International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol. 8, Issue.7, July 2020 E-ISSN: 2347-2693

A Review on Soar Programming

Nitin[1*], Brij Bhushan[2], S. Srinivasan[3]

[1,2]Department of Computer Science, Mewar University, Gangrar, Chittorgarh, Rajasthan, India
[3]Department of Computer Science & Applications, PDM University, Bahadurgarh, Haryana, India

[*]*Corresponding Author: bansal.nitin12@gmail.com, Tel.: +91-9467922442*

DOI: https://doi.org/10.26438/ijcse/v8i7.111115 | Available online at: www.ijcseonline.org

Received: 05/July/2020, Accepted: 20/July/2020, Published: 31/July/2020

Abstract— the objective of this paper is to review on soar cognitive architecture's programming aspect.to create the intelligent agents for solving any artificial intelligence problem. we discusses the procedure with sequential steps in which soar execution cycle executes within soar programming .The purpose of this paper is to give a programmer's fundamental understanding of how a running Soar program works, with as little theoretical baggage as possible. This will involve stepping through a program's behaviour and describing what's happening, with special attention to the "hidden" parts of Soar.

Keywords— VisualSoar, Soar Debugger, Production Rules, Chunking, datamap, Subgoaling, States .

I. INTRODUCTION

Cognitive architectures is a part of artificial intelligence which introduced in 1950s with the motive of creating programs that could cause behind problem across various domains, accommodate to new situations and flash on them. An integrated cognitive architecture stated as a single system that is efficient of generating all aspects of behaviour, however remaining constant across various domains and knowledge bases. This system would made up of many modules working together to generate a behaviour. These modules enclose representations of knowledge, memories for storage of content and processes utilizing and acquiring knowledge [2].

Cognitive architecture is the concept about the structure of the human mind and how they work together to handle intelligent behavior in complex environment. It is a blueprint for intelligent agents. Its main motive is to have artificial computational system processes that behave like human or natural cognitive systems. It contains short term memory and long term memory. In short term memory, information is structured as a symbolic graph structure and on the other hand, production rules are put down in the long term memory.

Different architectures offer varying features and benefits, which should be analyzed based on how and why they will be used. They all vary in performance, capabilities, infrastructure requirements, and cost, and all have their unique limitations and operating methodologies [17].

A Cognitive architecture composed of memories for storing knowledge, processing units that extract, choose, combine, store knowledge and languages for representing the knowledge that is stored and processed. This is a bare-bones introduction to the Soar programming language [9]. Expert System is a computer program with collection of facts, rule and knowledge about a particular domain. This emulates the decision making ability of a human expert and acts in all respects like a human expert [16].

The Soar theory was developed by Allen Newell, John Laird, and Paul Rosenbloom is about cognition i.e. how people think, how they solve problems and how they learn. The specialized programming language, based on the Soar theory is soar language. It describes that how the knowledge that people have about the world is mentally represented. The soar system "runs" models written in the Soar language. Soar is also used as an artificial intelligence system, which can control robots or other machines.

II. RELATED WORK

AI is making systems that can enable masters to give progressively right or capable assurance for undeniable conditions [19].

In variety of application of Artificial Intelligence such as speech recognition, bio informatics, computer vision, and Machine Learning (ML) [18], the cognitive architecture have been used efficiently for implementing expert system. Soar architecture is one of cognitive architectures which we studied for our work. It consists of following characteristics.-nit

A. Productions

In a Soar program, all knowledge about the world and what to do there is encoded as productions and so the Soar is a production system. Productions are in the form of if-then pairs. E.g.: IF it's cloudy THEN take Umbrella. IF it 1:30 p.m. THEN its lunch time. When soar system runs the Soar

program, the THEN side of each production is fired only when IF side matches with the current state of the world. By using the productions, a Soar model can solve any problem that a human could.

