European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 07, Issue 06, 2020

Analysis of Finite State Automata For Sequence Mining

Omkar Singh[1], Anant Tiwari[2], Ashutosh Jaiswal[3], Roushan Kumar[4]

*[1]Assistant Professor, Maharaja Agrasen College
Vasundhara Enclave, University of Delhi.
omkarsinghlodhi@gmail.com*

*[2]Assistant Professor, Mathematics department,
ARSD College, University of Delhi*

*[3]Assistant Professor, Maharaja Agrasen College
Vasundhara Enclave University of Delhi
sjaiswal111@gmail.com*

[4]Lecturer. MBIT , Nct of Delhi

Abstract
Finite state automata (FSA) have a very understandable mathematical model; data can be compactly displayed using a final state automaton as well as automatically compile the system components. Sequence Pattern mining is useful for data mining &is important for a wide range of applications including the buyout sequence of consumers. Sequence Mining (SM) means the sequence patterns of the broad dataset are identified. It finds common substrings from a dataset as patterns. Most industries are interested in scan sequence patterns in their databases with massive data continuously collected. Data mining (DM) is one of the methods by which hidden patterns linked to instant sequences are recovered. We extract sequence models in sequence mining that are larger or equivalent to min support threshold value of supported patterns. In this paper, we discussed various types of automata types, automata in data mining, finite state automata, sequence mining, etc.

Keywords -- Data Mining, Automata, Finite State Automata, Deterministic Finite Automata, Sequence Mining

I. INTRODUCTION

Data Mining (DM) is distinct as the method of analyzing huge databases it is consist of many fields, working in different areas including database, Statistics, pattern recognition, retrieval of information, bioinformatics, recognition, bioinformatics, machine learning, graph mining, knowledge-based systems, high-performance computing, visualization of data & high-performance computing. The extraction of information from large volumes of data is often demonstrated by DM [1]. DM is a sequence of techniques & procedures that can be used from various data resources, such as data store and partner repositories to flat, format files, using methodologies of statistical studies to predict or predict predictive confidence measures based on existing information. DM "is, according to Fayyad, a nontrivial method of true recognition invention that can support and understand comprehensible trends embedded in data" The data mining mechanism is often seen in the development of IT (Jiawei & Kamber, 2001). [2].

The growth of automaton theory is a historic consequence of the technological development of computer arts, but its commitments are significant in its development. It is a historical outcome from a particular point of view. The idea of the auto should however not be confused with a description of a machine; therefore, a theory of automaton is an ideally established principle. Her appropriate interaction with computer technology is fundamental to IT science [3].

Automaton is a discrete, dynamic system that performs calculations in an excellently dispersed way on a spatial lattice. Conway's life game, introduced by Gardner in Scientific American, is most certainly the best-known scenario for an automaton. Wolfram & others have typically found Automata. While individual cells in an automaton are incredibly basic, they can all contribute to complex new conduct and can produce many forms of self-associations. Automata is also used to measure small occurrence space attributes, e.g., purported thickness & demanding problems Automata was also used as a part of example acknowledgment to perform include extraction and acknowledgment [4].

Finite-state automates emerged as models of discrete information transducers, that is to say, models, interacting with their environment. An example of such an interaction is automatic picture languages and labyrinth automatons, automated communications, multicenter automatons, and a computer model in the form of an interaction between a controller. FSA term defines a class of models with a limited number of states. Information is encoded through symbol sequences. [5]
The accessibility of these databases has generated significant interest in the issue of extracting valuable information after data. Many data are in particular sequential, so sequence mining techniques are required to explore sequence patterns. Within this paper, we demonstrate how a sequential data set is structured & Why this model can then be implemented to offer effective frequency query options for various data patterns. [6].

II. AUTOMATA FOR DATA MINING

Automata are graded according to whether or not they will grow to be large at random. In what it can do, an automaton that develops but cannot become arbitrarily big turns out to be no more efficient than a nongrowing automaton, 2 interesting classifications than maybe infinite increasing automata ("growing automata" for short) and fixed automata.

They use automatons as a type of case-based learning, where cells are formed to speak to parts of example space. Cells are organized & connected to a range of property estimation. The space for this occasion will form a (multi-dimensional) gill that the Automata operates over. Training instances are set in the framework and the Automata can be applied. The state of each Automata cell in the case space speaks to the class task at that point. It is expected that cells will be divided into districts with a common classification. We use straightforward voting to determine that entropy is reduced locally.

