

Developing Efficient Web-Based Xml Tool

S.Devaraju, D.Saravana Prakash



Abstract— The “XML TOOL” is used to design and develop the efficient web based application. The main objective of this application is to design and develop web based software for creation of XML, DTD, Schema and Validation of XML against a DTD or Schema, thereby reducing the discrepancies of the existing manual system which is quite inefficient and time consuming. The user can create XML by giving the name of the file, the names of the tags, description of tags, its sub tags, the attributes and the various other values. The user can create DTD and XSD by providing values for the tag names, the sub tags names, the attribute names and the various rules associated with it. The XML, DTD and XSD files are created automatically based on the values provided by the user. The user can check if the XML is well formed or not. The user can manually edit the XML, DTD or XSD that is auto created. The user can validate the XML against either a DTD or an XSD. If the XML is valid then a success message is displayed in green color. If the XML is either not well formed or not valid, then the associated errors are displayed in red color.

Keywords— XML Tool, DTD, XSD

I. INTRODUCTION

The “XML Tool” is used to create, edit and view the XML, DTD and XSD files with minimal complexity. The application performs the following [1][2],

- The user can create XML by giving the name of the file, the names of the tags, description of tags, its sub tags, the attributes and the various other values.
- The user can create DTD and XSD by providing values for the tag names, the sub tags names, the attribute names and the various rules associated with it.
- The XML, DTD and XSD files are created automatically based on the values provided by the user.
- The user can check if the XML is well formed or not.
- The user can manually edit the XML, DTD or XSD that is auto created.
- The user can validate the XML against either a DTD or an XSD. If the XML is valid then a success message must be displayed in green color. If the XML is either not well formed or not valid, then the associated errors are displayed in red color.

A. The Problem Definition

The problem is to develop a web application that is capable of helping all kinds of users, even those who are not well versed in XML to work on it with ease.

The user provides the details of the tags to be included in the XML / DTD /XSD, name of the file to be created etc. The application should be able to create the corresponding file with the tag details specified by the user.

The user should also be able to parse the XML or validate it against a DTD or XSD.

B. Objectives

The main objective of the application “XML TOOL” is to provide a web interface with which anyone who wishes to work with XML can create XML, DTD and XSD files with ease. The user can edit the XML, DTD and Schema document that is auto created. The application also provides the user, an option to validate the XML files against a DTD or XSD.

In brief, the objective of the application can be summarized as follows

- The XML tool application should be user friendly.
- It should be efficient and effective.
- It should guide the user in creating XML, DTD and XSD and make them feel comfortable working with it.
- Validation of the XML against a DTD or XSD should be performed proficiently.

C. Motivation Study

This application is mainly developed to provide users, an easy platform to work with XML, DTD and XSD. The existing system allows only the users who have a brief outline of XML to work in it. In the proposed XML Tool, any user who has very little knowledge of XML can also easily create an XML, DTD or XSD file. Thus the main reason for developing the XML Tool is to provide users who have no in-depth knowledge on XML, an easy access to work on it and for those users who have good skill in XML, a still easier guide to work with XML [3] [4].

II. MODULE DESCRIPTION

A. Create XML

The Create XML module is used to create an XML document by providing the tag values and attribute values. This module contains the sub modules such as View XML, Edit XML and Parse XML. The View XML module allows the user to view the XML document that has been auto created. The Edit XML module is used to edit the XML document accordingly and overwrite the existing XML file with the edited content. The Parse XML module is used to check the well-formedness of the XML document and the errors are displayed if found. Table 1 shows the descriptions of create XML.

Manuscript published on 30 September 2019

* Correspondence Author

S.Devaraju*, Department of Computer Science and Applications, Sri Krishna Arts and Science College, Coimbatore-8, Tamil Nadu, India
E-mail Id: deva_sel@yahoo.com

D.Saravana Prakash, Department of Computer Science and Applications, Sri Krishna Arts and Science College, Coimbatore-8, Tamil Nadu, India E-mail Id: dsp.prakash@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

TABLE 1. USE CASE DESCRIPTIONS – CREATE XML

Title	Create Xml
Actor	Xml User
Description	Creating an Xml document
Flow of Events	<ul style="list-style-type: none"> The user enters the name of the root tag, sub tags and attributes. The Xml document is created based on the inputs provided. A DTD/Schema file is added to the xml file based on the user selection
User interface	The user interface is designed using JSF and JSP

B. Create DTD

The Create DTD module is used to create a DTD file by providing the various tag values, type and attribute values. Table 2 shows the descriptions of create DTD.

