system can be explained as a user account login/password authentication based access of a World Wide Web based information system. All management personnel is allocated their personal accounts, which is a page if we talk in terms of World Wide Web. Each user is authenticated through login for its access from the website of the organization. Each account page is a set of windows or drop down menus having some predefined templates for enquiries, access to the database(s) and information transfer (General and specific information) as shown in figure 2.

Programs developed by programmers, are maintained by DBA's library of processes and sub processes meant for the purpose. User may access, query, modify or update database according to the authority provided by DBA. Continually up gradation of library programs is required to fulfil the newly identified information requirements, facilities made available by innovations and new technology. Data warehouse is maintained at back end to collect the information from different sources scattered worldwide.

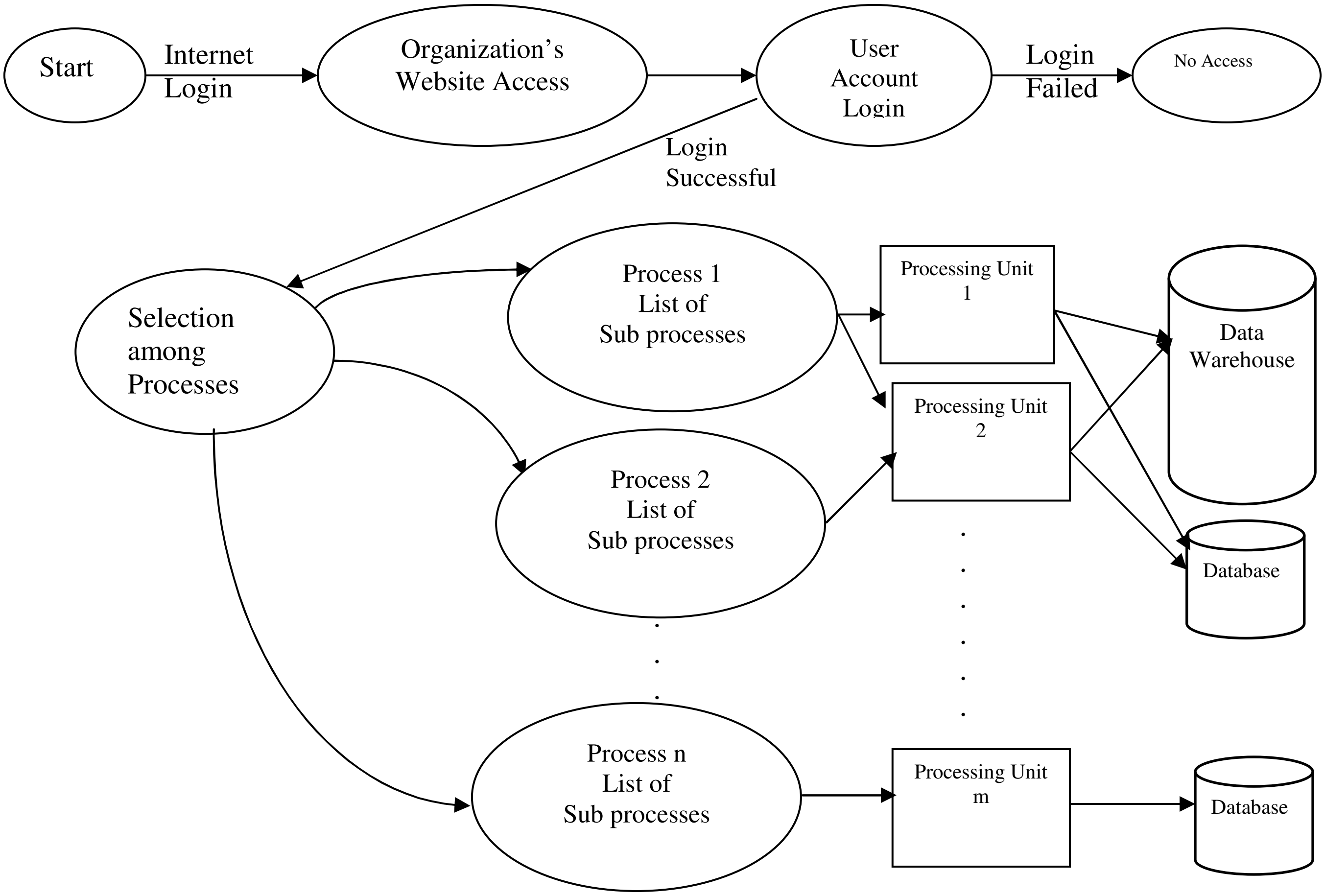


Figure 2: Accessing User account through Internet or Intranet.

User may access its account by connecting from any part of the world through Internet. Each user is provided information from internal and external sources of information of the organization. Key process areas need to be identified and information is collected in central database to furnish the required information, so that one may acquire up-to-date information related to the organization. Different internal/external sources of information are identified and bundled together to furnish the information from external environment of the organization. Key process areas are those critical operations upon which success of any business is dependent. Key Process Areas of any organization may be among Sales, Advertisement, Logistics, Raw Material, Labour, Finance or Production etc., whereas external sources of Information Government Agencies, Competitors, Industry, Suppliers, Customers and many others. These key success factors can be identified and information related to them can be furnished to the user according to their authority. External sources can be identified and their information may be made available directly or after required processing to the user. Organization's Information System Development Team of Analysts, DBAs, Programmers need to work together for identifying emerging information needs of users, considers the matter of inclusion of new technology etc. continuously. Most important aspect of this model is its open behaviour. Processes and sub processes are combined with their logic and access. Individual process is a complete system in all respects having required accessibility, functionality and logic as shown in figure 3. When a user is authorized to access any process or sub process, that process/ sub process is added to user's account, removal of process(s) or sub processes can be done by removing those from the user account.

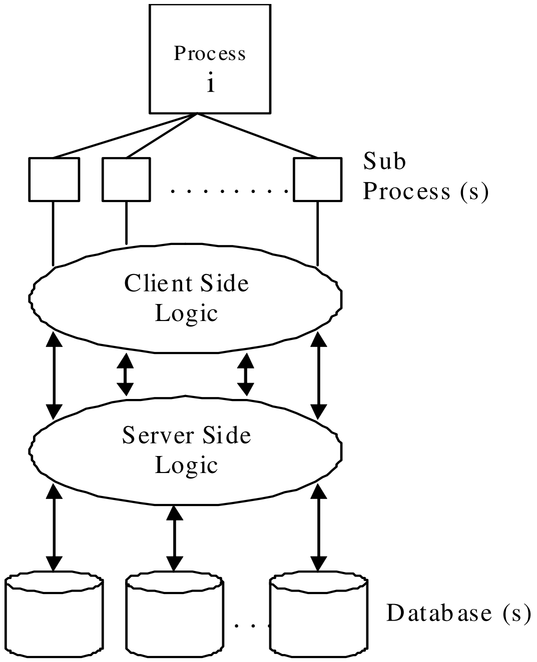


Figure 3: Connectivity of user to database through process/ sub process

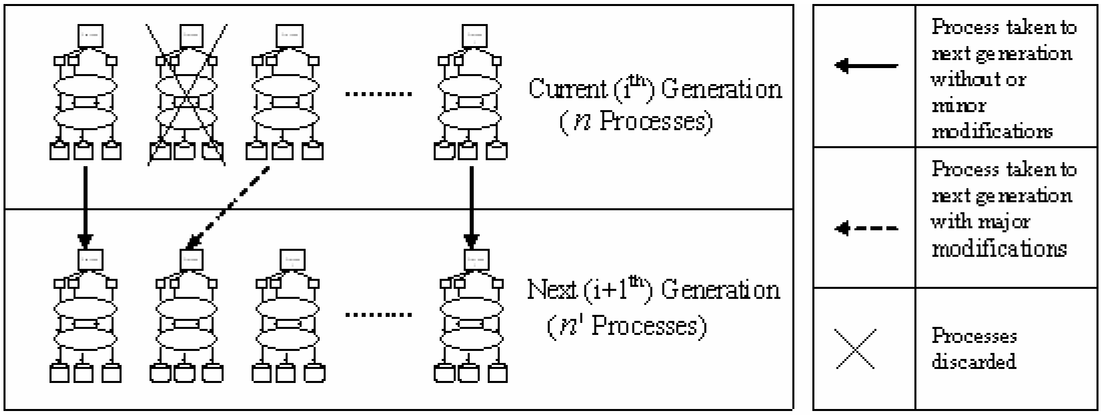

Current (i^th) Generation
(n Processes)
Next (i+1^th) Generation
(n' Processes)
Process taken to next generation without or minor modifications
Process taken to next generation with major modifications
Processes discarded

Figure 4: Transition of processes from current generation to next generation.

Different types of database environments may be combined to give access to the user. If data is to be accessed from two or more different type database environments and minimum access time is needed, then these may be combined in organizational data warehouse on regular basis. If real time access is required, then remote database(s) can be accessed with small delays as well. Providing heterogeneous access of database(s) to the user solves problem of adaptation of new technology and processes. Thorough Documentation is needed for the development and updating the set of processes, so that software reuse is maximized. Whenever a change is implemented, affected areas may be rectified and concerned documentation is done. In this way, problems related to maintenance might be controlled. Desired results can only be achieved, if the development and maintenance is carried out continuously. Set of processes is carried forward from current generation to next generation, after implementing the required modifications. Processes may undergo minor modification of its sub-processes, major modification of processes/sub processes, elimination of complete processes (i.e. discarding complete processes) or may require no change in earlier implementation as shown in figure 4.

6. EFFECTIVENESS OF PROPOSED MODEL

Organizational employees can understand and identify information requirements of their organization far better than the professionals hired for system development from outside organizations. People from outside agencies may be involved for technical expertise for software development, but Genetic information system development require involving more own organization’s employees as system developers. Different problems faced during software development and support provided by proposed model is as follows:

6.1 Cost: Software development is a process of major concern, but a major chunk of total software cost is consumed in maintenance. Genetic development and maintenance cost almost same expenditure, but it helps in developing personalized software, enhancing software reuse and early identification as well as improvement of newly generated information needs on continuous basis, which is profitable in long run. It is due to the reason that the organizational people may understand their organization better than anyone else from outside agency and continuous review enhances the system performance.

6.2 Functionality: Customized or personalized system development helps in achieving higher user satisfaction, as user requirements are understood more closely and system is updated according to these requirements.

6.3 Reliability: When user needs are understood well and are improved whenever changes are required, testing of system in real environment gives maximum fruitful results, if processes are implemented in a well manner and reliability is enhanced.

6.4 Maintainability: Genetic system development and maintenance inherently concentrate on the maintainability of the system.

6.5 Adaptability: System is opened to involve any type of technology and access of databases may be provided directly to the user or it may be connected to the centralized database and suitable changes may be implemented in concerned programs or logic of the system for different users.

6.6 Efficiency: Online/Internet based systems are as good as any offline systems. In case of Online Analytical Processing or Data Mining processing, a small delay in responses may be due to inability of state-of the-art technology to meet desired response time, but still current technology can meet all kinds of information requirements with negligible delays. Open connectivity of improved technology and continuous maintenance of system can help to achieve the desired efficiency.

6.7 Portability and Reuse: Internet connectivity is available with almost all type of technologies with little modifications, so portability is up to the mark for the system. Proposed model sufficiently assures the requirement of proper reuse of software.

6.8 Usability: Users may be trained by simple training programs to use the system and system needs to be developed more intuitively. Different activities of the user pages may be combined in separate windows or drop down menus and details can be accessed through in-depth navigation of the window or connected pages.

6.9 Security: Security of data is always a biggest question, while accessing Internet or online processing. Access to the database need to be secured and multiple levels of security layers may be involved, where authentication is required. Security levels may differ from process to process or database to database as per the requirement.

6.10 Quality: If above discussed parameters are successfully achieved, then quality of information system is assured, not only from the outputs of the system, but also from quality standards defined for the purpose as well.

Proposed model is implemented in almost twenty organizations operating in a variety of business domains. Genetic system development and maintenance model is helpful in finding the solution of many problems related to software development discussed above. However according to the scope of this paper, investigations for testing the model needs to be guided towards reliable maintainability and costs involved. Following case studies concentrate the investigations toward these parameters. Proposed model definitely show the improvement in adaptive and perfective maintenance, so our investigations need to be concentrated upon corrective maintenance/change management at low or no extra cost. So following section will concentrate our investigations in these directions.

7. CASE STUDIES AND HYTPOTHESIS TESTING

7.1 CASE STUDIES

7.1.1 CASE STUDY 1: PEPSI FOODS Ltd. is a renowned soft drink manufacturing, multinational company operating in most of the countries worldwide. Soft drink industry is marketing based business and purely dependent upon eating and drinking habits of the people. Pepsi Foods Ltd. has diversified its business in India not only in soft drinks but also in eatables goods. Information system of the company needs to be efficient enough to meet the dynamic information requirements of the industry. As soft drink industry is marketing operations intensive, its information requirements are highly dynamic. Changing information needs has created a lot of difficulty for the software developers, as maintenance costs rise beyond the estimates and has caused unreliability even after spending sufficient amounts on the system development and maintenance. Earlier software development methodologies and models used, could not meet the requirements.

7.1.2 CASE STUDY 2: Ranbaxy Laboratories Ltd. is a leading pharmaceutical multinational company of India. Rising competition from local and global players has caused the need of an efficient information system to meet the challenge of maintaining leadership and expansion of business globally. Dynamic information needs has forced the company to think over its information technology and information system strategy. Earlier information system development has failed to meet the challenge. Company is looking for any permanent solution for their information needs.

7.1.3 CASE STUDY 3: Usha Power Tec. is a company committed to meet the need for power conditioning in India. Company has record tremendous growth in future and has bright future prospects as well, but rising competition and dynamic information requirements of the information systems has forced the company to think on strengthening its information system. Earlier development is unable to meet the requirement and company is lifelong solutions for its information systems.

7.1.4 CASE STUDY 4: Megaleap Inc. is an online retailing organization. Company deals in a variety of e-Stores to provide a storefront to a variety of manufacturers and retailers. Company operates in a highly dynamic environment. Company has its information system operating, but dynamism of customer choices, retailers offers etc. has emerged the need for incorporating the information dynamism to be incorporated in its information system. So company is looking for long-term solution of its information system development and maintenance at optimized costs.

7.2 Hypothesis Testing

Teams of researchers and software developers were involved in the process of software development for the organizations discussed above. Genetic information system development and maintenance process was tried and the results were tested against data collected from earlier developments for the same organizations and actual data of the development process and results of case study 1 are presented here, similar results are observed for other organizations discussed above.

Research methodology for proposed research work includes implementation of proposed model in diversified business organizations, Collection of data from actual implementation, Collection of dataset from standard organizations, Comparison of results, Hypothesis testing and concluding with results etc.

For testing parameters like reliable maintenance of information system and total system costs are of major concern. Proposed model is highly adaptive due to adoption of Internet based technologies and team involved on continuous basis for system development work for the perfection of the system. So, Hypothesis testing is performed against the parameters related to reliable corrective maintenance and costs involved. Datasets are collected from the organizations' offices scattered worldwide, whereas Criteria for collection of data was based on Type of Business organization, System development/Maintenance Costs, Size of System, Environment of the system, Scale of operations for system etc. and these factors were also normalized according to the characteristics of the system under study.

7.2.1 Reliability testing through numbers of faults received after initial implementation of information systems.

Reliability of a system depends on the number of faults received and action taken within same week as faults received. First we test the hypothesis related to number of faults received after initial implementation of system. Null and alternate hypothesis for reliability testing in terms of number of faults received after initial implementation of system, are as follow:

H_0 = There is no significant difference between reliability trends of proposed model and other earlier models in terms of number of faults received after new system is implemented.

H_1 = There is a significant difference between reliability trends of proposed and other earlier models.

Significant level is kept 98% for the acceptance of Null hypothesis. On the basis on data collected from earlier implementations and data collected from proposed model's implementation, following graphical comparison is produced. Figure 5 include the difference between the faults received for benchmark system and system under study.

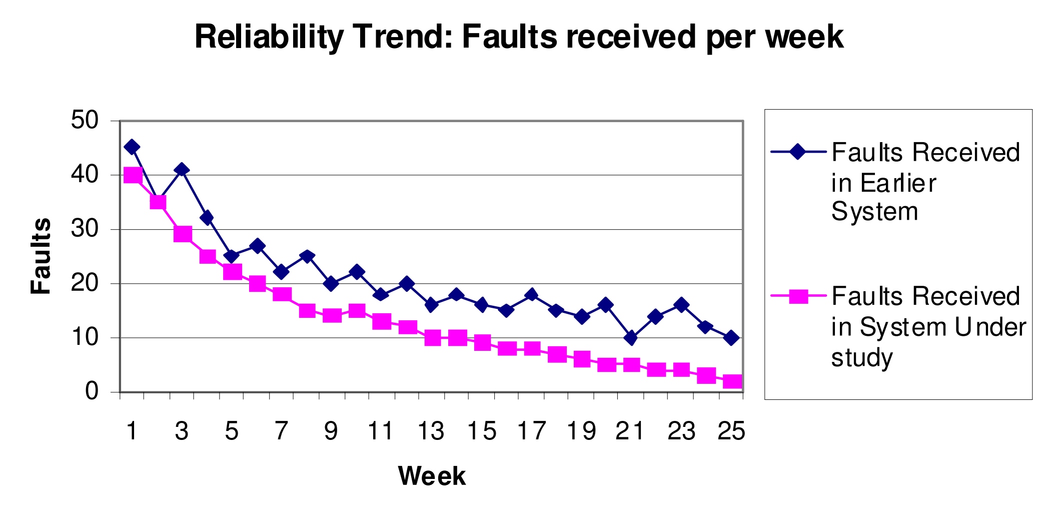


Figure 5: Reliability Trend: Faults received per week after initial implementation

Continuous decline in number of faults is recorded from proposed methodology as compared to the earlier system. Above data are applied to test the hypothesis, chi-square tests are performed for testing goodness of fit and results show the rejection of null hypothesis implying the acceptance of alternative hypothesis. i.e. there is a significant difference between faults received from current system and earlier implementations.

7.2.2 Reliable corrective maintenance testing through action taken within same week against the faults received after initial implementation of information system.

Reported faults need to be corrected as soon as possible. Faults received and action taken to correct them within same weeks as they received is the criteria used for testing reliable corrective maintenance. Null and alternate hypothesis for reliability testing are as follow:

H_0 = There is no significant difference between reliability trends of proposed model and other earlier models in terms of action taken against faults received after new system implemented.

H_1 = There is a significant difference between reliability trends of proposed and other earlier models.

Significant level is kept 98% for the acceptance of Null hypothesis. Figure 6 include the comparison of action taken against the reported faults within same week for proposed model.

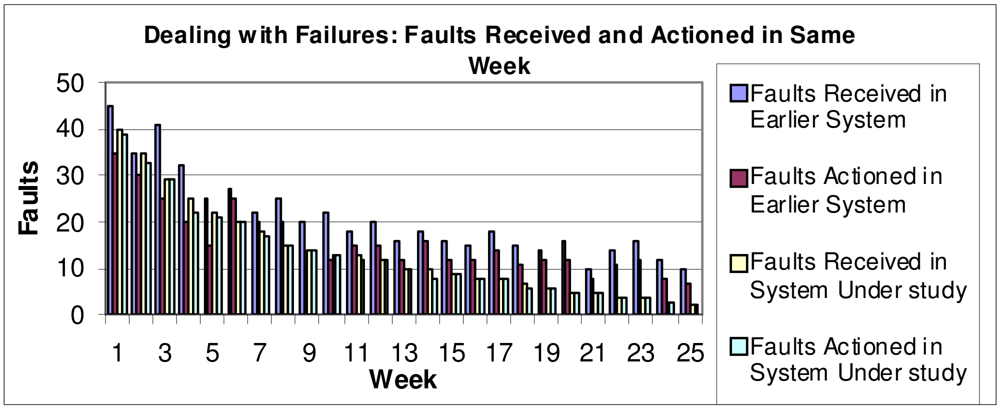


Figure 6: Dealing with failures: Faults Received and Action taken in Same Week

Figure 7 includes the comparison of percentage of faults corrected within the same week as fault(s) reported for earlier system development and for proposed model implementation.

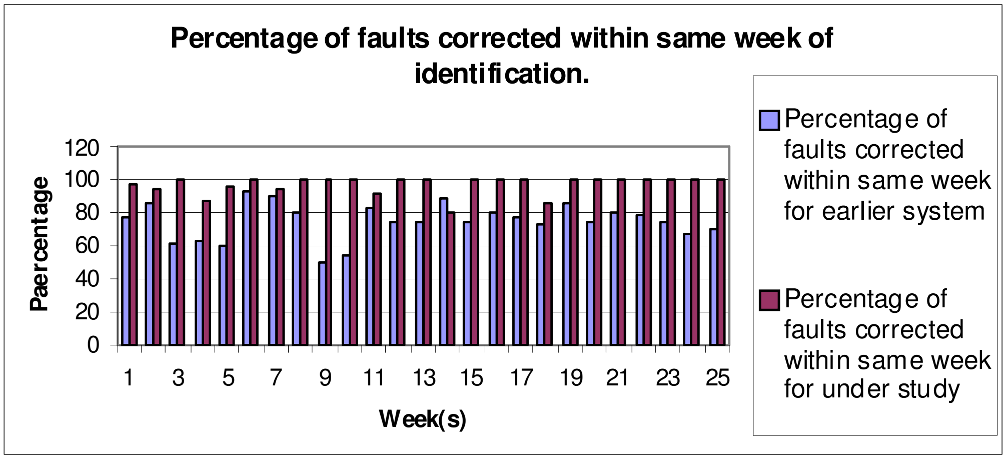


Figure 7: Percentage of faults corrected within same week of identification.