A. Finite Automata

Generally speaking, the output of an automaton does not only respond to the input then but also to the history of the input. So, such an automaton needs to provide memories in some way or another. One important theoretical principle in this regard is the idea of state: an automaton will exist in different states at different times and in response to something occurring inputs, it recalls what has occurred by moving from one state to another. The overall number of different states shows the amount of memory that the automaton has. Both memories were described in terms of states such that the value of every output at any time only depending upon input values at that time & on a state; only its determination of present state is entirely determined by the previous input history of the present value of the output. [3].

B. Deterministic Finite Automata

Deterministic Finite Automata (DFA) is quintuple M= (Q, ∑, δ, q_0, F), in which Q is a finite set of states, = = finite set is called alphabet, $q0 \in Q$, a distinct state known as the beginning state, F is a subset of Q known as final or accepting state, and δ: $Q \times \sum \rightarrow Q$, named transition function.

A broader transformation function can be defined. We describe functionα ltd. as $QX\sum^{*} \rightarrow Q$. Let M= (Q, ∑, δ, q_0, F) be DFA.

Aimed at some q$\in Q$, δ^{*} q, $\lambda = q$.

For any q $\in Q$,

$y \in \sum^{*} anda \in \sum then\delta^{*}$ q, $ya = \delta$ δ^{*} q, ya.

M if δ^{*} $q0$, $x \in F$, will consider string Hence, deterministic, finite automatons as abstract machines, language are recognized by DFA M is that group. DFA function is denoted as mechanisms in some popular computer systems [7,8,9].

III. FINITE STATE AUTOMATA (FSA)

At this point a short introduction to the concepts borrowed from the Finite State Automata theory is necessary. A Finite State Automaton is a computation model that comprises a series of states, a starting state, an input alpha, and a transitional function mapped to the next state by the input symbols as well as current conditions. Computation begins with an input string in the start state and varies according to the transition function to new states.

The variant used in the proposed methodology is a DFA, that has for every symbol and state at most one transition. A DFA *M* is defined as a 5-tuple of the form:

$M = (K, \Sigma, \delta, s, F)$, where: … (1)

K is set of states,

Σ is input alphabet,

s □ K is start state,

F □ K is set of end states and

δ is transition function as of K □ Σ to K.

An input string (composed of symbols contained in Σ) is accepted by the DFA when, while reading it, M goes over (according to δ: K □ $\Sigma \rightarrow K$) from state to state (starting from state s) and ends (exhausting the input string) to any state that belongs to the set of end states F. The most important aspect of the DFA, with regards to other finite-state variants, is that from a state q and using the same symbol, the next state is always a specific state q'. In other variants, such as the non-deterministic finite-state automata, the transition from a state q using the same symbol, may lead to more than one next state q'.

Finally, another form of DFA is the stochastic deterministic finite-state automaton (S-DFA). The main difference between a DFA and an S-DFA is that the latter introduces the concept of transition probability, i.e. probability of transition amid states according to δ: K □ $\Sigma \rightarrow K$. result of this probability is that some transitions are favored over others [10].

A. Estimating pattern frequencies from an FSA

During this section, we provide several samples for calculating the probabilistic finite-state automaton probability of particular patterns throughout sequences. Such possibilities are then used in SM algos.

Using the n-gram sequence of symbols, we can describe a pattern. If n-grams appear in proper order within a sequence, SM pattern. We usage corner brackets to mark patterns. As, pattern < x, yz, w > coincides with a sequence with x, a yz, or w in order symbols. Any sequence of an empty pattern is marked by □l. When p1 & p2 are 2 patterns, p1 + p2 leads to 2 patterns being combined.

For automatons, we want some notation. If S is an automobile & x is non-terminal, we'll write on any S transformation.

p(S, x) = probability (P) that x is symbol emitted,

q(S, x) = next state after x is emitted.

A different automated illustration program will also be used. Select transitions randomly with the same probability. They used an automated inference to recover an automated structure from the sample data. automatic inferred, as seen in fig 1, can calculate transition probabilities from the data:

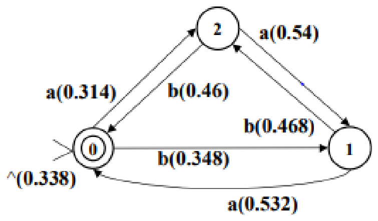

Figure 1: Inferred Automaton with Transition Probabilities

Example 1

“How much sequence does symbol x contain?”