TABLE 2. USE CASE DESCRIPTIONS – CREATE DTD

Title	Create DTD
Actor	Xml User
Description	Creating a DTD
Flow of Events	<ul style="list-style-type: none"> The user enters the names of the root tag, sub tag and attributes. The DTD is created based on the inputs provided.
User interface	The user interface is designed using JSF and JSP
Title	Create Xml Schema
Actor	Xml User
Description	Creating a Xml Schema
Flow of Events	<ul style="list-style-type: none"> The user enters the names of the root tag, sub tag and attributes. The Xml Schema is created based on the inputs provided.
User interface	The user interface is designed using JSF and JSP

C. Create Schema

The Create Schema module is used to create a XSD file by providing the various tag values, type and attribute values. Table 3 shows the descriptions of create XSD.

TABLE 3. USE CASE DESCRIPTIONS – CREATE XSD

Title	Create Xml Schema
Actor	Xml User
Description	Creating a Xml Schema
Flow of Events	<ul style="list-style-type: none"> The user enters the names of the root tag, sub tag and attributes. The Xml Schema is created based on the inputs provided.
User interface	The user interface is designed using JSF and JSP

D. Validate XML

The Validate XML module allows the user to validate an XML file against a DTD or XSD based on the user's choice. Errors are displayed as a table with error line, reason and location. The user can edit the XML file accordingly and overwrite the existing XML file with the edited content. Table 4 shows the descriptions of validate XML.

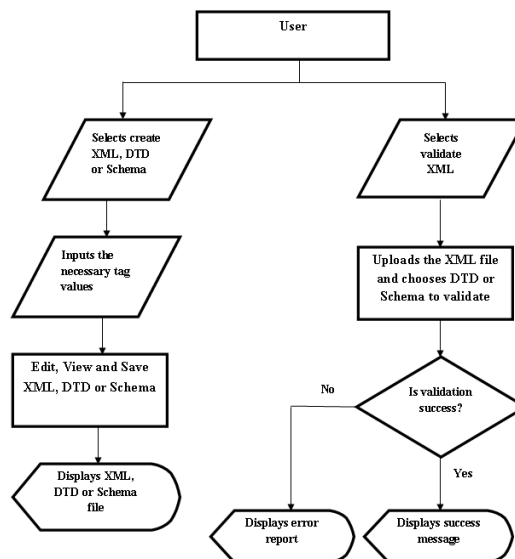
TABLE 4. USE CASE DESCRIPTIONS – VALIDATE XML

Title	Validate Xml
Actor	Xml User
Description	The Xml document is validated here
Flow of Events	<ul style="list-style-type: none"> The user selects an Xml document and a DTD/Xml Schema for validation. The Xml document is validated against the selected DTD/Xml Schema
User interface	The user interface is designed using JSF and JSP

E. System Flow Diagram

A System Flow Diagram (SFD) visually represents the general flow of the system. An overall representation of the system can be represented by using system flow diagram. It illustrates the sequence of process and work flow that is involved in the system. Figure 1 shows the System Flow Diagram [5][6].