Above data is applied to test the hypothesis, chi-square test for testing goodness of fit and results indicate the rejection of null hypothesis implying the acceptance of alternative hypothesis and indicating a significant improvement in taking action against faults received.

7.2.3 Testing the difference in cumulative costs for the systems based on other models and proposed model.

Genetic information system development and maintenance involves a team of company employees to look after the system continuously contributing to higher system development costs, but maintenance costs for such system development will be lowest possible or will hide behind development cost. So, cumulative costs (including system development and maintenance costs including effort involved) for the system will be very similar to the system developed and maintained through other models. Null and alternate hypothesis for testing the differences between cumulative development and maintenance costs for earlier models and proposed model are defined as follow:

H_0 = There is no significant difference between cumulative development and maintenance costs of proposed model and other earlier models.

H_1 = There is a significant difference between cumulative development and maintenance costs of proposed and other earlier models.

Significant level is kept 95% for the acceptance of Null hypothesis. On the basis collection of standard and actual datasets following graphical comparisons are produced. Figures 8 and 9 include the comparison of the systems based on costs involved.

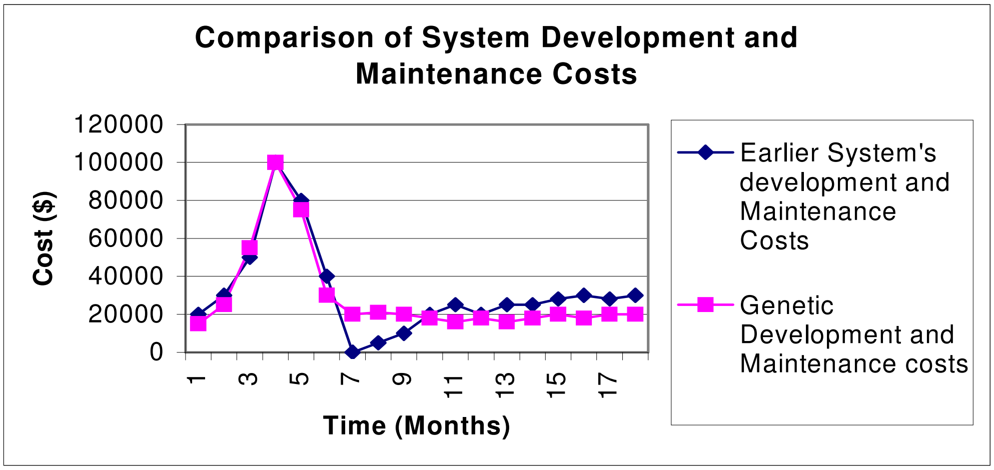


Figure 8: Comparison of System Development and Maintenance Costs

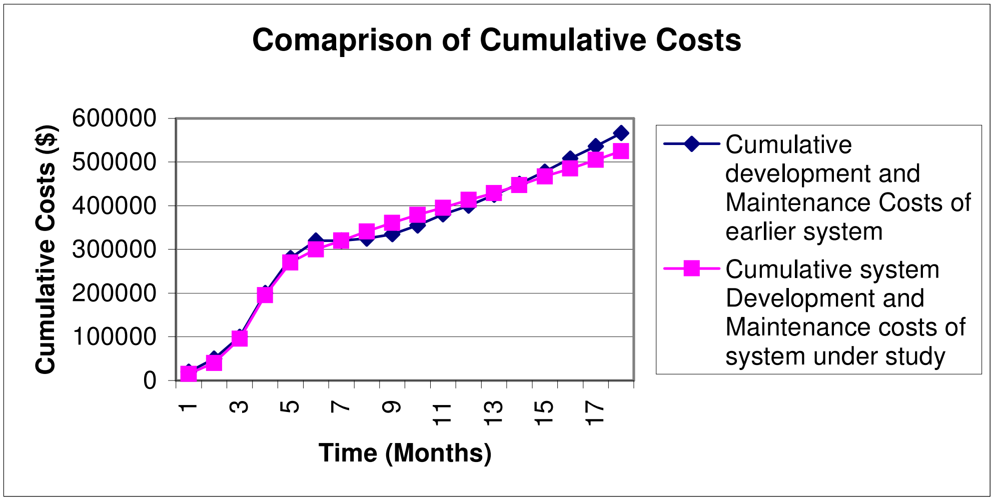


Figure 9: Comparison of Cumulative Costs

From comparison, we find that with time, maintenance costs for other benchmark system increases very rapidly, whereas it remains under control for proposed model. Cumulative costs for system development and maintenance are almost same for a newly developed system, followed by decline in costs for other models and then followed by tremendous increase in total costs of the system development and maintenance for benchmark system, whereas for proposed model cumulative costs are lesser in long run. Above data is applied to test the hypothesis chi-square test for goodness of fit and results indicate the acceptance of null hypothesis implying the rejection of alternative hypothesis, which implies that there is no significant difference between the total costs of the earlier models and proposed model.

7.2.4 Comparative analysis to check the effect of proposed model over software reuse

Effect of proposed model on software's reuse is studied and compared, based on the Lines of Code (LOC) reused during different revisions of software. The de-facto standard for measuring the reuse level (Poulin [44]), is the percentage of software (LOC) reused as compared to the Total size of software (LOC), i.e.

$$\frac{\text{Reused Software}}{\text{Total Software}} X\,100 \qquad (7)$$

Software reuse is compared for system under study with earlier used system in terms of percentage of lines of codes reused during each revision. Figure 10 displays the percentage of software reused during each revision.

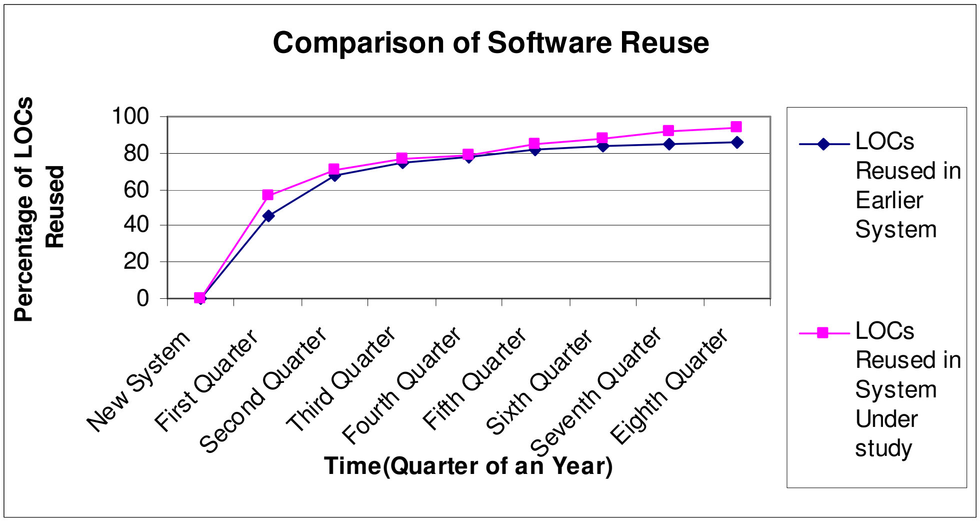


Figure 10: Comparison of Software Reused in Percentage of Lines of Code reused

Analysis of above comparative study shows improvement in software reuse for the system developed through proposed model.

7.2.5 Generalized comparison of proposed model with other models

Comparative study of different models based on general observations and expert opinion about different models, is summarized in Table 1, it contains the Comparison of models based on costs, reliability, maintainability, adaptability, portability and reusability.

Model(s) System Factors	Water Fall Model	Spiral Model	Iterative Model	Proto-typing model	Component Based Development	Genetic development and maintenance
Development cost	Moderate	Moderate	High	High	High	High
Maintenance cost	High	High	Moderate	Moderate	Moderate	Low
Reliability	Low	Moderate	High	Moderate	Moderate	High
Maintainability	Low	Moderate	Moderate	Moderate	Moderate	High
Adaptability	Low	Moderate	Moderate	High	High	High
Portability	Low	Moderate	Moderate	High	High	High
Reusability	Low	Moderate	Moderate	Moderate	High	High

Table 1: Comparison of different methodologies with cost, reliability, maintainability and adaptability of information systems

Above these advantages, many other aspects like user satisfaction, goal orientation, human and organizational factors are also observed to be better for proposed model as compared to any other implementation.

8. CONCLUSION AND FUTURE DIRECTIONS

From the paper, we conclude that Organization and Information Systems progress step by step together, so the development process of information systems cannot be restricted to a limited span of information system development project. Increasing complexities of business processes and information systems demand better change management, Maximum Software Reuse and continuous effort for system development and maintenance. Positive results from the implementation of proposed model indicate the future success of Genetic information system development and maintenance model for new age information systems and for the success of the organizations for improving their business processes. From results, we conclude that proposed model helps in software maintainability and software reuse significantly without extra expenditure. Authors are working on the implementation of the proposed process and model in other diversified business organizations and for its acceptance worldwide. We hope that problems related to software development and maintenance will be minimized, by using Genetic system development and maintenance model and organizations will be able to flourish their operations without any worry about the maintenance and cost related problems of the information systems.

9. REFERENCES

1. Alavi M. *"An Assessment of the Prototyping Approach to Systems".* Communications of the ACM, 27, 6, 556-564, 1985

2. Aoyama M., *"Component-Based Software Engineering: Can it Change the Way of Software Development?".* Proceedings Volume II of the 1998 International Conference on Software Engineering April, 1998

3. Armenise P., Bandinelli S., Ghezzi C. and Morzenti A. *"A survey and assessment of software process representation formalisms"*. International Journal of Software Engineering and Knowledge Engineering, Vol. 3 No. 3, pp. 410-26, 1993

4. Bandinelli S., Fugetta A. and Grigoli S. *"Process modelling in the large with SLANG".* Proceedings of the 2nd International Conference on Software Process, Berlin, pp. 75-93, 1993

5. Basili V.R. and Turner A.J. *"Iterative Enhancement A Practical Technique for Software Development"* IEEE Transactions on SoftwareEngineering 1, 4, 390-396, 1975

6. Bersoff E. *"Elements of software configuration management"*. IEEE Trans. Software Engg., 1984

7. Bider I., Johannesson P. and Perjons E. *"Goal-Oriented Patterns for Business Processes".* position paper for Workshop on Goal-Oriented Business Process Modeling (GBPM'02), 2002

8. Boehm B. W., Gray T. E. and Seewaldt T. *"Prototyping Versus Specifying: A Multiproject Experiment"*. IEEE Transactions on Software Engineering, SE-10, 4, 290 -402, 1984

9. Boehm Barry W. *"A spiral model of software development and enhancement"*. IEEE Computer, 21(5):61-72, 1988

10. Brandtweiner R and Scharl A. *"An institutional approach to modeling the structure and functionality of brokered electronic markets"*. International Journal of Electronic Commerce 3(3), 71-88, 1999

11. Bubenko J., Rolland C., Loucopoulos P. and De Antonellis V. *"Facilitating 'fuzzy to formal' requirements modelling".* Proceedings of the IEEE 1st Conference on Requirements Engineering, ICRE'94, Colorado Springs, CO and Taipei, pp. 154-8, 1994

12. Burrows C., George G. and Dart S. *"Configuration Management".* Ovum Ltd., 1996

13. Cho N. J. *"Internet business in Korea: the state-of-art"* Management and Computer 188-220, 1999

14. Conradi R. and Westfechtel B. *"Towards a Uniform Version Model for Software Configuration Management"*. Proceedings of the Seventh International Workshop on SoRware Configuration Management, New York., 1997

15. Dart S. *"Spectrum of Functionality in Configuration Management Systems".* Technical Report CMU/SEI-90-TR-11, Software Engineering Institute, Pittsburgh, Pennsylvania, December, 1990

16. Dart S. *"Concepts in Configuration Management Systems".* Proceedings of the Third International Workshop on Software Configuration Management, pages 1-18. ACM SIGSOFT, 1991

17. Darwin Charles. *"The Origin Of Species".* 6th Edition, John Murray, London, P. 278, 1902

18. Decker S., Daniel M., Erdmann M. and Studer R. *"An enterprise reference scheme for integrating model based knowledge engineering and enterprise modeling".* Proceedings of the 10th European Workshop on Knowledge Acquisition, Modeling and Management, EKAW'97, Springer-Verlag, Heidelberg, Lecture Notes in Artificial Intelligence, LNAI 1319, 1997

19. Drucker Peter F. *"The Practice of Management. Contributors"*. Publisher: Harper & Row., New York, 1954

20. Estublier J., Dami S. and Amiour M. *“High Level Process Modeling for SCM Systems”.* Proceedings of the Seventh International Workshop on Software Configuration Management, New York., 1997

21. Feller P. *“Configuration Management Models in Commercial Environments”.* Technical Report CMU/SEI-91-TR-7, Software Engineering Institute, Pittsburgh, Pennsylvania, March, 1991

22. Floyd C. *“A Systematic Look at Prototyping, in: Budde, R., Kuhlenkamp, K., Mathiassen, L. and Zullighoven, H. (Eds.) Approaches to Prototyping”.* Springer-Verlag: Heidelberg, 1-17, 1984

23. Forrest Stephanie. *“Genetic Algorithms”.* ACM Computing Surveys, Vol. 28, No. 1, 1996

24. Frakes William and Terry Carol. *“Software Reuse: Metrics And Models”*. ACM Computing Surveys, Vol. 28, No. 2, 1996

25. Gomaa H. *“The Impact of Rapid Prototyping on Specifying User Requirements”.* ACM SIGSOFT Software Engineering Notes 8, 2, 17-28, 1983

26. Gonsalves G.C., Lederer A.L., Mahaney R.C. And Newkirk H. E. *“A customer resource life cycle interpretation of the impact of the World Wide Web on competitiveness: expectations and achievements”.* International Journal of Electronic Commerce 4(1), 103-120, 1999

27. Gordon V.S. and Bieman J.M. *“Rapid Prototyping: Lessons Learned”.* IEEE Software, 12, 1, 85-95, 1994

28. Hamill J. *“The Internet and international marketing”*. International Marketing Review 14(5), 300-323, 1997

29. Hunt J., Lamers F., Renter J. and Tichy W.F. *“Distributed Configuration Management via Java and the World Wide Web”.* In Proceedings of the Seventh International Workshop on Software Configuration Management, New York., 1997

30. Hyde A. *“The Proverbs of Total Quality Management: Recharting the Path to Quality Improvement in the Public Sector”*. Public Productivity and Management Review, 16(1), 25-37, 1992

31. Ishikawa K. *“What is Total Quality Control?”.* The Japanese way. Englewood Cliffs, New Jersey, Prentice- Hall, 1985

32. Jacobson I., Booch G. and Rumbaugh J. *“Unified Software Development Process”.* Object Technology Series, Addison-Wesley, New York, 1999

33. Jarzabek S. and Ling T.W. *“Model-based support for business reengineering”.* Information and Software Technology, Vol. 38 No. 5, pp. 355-74, 1996

34. Kaushaar J.M. and Shirland L. E. *“A Prototyping Method for Applications Development End Users and Information Systems Specialists”.* MIS Quarterly, 9, 4, 189 -197, 1985

35. Khomyakov M. and Bider I. *“Achieving workflow flexibility through taming the chaos”.* paper presented at OOIS 2000 – 6th International Conference on Object Oriented Information Systems, Springer-Verlag, Berlin, pp. 85-92, 2000

36. Kramer Stefan And Kaindl Hermann. *“Coupling And Cohesion Metrics For Knowledge-Based Systems Using Frames And Rules”.* ACM Transactions On Software Engineering And Methodology, Vol. 13, No. 3, Pages 332–358, 2004

37. Kueng P. and Kawalek P. *“Goal-based business process models: creation and evaluation”*. Business Process Management Journal, Vol. 3 No. 1, pp. 17-38, 1997

38. Leblang D.B. *“Managing the Software Development Process with ClearGuide”*. Proceedings of the Seventh International Workshop on Software Configuration Management, New York., 1997

39. Lee S. *“Business use of internet-based information systems: the case of Korea”.* European Journal of Information Systems, Vol. 12, Issue 3, Sep, 2003

40. Marca D.A. and McGowan C.L. *“Business Process and Enterprise Modeling”.* Eclectic Solutions, Inc., San Diego, CA, 1993

41. Milewski B. *“Distributed Source Control System”.* In Proceedings of the Seventh International Workshop on Software Configuration Management, New York., 1997

42. Nauman J.D. and Jenkins M. *“Prototyping: The New Paradigm for Systems Development”.* MIS Quarterly, 6, 3, 29-44, 1982

43. Palvia P. and Nosek J. T. *“An Empirical Evaluation of System Development Methodologies”.* Information Resources Management Journal, 3,23-32, 1990

44. Poulin J. S. “*Measuring Software Reuse: principles, practices, and economic models”*. Reading, MA, Addison-Wesley, 1997

45. Rolland C. *“L’e-lyee: L’escritoire and Lyeeall”.* Information and Software Technology, Vol. 44, pp. 185-94, 2002

46. Rolland C. *“Reasoning with goals to engineer requirements”*. ICEIS’03– Fifth International Conference on Enterprise Information Systems, Angers, France, pp. 11-19, 2003

47. Royce W. W. *“Managing the Development of Large Software Systems: Concepts and Techniques”.* Proceedings WESCON, August, 1970

48. Rumbaugh J., Blaha M., Premerlani W., Eddy F. and Lorensen W. *“Object Oriented Modeling and Design”.* Prentice-Hall, New York, NY, 1991

49. Tate G. and Verner J. *“Case Study of Risk Management, Incremental Development and Evolutionary Prototyping”.* Information and Software Technology 42, 4, 207-214, 1990

50. TICHY W. F. *“Tools for software configuration management”*. Proceedings of the International Workshop on Software Version and Configuration Control (Grassau, Germany), J. F. H. Winkler, Ed., Teubner Verlag, 1–20, 1988

51. Weill P. *“The relationship between investment in information technology and firm performance: a study of the valve-manufacturing sector”*. Information Systems Research 3(4), 307-333, 1992

An Empirical Comparison Of Supervised Learning Processes

Sanjeev Manchanda* smanchanda@thapar.edu
School of Mathematics and Computer Applications,
Thapar University, Patiala-147004 (INDIA).
**Corresponding author*

Mayank Dave mdave67@yahoo.com
Department of Computer Engg.,
National Instt. Of Technology, Kurukshetra, India.

S. B. Singh sbsingh69@yahoo.com
Department of Mathematics,
Punjabi University, Patiala, India.

Abstract

Data mining as a formal discipline is only two decades old, but it has registered phenomenal development and has become a mature discipline in this short span. In this paper, we present an empirical study of supervised learning processes based on empirical evaluation of different classification algorithms. We have included most of the supervised learning processes based on different pre pruning and post pruning criteria. We have included ten datasets, collected from internationally renowned agencies. Different specific models are presented and results are generated. Issues related to different processes are analyzed suitably. We also present a comparison of our study with benchmark results of different datasets and classification algorithms. We have presented results of all algorithms with fifteen different performance measures out of a set of twenty three calculated measures, making it a comprehensive study.

Keywords: Data Mining, Knowledge Discovery in Databases, Supervised learning algorithms, Stacking, Classification, Regression etc.

1. Introduction

Knowledge discovery in databases (KDD) is the theme of many discussions for last two decades. A large number of techniques and algorithms have been developed for mining the knowledge from large databases. Supervised learning techniques are usually used for the solution of classification problems. Usually a general process is recommended for supervised learning. But practical implementation of a general process becomes difficult, when we need to implement this general process for some specific problem solving. There are possibly many processes that are used for supervised learning. Problem arises with finding a suitable process for extracting knowledge for problem at hand. This type of dilemma motivated us to analyze the environmental factor that affect the selection of a suitable process and to handle potential issues involved in such processing.

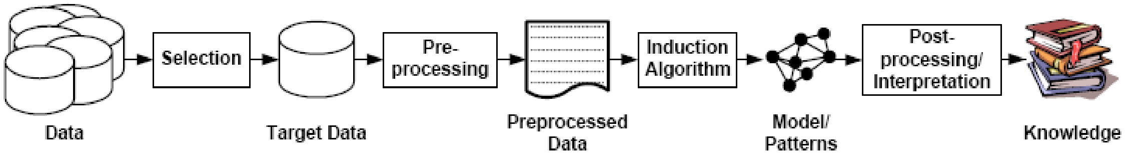


Figure 1: The KDD process (Fayyad et al. [10])

Present work is mainly motivated through following three objectives. First of all, supervised learning processes can vary from simple to very complex processing. No single process can fulfill all needs and suitability of any process depends upon many environmental factors. So, there is a need to analyze different processes by identifying different environmental factors. Secondly, Different techniques and algorithms are used to extract knowledge from data. These algorithms involve certain criteria to extract knowledge. Different techniques and algorithms are suitable for different types of problems. There is no unique technique/algorithm to solve all types of problems. So, there is a need to analyze suitability of different techniques/algorithms with specific domain of problems. Thirdly, Different performance metrics are considered appropriate for different domains, e. g. Precision/Recall measures are preferred metrics for information retrieval, ROC curves/area is preferred metric for the problems related to medical domain, Lift is preferred for marketing tasks etc. Each metric is dedicated to some specific nature of algorithm evaluation. No individual metric may be used for all domains. So, there is a need to test different learning algorithms based on a large set of metrics. Overall present paper is an effort to explore relationship between types of problems with specific technique/algorithm as well as with type of processing required for extracting knowledge based on different metrics. Experiments are performed through many suitable processes on a variety of supervised learning techniques and algorithms. Results are presented for fifteen different metrics out of generated results for twenty three metrics. Output of these experiments is compared with the results obtained from direct experimentation of classification algorithms and the results obtained through cross validations. Results are also compared with the available benchmark results of the problems involved for study. This paper includes a comprehensive study of different possible supervised learning processes. Internationally renowned datasets are chosen for evaluating six most important processes for study. These datasets are applied on these processes and comprehensive results are presented.