B. Subgoaling

Subgoaling is a powerful technique, and Soar does this whenever it can't decide what else to do. Whenever it has a problem deciding what to do next, Soar sets itself the subgoal of resolving that problem [13]. The technique of setting the main problem aside in order to work on a lesser problem that's blocking progress is called "subgoaling." When the soar system creates a subgoal, it changes the state of the world in its internal representation.

Example: 1. if A& B then C
2. If D then B.

To prove 1. We will subgoal 2.

C. Chunking

Soar learns from its deliberations. Every time Soar resolves a problem in a subgoal, it remembers the solution. The next time the same problem occurs, that memory is available and Soar can avoid the time and trouble of working in the subgoal [14]. It is stored as "chunks." Chunks are nothing more than productions produced by the soar system instead of the programmer [12].

D. Soar Input/Output

These models need some way of interacting with the world outside of the computer, in order to acquire their new knowledge. The soar system provides input and output routines for this

III. METHODOLOGY

The cognitive architecture soar's editor is used for this purpose. They're defined in the official Soar User's Manual. Here we tried to show each and every step starting from Artificial Intelligence problem formulation to complete execution.

In this, we use two editors Visual soar for writing code to create the agent and SOAR Debugger, which is used to run the created agent. In our work we tried to make the agent intelligent.

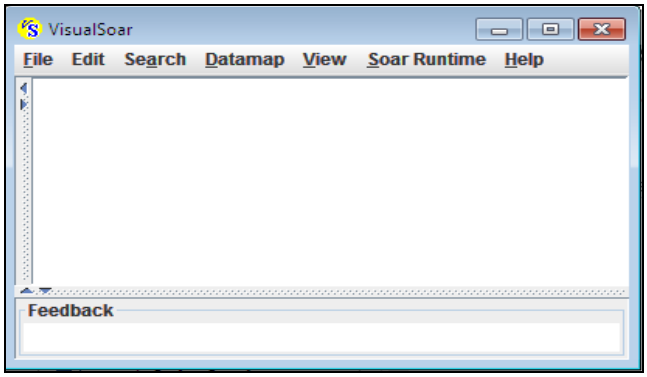

Figure 1. Visual SOAR: to write code for agent creation

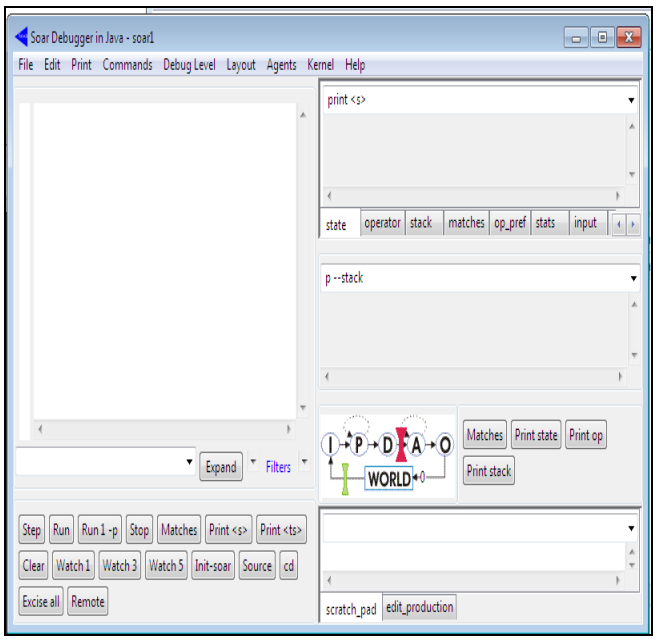

Figure 2. SOAR Debugger: to run the agent

- Go to visual soar
- Click on file …new project…. Give name of agent …..Click on new.

You will get this screen.