Perhaps the easiest question we might pose is this one. The question may be reassessed in terms of patterns asking about the probability of a pattern. This is how we can measure it. Firstly, we define S as well as x for each state.:

P(S,x) = P that random path from state S includes x.

Today track after S with an x starts by x (with probability p(S, x)) or somewhere else by a path that starts at the next level with an x. So, the formula can be written

$$P(S,x) = p(S,x) + \sum_{z \neq x} \left(p(S,z) \times P(q(S,z),x) \right) \cdots (1)$$

This formula for the automaton is shown in Fig 5 for representation a:

P(0,a) = 0.314 + 0.348×P(2,a)

P(1,a) = 0.54 + 0.46×P(0,a)

P(2,a) = 0.532 + 0.468×P(1,a)

Hence, We've got a system of linear equ. to find P(S, x) that can be solved. The original answer to the problem is just P(0,x).

Equ. (1) may be revised as occurs:

$$P(S,x)=p(S,x)+\sum\left(p(S,z)\times P(q(S,z),x)\right)$$

$$=p(S,x)+\sum_{T}\left(\sum_{\substack{z\neq x\\ q(S,x)=T}}p(S,z)\right)P(T,x)\cdots(2)$$

This can be written as follows in a matrix. Initially, set µ(x) by first

$$\rho_{S,T}(x)=\sum_{\substack{z\neq x\\ q(S,x)=T}}p(S,z)$$

Aimed at automaton in Fig 5, we get

$$\rho(a)=\begin{bmatrix}0 & 0.348 & 0\\ 0 & 0 & 0.468\\ 0.46 & 0 & 0\end{bmatrix}$$

P(x) is specified for P(S, x) & p(x) values matrix for p(S, x) values. (2) is then (2):

$$P(x)=p(x)+\rho(x)P(x)$$

Rearranging, we get

$$P(x)\ (I-\rho(x))\ p(x)$$

The matrix on the right side is useful, and J(x) is a useful matrix. We therefore reiterate the above formula as

$$P(x)=J\ (x)p(x)$$

We have a car in Figure 5:

$$J(a)=\begin{bmatrix}1.081 & 0.376 & 0.176\\ 0.233 & 1.081 & 0.506\\ 0.498 & 0.173 & 1.081\end{bmatrix}$$

So that:

$$P(a)=\begin{bmatrix}1.081 & 0.376 & 0.176\\ 0.233 & 1.081 & 0.506\\ 0.498 & 0.173 & 1.081\end{bmatrix}\times\begin{bmatrix}0.314\\ 0.54\\ 0.532\end{bmatrix}$$

$$=\begin{bmatrix}0.635\\ 0.921\\ 0.832\end{bmatrix}$$

Therefore, 0.635 of a lowered automaton is the estimated proportion of cables holding a symbol. In sample data gathered by counting, the proportion of sequences is 0.63. In Fig 1, the expected ratio of this original car is 0.63 to equal probabilities by each symbol. [11].

IV. SEQUENCE MINING AUTOMATA

Sequence Mining means the sequence patterns of the broad dataset are identified. It finds common substrings from a dataset as patterns. Numerous industries are interested in sequential patterns from their databases because massive volumes of data are continuously collected. Sequence pattern mining is among the most common methods and has wide-ranging applications including web analysis, consumer computational analysis, and medical record analyses.

A little example of our approach is given here. Suppose we have a four-sequence dataset drawn allegedly after larger array:

I = {abbaaaa, ab, ba, aaaaabb}.

Assume that a bigger set is a regular language, so a finite state automaton will generate. Figure 2 displays an automatically created state diagram.