Rest of the content of this paper is organized in following manner. Second section includes the literature review of related work. Third section includes the description of various processes for supervised learning. Fourth section includes the description of different techniques and algorithms included for study. Fifth section explains methodology of study. Sixth section includes experimental results of present study. Seventh section includes a comparison of present study with other studies. Eighth section concludes the study with future directions. Last but not the least, Ninth section lists the references used during present study.

2. Literature Review

Data mining has originated just two decades back. Within this short span, data mining has grown up as a mature discipline. Large numbers of techniques and algorithms have been developed for extraction of knowledge. Out of these algorithms, majority of algorithms are developed for supervised learning. Supervised learning is mostly performed for classification tasks. Data mining itself has emerged from other disciplines like Machine Learning, Artificial Intelligence and Statistics etc., so it is obvious to get initial references related to this study from its parent disciplines. Many researches were being performed before the time data mining was coined as a separate discipline for study.

In a study, the results of a point awarding approach were compared with the results obtained by the linear discriminant (Fahrmeir et al. [9]). One study reported that back-propagation outperformed nearest neighbour for classifying sonar targets (Gorman et al. [13]), whereas some Bayes algorithms were shown to be better on other tasks (Shadmehr et al. [30]). A symbolic algorithm, ID3 (Kirkwood et al. [16]) was developed, which performed better than discriminant analysis for classifying the gait cycle of artificial limbs.

The CART (Classification and Regression Trees) method (Breiman et al. [6]) was used to analyze consumer credit granting (Hofmann [14]). It concluded that CART had major advantages over discriminant analysis and emphasized the ability of CART to deal with mixed datasets containing both qualitative and quantitative attributes. However, on different tasks other researchers found that a higher order neural network (HONN) performed better than ID3 (Spikvoska et al. [32]) and back-propagation did better than CART (Atlas et al., [1]).

A study was conducted for a coordinated comparison of many algorithms on the MONK's problem (Mitchell et al. [20]). A diverse set of statistical methods, neural networks, and a decision tree classifier was compared on the Tsetse fly data (Ripley [28]). After many small comparative studies, STATLOG is known as first comprehensive study that analyzed different data mining algorithms (King et al. [15]). Another research work compared several learning algorithms (including SVMs) on a handwriting recognition problem using three performance criteria: accuracy, rejection rate, and computational cost (LeCun et al. [18]). One other study evaluated nearly a dozen learning methods on a real medical data set using both accuracy and an ROC-like metric (Cooper et al. [8]). In one other study, an impressive empirical analysis was presented about different ensemble methods such as bagging and boosting (Bauer et al. [3]). An empirical comparison of decision trees and other classification methods was performed using accuracy as the main criterion (Lim et al. [19]). An empirical study conducted comparison between decision trees and logistic regression (Perlich et al. [23]). One study examined the issue of predicting probabilities with decision trees, including smoothed and bagged trees (Provost et al. [25]). One research work presented the comparison of different tools and techniques of data mining (Witten et al. [33]). Recently, one study was conducted to rank different many learning algorithms (Caruana et al. [7]). Present research work is dedicated to analyze all type of classification techniques and algorithms on a variety of problems and to compare the results with earlier studies.

3. Various Processes for Supervised Learning

Supervised learning processes can vary from simple processing to very complex processing. Different techniques and algorithms are used to extract knowledge from data. These algorithms involve certain criteria to extract knowledge. Different techniques and algorithms are suitable for different types of problems. There is no unique technique/algorithm to solve all types of problem. Supervised learning involves training set to train algorithm for the creation of a model and then this model is applied on test set to generate and compare results. Different supervised learning processes are as follows:

3.1 Simple Supervised Learning: In its simplest form input data is applied to classification algorithm to generate a model, model is applied test data and result is generated. Such experimentation suffers with over fitting and under fitting of model and results may not fulfill the reliability criteria. So there is a need for preprocessing and post-processing of data.

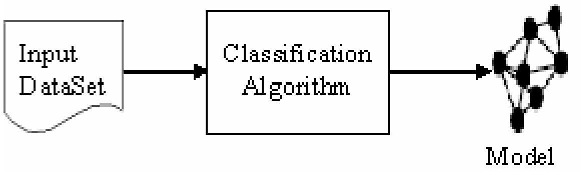


Figure 2: Simple Supervised Learning Process.

3.2 Preprocessing of the data: A data set collected is not directly suitable for induction (knowledge acquisition), it comprises in most cases noise, missing values, inconsistent data, data set is too large, and so on. Therefore, we need to minimize the noise in data, choose a strategy for handling missing (unknown) attribute values, use any suitable method for selecting and ordering attributes (features) according to their informativity (so-called attribute mining), discretize/fuzzify numerical (continuous) attributes, validating part of training data to be used for creating model and eventually process continuous classes.

3.2.1 Attribute Transformation: Input data may be nominal or numerical. Few classification algorithms like ID3 and Naïve Bayes operate only on discrete data, whereas regression based algorithms operate only on numerical data. So there may a requirement of transformation of data from one form to another to match the data with algorithmic requirements.

3.2.1.1 Categorical Attribute Transformation: Nominal or categorical data may be transformed into binary or scale values as follows:

(a) Categorical to Binary: Problems having more than two categories of class attribute are converted into Binary class problems. We have converted our datasets into binary class treating first half of class categories as negative class and last half as positive class.

(b) Dual Scaling: Dual scaling (Nishisato [22]) is a multivariate method for assigning scale values to the rows and columns of a table of data, with certain optimal properties.

3.2.1.2 Continuous Attribute Transformation: Continuous or real number based attributes may be transformed into discrete attributes as follows:

(a) Class-based discretization: Class-Attribute relationship is used to define discretization of any attribute, each attribute is discretized independently. Such discretization is useful for small number of attributes, but becomes complex for large number of attribute.

(b) Fixed-bin discretization: All the attributes to be discretized are considered collectively and a fixed number of bins are used for discretizing all attributes. We have used fixed bin discretization, so that the future researchers can utilize the results of this paper for their comparative analysis and it also helps in maintaining consistency of experimentation.

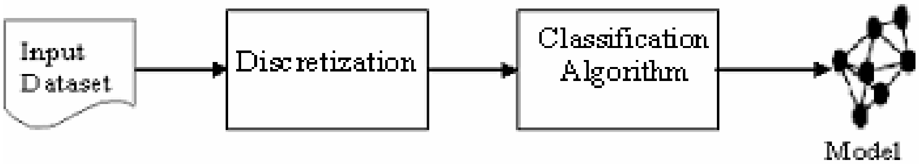


Figure 3: Discretized Supervised Learning Process.

(c) Principle component analysis: Principal component analysis is a useful tool for categorization of data, it separates the dominating features in the data set.

3.2.2 Data Sampling

(a) Progressive sampling: Progressive Sampling (PS) (Provost et al. [27]) incrementally constructs a training set from a larger dataset without decreasing the classification performance and without altering the initial format of the examples

(b) Random sampling: Samples are selected randomly for experimentation. Such a sampling makes the experimentation results to be unreliable as different sampling algorithms may select samples differently and results may vary significantly.

(c) Stratified sampling: Stratified sampling is based on re-sampling the original datasets in different ways: under-sampling the majority class or over-sampling the minority class.

3.2.3 Validation:

(a) Fixed Split Validation: Simplest form of experimentation is to divide dataset into two fixed length datasets of training set and test set to perform experiment directly. This kind of experimentation is meant for simple testing of algorithms. Biggest problem with fixed split is over fitting of training data e.g. tree based techniques may have too many branches that may reflect anomalies and result in poor accuracy of unseen samples. To overcome the limitations of fixed split two approaches used are prepruning and post pruning. Prepruning is performed through cross-validation, whereas many calibration methods have been proposed for post pruning. Following sections include the discussion about these methods.

(b) Cross Validation: To evaluate the robustness of the classifier, the normal methodology is to perform cross validation on the classifier. Ten fold cross validation has been proved to be statistically good enough in evaluating the performance of the classifier (Witten et al. [33]). For present study datasets are divided into training and test sets. Then training set is equally divided into 10 different subsets for ten fold cross validation. Nine out of ten of the training subsets are used to train the learner and the tenth subset is used as the test set. The procedure is repeated ten times, with a different subset being used as the test set. In this way cross validation is performed to calibrate the models and select the best parameters and then models are applied on the large Final test set.

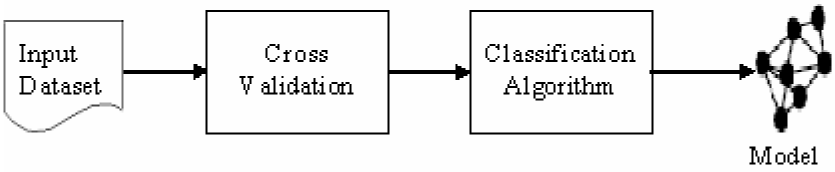


Figure 4: Cross-Validated Supervised Learning Process.

3.3 Post processing of the derived knowledge: The pieces of knowledge extracted in the previous step could be further processed. One option is to simplify the extracted knowledge. Also, we can evaluate the extracted knowledge, visualize it, or merely document it for the end user. They are various techniques to do that. Next, we may interpret the knowledge and incorporate it into an existing system, and check for potential conflicts with previously induced knowledge.

3.3.1 Calibration: Many learning algorithms do not predict probabilities. For example the outputs of an SVM are normalized distances to the decision boundary, whereas naive bayes models are known to predict poorly calibrated probabilities, because of the unrealistic independence assumption.

A number of methods have been proposed for mapping predictions to posterior probabilities. Platt Scaling (Platt [24]) is used for transforming SVM predictions to posterior probabilities by passing them through a sigmoid. Platt scaling also works well for boosted trees and boosted stumps (Niculescu et al.[21]). A sigmoid is also not the correct transformation for all learning algorithms.

Second method used for calibration is Logistic regression. Logit Boost algorithm is used for performing additive logistic regression. This algorithm performs classification using a regression scheme as the base learner, and can handle multi-class problems (Friedman et al. [11]) and it can also do efficient internal cross-validation to determine appropriate number of iterations.

Other method generally used for calibration is Isotonic Regression (Zadrozny et al. [35,36]; Robertson et al. [29]). It is used to calibrate predictions from SVMs, naive bayes, boosted naive bayes, and decision trees. Isotonic Regression is more general method, but its only restriction is that the mapping function used is isotonic (monotonically increasing). A standard algorithm for Isotonic Regression that finds a piecewise constant solution in linear time, is the pair-adjacent violators (PAV) algorithm (Ayer et al. [2]).

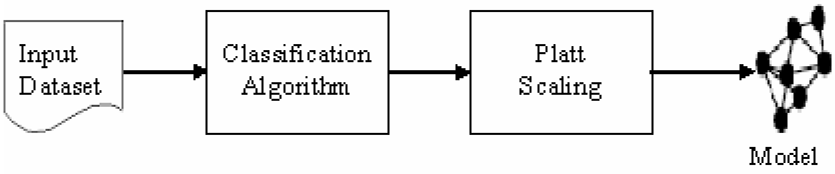


Figure 5: Post-Processed Supervised Learning Process.

3.3.2 Thresholding: The minimum acceptable value which, in the user's judgment, is necessary to satisfy the need. If threshold values are not achieved, program performance is seriously degraded, the program may be too costly, or the program may no longer be timely.

3.3.2.1 Class Probability Estimators (CPE) Thresholding: For a decision maker to act optimally it is necessary to estimate the probability of success. Because training information is costly, we would like to reduce the cost of inducing an estimation model that will render decisions of a given quality. One approach to reducing the cost of learning accurate CPEs is via traditional active learning methods, which are designed to improve the model's average performance over the instance space. if the probability of a successful outcome exceeds the threshold.

3.3.2.2 Regression Thresholding: Threshold regression refers to first-hitting-time models with regression structures that accommodate covariate data. The parameters of the process, threshold state and time scale may depend on the covariates.

3.4 Stacking: Stacking combines the output of a number of classifiers. Stacked Generalization, also known as Stacking in the literature, is a method that combines multiple classifiers by learning the way that their output correlates with the true class on an independent set of instances. At a first step, N classifiers Ci, i = 1..N are induced from each of N data sets Di, i = 1..N. Then, for every instance e_j , j = 1..L of an evaluation set E, independent of the Di data sets, the output of the classifiers Ci(e_j) along with the true class of the instance class(e_j) is used to form an instance m_j , j = 1..L of a new data set M, which will then serve as the meta-level training set. Each instance will be of the form: C1(e_j), C2(e_j), . . . , CN(e_j), class(e_j). Finally, a global classifier GC is induced directly from M. If a new instance appears for classification, the output of all local models is first calculated and then propagated to the global model, which outputs the final result. Any algorithm suitable for classification problems can be used for learning the Ci and GC classifiers. Independence of the actual algorithm used for learning Ci, is actually one of the advantages of Stacking, as not every algorithm might be available for each data set and not the same algorithm performs best for every data set. We have applied stacking of isotonic regression with other classification algorithms.

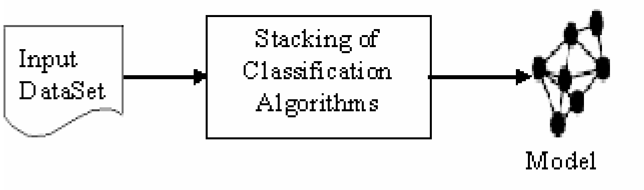


Figure 6: Stacked Supervised Learning Process.

3.5 Complex Processing: Different preprocessing, Post-processing and stacking of different algorithms may be combined to extract knowledge from databases. Such complex criteria may involve parallel processing of different algorithms as well. No encouraging results have been generated through such processing.

4. Description of Techniques and Algorithms used for study

Different techniques included for study with their specific algorithms are as follows:

4.1 Classification Techniques and Algorithms: A variety of classification algorithms were used for study. These techniques/algorithms are broadly described as follows:

4.1.1 Decision Trees: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Decision trees represent a series of IF…THEN type rules which are linked together and can be used to predict properties for observations based upon the values of various features. These are able to produce human-readable descriptions of trends in the underlying relationships of a dataset and can be used for classification and prediction tasks. The algorithms used for experimentation were Decision Stump and REPTree etc. Different parameters were set as follows: maximum tree depth was allowed to be infinite, minimum number of instance per leaf were set to 2, Confidence threshold was set to 0.25 and numbers of trees were allowed to be infinite.

4.1.2 Support Vector Machine: These are methods for creating functions from a set of labeled training data. These functions can be a classification function (the output is binary: is the input in a category) or the function can be a general regression function. For classification, SVMs operate by finding a hyper-surface in the space of possible inputs. This hyper-surface will attempt to split the positive examples from the negative examples. The split will be chosen to have the largest distance from the hyper-surface to the nearest of the positive and negative examples. Intuitively, this makes the classification correct for testing data that is near, but not identical to the training data. We have included LibSVM algorithm for study. Different parameters were set as follows: Different types of kernel functions were tried like linear, polynomial, radial basis function etc., Degree of kernel function set to 3 and Tolerance parameter set to 0.001.

4.1.3 Genetic Algorithms: Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of evolution. Genetic algorithms should be used, when no other option is left. We have not included any genetic algorithm, but learning processes are always based on genetic processing, so indirect contribution of genetic processing can not be neglected.

4.1.4 Neural Networks: Inspired by the structure of the brain, a neural network consists of a set of highly interconnected entities, called *nodes* or *units*. Each unit is designed to mimic its biological counterpart, the neuron. Each accepts a weighted set of inputs and responds with an output. An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. We have included Multi Layer Perceptron algorithm for study. Different parameters were set as follows: Learning rate of back propagation set to be 0.3, Momentum rate 0.2 etc.

4.1.5 K-nearest neighbor: Among the various methods of supervised statistical pattern recognition, the Nearest Neighbor rule achieves consistently high performance, without *a priori* assumptions about the distributions from which training examples are drawn. It involves a training set of both positive and negative cases. A new sample is classified by calculating the distance to the nearest training case; sign of that point then determines the classification of the sample. The IBk classifier included in present study extends this idea by taking the *k* nearest points and assigning the sign of the majority. It is common to select *k* small and odd to break ties (typically 1, 3 or 5). Larger *k* values help reduce the effects of noisy points within the training data set, and the choice of *k* is often performed through cross-validation. Different parameters were set as follows: Different values for k were tried ranging 1 to 10.

4.1.6 Rule Induction: The extraction of useful if-then rules from data based on statistical significance. We have included Decision Table and ZeroR algorithms for study. Different parameters were set as follows: Confidence threshold set to 0.25.

4.2 Boosting/Bagging: These methods create a set or ensemble of classifiers from a given dataset. Each classifier is generated with a different training set obtained from the original using re-sampling techniques. The final output is obtained by voting.

Boosting: The idea of Boosting is to combine simple rules to form an ensemble such that the performance of the single ensemble member is improved i.e. Boosted. AdaBoostM1 algorithm was used for boosting trees (Yoav et al. [34]). Different parameters were set as follows: Number of iterations allowed was 10 and 100 percentage of weight mass being used.

Bagging (**B**ootstrap A**gg**regat**ing**): It produces replications of the training set by sampling with replacement. Each replacement of the training set has the same size as the original set, but some examples can appear more than once while other don't appear at all. A classifier is generated from each replication. All classifiers are used to classify each sample from the test set using a vote scheme (Breiman [6]). We have applied Bagging and Boosting on Decision Stump and REPTree algorithms. Experimental results of both Boosting and Bagging are really enthusiastic. Different parameters were set as follows: Size of each bag being set to 100 and number of iterations allowed were 10.

5. Methodology

5.1 Datasets: Present study compares supervised learning algorithms on ten binary classification problems. ADULT, COV_TYPE, LETTER, PEN_DIGITS, SHUTTLE, SATELLITE and TIC2000 are the problems from UCI repositories (Blake et al. [5]). COV_TYPE has been converted to a binary problem by treating the largest four classes as positives and the rest three as negatives. LETTER is converted by replacing alphabets A-M as negatives and N-Z as positives. PEN_DIGITS is converted by replacing top five digits (5 to 9) into positive class whereas lower five into negative class (0 to 5). SATELLITE and SHUTTLE are the problems from STATLOG. SHUTTLE has been converted to a binary problem by treating largest two classes as positives and rest three classes as negatives. SATELLITE conversion is treated by converting largest three classes (i. e. 4, 5, 7) as positives (class 6 was absent), whereas smallest three classes as negatives (i.e. 1, 2, 3). ACC_CELE and ACC_DROSO are biological sequence datasets (Sonnenburg et al.[31]). DS1_100 is outcome of biological and chemistry experiments (Komarek et al. [17]). Table 1 includes the description about the datasets.

Problem	Number of Attributes	Size of Datasets		
		Train Set	Test Set	Total
ADULT	14	9768	39074	48842
COV_TYPE	54	10000	40000	50000
ACC_CELE	141	10000	40000	50000
ACC_DROSO	141	10000	40000	50000
DS1_100	100	10000	16734	26734
LETTER	16	10000	10000	20000
PEN_DIGITS	16	5000	5992	10992
SATELLITE	36	3000	3435	6435
SHUTTLE	9	10000	40000	50000
TIC2000	85	5000	4822	9822

Table 1: Description of Problems

5.2 Experimentation: Experimentation is the most important part of any empirical study. We have included all the ways of experimentation developed so far for supervised learning. In this study Pre-processing through Fixed split validation and Cross validation have been performed, whereas three calibration methods viz. Platt scaling, Logit Boost and Additive Regression have been used for experimentation and Isotonic Regression has been applied through Stacking of algorithms. Discretization has been applied for ID3 and Naïve Bayes algorithms.

5.3 Metrics for evaluation: Learning techniques and algorithms are used in a variety of domains. Different performance metrics are considered appropriate for different domains, e. g. Precision/Recall measures are preferred metrics for information retrieval, ROC curves/area is preferred metric for the problems related to medical domain, Lift is preferred for marketing tasks etc. Each metric is dedicated to some specific nature of algorithm evaluation. No individual metric may be used for all domains. So, there is a need to test different learning algorithms based on a large set of metrics. Metrics used for testing algorithms are broadly categorized as follows (Same metric may belong to more than one broader category depending on their nature belonging to multiple categories):

5.3.1 Confusion Matrix Based Metrics: Outcome of all classification tasks produces four types of output i.e. two from each instance is mapped to one element of the set { Positive, Negative} from actual positive and negative class labels, whereas other two labels {Positive, Negative} from the class predictions produced by a model. Different statistics like Accuracy, Precision, Recall, Fallout, F-measure, Margin etc. are directly derived from the confusion matrix (Provost et al. [26,27]), whereas Lift, AUC are derived from it.