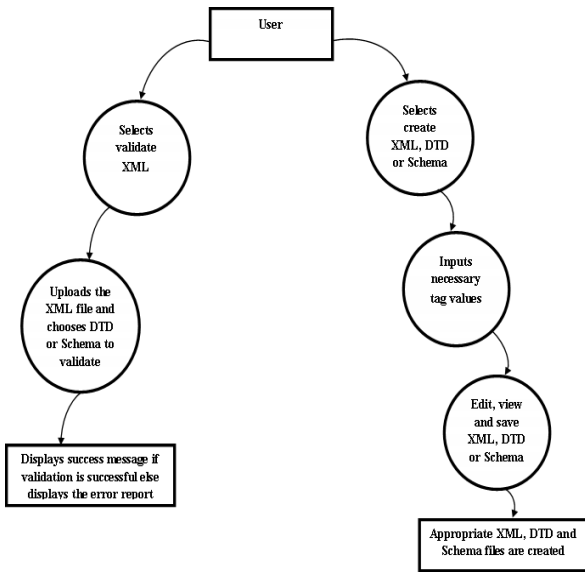
FIGURE 1. SYSTEM FLOW DIAGRAM-XML TOOL



F. Data Flow Diagram

Data Flow Diagram (DFD) is an important technique for modelling a system's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFDs reveal relationships among and between the various components in a program or system. DFDs consist of four major components: entities, processes, data stores and data flow. The Data Flow Diagram is commonly used also for the visualization of structured design data processing. The normal flow is represented graphically. A designer typically draws context level DFD first showing interaction between the system and the outside entities. Then this context level DFD will then be exploded in order to further show the details of system being modelled. Figure 2 shows the Data Flow Diagram [7][8].

FIGURE 2. LEVEL 1 DFD-XML TOOL



III. RESULTS AND DISCUSSION

A. Input Design

Input facilitates the entry of data into the computer system. Input design involves the selection of the best strategy for getting data into the computer system at the right time and as accurately as possible. This is because the most difficult aspect of input design in accuracy. The use of well-defined documents can encourage users to record data accurately without omission [9][10]. For example, if the user needs to create a root tag, then he needs to give in the root tag name and its attributes if any as input. The corresponding form in the CreateXML page should have proper text input fields with label clearly defined such as “Root Tag name”, “Attribute Name”, ”Attribute Value” etc. Input design must capture all the data that the system needs, without introducing any errors. Input errors can be greatly reduced when inputting directly by using appropriate forms for data capture and well-designed computer screen layout. Table 5 shows the input designs.

TABLE 5. INPUT DESIGNS

Input	Description
Name of the File	Name of the XML, DTD or Schema to be created and the XML to be validated
Root tag name	Name of the root tag to be created in respective XML, DTD or Schema file
No. of Sub tags, No. of attributes, No. of child elements	Number of attributes, sub tags and child elements for the respective tags
Selection of DTD/Schema	Selecting DTD/Schema to validate against XML and including reference of DTD/Schema in the XML file

B. Output Design

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. In any system, the output design determines the input to be

given to the application. Table 6 shows the output designs [11][12].

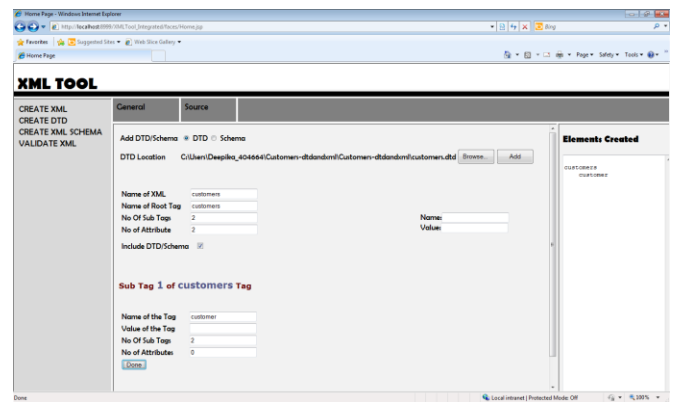
TABLE 6. OUTPUT DESIGNS

Output	Description
Source view	XML, DTD and Schema files are created and shown as output
Edit view	The files that are auto created can be edited and saved further.
List of Errors	The errors if any are listed in a data table

C. Results of XML Tool

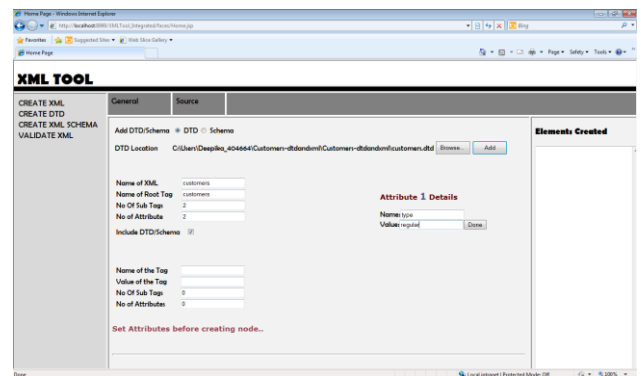
Create XML – Tag Creation: The screenshot depicts how the details of the sub tag are fetched from the user. The panel on the right shows the tree structure of the XML tags that are created so far. Figure 3 shows the tag creation of create XML [13][14].