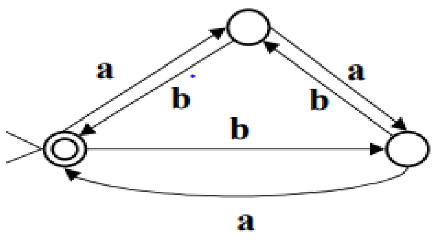

Figure 2: Generating Automaton for I

The fig depicts circles, & marked arcs reflect transitions amid them. We survey standard practice to mark the beginning state with a">" & for FS, use double circles. From the start point, paths are traced by transitioning from one point to the next through the graph. As transformations pass through, the corresponding symbols are released. In any final state, paths can stop. If the figure shows the right automatically, we can deduce information from the larger collection. For instance, the number of a's in a single sequence & no. of b's is congruent with modulo 3 (in this case, this property defines a larger collection). Ex., we can deduce the predictable proportion of sequences initial from symbol an or proportion of a sequence occurred later in b if the likelihoods were given to automatic transitions (later we will see how these calculations can be made).

We usually look for patterns with the lowest possible degree of support in the data mining environment (no. of confidence (a proportion of such sequences corresponding to the pattern). Support and confidence may be approximated via predicted proportions calculated [11].

A. A Sequence Mining Algorithm

SMA using formulae is obtained in this section. We'll look for rules such as

If a sequence has x & a y, then a subsequent z may be included.

This rule is to be written <x, y> □ z.

We'll more commonly detect patterns that display certain symbols. We will need sufficient support for a rule (the probability that the model will happen& confidence (the probability that subsequent Z takes place, presented that model happened). In notation applied in example 2, support (Sup) of the rule above is *P(<x,y>),* whereas confidence (Con) is *P(<x,y,z>)/P(<x,y>).* In Example 2 above, we can use a formula to measure the probabilities for such patterns.

Assume the level of support, s, as well as confidence, c were indicated. The wide pattern is described as one with S of as a min s. Large patterns have an Apriori property so that we can make searching efficient using an Apriori-like method.

Algo Apriori-Pattern

```
Inputs:
        A probabilistic FSA
        S (Sup)
        C (Con)
Output:
        Set of rules S
begin
    M = {}
    $K_1$ = {}
    do ( □s □ □ )
compute F(s) and P(s) for automaton
 if P(start, s) □ Con then
    M = M □ {□ □ s}
if P(start, s) ) □ Con then
     $K_1$ = $K_1$ □ {<s>}
end do n = 1
    while $L_n$ is not empty do begin
    $L_{n+1}$ = {}
    do ( □□ □ $L_n$ )
do ( □s: <s> □ $L_1$ )
              □' = □ + <s> P(□') = F(□)P(s)
          if P(start, □')/P(start, □) □ Con then
           M = M □ {□ □ s }
          if P(start, □') □ Sup then $L_{n+1}$ = $L_{n+1}$ □ {□'}
    end do end do
    n =n+1 end while
end
```

V. LITERATURE SURVEY

Majed AbuSafiya [2020] This paper measures a text document by identifying 3 sets of words in a pair of text documents: words in the first but not in the second, but not in first, but works in both. 2 DFA are formed to recognize the terms of these 2 documents. These final state automatons are used for the identification of these three terms. The proposed method and comparison to other peer text similarity measures were conducted in an experimental study: jacard and cosine distance. The proposed approach provided a more rational outcome than Jaccard however more inflation than cosine space. [12].

Joseph Marvin R. Imperial [2019] The concept of a Tagalog spell checker has been revolutionary in this article, however, utilizes a list of words with 300 random root words & three inflected forms with DFA combined as training data and a two-layer layout of the Levenshtein editing distance. DFA is used to determine whether the binary outcome of accepting or rejecting in a given language is used for the processing of strings. The number(k) to be inserted, updated, entered between the two-character sequences is changed by Levenshtein. Results of qualified sampled wordlist demonstrate that the value of one may be fruitful in the wording of Tagalog phrases for edit distance (k). Any value greater than 1 may include words, but word spelling is correct since certain characters like a, n, g, t, s, l are written selectively and prominently in the Tagalog language. [13].

Trasarti et al. [2018] Consider the issue of the frequent mining sequences that meet a certain ordinary phrase. Previous strategies were based on the search field and the basic expression offered (in certain respects) was forced to prune promising candidate patterns. Rather than, they concentrate completely on data presented and on standard speech. They use SMA, a specialized Petri Net type, to generate all the sequential patterns for each sequence such that the sequences are interpreted with a certain regular expression. On this automated system, we build an algorithm family centered. Our thorough analysis into different data sets and application areas helps our approaches to frequent sequence mining algos in many cases overpower current state-of-the-art with regular expressions (in cases of magnitude orders). [14].