		Actual Class	
		Positive	**Negative**
Predicted Class	**Positive**	**T**rue **P**ositives	**F**alse **P**ositives
	Negative	**F**alse **N**egatives	**T**rue **N**egatives

Table 2: A contingency table for a binary class problem

5.3.2 Threshold metrics: The threshold metrics are accuracy, F-score and Lift (Giudici, [12]). A fixed threshold 0.5 is used for Accuracy and F-Score. For lift, percent p of cases is predicted as positive and the rest as negative, for present study p is selected to be 25%. Predictions may have a significant distance from these thresholds.

5.3.3 Rank Metrics: The rank metrics used are **A**rea **U**nder the RO**C** curve (i. e. AUC) (Provost et al. [26]), Average Precision and Recall.

5.3.4 Errors: Different types of errors have been involved in this study. Absolute Error, Relative Error, Root Mean Squared Error, Squared Error and Fallout etc. have been calculated for all the algorithms and problems involved for present study. Classification error has been omitted from the table because it can be calculated from the accuracy measure by subtracting accuracy from one.

5.3.5 Probability Metrics: Probability metrics, Root mean squared error and Mean crossed-entropy, interpret the predicted value of each case as a conditional probability of that case being in the positive class.

5.3.6 Other Metrics: Other metrics like kappa and correlation are calculated**.** The kappa coefficient measures pair wise agreement among a set of coders making category judgments, correcting for expected chance agreement (Berry [4]), whereas correlation calculates the degree of relationship between attributes.

6. Experimental Results

This section includes the experimental results of present study. Experimental results are divided into two categories viz. Major study and Minor study.

6.1 Major Study: Major study includes twenty two algorithms and experiments are performed over fixed split, cross validation and platt scaling. For Fixed Split validation original dataset is divided into train set and test set, then experiments are performed. For Cross Validation dataset is again divided into Train set and test set, Train set is further divided into ten fold datasets, Experiments are performed over one fold with the help of others and dataset with minimum squared error is selected for testing the performance over test set. For Platt Scaling, Cross validated model is passed through a sigmoid and probabilities based predictions are performed.

6.1.1 Performance by Problem: Table 3 includes accuracy of twenty two algorithms involved for study and are ranked in descending order based on their average performances. Random Forest algorithms have topped the chart, whereas J48, PART, Multi Layer Perceptron, IBk, REPTree and ADTree algorithms are close to the top positions. Fixed Split has performed better than Cross Validated and Platt Scaling preprocessing and post processing algorithms. As Cross validation restricts the over fitting of algorithms, so the performance over cross validation and platt scaling is the corrected performance of the algorithms. Even for Cross validated and Platt scaling results random forest algorithms perform far better than other algorithms. Bagging (**B**ootstrap A**gg**regat**ing**) has performed better than alone algorithm and with boosting. Fro ADTree Boosting seems to perform better than others and has enhanced the performance rapidly. ZeroR has performed very badly and has secured lowest positions as compared to others.

6.1.2 Performance by metrics: Table 4 includes averages of fifteen metrics involved in study and are positioned in descending order according to average accuracy. For few metrics output was generated to be NaN (**N**ot **a** **N**umber), for such metrics averages are calculated over remaining values, excluding the count of such values. These values are pointed with an asterisk (*) and if all the values (for all ten problems) are NaN, such values are represented by NaN*.

6.2 Minor Study: Minor study includes five algorithms and experiments are performed over other calibration methods like Additive Regression, Logistic Regression and Isotonic Regression. Limitation for regression based methods is that these require fully numeric values, so all the datasets are converted into numeric values except for the Logit Boost algorithm (i. e. Logistic Regression). On Additive regression ten fold cross validation has been applied and study is performed through meta classifiers. Logit Boost involves internal cross validation, so fixed split experimentation is performed. For Isotonic regression, stacking is done in conjunction with other algorithms and ten fold cross validation is performed.

6.2.1 Performance by Problem: Table 5 includes accuracy of five algorithms for all ten datasets that are involved for study and are ranked in descending order, based on their average accuracy. Five algorithms used for study are IBk, Decision Stump, Decsion Table, LibSVM and ZeroR across six dimensions i. e. Fixed Split, Cross Validation, Platt Scaling, Additive Regression, Logit Boost and Isotonic Regression. Results indicate the better performances through Logit Boost calibration, followed by Additive Regression. Isotonic Regression has degraded the performances of the algorithms. IBk and Decision Table has topped the chart with calibration and individually. Additive Regression has enhanced the performance of ZeroR algorithm and has uplifted its performance significantly.

6.2.2 Performance by metrics: Table 6 includes averages of fifteen metrics involved in study and are positioned in descending order according to average accuracy. For few metrics output was generated to be NaN (**N**ot **a** **N**umber) and Infinity, for such metrics averages were calculated over remaining values, excluding the count of such values. These values are pointed with an asterisk (*) for NaN, a plus sign (+) for Infinity and if all the values (for all ten problems) are NaN or Infinity, such values are represented by NaN* or Inf+.

6.3 Graphical Comparison: A graphical comparison involving ROC and Precision/Recall graphs of algorithms is prepared for Cross Validated experiments on Adult dataset.

6.3.1 ROC Curves: An ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC graphs are two-dimensional graphs in which True Positive rate is plotted on the Y axis and False Positive rate is plotted on the X axis. An ROC graph is compared on the basis of behavior of the curve in graph. A curve sharply rising towards Y axis is considered to be better than the diagonal or a curve sharply bending towards X axis. Clearly, in figure 7 better performing algorithms like random forests, boosted decision stumps etc. have their curves rising towards Y-axis marking better performance for them, SMO Decision Stump etc. are rising diagonally indicate average performance.

6.3.2 Precision/Recall Curves: Precision is the ratio of True Positives to the Sum of True Positives and False Positives, Recall is the ratio of True Positives to the Sum of True Positives and False Positives. A Precision/Recall curve bending towards origin is considered to be worst performances, whereas a curve rising away from origin towards 1 for X and Y axis collectively, is considered to be better performances. Clearly, algorithms like Random Forests, ADTrees etc. are rising away from origin indicate better performances, whereas diagonal curves for algorithms like SMO, ID3 etc. indicate average performances. These graphs are indicating the scenario of Adult problem in figure 8, curves can dramatically change for other problems depending upon their results.

Algorithm	Validation/ Calibration	Adult	CovType	Celegans	Droso	DS1_100	Letter	PenDigits	Satellite	Shuttle	Tic2000	Average
Random-Forest-Bagging	Fixed Split	0.8436	0.9804	0.9421	0.9832	0.9796	0.9632	0.9938	0.9485	0.9998	0.9287	0.9563
Random-Forest-Boosting	Fixed Split	0.8343	0.9788	0.9452	0.9832	0.9799	0.9671	0.9902	0.9517	0.9997	0.9243	0.9554
Random-Forest	Fixed Split	0.8328	0.9774	0.9446	0.9832	0.9788	0.9519	0.9903	0.9412	0.9997	0.9247	0.9525
J48	Fixed Split	0.8533	0.9789	0.9589	0.9832	0.9751	0.9183	0.9718	0.9191	0.9992	0.9384	0.9496
PART	Fixed Split	0.8452	0.9758	0.9579	0.9760	0.9785	0.9192	0.9813	0.9301	0.9997	0.9081	0.9472
MultiLayerPerceptron	Fixed Split	0.8064	0.9755	0.9775	0.9856	0.9786	0.8906	0.9887	0.9360	0.9986	0.9231	0.9460
IB-k	Fixed Split	0.7889	0.9789	0.9304	0.9765	0.9727	0.9715	0.9968	0.9426	0.9991	0.9007	0.9458
REPTree	Fixed Split	0.8409	0.9741	0.9579	0.9830	0.9775	0.8901	0.9631	0.9185	0.9982	0.9405	0.9444
ADTree-Boosting	Fixed Split	0.8509	0.9691	0.9617	0.9810	0.9776	0.8415	0.9786	0.9275	0.9998	0.9399	0.9428
Random-Forest-Bagging	Platt	0.8463	0.9823	0.9420	0.9839	0.9776	0.9597	0.9830	0.8210	0.9994	0.9305	0.9426
Random-Forest-Bagging	Cross Val	0.8463	0.9822	0.9420	0.9839	0.9774	0.9594	0.9823	0.8221	0.9995	0.9305	0.9426
Random-Forest-Boosting	Cross Val	0.8209	0.9836	0.9447	0.9839	0.9754	0.9613	0.9791	0.8282	0.9994	0.9270	0.9403
Random-Forest	Cross Val	0.8449	0.9803	0.9431	0.9839	0.9774	0.9452	0.9783	0.8169	0.9996	0.9274	0.9397
Random-Forest	Platt	0.8446	0.9809	0.9437	0.9839	0.9797	0.9468	0.9821	0.8070	0.9995	0.9195	0.9388
Random-Forest-Boosting	Platt	0.8209	0.9835	0.9447	0.9839	0.9789	0.9613	0.9815	0.8282	0.9994	0.8946	0.9377
MultiLayerPerceptron	Cross Val	0.8170	0.9760	0.9775	0.9856	0.9740	0.8847	0.9825	0.8489	0.9981	0.9324	0.9377
J48	Cross Val	0.8525	0.9807	0.9583	0.9839	0.9724	0.9141	0.9548	0.7991	0.9989	0.9382	0.9353
J48	Platt	0.8525	0.9807	0.9582	0.9839	0.9724	0.9141	0.9548	0.7991	0.9989	0.9382	0.9353
Decision-Table	Fixed Split	0.8516	0.9777	0.9421	0.9832	0.9773	0.8487	0.9196	0.9019	0.9989	0.9405	0.9341
PART	Cross Val	0.8187	0.9742	0.9571	0.9787	0.9765	0.9113	0.9678	0.8378	0.9991	0.9123	0.9333
PART	Platt	0.8184	0.9742	0.9571	0.9787	0.9765	0.9113	0.9678	0.8378	0.9991	0.9123	0.9333
IB-k	Cross Val	0.7851	0.9761	0.9306	0.9761	0.9656	0.9706	0.9890	0.8256	0.9990	0.9058	0.9324
ADTree-Boosting	Cross Val	0.8550	0.9801	0.9637	0.9821	0.9752	0.8314	0.9676	0.8236	0.9994	0.9384	0.9316
REPTree	Cross Val	0.8371	0.9767	0.9569	0.9826	0.9721	0.8869	0.9556	0.8084	0.9990	0.9390	0.9314
REPTree	Platt	0.8361	0.9767	0.9569	0.9826	0.9721	0.8860	0.9556	0.8084	0.9990	0.9390	0.9312
MultiLayerPerceptron	Platt	0.8166	0.9760	0.9655	0.9220	0.9796	0.8847	0.9825	0.8489	0.9980	0.9367	0.9311
ADTree-Bagging	Fixed Split	0.8515	0.9678	0.9634	0.9832	0.9789	0.7568	0.9307	0.9319	0.9996	0.9405	0.9304
IB-k	Platt	0.7851	0.9761	0.9161	0.9691	0.9651	0.9703	0.9890	0.8242	0.9990	0.8946	0.9289
ADTree	Fixed Split	0.8517	0.9678	0.9581	0.9832	0.9788	0.7404	0.8900	0.8961	0.9997	0.9405	0.9206
ADTree-Bagging	Cross Val	0.8538	0.9856	0.9637	0.9842	0.9742	0.7680	0.8979	0.8215	0.9996	0.9390	0.9187
ADTree-Bagging	Platt	0.8526	0.9848	0.9546	0.9839	0.9754	0.7614	0.8975	0.8358	0.9985	0.9390	0.9184
ADTree-Boosting	Platt	0.8535	0.9801	0.9235	0.9063	0.9490	0.8324	0.9658	0.8306	0.9994	0.9384	0.9179
Decision-Table	Cross Val	0.8515	0.9379	0.9420	0.9839	0.9795	0.8318	0.9087	0.8041	0.9982	0.9386	0.9176
Decision-Table	Platt	0.8518	0.9382	0.9420	0.9839	0.9795	0.8275	0.9062	0.8073	0.9982	0.9386	0.9173
SimpleLogistic	Fixed Split	0.8503	0.9733	0.9772	0.9853	0.9808	0.7321	0.8418	0.9258	0.9585	0.9405	0.9166
SMO	Fixed Split	0.8470	0.9725	0.9682	0.9800	0.9805	0.7358	0.8460	0.9269	0.9564	0.9405	0.9154
ADTree	Cross Val	0.8522	0.9848	0.9601	0.9836	0.9702	0.7568	0.8773	0.8026	0.9996	0.9392	0.9126
ADTree	Platt	0.8507	0.9848	0.9601	0.9839	0.9713	0.7416	0.8730	0.8082	0.9985	0.9390	0.9111
Decision-Stump-Boosting	Fixed Split	0.8420	0.9556	0.9567	0.9832	0.9713	0.6992	0.8518	0.9004	0.9980	0.9405	0.9098
SimpleLogistic	Cross Val	0.8491	0.9819	0.9745	0.9850	0.9824	0.7237	0.8289	0.8370	0.9594	0.9388	0.9061
ID3	Fixed Split	0.7967	0.9741	0.9411	0.9712	0.9637	0.8343	0.7937	0.8719	0.9990	0.9054	0.9051
BayesNetGenerator	Platt	0.8534	0.9789	0.9459	0.9464	0.9796	0.7705	0.8211	0.7907	0.9937	0.9370	0.9017
Decision-Stump-Boosting	Cross Val	0.8421	0.9781	0.9578	0.9831	0.9582	0.6962	0.8408	0.8122	0.9973	0.9390	0.9005
BayesNetGenerator	Cross Val	0.8515	0.9811	0.9781	0.9772	0.9786	0.7605	0.8233	0.7907	0.9931	0.8675	0.9002
BayesNetGenerator	Fixed Split	0.8308	0.9385	0.9782	0.9780	0.9765	0.7703	0.8296	0.8725	0.9918	0.8345	0.9001
SMO	Platt	0.8038	0.9334	0.9686	0.9814	0.9830	0.7299	0.8306	0.8565	0.9469	0.9390	0.8973
Decision-Stump-Boosting	Platt	0.8417	0.9781	0.9372	0.9704	0.9581	0.6962	0.8358	0.8148	0.9984	0.9390	0.8970
SimpleLogistic	Platt	0.8269	0.9792	0.9379	0.9577	0.9775	0.7243	0.8263	0.8151	0.9182	0.9380	0.8901
SMO	Cross Val	0.8038	0.9334	0.9686	0.9814	0.9830	0.7299	0.8306	0.8565	0.8565	0.9390	0.8883
Decision-Stump-Bagging	Fixed Split	0.7608	0.9220	0.9421	0.9832	0.9699	0.6712	0.7176	0.8789	0.9266	0.9405	0.8713
Decision-Stump	Fixed Split	0.7608	0.9220	0.9421	0.9832	0.9699	0.6712	0.7101	0.8789	0.9266	0.9405	0.8705
Naïve-Bayes-Simple	Fixed Split	0.8332	0.9382	0.9783	0.9775	0.9430	0.7157	0.7762	0.8771	0.8956	0.7553	0.8690
Decision-Stump	Cross Val	0.7596	0.9524	0.9420	0.9839	0.9581	0.6678	0.7063	0.8230	0.9279	0.9390	0.8660
Decision-Stump	Platt	0.7596	0.9524	0.9420	0.9839	0.9581	0.6678	0.7063	0.8230	0.9279	0.9390	0.8660
Decision-Stump-Bagging	Platt	0.7596	0.9524	0.9420	0.9839	0.9572	0.6678	0.7063	0.8230	0.9279	0.9390	0.8659
Decision-Stump-Bagging	Cross Val	0.7596	0.9524	0.9420	0.9839	0.9581	0.6678	0.7063	0.8105	0.9279	0.9390	0.8647
Naïve-Bayes-Simple	Cross Val	0.8273	0.9811	0.9782	0.9767	0.9540	0.7040	0.7547	0.8160	0.8974	0.7553	0.8645
ID3	Platt	0.7864	0.9770	0.9439	0.9659	0.9475	0.6435	0.8059	0.8082	0.8279	0.9231	0.8629
Naïve-Bayes-Simple	Platt	0.8273	0.9788	0.9451	0.9449	0.9713	0.7040	0.7552	0.8160	0.8974	0.7721	0.8612
LibSVM	Fixed Split	0.7609	0.9671	0.9421	0.9832	0.9815	0.9713	0.5045	0.5525	0.9482	0.9399	0.8551
ID3	Cross Val	0.7866	0.9774	0.9439	0.9659	0.9476	0.5029	0.8059	0.8082	0.8279	0.9204	0.8487
LibSVM	Platt	0.7597	0.9809	0.9420	0.9839	0.9831	0.9717	0.5113	0.3584	0.9427	0.9390	0.8373
LibSVM	Cross Val	0.7597	0.9809	0.9420	0.9839	0.9831	0.9717	0.5113	0.3584	0.3584	0.9390	0.7788
ZeroR	Fixed Split	0.7608	0.8286	0.9421	0.9832	0.9699	0.5013	0.5045	0.5525	0.7887	0.9405	0.7772
ZeroR	Cross Val	0.7596	0.0765	0.9420	0.9839	0.9713	0.4971	0.5113	0.3584	0.7885	0.9390	0.6828
ZeroR	Platt	0.7596	0.0765	0.9420	0.9839	0.9713	0.4971	0.5113	0.3584	0.7885	0.9390	0.6828

Table 3: Accuracy of all algorithms over ten problems and their mean performances in descending order