FIGURE 3. CREATE XML – TAG CREATION



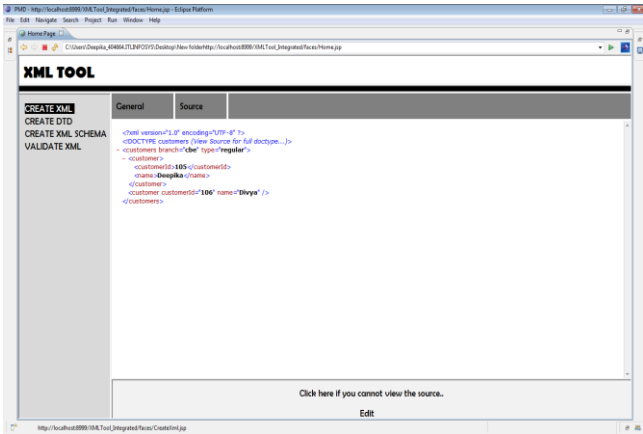
Create XML- Attribute Creation: The screenshot displays how the attribute values are got from the user. The message in the below tells the user that the tag creation cannot be done before creating attributes, since the count of attributes is greater than zero. Figure 4 shows the attribute creation of create XML [15][16][17].

FIGURE 4. CREATE XML - ATTRIBUTE CREATION



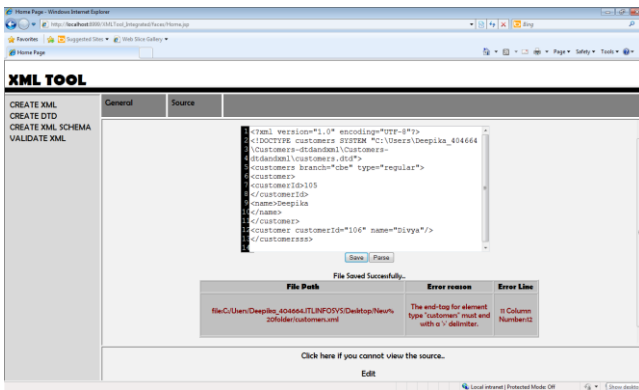
Create XML - Source View: The screenshot shows the created XML file when the Source tab is chosen in a read only format. For editing, an option is provided at the bottom of the page, where the source will be opened in the writable format. Figure 6 shows the source view of create XML.

FIGURE 5. CREATE XML - SOURCE VIEW



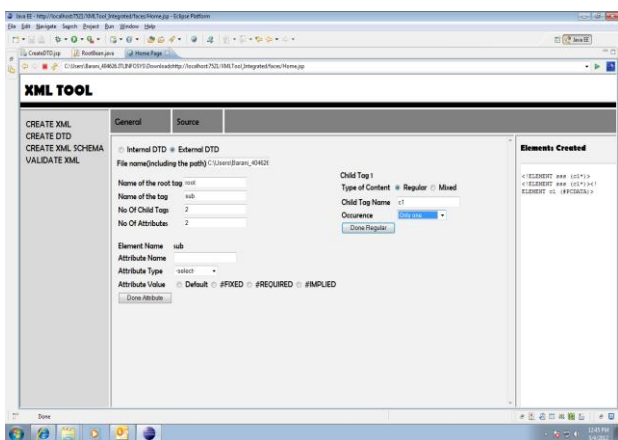
Create XML- Parse: The screenshot shows a table with the error details when the XML is parsed. It shows the location of the file, error reason and error line. Figure 6 shows the parse view creation of create XML.

FIGURE 6. CREATE XML – PARSE VIEW



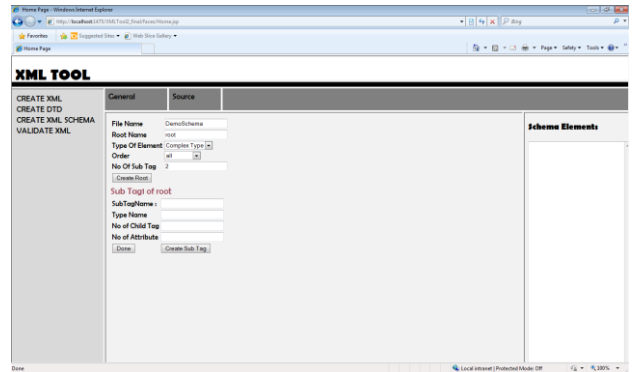
Create DTD- Tag Creation: The screenshot depicts how the details of the sub tag are fetched from the user. The panel on the right shows the tree structure of the DTD file that has been created so far. Figure 7 shows the tag creation of create DTD.