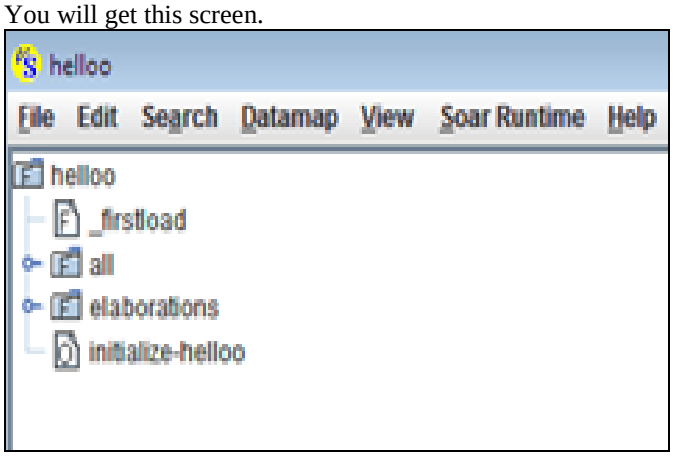

Figure 3. Agent Created

You will see a new window on the left that should contain 5 things in the form of a tree. This is called the operator window [4]. At the root is the name of your project, and the four things below it are the default files that are automatically created.

In first load file type the soar production (Basic Syntax) as :

Sp {rule*name
(condition)
(condition)…..
-→
(action)
(action)…..}

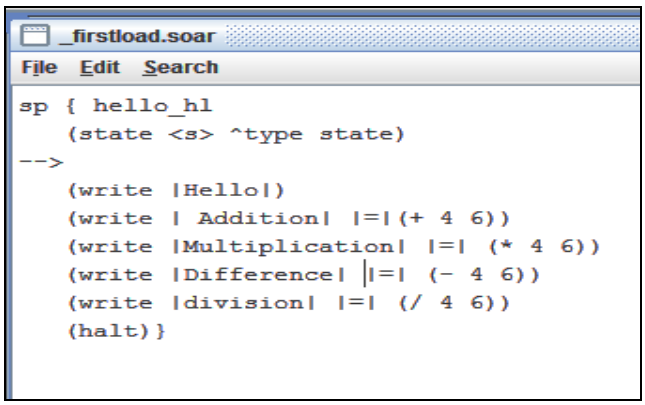

Figure 4. Soar production for arithmetic operations

Then save the file and Go to soar debugger to run the file. Alternatively, click on the "source" button at the bottom of the interaction window.

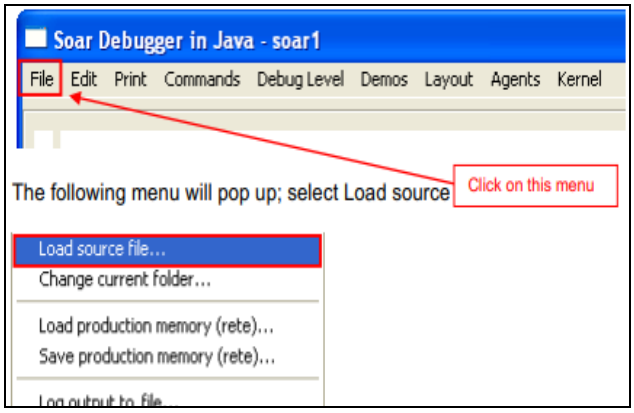

Figure 5. running of an agent

You can navigate in this menu to get to the correct directory (/Agents/) and then select the file you want to load (e.g. hello-world-rule.soar).