Algorithm	Val/Cali	Abs_Err	Rel_Err	RMSE	Sqr_Err	Corr.	Pre_Avg	AUC	Margin	Kappa	Preci.	Recall	LIFT	Fallout	F_Mea.	Acc.
Random-Forest-Bagging	FixedSplit	0.078	0.078	0.166	0.034	0.738*	0.202	0.934	0.033	0.583	0.836*	0.578	6.655*	0.016	0.758*	0.956
Random-Forest-Boosting	FixedSplit	0.047	0.047	0.183	0.043	0.682*	0.202	0.896	0.020	0.595	0.843*	0.587	7.589*	0.016	0.689*	0.955
Random-Forest	FixedSplit	0.076	0.076	0.174	0.038	0.667*	0.200	0.892	0.000	0.581	0.834*	0.574	7.506*	0.018	0.676*	0.952
J48	FixedSplit	0.069	0.069	0.192	0.044	0.688*	0.206	0.824	0.001	0.616	0.768*	0.623	5.964*	0.026	0.721*	0.950
PART	FixedSplit	0.062	0.062	0.194	0.046	0.655	0.212	0.851	0.000	0.652	0.730	0.661	7.239	0.029	0.688	0.947
MultiLayerPerceptron	FixedSplit	0.057	0.057	0.187	0.046	0.686	0.197	0.927	0.000	0.676	0.784	0.675	9.462	0.022	0.709	0.946
IB-k	FixedSplit	0.056	0.056	0.199	0.054	0.613	0.219	0.819	0.000	0.612	0.658	0.641	5.350	0.033	0.647	0.946
REPTree	FixedSplit	0.080	0.080	0.197	0.046	0.685*	0.203	0.878	0.006	0.603	0.820*	0.608	9.243*	0.028	0.711*	0.944
ADTree-Boosting	FixedSplit	0.077	0.077	0.186	0.043	0.645	0.217	0.931	0.002	0.643	0.712	0.667	7.755	0.041	0.760*	0.943
Random-Forest-Bagging	Platt	0.095	0.095	0.191	0.046	0.684*	0.190	0.922	0.021	0.533	0.804*	0.550	7.732*	0.034	0.708*	0.943
Random-Forest-Bagging	Cross Val	0.094	0.094	0.188	0.045	0.683*	0.190	0.922	0.018	0.532	0.804*	0.548	7.722*	0.033	0.706*	0.943
Random-Forest-Boosting	Cross Val	0.063	0.063	0.209	0.058	0.613*	0.182	0.879	0.000	0.519	0.831*	0.527	8.732*	0.029	0.613*	0.940
Random-Forest	Cross Val	0.091	0.091	0.194	0.048	0.617*	0.189	0.874	0.000	0.535	0.797*	0.551	8.157*	0.035	0.635*	0.940
Random-Forest	Platt	0.091	0.091	0.194	0.048	0.569	0.198	0.874	0.000	0.550	0.702	0.573	7.147	0.042	0.655*	0.939
Random-Forest-Boosting	Platt	0.069	0.069	0.211	0.060	0.633*	0.188	0.879	0.001	0.545	0.812*	0.558	8.148*	0.033	0.646*	0.938
MultiLayerPerceptron	Cross Val	0.067	0.067	0.205	0.055	0.655	0.193	0.921	0.000	0.651	0.713	0.688	9.310	0.039	0.692	0.938
J48	Cross Val	0.082	0.082	0.215	0.058	0.653*	0.199	0.809	0.002	0.583	0.733*	0.623	6.507*	0.046	0.697*	0.935
J48	Platt	0.082	0.082	0.215	0.058	0.653*	0.199	0.809	0.002	0.583	0.732*	0.624	6.496*	0.046	0.697*	0.935
Decision-Table	FixedSplit	0.091	0.091	0.215	0.056	0.760*	0.199	0.784	0.013	0.524	0.892*	0.541	6.731*	0.038	0.810*	0.934
PART	Cross Val	0.074	0.074	0.219	0.061	0.611	0.184	0.819	0.000	0.600	0.713	0.615	7.903	0.034	0.642	0.933
PART	Platt	0.074	0.074	0.219	0.061	0.611	0.184	0.819	0.000	0.600	0.713	0.615	7.900	0.034	0.642	0.933
IB-k	Cross Val	0.070	0.070	0.225	0.068	0.575	0.212	0.804	0.000	0.572	0.612	0.632	5.612	0.052	0.617	0.932
ADTree-Boosting	Cross Val	0.088	0.088	0.205	0.054	0.608	0.197	0.915	0.003	0.600	0.717	0.627	8.891	0.049	0.648	0.932
REPTree	Cross Val	0.090	0.090	0.214	0.057	0.658*	0.201	0.866	0.006	0.587	0.729*	0.631	8.442*	0.051	0.705*	0.931
REPTree	Platt	0.090	0.090	0.214	0.057	0.659*	0.202	0.866	0.006	0.588	0.728*	0.633	8.434*	0.052	0.706*	0.931
MultiLayerPerceptron	Platt	0.099	0.099	0.219	0.057	0.640	0.201	0.921	0.001	0.624	0.670	0.736	6.983	0.046	0.666	0.931
ADTree-Bagging	FixedSplit	0.155	0.155	0.216	0.058	0.735*	0.212	0.928	0.043	0.578	0.868*	0.604	7.704*	0.054	0.789*	0.930
IB-k	Platt	0.091	0.091	0.229	0.068	0.580	0.217	0.805	0.015	0.577	0.604	0.651	5.415	0.056	0.625	0.929
ADTree	FixedSplit	0.160	0.160	0.226	0.063	0.703*	0.197	0.909	0.033	0.554	0.854*	0.574	7.540*	0.047	0.766*	0.921
ADTree-Bagging	Cross Val	0.164	0.164	0.233	0.067	0.629*	0.212	0.915	0.027	0.549	0.769*	0.611	10.975*	0.078	0.681*	0.919
ADTree-Bagging	Platt	0.171	0.171	0.238	0.067	0.636*	0.212	0.915	0.041	0.551	0.768*	0.624	10.670*	0.078	0.683*	0.918
ADTree-Boosting	Platt	0.138	0.138	0.232	0.063	0.591	0.216	0.915	0.019	0.572	0.625	0.720	5.406	0.066	0.625	0.918
Decision-Table	Cross Val	0.109	0.109	0.243	0.071	0.616*	0.193	0.786	0.008	0.480	0.739*	0.542	6.739*	0.057	0.669*	0.918
Decision-Table	Platt	0.109	0.109	0.243	0.071	0.616*	0.197	0.786	0.008	0.480	0.736*	0.546	6.733*	0.061	0.670*	0.917
SimpleLogistic	FixedSplit	0.163	0.163	0.251	0.080	0.682*	0.203	0.894	0.051	0.605	0.828*	0.627	11.102*	0.056	0.745*	0.917
SMO	FixedSplit	0.085	0.085	0.265	0.085	0.674*	0.203	0.791	0.000	0.601	0.794*	0.639	9.390*	0.057	0.740*	0.915
ADTree	Cross Val	0.166	0.166	0.240	0.070	0.555	0.207	0.898	0.016	0.541	0.716	0.604	9.097	0.077	0.610	0.913
ADTree	Platt	0.179	0.179	0.247	0.073	0.704*	0.218	0.898	0.036	0.486	0.791*	0.571	5.199*	0.092	0.794*	0.911
Decision-Stump-Boosting	FixedSplit	0.134	0.134	0.230	0.065	0.627*	0.182	0.891	0.015	0.484	0.858*	0.507	7.824*	0.042	0.683*	0.910
SimpleLogistic	Cross Val	0.133	0.133	0.237	0.068	0.583	0.194	0.904	0.002	0.572	0.741	0.615	10.959	0.072	0.646	0.906
ID3	FixedSplit	0.148	0.148	0.342	0.147	0.527	0.188	0.758	0.000	0.522	0.636	0.557	4.928	0.044	0.590	0.905
BayesNetGenerator	Platt	0.154	0.154	0.255	0.078	0.587	0.208	0.911	0.009	0.561	0.686	0.691	8.212	0.082	0.635	0.902
Decision-Stump-Boosting	Cross Val	0.142	0.142	0.245	0.075	0.578*	0.180	0.884	0.010	0.511	0.702*	0.561	8.087*	0.062	0.647*	0.900
BayesNetGenerator	Cross Val	0.113	0.113	0.249	0.080	0.621	0.215	0.911	0.000	0.615	0.690	0.715	8.804	0.087	0.693	0.900
BayesNetGenerator	FixedSplit	0.112	0.112	0.251	0.076	0.630	0.239	0.919	0.000	0.625	0.685	0.740	7.883	0.087	0.702	0.900
SMO	Platt	0.108	0.108	0.293	0.102	0.609*	0.184	0.767	0.006	0.535	0.736*	0.603	9.765*	0.069	0.674*	0.897
Decision-Stump-Boosting	Platt	0.182	0.182	0.263	0.079	0.587*	0.190	0.884	0.030	0.519	0.655*	0.623	6.355*	0.069	0.658*	0.897
SimpleLogistic	Platt	0.188	0.188	0.266	0.079	0.546	0.192	0.904	0.020	0.523	0.655	0.644	7.325	0.079	0.602	0.890
SMO	Cross Val	0.112	0.112	0.308	0.112	0.594*	0.207	0.763	0.000	0.521	0.717*	0.610	9.509*	0.083	0.668*	0.888
Decision-Stump-Bagging	FixedSplit	0.188	0.188	0.282	0.091	0.522*	0.179	0.810	0.074	0.309	0.756*	0.402	2.503*	0.073	0.771*	0.871
Decision-Stump	FixedSplit	0.187	0.187	0.285	0.094	0.623*	0.175	0.774	0.082	0.308	0.759*	0.397	2.508*	0.070	0.767*	0.871
Naïve-Bayes-Simple	FixedSplit	0.143	0.143	0.303	0.105	0.557	0.238	0.900	0.000	0.545	0.629	0.702	6.546	0.108	0.640	0.869
Decision-Stump	Cross Val	0.211	0.211	0.305	0.106	0.516*	0.161	0.742	0.073	0.300	0.697*	0.376	5.283*	0.078	0.633*	0.866
Decision-Stump	Platt	0.225	0.225	0.308	0.107	0.516*	0.161	0.742	0.099	0.300	0.697*	0.376	5.283*	0.078	0.633*	0.866
Decision-Stump-Bagging	Platt	0.225	0.225	0.307	0.106	0.516*	0.161	0.756	0.098	0.300	0.696*	0.376	5.240*	0.078	0.633*	0.866
Decision-Stump-Bagging	Cross Val	0.212	0.212	0.304	0.105	0.514*	0.164	0.756	0.075	0.298	0.692*	0.379	5.268*	0.081	0.632*	0.865
Naïve-Bayes-Simple	Cross Val	0.145	0.145	0.303	0.110	0.540	0.219	0.890	0.000	0.527	0.631	0.674	7.458	0.114	0.626	0.864
ID3	Platt	0.137	0.137	0.342	0.137	0.398	0.123	0.683	0.000	0.372	0.593	0.404	4.895	0.038	0.450	0.863
Naïve-Bayes-Simple	Platt	0.183	0.183	0.305	0.106	0.482	0.218	0.890	0.008	0.456	0.657	0.641	8.310	0.113	0.555	0.861
LibSVM	FixedSplit	0.145	0.145	0.319	0.145	0.474*	0.084	0.652	0.000	0.325	0.761*	0.308	8.570*	0.004	0.557*	0.855
ID3	Cross Val	0.337	0.337	0.518	0.337	0.401*	0.104	0.669	0.000	0.343	0.559*	0.370	5.225*	0.034	0.447*	0.849
LibSVM	Platt	0.163	0.163	0.328	0.163	0.661*	0.175	0.654	0.000	0.326	0.770*	0.413	8.217*	0.104	0.646*	0.837
LibSVM	Cross Val	0.221	0.221	0.384	0.221	0.620*	0.259	0.618	0.000	0.246	0.663*	0.440	7.596*	0.204	0.593*	0.779
ZeroR	FixedSplit	0.279	0.279	0.348	0.139	NaN*	0.100	0.500	0.221	0.000	0.501*	0.100	1*	0.100	0.668*	0.777
ZeroR	Cross Val	0.307	0.307	0.366	0.157	NaN*	0.300	0.500	0.250	0.000	0.311*	0.300	1*	0.300	0.445*	0.683
ZeroR	Platt	0.307	0.307	0.366	0.157	NaN*	0.300	0.500	0.250	0.000	0.311*	0.300	1*	0.300	0.445*	0.683

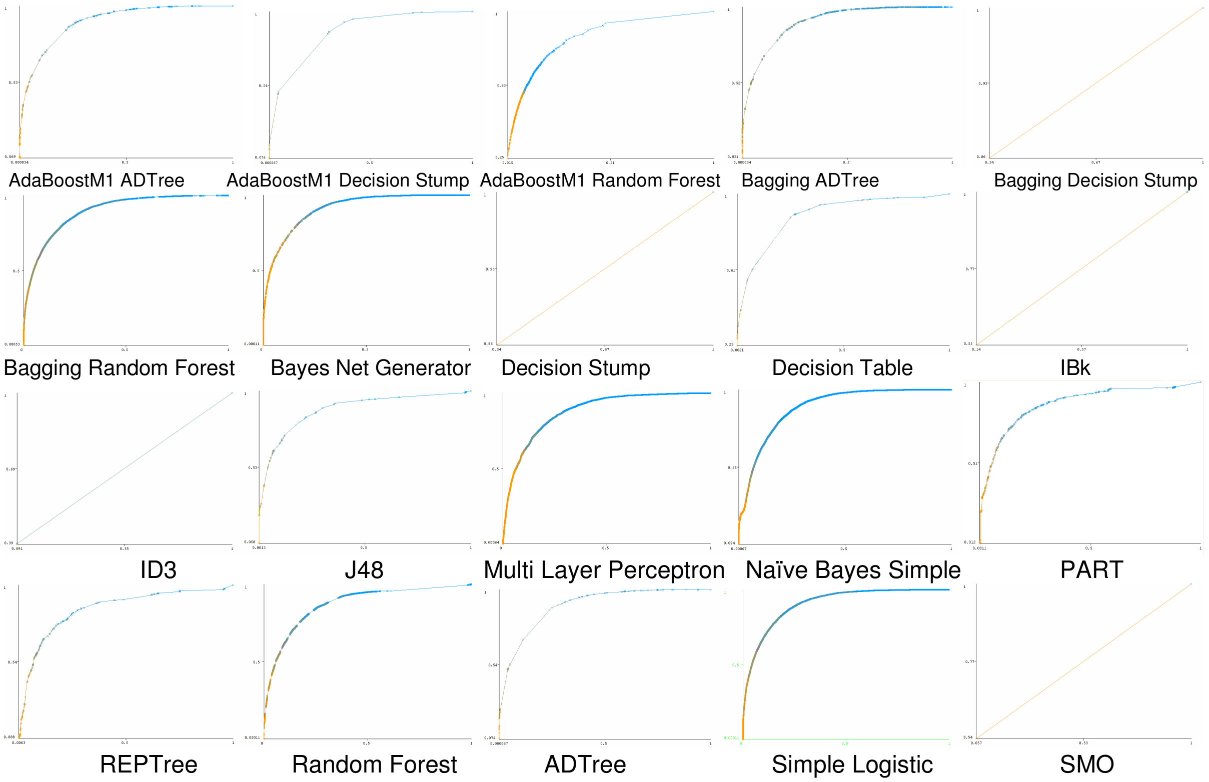
Table 4: Average performances for each learning algorithm by metric (average over ten problems)

Algorithm	Val./Cal.	Adult	CovType	Celegans	Droso	DS1_100	Letter	PenDigits	Satellite	Shuttle	Tic2000	Average
IB-k	FixedSplit	0.7889	0.9789	0.9304	0.9765	0.9727	0.9715	0.9968	0.9426	0.9991	0.9007	0.9458
Decision Table	LogitBoost	0.8449	0.9797	0.9670	0.9837	0.9762	0.9156	0.9594	0.8032	0.9798	0.9349	0.9344
Decision-Table	FixedSplit	0.8516	0.9777	0.9421	0.9832	0.9773	0.8487	0.9196	0.9019	0.9989	0.9405	0.9341
Ibk	LogitBoost	0.7833	0.9765	0.9304	0.9765	0.9672	0.9717	0.9898	0.8294	0.9992	0.9054	0.9330
Ibk	AddReg	0.7834	0.9673	0.9383	0.9813	0.9690	0.9710	0.9908	0.8250	0.9992	0.9036	0.9329
IB-k	CrossVal	0.7851	0.9761	0.9306	0.9761	0.9656	0.9706	0.9890	0.8256	0.9990	0.9058	0.9324
IB-k	PlattScaling	0.7851	0.9761	0.9161	0.9691	0.9651	0.9703	0.9890	0.8242	0.9990	0.8946	0.9289
Decision-Table	CrossVal	0.8515	0.9379	0.9420	0.9839	0.9795	0.8318	0.9087	0.8041	0.9982	0.9386	0.9176
Decision-Table	PlattScaling	0.8518	0.9382	0.9420	0.9839	0.9795	0.8275	0.9062	0.8073	0.9982	0.9386	0.9173
Decision Stump	LogitBoost	0.8519	0.9784	0.9567	0.9841	0.9700	0.7383	0.8650	0.8125	0.9984	0.9390	0.9094
Decision-Stump	FixedSplit	0.7608	0.9220	0.9421	0.9832	0.9699	0.6712	0.7101	0.8789	0.9266	0.9405	0.8705
Decision-Stump	CrossVal	0.7596	0.9524	0.9420	0.9839	0.9581	0.6678	0.7063	0.8230	0.9279	0.9390	0.8660
Decision-Stump	PlattScaling	0.7596	0.9524	0.9420	0.9839	0.9581	0.6678	0.7063	0.8230	0.9279	0.9390	0.8660
Decision Table	AddReg	0.7802	0.9437	0.9487	0.9839	0.9704	0.6745	0.7934	0.6885	0.9276	0.9382	0.8649
LibSVM	FixedSplit	0.7609	0.9671	0.9421	0.9832	0.9815	0.9713	0.5045	0.5525	0.9482	0.9399	0.8551
LibSVM	PlattScaling	0.7597	0.9809	0.9420	0.9839	0.9831	0.9717	0.5113	0.3584	0.9427	0.9390	0.8373
LibSVM	LogitBoost	0.7595	0.9743	0.9432	0.9832	0.9813	0.9772	0.5117	0.3584	0.9490	0.9224	0.8360
Decision Stump	AddReg	0.7888	0.9249	0.9420	0.9839	0.9713	0.5215	0.6225	0.8277	0.7885	0.9390	0.8310
LibSVM	AddReg	0.7596	0.9235	0.9420	0.9839	0.9713	0.5029	0.5113	0.6416	0.7885	0.9390	0.7964
ZeroR	AddReg	0.7596	0.9235	0.9420	0.9839	0.9713	0.5029	0.5113	0.6416	0.7885	0.9390	0.7964
Decision Stump	IsoReg	0.7596	0.9235	0.9420	0.9839	0.9713	0.5029	0.5113	0.6416	0.7885	0.9390	0.7964
Decision Table	IsoReg	0.7596	0.9235	0.9420	0.9839	0.9713	0.5029	0.5113	0.6416	0.7885	0.9390	0.7964
Ibk	IsoReg	0.7596	0.9235	0.9420	0.9839	0.9713	0.5029	0.5113	0.6416	0.7885	0.9390	0.7964
LibSVM	IsoReg	0.7596	0.9235	0.9420	0.9839	0.9713	0.5029	0.5113	0.6416	0.7885	0.9390	0.7964
ZeroR	IsoReg	0.7596	0.9235	0.9420	0.9839	0.9713	0.5029	0.5113	0.6416	0.7885	0.9390	0.7964
LibSVM	CrossVal	0.7597	0.9809	0.9420	0.9839	0.9831	0.9717	0.5113	0.3584	0.3584	0.9390	0.7788
ZeroR	FixedSplit	0.7608	0.8286	0.9421	0.9832	0.9699	0.5013	0.5045	0.5525	0.7887	0.9405	0.7772
ZeroR	CrossVal	0.7596	0.0765	0.9420	0.9839	0.9713	0.4971	0.5113	0.3584	0.7885	0.9390	0.6828
ZeroR	PlattScaling	0.7596	0.0765	0.9420	0.9839	0.9713	0.4971	0.5113	0.3584	0.7885	0.9390	0.6828
ZeroR	LogitBoost	0.7596	0.0765	0.9420	0.9839	0.9713	0.4971	0.5113	0.3584	0.7885	0.9390	0.6828

Table 5: Accuracy of selected algorithms across Fixed Split, Cross Validation and all four types of Calibration methods over ten problems and their mean performances in descending order

Algorithm	Val/Cal.	Abs_Err	Rel_Err	RMSE	Sqr_Err	Corr.	Pre_Avg	AUC	Margin	Kappa	Preci.	Recall	LIFT	Fallout	F_Mea.	Acc.
IB-k	FixedSplit	0.056	0.056	0.199	0.054	0.613	0.219	0.819	0.000	0.612	0.658	0.641	5.350	0.033	0.647	0.9458
Decision Table	LogitBoost	0.089	0.089	0.205	0.050	0.622	0.208	0.917	0.002	0.604	0.755	0.621	9.641	0.047	0.649	0.9344
Decision-Table	FixedSplit	0.091	0.091	0.215	0.056	0.760*	0.199	0.784	0.013	0.524	0.892*	0.541	6.731*	0.038	0.810*	0.9341
Ibk	LogitBoost	0.068	0.068	0.223	0.067	0.577	0.211	0.800	0.000	0.574	0.615	0.631	5.682	0.051	0.619	0.9330
Ibk	AddReg	0.068	Inf $^{+}$	0.220	0.066	0.570	0.211	0.000	1.000	0.552	0.564	0.624	5.108	0.069	0.565	0.9329
IB-k	CrossVal	0.070	0.070	0.225	0.068	0.575	0.212	0.804	0.000	0.572	0.612	0.632	5.612	0.052	0.617	0.9324
IB-k	PlattScaling	0.091	0.091	0.229	0.068	0.580	0.217	0.805	0.015	0.577	0.604	0.651	5.415	0.056	0.625	0.9289
Decision-Table	CrossVal	0.109	0.109	0.243	0.071	0.616*	0.193	0.786	0.008	0.480	0.739*	0.542	6.739*	0.057	0.669*	0.9176
Decision-Table	PlattScaling	0.109	0.109	0.243	0.071	0.616*	0.197	0.786	0.008	0.480	0.736*	0.546	6.733*	0.061	0.670*	0.9173
Decision Stump	LogitBoost	0.138	0.138	0.238	0.071	0.534	0.187	0.893	0.005	0.516	0.730	0.562	10.203	0.062	0.583	0.9094
Decision-Stump	FixedSplit	0.187	0.187	0.285	0.094	0.623*	0.175	0.774	0.082	0.308	0.759*	0.397	2.508*	0.070	0.767*	0.8705
Decision-Stump	CrossVal	0.211	0.211	0.305	0.106	0.516*	0.161	0.742	0.073	0.300	0.697*	0.376	5.283*	0.078	0.633*	0.8660
Decision-Stump	PlattScaling	0.225	0.225	0.308	0.107	0.516*	0.161	0.742	0.099	0.300	0.697*	0.376	5.283*	0.078	0.633*	0.8660
Decision Table	AddReg	0.122	Inf $^{+}$	0.238	0.067	0.600	0.215	0.000	1.000	0.274	0.204	1.000	1.000	1.000	0.304	0.8649
LibSVM	FixedSplit	0.145	0.145	0.319	0.145	0.474*	0.084	0.652	0.000	0.325	0.761*	0.308	8.570*	0.004	0.557*	0.8551
LibSVM	PlattScaling	0.163	0.163	0.328	0.163	0.661*	0.175	0.654	0.000	0.326	0.770*	0.413	8.217*	0.104	0.646*	0.8373
LibSVM	LogitBoost	0.166	0.166	0.311	0.142	0.439*	0.179	0.695	0.024	0.336	0.698*	0.426	7.067*	0.107	0.445*	0.8360
Decision Stump	AddReg	0.167	Inf $^{+}$	0.260	0.079	0.542	0.223	0.000	1.000	0.108	0.204	1.000	1.000	1.000	0.304	0.8310
LibSVM	AddReg	0.485	Inf $^{+}$	0.526	0.341	1.38E-07*	0.078	0.000	1.000	0.000	0.204	1.000	1.000	1.000	0.304	0.7964
ZeroR	AddReg	0.485	Inf $^{+}$	0.526	0.341	1.38E-07*	0.078	0.000	1.000	0.000	0.204	1.000	1.000	1.000	0.304	0.7964
Decision Stump	IsoReg	0.263	Inf $^{+}$	0.366	0.157	1.02E-07*	0.271	0.000	1.000	0.000	0.204	1.000	1.000	1.000	0.304	0.7964
Decision Table	IsoReg	0.263	Inf $^{+}$	0.366	0.157	1.02E-07*	0.271	0.000	1.000	0.000	0.204	1.000	1.000	1.000	0.304	0.7964
Ibk	IsoReg	0.263	Inf $^{+}$	0.366	0.157	1.02E-07*	0.271	0.000	1.000	0.000	0.204	1.000	1.000	1.000	0.304	0.7964
LibSVM	IsoReg	0.263	Inf $^{+}$	0.366	0.157	1.02E-07*	0.271	0.000	1.000	0.000	0.204	1.000	1.000	1.000	0.304	0.7964
ZeroR	IsoReg	0.263	Inf $^{+}$	0.366	0.157	1.02E-07*	0.271	0.000	1.000	0.000	0.204	1.000	1.000	1.000	0.304	0.7964
LibSVM	CrossVal	0.221	0.221	0.384	0.221	0.620*	0.259	0.618	0.000	0.246	0.663*	0.440	7.596*	0.204	0.593*	0.7788
ZeroR	FixedSplit	0.279	0.279	0.348	0.139	NaN*	0.100	0.500	0.221	0.000	0.501*	0.100	1*	0.100	0.668*	0.7772
ZeroR	CrossVal	0.307	0.307	0.366	0.157	NaN*	0.300	0.500	0.250	0.000	0.311*	0.300	1*	0.300	0.445*	0.6828
ZeroR	PlattScaling	0.307	0.307	0.366	0.157	NaN*	0.300	0.500	0.250	0.000	0.311*	0.300	1*	0.300	0.445*	0.6828
ZeroR	LogitBoost	0.307	0.307	0.366	0.157	NaN*	0.300	0.500	0.250	0.000	0.311*	0.300	1*	0.300	0.445*	0.6828