FIGURE 7. CREATE DTD- TAG CREATION



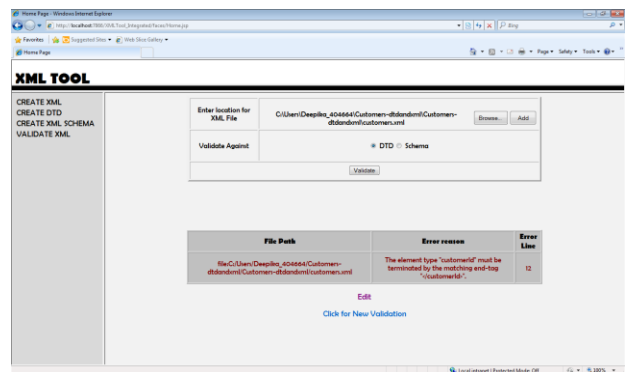
Create Schema-Tag Creation: The screenshot shows tag creation details for creating a XSD. The message displays which tag values the user is supposed to enter next. Figure 8 shows the tag creation of create schema. Figure 8 shows the tag creation of create schema.

FIGURE 8. CREATE SCHEMA-TAG CREATION



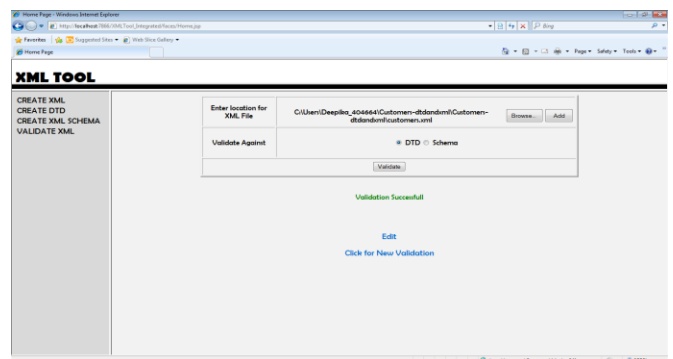
Validate XML- Validation with Errors: The figure shows the screenshot when the validation of XML against a DTD/XSD shows errors. Here again, a table with the list of errors is displayed. Figure 9 shows the validation with errors of validate XML.

FIGURE 9. VALIDATE XML- VALIDATION WITH ERRORS



Validate XML-Validation Success: The screenshot shows the success message screen when validation of XML is successful. Figure 10 shows the validation success of validate XML.

Figure 10. Validate XML-Validation SUCCESS



IV. CONCLUSION

Thus, the “Xml Tool” application created provides an interface that is simple to work with XML, DTD and XSD. Each program is tested individually at the time of development using the data and has verified that this program linked together in the way that is specified in the programs specification,



the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user. And so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly. All the suggestions forwarded in the software proposal have been completed and the final threshold of the application has been crossed. Error correction and modification can be done very easily for the future enhancement. This software may be modified as per management requirements and future needs.

In short, the proposed system acts efficiently and to add more to its features, enhancements can be made in the following areas,

- Provision to create the XML file for a particular DTD / XSD can be included for better performance.
- Editors to directly create XML, DTD, XSD files can be added to improve the efficacy of the application.