Manko et al. [2017] This paper gives a strategy for planning a sequence of independent robots in a poorly defined environment to address a multi-phase task unless the operational scenario is previously established. The multi-stage finite automaton model is specified and they give a design algorithm to build and execute the scenario in a dynamic way. The implications of a network of end-of-state machines do not allow for the planning of robotic acts but allow progress to be tracked in real-time. The experimental data of the paper thoroughly confirm the reliability of the proposed method [15].

Reddy et al. [2016] This paper introduce a modern pattern analysis algorithm based on a symbolic pattern approach. The classification proposal is based on the symbolic method and the FSA model. This algo represents data as well as divides the images into two dimensions. Features of a symbolic picture and finite automatic model are extracted. The texture images classification is a probabilistic based classifier. The experiment shows that better classification accuracy is given for the proposed method for texture imaging for different samples [16].

Xiaolin et al. [2014] The paper below provides an introduction to software behavior modeling, which relies on modern behavioral modeling and focuses on shortcomings of data values and traces of interaction between the software components. They merge daikon &

ESC/JAVA tools to increase model accuracy to achieve the constraints on longer final automated state boundaries. This technique was used for the simulation of machine behavior. Experiments show that this model can obtain much more precise requirements & data for the analysis, verification, and testing of software [17].

VI. COMPARATIVE ANALYSIS

To validate the correctness of the protea's method, the experimental procedure including random forest and decision tree were used.

Table I. Comparison of various methods in accuracy

No. of classes	Proteas	Gen-Miner	Alergia algorithm
10	93.71	80.13	81.13
20	94.31	84.27	78.27
30	75.99	74.5	77.5
40	70.81	68.3	66.3

The experiment was performed multiple times over the number of classes and the results shown in Table I.

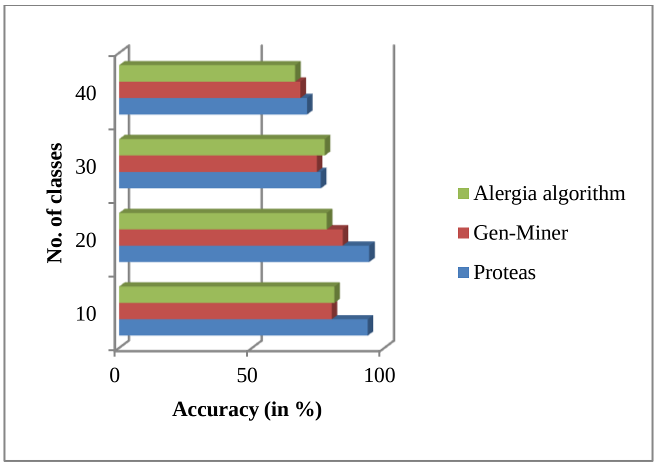

Fig. 3: Accuracy comparison over no. of classes

Fig. 2 describes that the different number of classes including various algorithms have different accuracy. Among them, the proteas method has the highest accuracy.

Each time a different portion of the initial dataset was used for training and the same experiment was performed multiple times. The average results are presented in Table II.

Table II. Comparative experiment

Train set (%)/Test set (%)	Proteas	Random forest	Decision tree algorithm
10/100	96.223	80.13	94
20/100	96.525	84.27	95.1
30/100	99.546	74.5	96.5
40/100	99.697	68.3	98.4

50/100	99.546	80.13	99
60/100	99.679	84.27	98.7
70/100	99.687	74.5	98.5
80/100	99.884	68.3	98.3
99/100	99.697	99.9	99.8

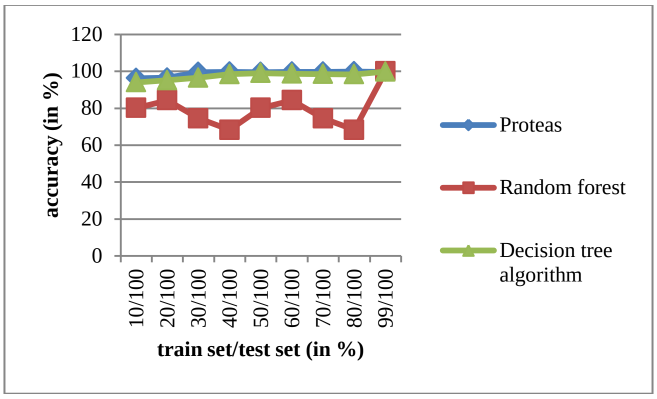

Fig. 4: Accuracy comparison over no. of classes

Fig. 3 depicted the train set percentage out of 100% test set. On each train set, the proteas, random forest, and decision tree have been compared. Each algorithm has a different accuracy. Among them, the proteas method has the highest accuracy i.e. approx 99.5 %.