Table 6: Average performances for selected learning algorithm by metric, across Fixed Split, Cross Validation and all four types of Calibration methods, (average over ten problems)



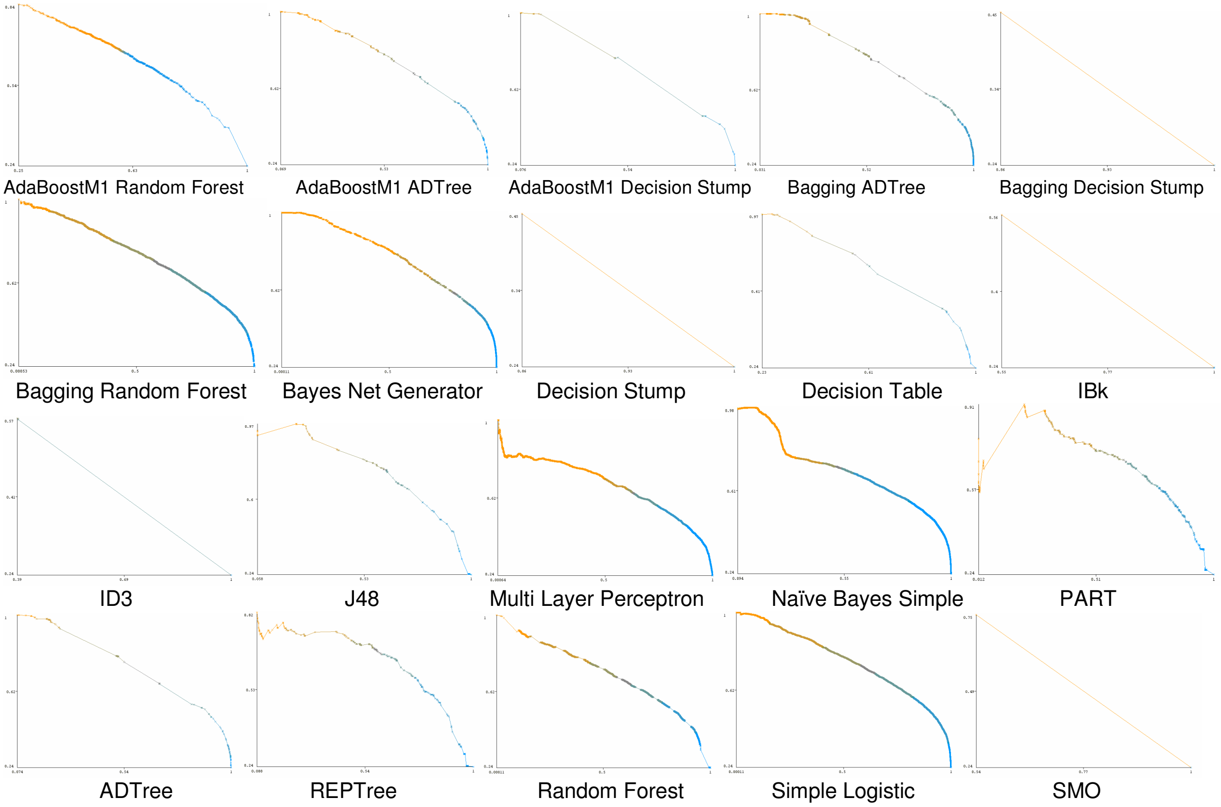
AdaBoostM1 ADTree AdaBoostM1 Decision Stump AdaBoostM1 Random Forest Bagging ADTree Bagging Decision Stump

Bagging Random Forest Bayes Net Generator Decision Stump Decision Table IBk

ID3 J48 Multi Layer Perceptron Naïve Bayes Simple PART

REPTree Random Forest ADTree Simple Logistic SMO

Figure 7: ROC graphs of twenty algorithms for Adult problem (X Axis-False Positive Rate, Y-Axis True Positive Rate).



AdaBoostM1 Random Forest AdaBoostM1 ADTree AdaBoostM1 Decision Stump Bagging ADTree Bagging Decision Stump

Bagging Random Forest Bayes Net Generator Decision Stump Decision Table IBk

ID3 J48 Multi Layer Perceptron Naïve Bayes Simple PART

ADTree REPTree Random Forest Simple Logistic SMO

Figure 8: Precision/Recall graphs of twenty algorithms for Adult problem (X Axis-Recall, Y Axis-Precision).

7. Comparison of results

Results of supervised learning techniques depend upon many things like type of dataset, number of instances in dataset, algorithm used for testing, process used for producing output etc. Data mining is a study of knowledge discovery in large datasets. First of all we present the comparison of different datasets based on average accuracy through different processes followed by the performance of different algorithm based on their average accuracy over three processes of major study. Figure 9 includes average performance of different problems from major study. Droso problem has performed best with an average accuracy of 97.78%, whereas Letter problem has performed worst with average performance with 80.57% average accuracy.

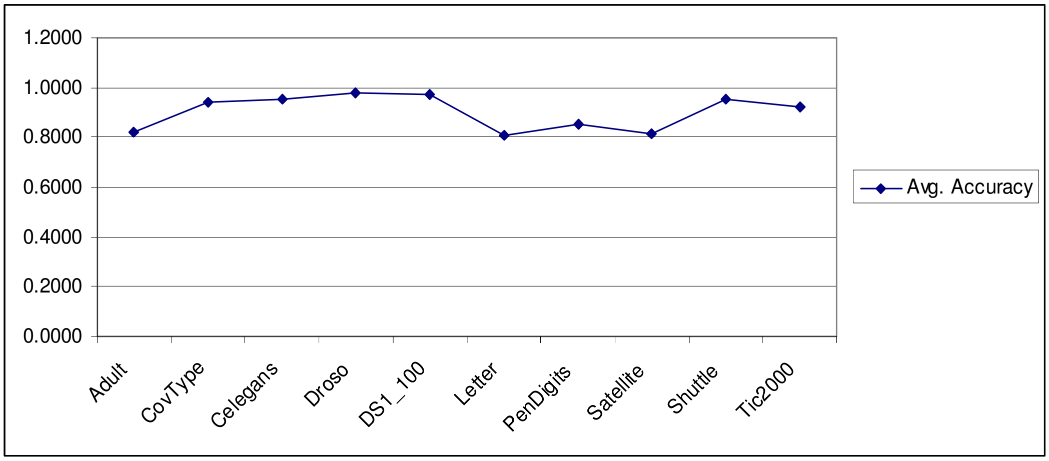


Figure 9: Average performance of different processes for different problems

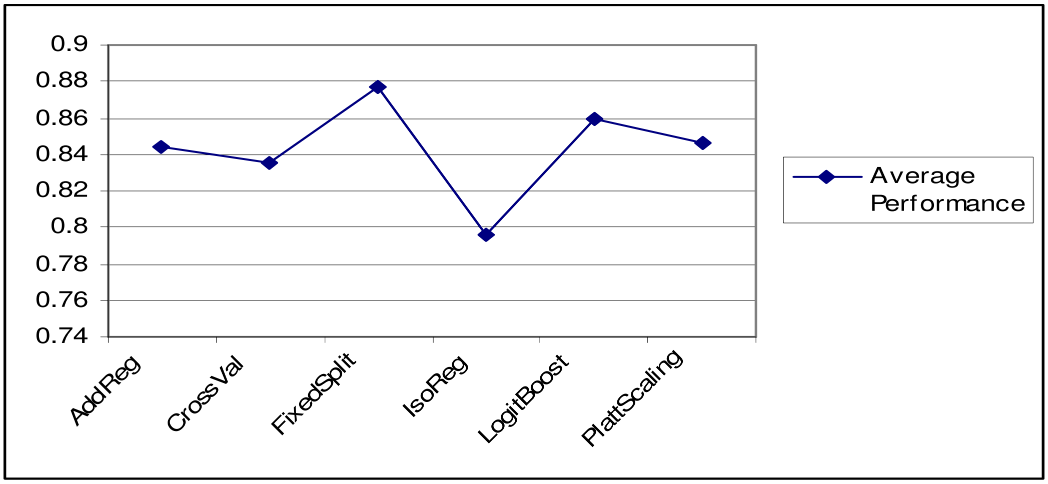


Figure 10: Average of average performances from five algorithms over all ten datasets included in minor study.

Figure 10 includes the average performance of all the processes from the average of five algorithms over all ten datasets included in minor study. Fixed split is highest performer, but its performance is over fitted, whereas post pruning through isotonic regression is least performer with 79.64% performance. Cross Validation and other three post pruning methods have pruned the models more appropriately. Among these four LogitBoost has performed best, whereas cross validation has performed least. Reason behind low performance of cross validation is exclusion of one tenth of training dataset while processing final model.

Other dimension of comparison includes two comprehensive studies that have been performed yet. First one is Statlog (King et al. [15]) and other is recent one (Caruana et al. [7]). One of the major differences between earlier two studies and current study, is about the selection of datasets i. e. earlier studies were mostly based on small datasets, whereas present study includes most of the datasets that are bigger in size and simple rule of probability states that increasing number of instances produces more accurate results and minimizes the chances of deviation. When Statlog study was conducted, algorithms like

Random Forest etc. were not being developed and data mining was in its initial phase of development. During last two decades data mining field has become mature enough. Statlog study presented the results for individual datasets. We compiled and processed the data for comparison and found that piecewise linear classifier DIPOL92 to be performing best for their tests, whereas Decision Tree was ranked second followed by the Back Propagation and kNN (k Nearest Neighbor) etc. Clearly, the absence of better algorithms like Random Forest at that time kept the high quality performances far away from current standards. Today, we have far better results than the results presented in Statlog.

Other recently conducted study (Caruana et al. [7]) presented the results that Boosted decision tree with platt scaling algorithm is the best performer, whereas Random Forest with platt scaling is the second best performer. Bagging and Boosting was not applied upon Random Forest. Experiments were performed through cross validation, Platt Scaling and Isotonic Regression. Top performers were Boosted Decision Tree, Random Forest, Bagged decision tree, SVM (Support Vector Machine), ANN (Artificial Neural Network) etc. Results of our study have marked Bagged Random Forest to be the best performer followed by J48, PART, Multi Layer Perceptron and IBk etc.

Finally results of present study are compared with the best known results ever claimed for problems included in study. For adult dataset best possible result is claimed for FSS Naïve Bayes in the description of datasets of UCI repositories (Blake et al. [5]) having 85.95% accuracy, where 32561 instances were used for training and 16281 instances for testing. Present study has used 9768 instances for training and 39074 instances for testing and best result is 85.50% for ADTree-Boosting algorithm with cross validation, which confirms our claim that training with twenty percent training instances for large datasets achieve significant maturity in results. For other problems as well results are up to the mark with best possible results ever being obtained.

8. Conclusion and Future Directions

Data mining has marked substantial progress in last two decades. Learning methods such as boosting, random forests, bagging and IBk etc. have achieved excellent performance that would have been difficult to obtain just fifteen years ago. Calibration with either Platt's method, Logit Boost, Additive Regression or Isotonic Regression is remarkably effective at obtaining excellent performance on the probability metrics from learning algorithms that performed well on the ordering metrics. Calibration dramatically improves the performance of Random Forests, ADTree, Decision stumps and Naive Bayes etc. and provides a noticeable improvement for random forests. With excellent performance over all fifteen metrics, calibrated Random Forest trees were the best learning algorithms overall. ADTree, IBk, J48 and MultiLayer Perceotron were quite close to it. Algorithm ZeroR has registered worst performance, but has registered a little improvement through Additive Regression based calibration. As the environmental factors like type of problems, size of dataset etc. may affect the performance of the algorithm, even better algorithms sometimes may result in bad results. Even after having a significant margin between best and worst performances, there exist chances for improvement. Authors will continue to work for the improvement of the processing environment of badly performing algorithms and for the improvement of the best algorithms as well as for the development of new algorithms for the field.

9. References

1. Atlas L., Connor J., and Park D. *"A performance comparison of trained multi-layer perceptrons and trained classification trees"*. In *Systems, man and cybernetics: proceedings of the 1989 IEEE international conference*, pages 915–920, Cambridge, Ma. Hyatt Regency, 1991

2. Ayer M., Brunk H., Ewing G., Reid W. & Silverman E. *"An empirical distribution function for sampling with incomplete information"*. Annals of Mathematical Statistics, 5, 641-647, 1955

3. Bauer E. and Kohavi R. *"An empirical comparison of voting classification algorithms: Bagging, boosting, and variants"*. Machine Learning, 36, 1999

4. Berry C. C. *"The kappa statistic". Journal of the American Medical Association, Linguistics (COLING-90),* volume 2, pages 251-256, 1992

5. Blake C. and Merz C., UCI repository of machine learning databases, 1998

6. Breiman L., Friedman J. H., Olshen R. A. and Stone C. J. *“Classification and Regression Trees”*. Wadsworth and Brooks, Monterey, CA., 1984

7. Caruana Rich and Niculescu-Mizil Alexandru. *“An Empirical Comparison of Supervised Learning Algorithms”.* Proceedings of the 23 rd International Conference on Machine Learning, Pittsburgh, PA, 2006

8. Cooper G. F., Aliferis C. F., Ambrosino R., Aronis J., Buchanan B. G., Caruana R., Fine M. J., Glymour C., Gordon G., Hanusa B. H., Janosky J. E., Meek C., Mitchell T., Richardson T. and Spirtes P. *“An evaluation of machine learning methods for predicting pneumonia mortality”.* Artificial Intelligence in Medicine, 9, 1997

9. Fahrmeir, L., Haussler, W., and Tutz, G. *“Diskriminanz analyse”.* In Fahrmeir, L. and Hamerle, A., editors, *Multivariate statistische Verfahren*. Verlag de Gruyter, Berlin, 1984

10. Fayyad U., Piatetsky-Shapiro G. and P. Smyth. *“The KDD process for extracting useful knowledge from volumes of data”.* CACM 39 (11), pp. 27-34, 1996

11. Friedman J., Hastie T. and Tibshirani R. *“Additive Logistic Regression: a Statistical View of Boosting”*. Stanford University,1998

12. Giudici P. *“Applied data mining”.* John Wiley and Sons. New York, 2003

13. Gorman R. P. and Sejnowski T. J. *“Analysis of hidden units in a layered network trained to classify sonar targets”*. Neural networks, 1 (Part 1):75–89, 1988

14. Hofmann H. J. *“Die anwendung des cart-verfahrens zur statistischen bonitatsanalyse von konsumentenkrediten”.* Zeitschrift fur Betriebswirtschaft, 60:941–962, 1990

15. King R., Feng C. and Shutherland A. *“Statlog: comparison of classi_cation algorithms on large real world problems”.* Applied Artificial Intelligence, 9, 1995

16. Kirkwood C., Andrews B. and Mowforth P. *“Automatic detection of gait events: a case study using inductive learning techniques”*. Journal of biomedical engineering, 11(23):511–516, 1989

17. Komarek P., Gray A., Liu T. and Moore A. *“High Dimensional Probabilistic Classification for Drug Discovery”*, Biostatics, COMPSTAT, 2004

18. LeCun Y., Jackel L. D., Bottou L., Brunot A., Cortes C., Denker J. S., Drucker H., Guyon I., Muller U. A., Sackinger E., Simard P. and Vapnik V. *“Comparison of learning algorithms for handwritten digit recognition”*. International Conference on Artificial Neural Networks (pp. 53{60).Paris, 1995

19. Lim T. S., Loh W.-Y. and Shih Y. S. *“A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms”*. Machine Learning, 40, 203-228, 2000

20. Mitchell T., Buchanan B., DeJon G., Dietterich T., Rosenbloom P. and Waibel A. *"Machine Learning".* Annual Review of Computer Science, vol. 4, pp. 417-433, 1990

21. Niculescu-Mizil A. and Caruana R. *“Predicting good probabilities with supervised learning”*. Proc. 22nd International Conference on Machine Learning (ICML'05), 2005

22. Nishisato S. *“Analysis of Categorical Data: Dual Scaling and its Applications”*. University of Toronto Press, Toronto, 1980

23. Perlich C., Provost F. and Simono J. S. *“Tree induction vs. logistic regression: a learning-curve analysis”.* J. Mach. Learn. Res., 4, 211-255, 2003

24. Platt J. *"Probabilistic outputs for support vector machines and comparison to regularized likelihood methods"*. Adv. in Large Margin Classifiers, 1999

25. Provost F. and Domingos P. *"Tree induction for probability-based rankings"*. Machine Learning, 2003

26. Provost Foster J. and Kohavi Ron, *"On Applied Research in Machine Learning"*. Machine Learning 30 (2-3): 127-132, 1998

27. Provost F., Jensen D. and Oates T. *"Efficient progressive sampling"*. Fifth ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining. San Diego, USA. 1999

28. Ripley B. *"Statistical aspects of neural networks"*. Chaos and Networks - Statistical and Probabilistic Aspects. Chapman and Hall, 1993

29. Robertson T., Wright F. and Dykstra R. *"Order restricted statistical inference"*. John Wiley and Sons, New York, 1988

30. Shadmehr R. and D'Argenio Z. *"A comparison of a neural network based estimator and two statistical estimators in a sparse and noisy environment"*. In *IJCNN-90: proceedings of the international joint conference on neural networks*, pages 289–292, Ann Arbor, MI. IEEE Neural Networks Council, 1990

31. Sonnenburg S, Rätsch G. and Schäfer C. *"Learning interpretable SVMs for biological sequence classification".* Research in Computational Molecular Biology, Springer Verlag, pages 389-407, 2005

32. Spikovska L. and Reid M. B., *"An empirical comparison of id3 and honns for distortion invariant object recognition*". In TAI-90: tools for artificial intelligence: proceedings of the 2nd international IEEE conference, Los Alamitos, CA. IEEE Computer Society Press, 1990

33. Witten I. H. and Frank E. *"Data Mining: Practical machine learning tools and techniques with java implementations"*. Morgan Kaufmann, 2000

34. Yoav Freund, Robert E. Schapire. *"Experiments with a new boosting algorithm"*. Thirteenth International Conference on Machine Learning, San Francisco, 148-156, 1996

35. Zadrozny B. and Elkan C. *"Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers"*. ICML, 2001

36. Zadrozny B. and Elkan C. *"Transforming classifier scores into accurate multi-class probability estimates"*. KDD, 2002

A Mode Switching Sliding-mode Controller with Observer-based State-Dependent Boundary Layer and Its Application

Liang-Chun Yao d917701@oz.nthu.edu.tw
Department of Power Mechanical Engineering
National Tsing-Hua University
HsinChu, 30013, Taiwan

Jian-Shiang Chen jschen@pme.nthu.edu.tw
Department of Power Mechanical Engineering
National Tsing-Hua University
HsinChu, 30013, Taiwan

Chao-Yu Hsu g913760@oz.nthu.edu.tw
Department of Power Mechanical Engineering
National Tsing-Hua University
HsinChu, 30013, Taiwan

Abstract

This paper presents a mode-switching sliding-mode control (MSMC) scheme that combines different sliding-mode control schemes to alleviate adverse effect while achieving precise control tasks. To achieve certain robustness and chattering alleviation, a design of disturbance observer based state-dependent boundary layer is proposed. The proposed method will provide a state-dependent boundary-layer in which the unknown dynamics is estimated a disturbance observer and then utilize it to calculate the width of boundary layer on-line. The convergent analysis of this state-dependent boundary-layer is provided with two theorems. Finally, its efficacy is further validated through experiments on the regulation control of a maglev platform.

Keywords: sliding mode, mode-switching, boundary layer, maglev platform.