REFERENCES

1. Brett McLaughlin, Justin Edelson (2003). Java and XML, Third Edition, Solutions to Real-World Problems, Publisher: O'Reilly Media.
2. Ramakrishnan, S., & Devaraju, S. (2017). Attack's Feature Selection-Based Network Intrusion Detection System Using Fuzzy Control Language, Springer-. International Journal of Fuzzy Systems, 19(2), 316–328.
3. Chaudhuri, S., Chen, Z., Shim, K., & Yu, Y. (2007). U.S. Patent No. 7,228,312. Washington, DC: U.S. Patent and Trademark Office.
4. Damiani, E., Samarati, P., Di Vimercati, S. D. C., & Paraboschi, S. (2001). Controlling access to XML documents. IEEE Internet Computing, 5(6), 18–28.
5. Devaraju, S., & Ramakrishnan, S. (2011). Performance Analysis of Intrusion Detection System Using Various Neural Network Classifiers. IEEE Proceedings of the International Conference on International Conference on Recent Trends in Information Technology (ICRTIT 2011), Madras Institute of Technology, Anna University, Chennai, India (pp. 1033-1038). IEEE.
6. Joshi, D., & Patel, S. V. (2016). AXSM: An Automated Tool for XML Schema Matching. In Proceedings of International Conference on ICT for Sustainable Development (pp. 299-308). Springer, Singapore.
7. Roger S. Pressman, (2001). Software Engineering-A Practitioner's Approach, Fifth Edition, McGraw Hill International Edition.
8. Devaraju, S., & Ramakrishnan, S. (2013). Performance Comparison of Intrusion Detection System using Various Techniques – A Review. ICTACT Journal on Communication Technology, 4(3), 802–812.
9. Lesch, V., Diddens, D., Bernardes, C. E., Golub, B., Dequidt, A., Zeindlhofer, V., ... & Schröder, C. (2017). ForConX: A forcefield conversion tool based on XML. Journal of computational chemistry, 38(9), 629-638.
10. Naito, O. (2017). A browser-based tool for conversion between Fortran NAMELIST and XML/HTML. SoftwareX, 6, 25-29.
11. Devaraju, S., & Ramakrishnan, S. (2014). Performance Comparison for Intrusion Detection System using Neural Network with KDD Dataset. ICTACT Journal on Soft Computing, 4(3), 743–752.
12. Devaraju S. & Ramakrishnan S. (2015). Detection of Attacks for IDS using Association Rule Mining Algorithm. IETE Journal of Research, 61(6), 624-633.
13. Deng, J., & Hongxing, L. (2018, October). Research and Development of a XML Modeling Tool. In 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES) (pp. 318-321). IEEE.
14. Gaiffe, B. (2017, August). A Tool for Mixing XML Annotations. In Proceedings of the 2017 ACM Symposium on Document Engineering (pp. 105-108). ACM.
15. Devaraju S. & Ramakrishnan S. (2013). Detection of Accuracy for Intrusion Detection System using Neural Network Classifier,

International Journal of Emerging Technology and Advanced Engineering, 3(1),338-345.

16. Jillings, N., Man, B. D., Moffat, D., & Reiss, J. D. (2015). Web Audio Evaluation Tool: A browser-based listening test environment.
17. Devaraju S. (2019), Evaluation of Efficiency for Intrusion Detection System using Gini Index C5 Algorithm, International Journal of Engineering and Advanced Technology (IJEAT), 8(6), 2196-2200.

AUTHORS PROFILE



Dr. S. Devaraju received the B.Sc degree in Chemistry in 1997 from the University of Madras, Chennai, and the M.C.A. degree in Computer Applications in 2001 from the Periyar University, Salem, and the M.Phil. degree in Computer Science in 2004 from Periyar University, Salem and also received M.B.A. degree in Human Resource from Madurai Kamaraj University, Madurai in 2007. He received his Ph.D degree in

Science and Humanities from Anna University, Chennai in 2017. He has 16+ years of teaching experience and 2 years industry experience. He is an Associate Professor, Department of Computer Science and Applications, Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India. Dr.S.Devaraju is a Reviewer for various reputed Journals and Conferences. He has published more than 10 papers in international journals and conference proceedings. His area of research includes Network Security, Intrusion Detection, Soft Computing, and Wireless Communication.



Mr. D. Saravana Prakash, received his M.Sc., degree in computer Science from Jamal Mohamed College, Tiruchirappalli, India in 2001. He completed his M.Phil., degree in computer Science from Bharathidasan University, Tiruchirappalli, India on 2012. Presently he is pursuing PhD degree in Computer Science in Dr.G.R.D. College of Science, under Bharathiar University, Coimbatore. He served as a Faculty of

Computer Science at various institutions since June 2001. Presently he is working as Assistant Professor in Computer Science and Applications at Sri Krishna Arts and Science College, Coimbatore, India. He has presented papers in International/National conferences and published three papers in International journal. He also published a text book for MCA under Gujarat Technological University, Gujarat. His research focuses on Test Case Optimisation for Big Data Performance.