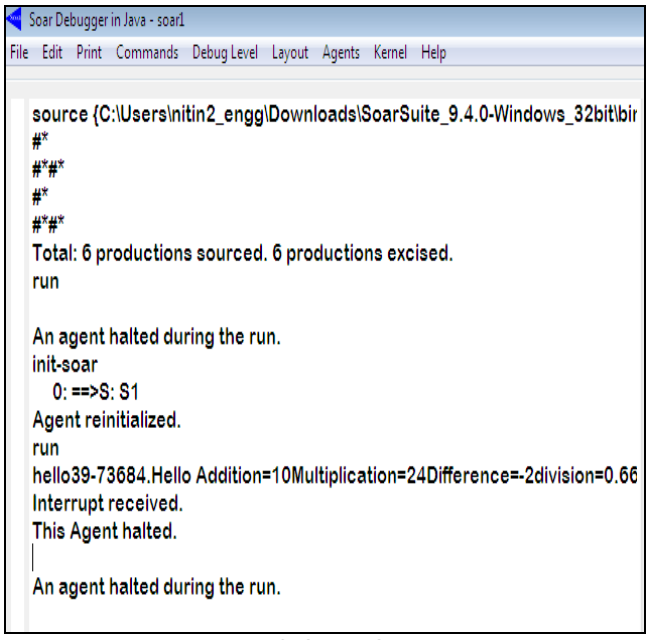

Figure 6. Output Screen

Visual Soar has a structure called the Data map for describing working memory structure [10]. In some ways, it is like making type definitions in other languages. To observe the Data map, right-click on the root of the tree entitled "production name" and click on Open Data map in the pop-up window.

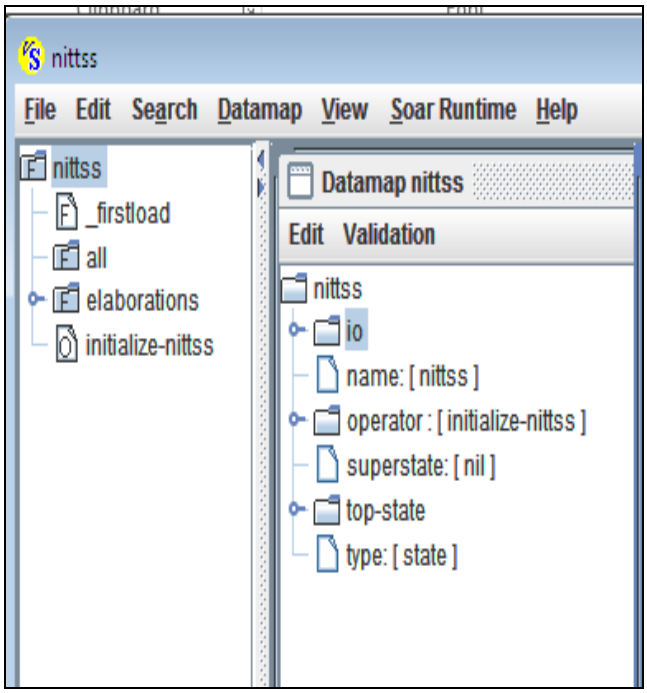

Figure 7. Data Map

Thus, states will include the following structures:
An object for each number (^num).
The 'a' value of water number (^val).

Production will be :

(state <s> ^num <n1>
^num <n2>)
(<n1> ^val 23)
(<n2> ^val 34)

As with every operator, we must define two types of rules: one to propose the operator and one to apply the operator. This operator should be proposed only at the beginning, before any task is selected [7].

In the 'initialize-nittss' rule window you will notice that Visual Soar has attempted to write this rule for you also. The application of initializing is not as standard as the proposal and Visual Soar is not able to initialize all of the attributes that this project will need. Modify the second rule in the rule window to match the rule below. Now, at the top of Visual Soar, click on Data map | Check All Productions Against the Data map.