From the results, it is obvious that classifiers' accuracy for the protein family available in prosite is not satisfactory. However, it must be noted that the classifier was built using the simplest rules possible owing to the high time complexity of the methodology. Notwithstanding this fact, the proteas method succeeds in accurately classifying.

VII. CONCLUSION

FA is a mathematical model of computing based on ideas, which is also known as FSA. We show how often sequential patterns from large datasets are found by initially triggering a finite automatic status model that describes data. Sequential pattern mining Algos in the vertical format is very successful because they can use expensive database scans to calculate support for candidate patterns. In this paper, we have compared various classifier methods to test accuracy. For these multiple experiments have been performed. From experimental outcomes observed the proteas method is the best one among all methods in terms of accuracy.

REFERENCES

[1] J . Han, M. Kamber, and J Pei. Data mining: concepts and techniques, Morgan Kaufmann, USA, 2006.

[2] Andrés Villanueva Manjarres," Data mining techniques applied in educational environments:", Digital Education Review - Number 33, June 2018- http://greav.ub.edu/der/

[3] https://sci-hub.tw/https://doi.org/10.1016/S0065-2458(08)60144-8

[4] https://www.airo.co.in/paper/admin/upload/international_volume/5508ROSHAN%20KUMAR%20000.pdf

[5] A. Mazurkiewicz, Trace theory In Petri Nets: Applications and Relationships to Other Models of Concurrency, Springer, pp.278-324, 1987.

[6] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.5295&rep=rep1&type=pdf

[7] S. A. ALI," Implementation of Automata Theory to Improve the Learning Disability", *Vol. 45 (1):1193-196 (2013),*

[8] https://sci-hub.tw/https://doi.org/10.1016/S0065-2458(08)60144-8

[9] Roushan Kumar," A Study Of Automata Based Data Mining Techniques", https://www.airo.co.in/paper/admin/upload/international_volume/5508ROSHAN%20KUMAR%20000.pdf

[10] https://pdfs.semanticscholar.org/b0e7/72950fcc8f0141e5a28832e3df1673f3da28.pdf?_ga=2.67484377.951520308.1591596494-212309397.1528353887

[11] Philip Hingston," Using Finite State Automata for Sequence Mining", DOI: 10.1145/563857.563814 https://www.researchgate.net/publication/49278896

[12] Majed AbuSafiya," Measuring Documents Similarity using Finite State Automata", 2020 International Conference on Mathematics and Information Technology, Adrar, Algeria, February 18-19, 2020 208

[13] Joseph Marvin R. Imperial," An experimental Tagalog Finite State Automata spellchecker with Levenshtein edit-distance feature", 2019 International Conference on Asian Language Processing (IALP)

[14] Trasarti, R., Bonchi, F., & Goethals, B. (2018). *Sequence Mining Automata: A New Technique for Mining Frequent Sequences under Regular Expressions. 2018 Eighth IEEE International Conference on Data Mining.* doi:10.1109/icdm.2008.111

[15] Manko, S., Diane, S., & Lokhin, V. (2017). *Task planning in robot groups for problems with implicitly defined scenarios based on finite-state automata technique. 2017 XX IEEE International Conference on Soft Computing and Measurements (SCM).* doi:10.1109/scm.2017.7970581

[16] Reddy, R. O., Reddy, B. E., & Reddy, E. K. (2016). A Probabilistic Finite State Architecture to Classify Texture Images. 2016 IEEE 6th International Conference on Advanced Computing (IACC). doi:10.1109/iacc.2016.60

[17] Xiaolin Zhao, Jingfeng Xue, Shanshan Zhang, Rui Ma, & Changzhen Hu. (2014). *Research on software behavior modeling based on extended finite-state automata. 2014 Communications Security Conference (CSC 2014).* doi:10.1049/cp.2014.0744