1. INTRODUCTION

Sliding mode control (SMC), a high-speed switching feedback control methodology, has received much attention both in theory and applications for the last decades. Generally, one of its salient features is known to be its robustness against uncertainties both in system parameters and dynamics [1][2]. However, one of the major drawbacks of the sliding mode control is the adverse chatters while the controller input undergoes very fast activity of switching [3]. Nevertheless most SMC designs prove its efficacy in maintaining both stability and robust performance in counteracting modeling imprecision and external disturbances as well. The chatters can always be alleviated by the *so-called* boundary layer approach [4], in which the discontinuous control activity is replaced by a continuous control effort inside a preset boundary layer around the switching surface. However, its robust performance would inevitably be deteriorated with this augmented boundary-layer. In general, the layer thickness or width is either fixed or time invariant, tradeoffs between alleviated chatters and robust performance could highly depend on the choice of layer thickness. Physically speaking, the boundary layer can be approximated as a low-pass filter for a high frequency switching output, the cut-off frequency of this filter would be very difficult to be determined, however. In [5], the controller design adopted physical properties of a robot manipulator and a set of time-varying switching gains and boundary layer are incorporated in the sliding mode controller to accelerate the state trajectories moving toward the sliding hyper-plane, the design turned out to be much complicated and the system dynamics must be known as a *priori.*

A state-dependent boundary layer approach were proposed by [6][7], in which the thickness of boundary layer can be adjusted online based-on the state norm for a class of uncertain linear systems. On the other hand, aimed at nonlinear systems, a mode-switching control (MSC) scheme is usually adopted to improve the accuracy and robustness of controller design. In [8], Iwasaki, Sakai and Matsui applied the MSC in a two-degree-of -freedom position control system to achieve both fast response and high accuracy. And Takashi, Hidehiko and Hiromu [9] proposed the MSC with initial value compensation to determine the optimal switching conditions for the disk drives. In[10], different SMC schemes based on mode switching was reported, it is noted that difficulties were encountered in the compromising between compatibility and robustness while mode switching took place.

The SMC design provides an effective approach in maintaining stability and robust performance due to modeling imprecision. However, the performance of controller will be deteriorated with augmented boundary-layer approach, and the steady-state error will occur. The integral SMC can reduce the steady-state error and chattering as compared to boundary-layer approach, but the additional integral term could cause the actuator's windup. This paper is aimed to propose a mode-switching sliding-mode control (MSMC) scheme that combines different SMC schemes to alleviate adverse effect while achieving precise control task. Here, both compatibility and robustness are resolved by a disturbance observer based state-dependent boundary layer design incorporated with MSMC is proposed. This proposed scheme can estimate uncertainties by a disturbance observer and then utilize it to calculate the thickness/width of the boundary layer on-line. Finally, to demonstrate the efficacy and feasibility of the proposed method, a maglev platform is devised to validate the proposed schemes through experiment studies.

2. PROBLEM STATEMENT

Consider an uncertain SISO system with matched uncertainties [10] and described as

$$q^{(n)} = f(t,\boldsymbol{q}) + b(t,\boldsymbol{q})u + d \tag{1}$$

where q is the output, u is the control input, and $\boldsymbol{q} = [q \ \ \dot{q} \cdots q^{(n-1)}]^T$ is the corresponding state vector. *b(t,**q**)* is a non-zero function and of known sign as a *priori*. ***f(t,q)*** and *b(t,**q**)* are bounded uncertain functions, *d* is bounded disturbance and satisfies the following inequalities.

$$\left| f(t,q) - \bar{f}(t,q) \right| \le F; \ \left| b(t,q) - \bar{b}(t,q) \right| \le B \ ; \ \left| d \right| \le D \tag{2}$$

Where $\bar{f}(t,\boldsymbol{q})$ and $\bar{b}(t,\boldsymbol{q})$ are nominal functions. If we further assume that $\boldsymbol{q_d}$ is the command vector to be tracked, and its initial state vector is known as $\boldsymbol{q_d}\mathbf{(0)}=\boldsymbol{q}\mathbf{(0)}$. We can then define $\tilde{\boldsymbol{q}} = \boldsymbol{q} - \boldsymbol{q}_d$ to be the tracking error, thus $\tilde{\boldsymbol{q}} = \boldsymbol{q} - \boldsymbol{q}_d = [\tilde{q} \ \ \dot{\tilde{q}} \cdots \tilde{q}^{(n-1)}]^T$ being the tracking error vector.

Lemma 1 Slotine(1983)[5]:
A time-varying surface *s(t)* defined in $\mathbf{R}^{(n)}$ can be defined by a scalar equation $s(\tilde{q},t)$ =0 and shown as below

$$s(\tilde{q},t)=(\frac{d}{dt}+\lambda)^{n-1}\tilde{q} \tag{3}$$

In which *λ* is a positive constant. Furthermore, bounds on *s* can be directly related to bounds on the tracking error vector $\tilde{q}$, therefore $s(\tilde{q},t)$ represents a true measure of the tracking performance. Specifically, assuming $\tilde{q}$ (0)=**0**, the bound of tracking error can be written as

$$\left|\tilde{q}^{(i)}(t)\right|\le(2\lambda)^{i}\varepsilon \quad i=0,1,...,n-1;\ \forall t\ge 0 \tag{4}$$

where $\varepsilon=\Phi/\lambda^{n-1}$ is the boundary layer width.

Lemma 2 Slotine(1983)[5]:
To reduce chatters induced by imperfect switching, the discontinuous control can be approximated inside a boundary layer located around the switching surface, while Φ , the thickness of the boundary layer is state-dependent or time-varying and a filtered output of a pre-specified trajectory *k*($\boldsymbol{q_d}$) and sliding motion is asymptotically stable. And,

$$s\dot{s}\le\left(\dot{\Phi}-\eta\right)|s| \tag{5}$$

$$\dot{\Phi}+\lambda\Phi=k\left(\boldsymbol{q}_d\right) \tag{6}$$

Supposed that we wish to design a sliding-mode control system with at least two sliding surfaces and both accompany with state-dependent boundary layer, a Mode-Switching Control scheme is thus augmented to achieve fast transient response, less chatters with better robustness. Without the loss of generality, we consider the switching between two sliding-mode schemes, e.g. a sliding-mode control (SMC) scheme and an integral sliding-mode control (ISMC) scheme, they are depicted as follows.

$$\begin{cases} s=(\frac{d}{dt}+\lambda)^{n-1}\tilde{q} & |s|\ge\Phi \\ s_I=(\frac{d}{dt}+\lambda_I)^{n}\int_0^t\tilde{q}dt & |s_I|<\Phi \end{cases} \tag{7}$$

Here *s* and s_I are the sliding variables, $\tilde{q}=q-q_d$, q is the generalized coordinate, q_d is the desired output; Φ is the pre-specified layer thickness, while λ and λ_I are the corresponding eigenvalues of the sliding mode control and integral sliding mode control, respectively. From (3), for *n*=2 we will have the following estimates of the bounds ε and ε_I on errors,

$$\varepsilon=\frac{\Phi}{\lambda} \text{ and } \varepsilon_I=\frac{\Phi}{2\lambda_I} \tag{8}$$

If $2\lambda_I\le\lambda$, ε will be equal to or less than ε_I which implies that one cannot ensure ISMC will t switching back to SMC, while $2\lambda_I>\lambda$, ISMC will converge to $0<|\tilde{q}|<\varepsilon_I$ provided that no disturbance is encounter. Consequently, It is concluded that if $\lambda\ne2\lambda_I$, sliding variable will encounter compatibility problem while switching occurs between $|s|<\Phi$ and $|s_I|>\Phi$. Instead of switching based on the sliding variable, for *n*=2, the mode-switching condition needs to be modified as follows [9].

$$s=\begin{cases}\dot{\tilde{q}}+\lambda\tilde{q} \\ \dot{\tilde{q}}+2\lambda_I\tilde{q}+\lambda_I^2\int_{t_s}^t\tilde{q}dt\end{cases} \text{switching at } t_s \text{ and} |\tilde{q}|=\varepsilon \tag{9}$$

Here t_s is the pre-specified time of switching, $\varepsilon=\Phi/\lambda$ is the pre-specified layer width while switching occurs. The choice of thickness will affect the robustness As described by [7], a state-dependent boundary layer control is capable of ensuring effective chattering alleviation with state-dependent uncertainties but the adverse effect of external disturbances still exist, in other words, ineffective switching might occur on (9) without a better estimation on the bounds of disturbances.

3. A STATE-DEPENDNET BOUNDARY LAYER WITH DISTURBANCE OBSERVER

As shown in Fig. 1, consider the dynamic system as described in (1), we wish to track a desired command ***q_d*(*t*)**, a pole-placement design using feedback-linearization technique leads to the following control law[11].

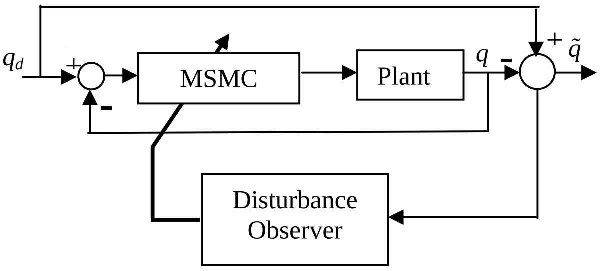


FIGURE 1: Block diagram of the proposed controller with disturbance observer

$$u = u_{pa} = \frac{1}{\hat{b}(t,q)}[-\hat{f}(t,q) + q_d^{(n)} - \sum_{i=0}^{n-1} c_i \tilde{q}^{(n)}] \tag{10}$$

in which u_{pa} represents the control effort using the design in Eq.(9); $\hat{f}(t,\mathbf{q})$ and $\hat{b}(t,\mathbf{q})$ are the estimates of functions $f(t,\boldsymbol{q})$ and $b(t,\boldsymbol{q})$, respectively. With $p \equiv d/dt$, the coefficients, c_i, i = 0, 1, 2,…, n-1, of the desired characteristic equation are rewritten as

$$p^n + c_{n-1}p^{n-1} + c_{n-2}p^{n-2} + \cdots + c_1 p + c_0 = 0 \tag{11}$$

has the desired multiple roots, at $-\lambda$, leading to a desired exponentially stable error dynamics. And,

$$\tilde{q}^{(n)} + c_{n-1}\tilde{q}^{(n-1)} + c_{n-2}\tilde{q}^{(n-2)} + \cdots + c_1\dot{\tilde{q}} + c_0\tilde{q} = 0 \tag{12}$$

provided that the bounded uncertainties, d vanishes and the nominal functions, $\hat{f}(t,\mathbf{q})$ and $\hat{b}(t,\mathbf{q})$ would coincide with $f(t,\boldsymbol{q})$ and $b(t,\boldsymbol{q})$. However, perturbation often arises and causes the resultant error dynamics to deviate from the desired one in an adverse way.

This effect can be revealed further by manipulating Eq. (1) and substituting the control law, Eq. (10) to yield

$$q^{(n)} = q_d^{(n)} - \sum_{i=0}^{n-1} c_i \tilde{q}^{(i)} + [f(t,q) - \hat{f}(t,q)] + [b(t,q) - \hat{b}(t,q)]u + d \tag{13}$$

Rearranging Eq.(13) to yield,

$$\tilde{q}^{(n)} + c_{n-1}\tilde{q}^{(n-1)} + c_{n-2}\tilde{q}^{(n-2)} + \cdots + c_1\dot{\tilde{q}} + c_0\tilde{q} = \psi \tag{14}$$

where $\psi = \left(f(t,q) - \hat{f}(t,q)\right) + \left(b(t,q) - \hat{b}(t,q)\right)u + d(t)$ is noted as a *lumped-perturbation* in the controller design and may cause an undesirable overshoot or, more severely, system instability. To compensate for the perturbation, the control law from Eq.(10) is redesigned with an extra compensation term, i.e.

$$u=u_{pa}+u_{pc} \tag{15}$$

Consequently, Eq.(1) becomes

$$q^{(n)}=q_d^{(n)}-\sum_{i=0}^{n-1}c_i\tilde{q}^{(i)}+\hat{b}(t,q)u_{pc}+\psi \tag{16}$$

To compensate for ψ , we would like to have $\hat{b}(t,q)u_{pc}=-\hat{\psi}$ to yield

$$q^{(n)}=q_d^{(n)}-\sum_{i=0}^{n-1}c_i\tilde{q}^{(i)}+\psi-\hat{\psi} \tag{17}$$

Ideally, $\hat{\psi}$ should be set equal to ψ , then the error dynamics would follow Eq.(12). To obtain the estimate of the perturbation, $\hat{\psi}$, an auxiliary process is adopted and defined as

$$\dot{w}=q_d^{(n)}-\sum_{i=0}^{n-1}c_i\tilde{q}^{(i)}+\hat{b}(t,q)u_{pc}+\Psi\,\text{sgn}(\sigma_q) \tag{18}$$

And, based on Eq.(2), we have

$$\Psi=-\left[F(t,q)+B(t,q)|u|+D(t)\right]$$

Furthermore, a switching function is defined as

$$\sigma_q=q^{(n-1)}-w \tag{19}$$

Without the loss of generality, we adopted the case in Eq.(9), for the switching of two different SMCs, we should have the following auxiliary process,

$$\dot{w}=\begin{cases}\ddot{q}_d-\lambda_q\dot{\tilde{q}}+\Psi\,\text{sgn}(\sigma_q)-\hat{\psi}_1 & |\tilde{q}|>\varepsilon_q\\ \ddot{q}_d-2\lambda_q\dot{\tilde{q}}-\lambda_q^2\tilde{q}+\Psi\,\text{sgn}(\sigma_q)-\hat{\psi}_2 & |\tilde{q}|\le\varepsilon_q\end{cases} \tag{20}$$

Here, $\hat{\psi}_1$ and $\hat{\psi}_2$ are perturbations estimated for ψ_1 and ψ_2 , respectively.

Theorem 1:
Let $\hat{\psi}_1$ and $\hat{\psi}_2$ be defined as in Eq.(20), and

$$\hat{\psi}=\hat{\psi}_1+\hat{\psi}_2 \tag{21}$$

A sliding function with n=2 in (19) that defined a global sliding mode is established using auxiliary process as described in Eq.(20). And let $\hat{\psi}_1$ and $\hat{\psi}_2$ be estimated from the following,

$$\dot{\hat{\psi}}_1=K_{c1}\Psi\,\text{sgn}(\sigma_q) \tag{22}$$

$$\dot{\hat{\psi}}_2=-K_{c2}\psi_2+K_{c2}\Psi\,\text{sgn}(\sigma_q) \tag{23}$$

where K_{c1} and K_{c2} are constants to be determined based on the range of bandwidth of interest. Then, we have an invariant condition

$$\frac{\hat{\psi}(p)}{\psi(p)}=\frac{(K_{c2}+K_{c1})p+K_{c2}K_{c1}}{(p+K_{c2})(p+K_{c1})} \tag{24}$$

Proof:

From Eq.(19), with n=2, we have

$$\dot{\sigma}_q = \ddot{q} - \dot{w} = \psi - \hat{\psi} - \Psi\,\mathrm{sgn}(\sigma_q) + \hat{\psi}_2 = \psi - \hat{\psi}_1 - \Psi\,\mathrm{sgn}(\sigma_q) \qquad (25)$$

with $\dot{\sigma}_q = 0$, we could find

$$\frac{\hat{\psi}_1(p)}{\psi(p)} = \frac{K_{c1}}{p + K_{c1}}$$

and $$\frac{\hat{\psi}_2(p)}{\psi(p)} = \frac{K_{c2}p}{(p + K_{c1})(p + K_{c2})}$$

Therefore,

$$\frac{\hat{\psi}(p)}{\psi(p)} = \frac{(K_{c2} + K_{c1})p + K_{c2}K_{c1}}{(p + K_{c2})(p + K_{c1})}$$

■

This relationship is invariant to variations of system parameters, and the estimate $\hat{\psi}$ is intrinsically a low-pass-filtered version of disturbance ψ with the filter's bandwidth determined by the constants K_{c1} and K_{c2}. Ideally, $\hat{\psi}$ should be set equal to ψ. Increasing the value of K_{c1} and K_{c2} approaches this ideal case, improves the effectiveness of disturbance compensation, but may increase the chatter level in control input. As a rule of thumb, the filter's bandwidth is usually chosen to be about ten times that of the closed-loop system, that is, $K_{c1} = K_{c2} = 10\omega_n$.

Remark1:

w is the state variable of the auxiliary process, the switching function σ_q is defined as (19) and the switching gain Ψ is assigned so that $|\psi - \hat{\psi}_1| < \Psi$. Here, $\mathrm{sgn}(\cdot)$ denotes the sign function. To ensure a sliding regime $\sigma_q = 0$, the sliding condition

$$\lim_{\sigma_q \to 0} \sigma_q \dot{\sigma}_q < 0$$

should be satisfied. Consideration here of Lyapunov candidate $V = 0.5\sigma_q^2$. Taking the derivative V with respect to time and substituting (20), (17) and (25) into the resulting equation gives. Multiplying both sides of (25) by σ_q and noting that $|\psi - \hat{\psi}_1| < \Psi$, we have

$$\dot{V} = \sigma_q \dot{\sigma}_q = (\psi - \hat{\psi}_1)\sigma_q - \Psi|\sigma_q| < 0, \qquad \text{if } \sigma_q \neq 0$$

which implies the satisfaction of the sliding condition and the existence of the sliding regime $\sigma_q = 0$ after some time. Subsequently, assigning the initial state of the auxiliary process as

$$w(t = 0) = \dot{q}(t = 0)$$

gives $\sigma_q = 0$ at $t = 0$.

This together with the satisfaction of the sliding condition implies that $\sigma_q = 0$ for all $t \geq 0$.

Thus, the sliding regime $\sigma_q = 0$ exists for the disturbance estimation during an entire response despite the presence of system disturbance which is desired to be estimated.

Theorem 2

Assuming that boundary layer Φ can be determined based on the lumped disturbance estimated by $\hat{\psi}$, and Eq.(6) is rewritten as

$$\dot{\Phi} + \lambda_{\Phi}\Phi = \hat{\psi} \tag{26}$$

Then, the bandwidth of state-dependent boundary layer can be chosen accordingly, i.e.

$$\lambda_{\Phi} = \frac{K_{c2}K_{c1}}{K_{c2} + K_{c1}} \tag{27}$$

And, the state-dependent boundary layer with observer is bounded and a quasi-sliding mode is assured.

Proof:

From Eq.(26) and Eq.(24), we have

$$\Phi(p) = \frac{1}{p + \lambda_{\Phi}}\hat{\psi}(p) \tag{28}$$

$$\hat{\psi}(p) = \frac{(K_{c2} + K_{c1})p + K_{c2}K_{c1}}{(p + K_{c2})(p + K_{c1})}\psi(p) \tag{29}$$

Combine (27) and (28) to yield

$$\Phi(p) = \frac{(K_{c2} + K_{c1})\left(p + \frac{K_{c2}K_{c1}}{K_{c2} + K_{c1}}\right)}{(p + \lambda_{\Phi})(p + K_{c2})(p + K_{c1})}\psi(p) \tag{30}$$

Consequently, using $\lambda_{\Phi} = \frac{K_{c2}K_{c1}}{K_{c2} + K_{c1}}$, Eq. (30) is reduced to

$$\frac{\Phi(p)}{\psi(p)} = \frac{(K_{c2} + K_{c1})}{(p + K_{c2})(p + K_{c1})} \tag{31}$$

Since K_{c1} and K_{c2} are positive constants, if the perturbation is bounded, then the state-dependent boundary based on disturbance observer is bounded and asymptotically stable is assured.

Remark2: The thickness of boundary layer $\Phi(p)$ is the filter output of the system perturbation ψ through an over-damped second order low-pass filter with pre-specified bandwidth. Accordingly, even if ψ is with high-frequency content or with discontinuous jump, only low-frequency part of Φ will be preserved. Furthermore, ψ can be estimated from $\hat{\psi}$, and a bounded Φ is assured.

4. EXPERIMENTAL STUDIES

The Experimental Setup [10]

As shown in Fig. 2, a two-D.O.F maglev platform under study consists of two electromagnets M_{zl} and M_{zr} with μ_r = 6000 and area of pole-face equals to 4.05 mm^2 wounded with 700 turns of coil to levitate the platform. For each electromagnet poles, both equipped with an optical-sensor which is manufactured by MTI Instrument with the probe having the capability of emitting a light to the surface and receiving the projection with a sensitivity of 0.868 *mm/V* and a linear range of 2 *mm* with a bandwidth of 140 *KHz* to measure its corresponding air gaps.

For each electromagnet poles, both equipped with an optical-sensor which is manufactured by MTI Instrument with the probe having the capability of emitting a light to the surface and receiving the projection with a sensitivity of 0.868 *mm/V* and a linear range of 2 *mm* with a bandwidth of 140 *KHz* to measure its corresponding air gaps. The signals were then send to a Pentium PC through a 12-bit high speed A/D converter with a conversion rate at 90K samples/sec and conversion range of ± 10 *volt*.

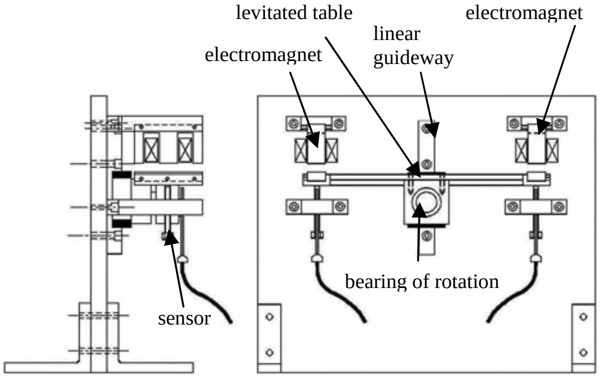


FIGURE 2: Schematic drawing of a two-DOF maglev platform

The control law was implemented using C-language at a sampling rate of 1*ms*.The control effort was then send out through a D/A converter with the conversion rate at 15 K samples/sec, and a range of 0 to 5 *volt* to a current source with current gain equals to 2 *A/V* with the maximum output of 2 *A* and a bandwidth of 10 *KHz* to drive the two electromagnets, M_{zl} and M_{zr} which levitated the platform thus closing the feedback loop.