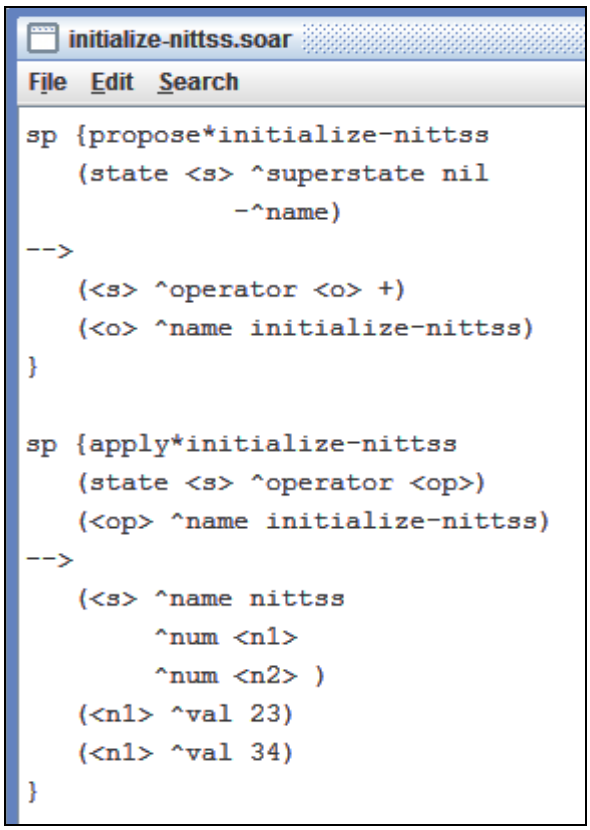

Figure 8. Initialize production Rule

You must now add entries into the Data map for the water-jug state structure.

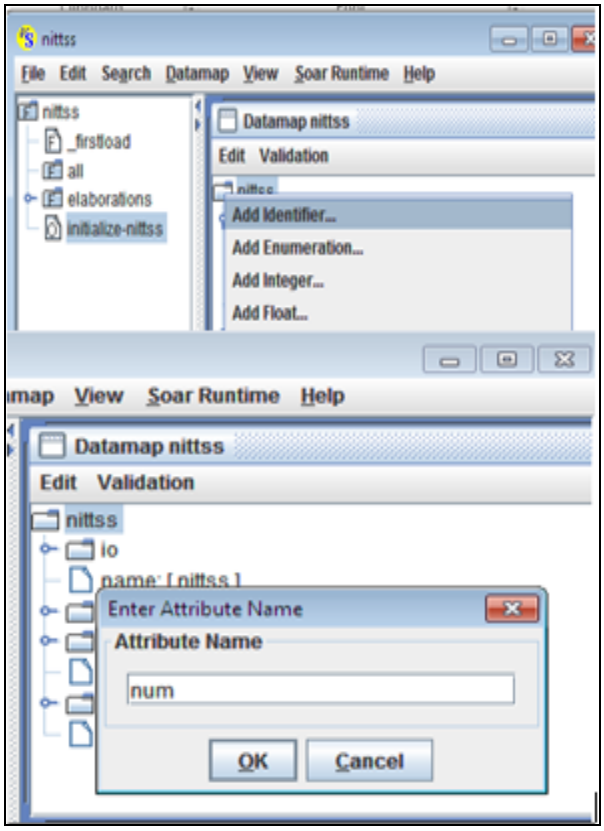

Figure 9. Step to add identifier

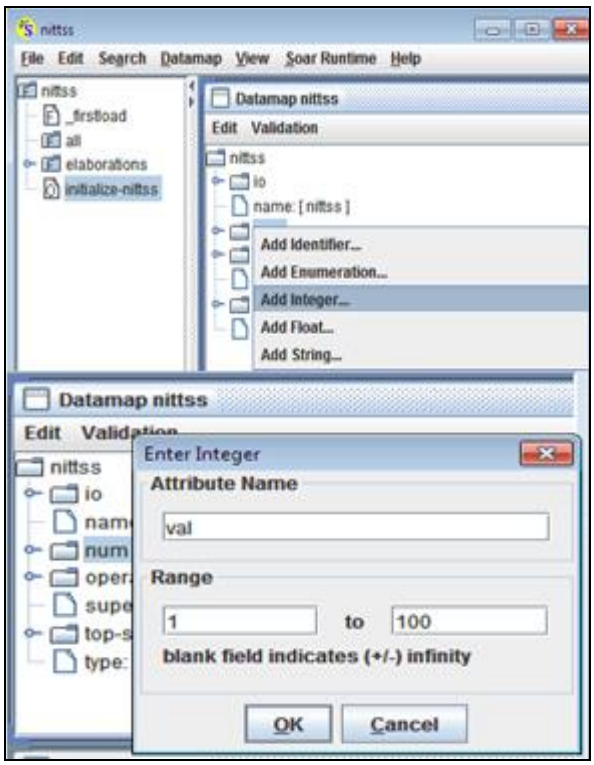

Figure 10. Step to add value to identifier

Now check all productions against the Datamap in visual soar by using the Datamap tab. you should have no errors.

Create elaboration file by following step.

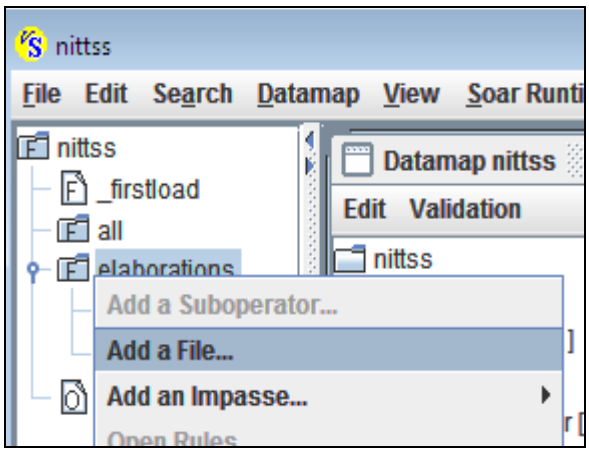

Figure 11. Step to create elaboration file

The state elaboration will compute the total of two numbers: elaborations*elaborate*calc
If the state is named nittss and a num1 has Val v1 and num2 has Val v2 then add v1+v2.
Visual Soar has a folder named elaborations in the operator window to hold files for state elaborations. Right-click on this folder and click "Add a File…"

You can add the rule from scratch and just type it in, or you can use a template [8]. To use a template, click on Insert Template in the toolbar and select elaborate-state.

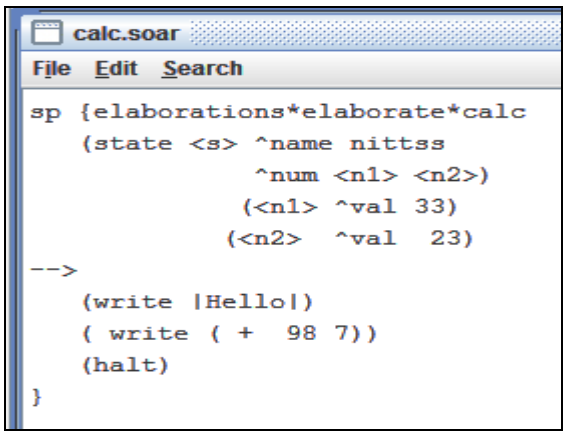

Figure 12. Soar production for elaboration to add two numbers

Now run the file in soar debugger.

IV. CONCLUSION

Our goal is to look at the knowledgeable overview on the programming structure and working of soar cognitive architecture. This paper offers the principal arrangement that would facilitate a system to accomplish the cognitive tasks, providing the methods for problem and task appropriate representations, and tried to acquire all traits of the tasks and also their performance.

This paper, presented a review in the field of soar programming by demonstrating addition of two numbers. From here, we can get knowledge to write programs in soar. For general intelligent agents a fixed computational building block is necessary. To map with the human's cognitive capability the intelligent agents can perform various tasks and learn, use and encode all the knowledge used by humans.

REFERENCES

[1] Laird, J., "Extending the Soar Cognitive Architecture", Artificial General Intelligence Conference, Pp **224-235**, and ISBN: **978-1-58603-833-5, 2008**.
[2] Laird, J. and Rosenbloom, P., "The Evolution of the Soar Cognitive Architecture", *Mind Matters*, T. Mitchell (Ed.),Pp-**1-50, 1996.**
[3] Nuxoll, A. and Laird, J., "Extending Cognitive Architecture with Episodic Memory", **22**nd National Conference on Artificial Intelligence **(AAAI), 2007**.
[4] John rieman., "an introduction to soar programming", mrc-apu **15 march 1995**.
[5] John laird, A Soar's eyeview of ACT-R, 24th soar workshop, **june 2004**.
[6] James, Working memory element (WME) Decay in Soar: University of Michigan, Pp **1-50**
[7] Rosenbloom, P. S., Laird, J. E., & Newell, A. (1993) The Soar papers: Research on Integrated Intelligence.MIT Press, **Cambridge**, MA, **1993**
[8] Nuxoll, Laird and james, Comprehensive working memory activation in SOAR, **Pp226-230**.
[9] Stuart J. Russell, Peter Norvig ,"Artificial Intelligence: Modern Approach" by Practice Hall Series in Artificial Intelligence, Pp-**22-40, 2006.**
[10] Laird, J. E., "Soar9 Tutorial Part, the Soar 9 Tutorial", University of Michigan, **1—44, 2014.**
[11] Bansal. N,Srinivasan. S., March 30, 2013, "Multi–Memory System: The Cognitive Architecture SOAR", **NCACT-2013.**
[12] Nason, S., & Laird, J. E. (2005) Soar-RL: Integrating reinforcement learning with Soar. Cognitive Systems Research, 6(1),Pp **51-59, 2005.**
[13] Nuxoll, A. & Laird, J. (2004). A Cognitive Model of Episodic Memory Integrated With a General Cognitive Architecture. International Conference on Cognitive Modeling **2004**.
[14] Tulving, E. (1983) Elements of Episodic Memory. Oxford: Clarendon press, **1983**.
[15] Kieras, D. & Meyer, D. E. (1997) an overview of the EPIC architecture for cognition and performance with application to human-computer interaction. Human-Computer Interaction, 12, Pp:**391-438**, **1997**.
[16] M.M. Mastoli, U.R. Pol, Rahul D. Patil, "Reasoning with Certainty Factor for Prediction of Diabetes Disease on Machine Learning Platform," *International Journal of Scientific Research in Computer Science and Engineering*, Vol.**8,** Issue.**1,** pp.**93-97**, **2020**.
[17] A.K. Bhatia, H. Kaur, "*Security and Privacy in Biometrics: A Review*", International Journal of Scientific Research in Computer Science and Engineering, Vol.**1**, Issue.**2**, pp.**33-35**, **2013**
[18] N. SelvaKumar, M. Rohini, C. Narmada, M. Yogeshprabhu, "Network Traffic Control Using AI," *International Journal of Scientific Research in Network Security and Communication*, Vol.**8,** Issue.**2**, pp.**13-21, 2020**
[19] Hemant Kumar Soni, "Machine Learning â€" A New Paradigm of AI," *International Journal of Scientific Research in Network Security and Communication*, Vol.**7**, Issue.**3**, pp.**31-32, 2019**

AUTHORS PROFILE

Ms..Nitin is a PhD student in the Department of Computer Science, Mewar University. She earned his MCA & M.Tech degrees from Maharshi Dayanand University, Rohtak, India. And M.Phil from Chaudhary Devi Lal University, Sirsa, India. Her research interests include Artificial intelligence and Intelligent Agents.

Dr Brij Bhushan, is Ph.D. in Computational Mathematics from Indian Institute of Technology, Kanpur (India) Presently he is a Professor **at** Mewar University, has served India Meteorological Department, National Informatics Centre, Ministries of Agriculture, Animal Husbandry, Education and Industry in India and written 45 proposals and papers on Information Technology projects. Dr Bhushan introduced concept of N-Governance, Dynamic Ranking of Institutes, Human Transformation Index, MetGIS and VetGIS. Dr Bhushan has guided a number of Ph.D. students in the fields of Mathematics, Disaster Management and Computer Applications.

Dr. S. Srinivasan e completed his PhD from the Madurai Kamraj University, Madurai in 1978. Presently, he is a Professor at the PDM University, Bahadurgarh, Haryana – 124507. His research interests include artificial intelligence, software engineering, software testing etc.