The Dynamic Model of a 2-D.O.F Maglev

Assuming the mass center of the platform and its geometric center coincides, and it operates with small angular motions, the linearized equation of motion can thus be written as

$$\mathbf{H}\ddot{\mathbf{q}} = \mathbf{Q}$$
$$\mathbf{H} = \begin{bmatrix} M & 0 \\ 0 & I_{yy} \end{bmatrix},\ \mathbf{q} = \begin{bmatrix} z_c & \theta \end{bmatrix}^T \text{ and } \mathbf{Q} = \begin{bmatrix} F_z & \tau_{yy} \end{bmatrix}^T \tag{32}$$

Where M = 545±0.5*g* is the total mass of the levitated platform, while I_{yy} = 0.01915 *Kg-m*^{2} denotes the moment of inertia around the y-axis, respectively. z_c is the mass center moving in *z*-direction, and θ denotes the table's rotation in the z-direction. F_z and τ_{yy} are the electromagnetic force and torque exerting on the 2-D maglev. Let air gaps be z_l and z_r, respectively. And, L = $199 \pm 0.5\,mm$ is the table length. The control force, f_l and f_r, on each side and torque generated by the two electromagnets are expressed as:

$$\begin{cases} f_l = \dfrac{\alpha I^2}{z_l^2} \\ f_r = \dfrac{\alpha I^2}{z_r^2} \\ F_z = f_l + f \\ \tau_{yy} = (f_r - f_l)L \end{cases} \tag{33}$$

with $\alpha = \dfrac{\mu_o N^2 A}{4}$.

Each optical sensor aligned with the center of the two electromagnets (M_{zl}, M_{zr}) can measure its corresponding air gaps. The linearized relationship between the displacement of the mass center (z_c) and pitch angle (θ) can be described as

$$\begin{cases} z_c = -\dfrac{z_l - z_r - w_z}{2} \\ \theta = -\dfrac{z_l - z_r}{L} \end{cases} \tag{34}$$

Because electromagnetic force and gravitational force are the external forces, the equations of motion can be expressed as

$$\begin{aligned} M\ddot{Z}_c &= F_Z - Mg + d_Z(t) \\ I_{yy}\ddot{\theta} &= \tau_{yy} + d_\theta(t) \end{aligned} \tag{35}$$

Where d_z and d_θ are lumped matched uncertainties. Moreover, based on the geometric relationship, the total electromagnetic forces f_l and f_r on two sides of the platform can be expressed in terms of F_z and τ_{yy}.

$$\begin{cases} f_l = \dfrac{1}{2}F_z - \dfrac{\tau_{yy}}{2L} \\ f_r = \dfrac{1}{2}F_z + \dfrac{\tau_{yy}}{2L} \end{cases} \tag{36}$$

Experimental Studies

Experiments were performed to verify the proposed scheme. In this sub-section, MSMC schemes utilize mode-switching between SMC and ISMC with fixed and state-dependent boundary layer were performed for comparison. It is also noted that the state-dependent boundary layer is determined based on a disturbance observer as described in the previous section.

Case 1: *The effect of controller selection*

From Fig. 3 and Fig. 4, SMC can obtain satisfactory settling time, while it enters the boundary layer, there exists a significant steady-state error, however. It is noted that while entering the switching region which is set at *t*=0.05 sec, ISMC has demonstrated its efficacy in reducing the steady state error, but it requires longer settling time and larger overshoot that will inevitably cause integral wind-up. It is seen that the response in pitch angle experienced a much violent vibration than that of vertical displacement. The mode-switching sliding mode(MSMC) has demonstrated its capability in the application on this Maglev platform but with the price of the fine tuning of proper switching region.

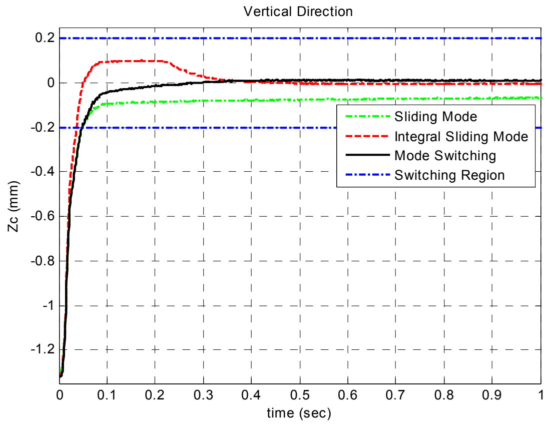


FIGURE 3: Response in Z-direction with different controllers

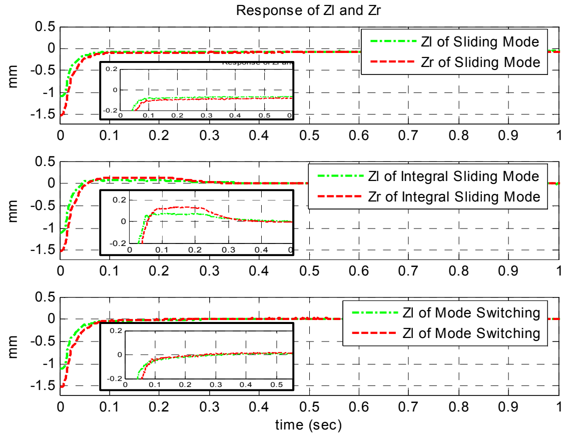


FIGURE 4: Response of left- and right-tip with different controllers

Case 2: *The effect of boundary layer selection*

Here, the MSMC is set to satisfy fixed switching instant, t_s=0.04 sec, but different boundary layers to investigate its effect on the control performance. The MSMC controller is designed to switch between a SMC and ISMC under pre-specified condition and its corresponding control parameters are listed as follows.

1. $\left|\tilde{Z}_c(0.04)\right| \leq 0.1mm$, λ_z=64.164 sec^{-1}, λ_{zI}=32.083 sec^{-1}.
2. $\left|\tilde{Z}_c(0.04)\right| \leq 0.2mm$, λ_z=46.835 sec^{-1}, λ_{zI}=23.418 sec^{-1}.
3. $\left|\tilde{Z}_c(0.04)\right| \leq 0.3mm$, λ_z=36.699 sec^{-1}, λ_{zI}=18.350 sec^{-1}.

Where Z_c were calculated using Eq. (34) that combines Z_l with Z_r .

It can be seen from the experimental results as shown in Fig. 5 and Fig. 6 that if the boundary layer is set at $\left|\tilde{Z}_c\right| \leq 0.2mm$. It could have the best performance among the three conditions in term of overshoot and steady state error. Moreover, the responses of left- and right-tip further revealed the efficacy of the MSMC scheme. Consequently, the mode-switching sliding mode has demonstrated its capability in the application on this Maglev platform provided that a fine tuning of proper boundary layer is needed.

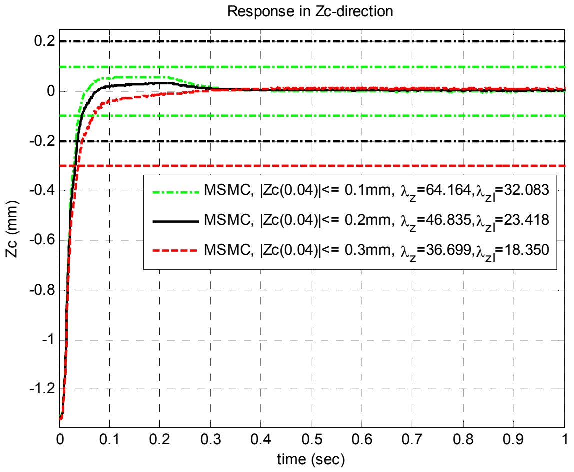


FIGURE 5: Response in Z-direction with different boundary layer

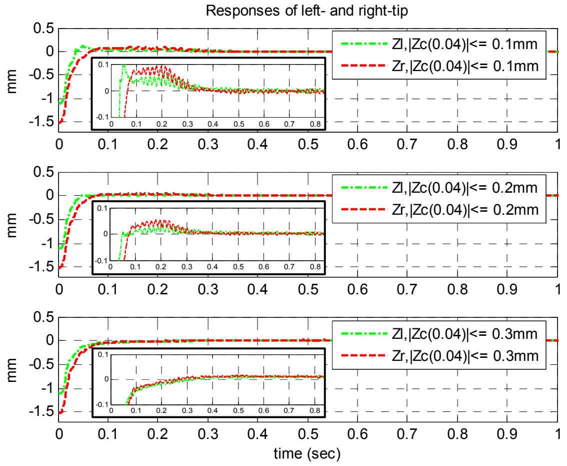


FIGURE 6: Responses of left- and right-tip with different boundary layer

Case 3: *The effect of switching time selection*

Here, the MSMC is set to satisfy different switching instant at fixed boundary layer, $\tilde{q}$ =0.2 *mm*, to investigate its effect on the control performance. The MSMC controller is designed to switch between a SMC and ISMC under pre-specified condition and its corresponding control parameters are listed as follows.

1. $\left|\tilde{Z}_c(0.06)\right| \le 0.2mm$, λ_z=31.223 sec^{-1}, λ_{zl}=15.612 sec^{-1}.
2. $\left|\tilde{Z}_c(0.05)\right| \le 0.2mm$, λ_z=37.467 sec^{-1}, λ_{zl}=18.734 sec^{-1}.
3. $\left|\tilde{Z}_c(0.04)\right| \le 0.2mm$, λ_z=46.835 sec^{-1}, λ_{zl}=23.418 sec^{-1}.

Where Z_c were calculated using Eq. (34) that combines Z_l with Z_r .

It can be seen from the experimental results as shown in Fig. 7 that if the switching instant is set at *t*=0.05 sec. It could have the best performance among the three conditions in terms of overshoot and steady state error. Furthermore, as shown in Fig. 8, the responses of left- and right-tip further revealed the efficacy of the MSMC scheme. Consequently, the mode-switching sliding mode has demonstrated its capability in the application on this Maglev platform provided that a fine tuning of proper time is needed. It can be concluded from results of Case 1 and Case 2 that one can pre-specify $\left|\tilde{Z}_c\right| \leq 0.2mm$ with switch instant at *t*=0.05 sec to have the best result among all tested conditions.

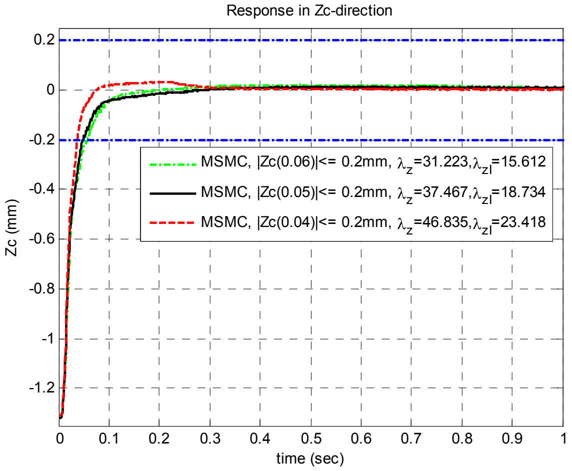


FIGURE 7: Response of in Z-direction with different switching time

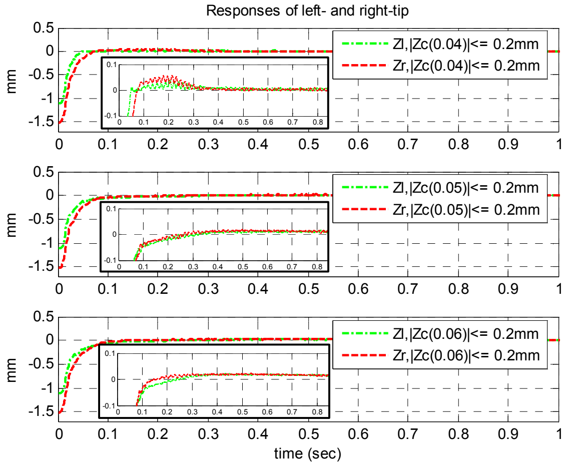


FIGURE 8: Responses of left- and right-tip with different switching time

Case 4: *MSMC with state-dependent boundary layer*

It is seen from the previous experimental results that we can select the switching region (in terms of layer thickness and switching instant) based on previous results through fining tuning process or engineering sense. As depicted in the previous section, a disturbance observer with state-dependent boundary layer for the on-line switching region selection for the MSMC scheme would perform the same result without trial and error. Finally, the experiments are performed to demonstrate its efficacy.

Test I: MSMC with state-dependent boundary layer controller under pre-specified condition and its on-line switching region selection are listed as follows.

1. $\left|\tilde{Z}_c(0.04)\right| \le 0.1mm$, λ_z=64.164 sec^{-1}, λ_{zi}=32.083 sec^{-1}, K_{c1} =K_{c2} =320 sec^{-1}.
2. $\left|\tilde{Z}_c(0.05)\right| \le 0.2mm$, λ_z=37.467 sec^{-1}, λ_{zi}=18.734 sec^{-1}, K_{c1} =K_{c2} =187 sec^{-1}
3. $\left|\tilde{Z}_c(0.06)\right| \le 0.2mm$, λ_z=31.223 sec^{-1}, λ_{zi}=15.612 sec^{-1}, K_{c1} =K_{c2} =156 sec^{-1}

where K_{c1} and K_{c2} is based on the uncertain bounds of system in Eq.(30) and Eq.(31), respectively.

As shown in Fig. 9, the state-trajectory of vertical displacement would be constrained inside the state-dependent boundary layer as expected. On the other hand, the tips' responses are also shown in Fig. 10. Obviously, the different gaps of left- and right-tip have been estimated and compensated on-line using the MSMC with state dependent boundary layer.

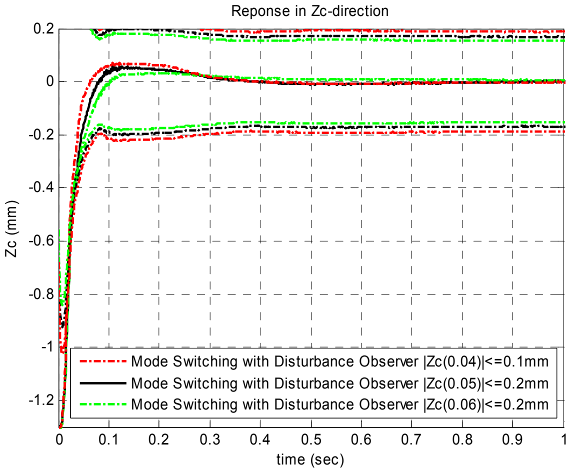


FIGURE 9: Responses in Z-direction with state dependent boundary layer

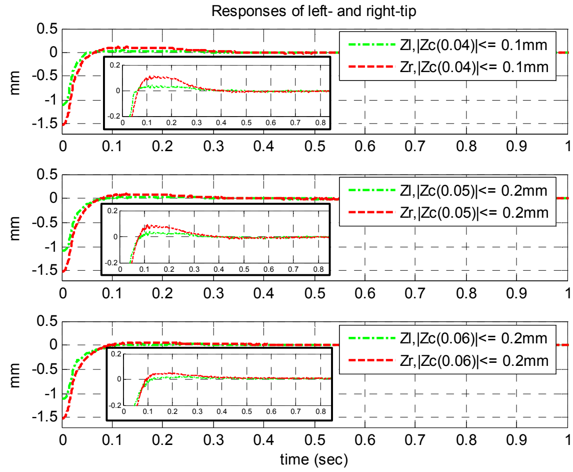


FIGURE 10: Responses in left- and right-tip with state dependent boundary layer

<u>*Test II*</u>: Compare with their counterparts in Case 1 and Case 2 with the same control laws and the same switching region,

$\left|\tilde{Z}_c(0.05)\right| \le 0.2mm$, λ_z=37.467 sec^{-1}, λ_{zi}=18.734 sec^{-1}, K_{c1} =K_{c2} =187 sec^{-1}

As Fig. 11 and Fig. 12 show, the experimental result on vertical direction is seen to be equipped with the augmented disturbance observer; a faster settling time and smaller steady-state error were achieved. It is also noted that the switching region is time-varying due to its state-dependent nature but is adjusted on-line based on the estimated result from the augmented disturbance observer.

Consequently, MSMC with state-dependent boundary layer have the best performance among the four conditions in terms of smoothness, overshoot and steady state error.

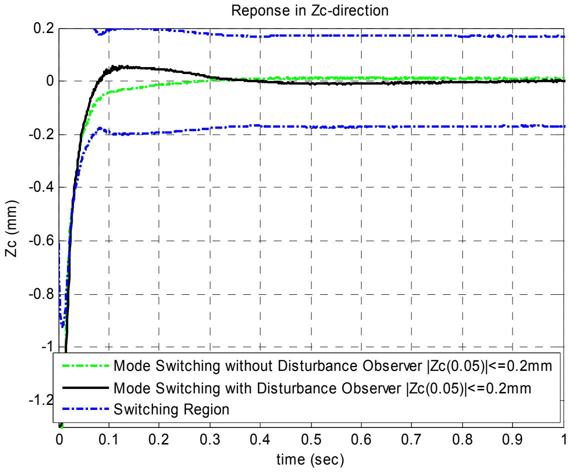


FIGURE 11: Responses in *Z*-direction with state dependent boundary layer

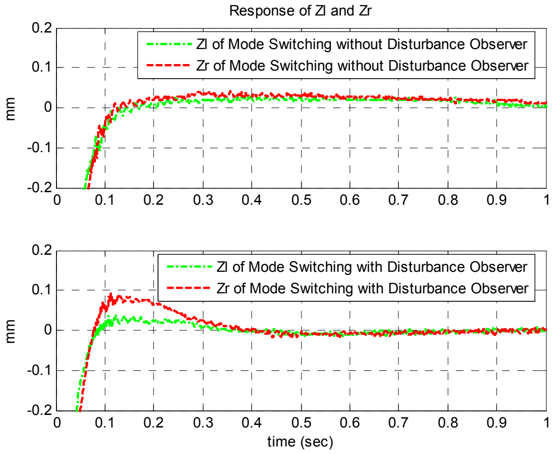


FIGURE 12: Responses in left- and right-tip with state dependent boundary layer

5. CONSLUSION

This paper presented a Mode-Switching Sliding-mode Control (MSMC) scheme that can switch between different sliding-mode control schemes. Switching would occur while the states entering the vicinity of a preset operating point. MSMC can provide better positioning performance than SMC and ISMC alone. The proposed state-dependent boundary layer based-on a disturbance observer can not only precisely compensate for system perturbation within the pre-specified frequency range, but also reduce the adverse effect due to chattering. MSMC with disturbance observer using state-dependent boundary layer design has been successful applied to a two-DOF Maglev platform. The experiment results also showed that the maglev system can track the reference input within the pre-specified errors, i.e. $\tilde{z}_c$ and $\tilde{\theta}$ as expected, and it can also provide certain robust performance for systems subjected to uncertainties from both parameters and external disturbance with an auto-tuned switching region based on a state-dependent boundary layer incorporated with disturbance observer design.

6. REFERENCES

1. K. D. Young, V. I. Utkin and U. Ozguner. *“A control engineer’s guide to sliding model control”.* IEEE Trans. on Control System Technology, 7(3):328-342, 1999
2. J. J. E. Slotine and W. Li. *“Applied Nonlinear Control*, Prentice-Hall”, New Jersey, pp. 276-310(1991)
3. Utkin, Jurgen Guldner and Jingxin Shi. *“Sliding Mode Control in Electromechanical Systems”*, CRC Press, pp. 131-153(1999)
4. J. J. E. Slotine and S. S. Sastry. “*Tracking control of non-liner systems using sliding surface, with application to robot manipulators”.* Int. J. of Control, 38(2):465-492, 1983
5. B. W. Bekit, J. F. Whidborne and L. D. Seneviratne, “*Sliding mode control for robot manipulators using time-varying switching gain and boundary layer”.* Control '98. UKACC International Conference on, London, United Kingdom, 1998.
6. M. -S. Chen, Y. R. Hwang and M. Tomizuka. “*A state-dependent boundary layer design for sliding mode control”.* IEEE Trans. on Automatic Control, 47(10):1677-1681, 2002
7. M. -S. Chen, Y. R. Hwang and M. Tomizuka, “*Sliding mode control reduced chattering for systems with dependent uncertainties”.* Networking, Sensing and Control, 2004 IEEE International Conference on, Taipei, Taiwan, 2004
8. M. Iwasaki, K. Sakai and N. Matsui, *“High-speed and high-precision table positioning system by using mode switching control”.* Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE**,** Nagoya, Japan 1998.
9. T. Yamaguchi, H. Numasato, H. Hirai, “*A mode-switching control for motion control and its application to disk drives: design of optimal mode-switching conditions”.* IEEE/ASME Trans. on Mechatronics, 3(3):202-209, 1998
10. C. -Y. Hsu. “*Application of Switching Law to A 2-DOF Maglev Platform”.* Master Thesis, National Tsing Hua University, June 2005
11. Y. -S. Lu and J. -S. Chen. “*Design of a perturbation estimator using the theory of variable-structure system and its application to magnetic levitation systems”.* IEEE Trans. on Industrial Electronics, 43(3):281-289, 1995

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA