Journal of Theoretical and Applied Information Technology
31st January 2023. Vol.101. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN OPTIMAL MACHINE LEARNING MODEL BASED ON SELECTIVE REINFORCED MARKOV DECISION TO PREDICT WEB BROWSING PATTERNS

Dr. V. V. R. MAHESWARA RAO[1], N. SILPA[2], Dr. GADIRAJU MAHESH[3], SHIVA SHANKAR REDDY[4], Dr. VIJAYA KUMAR[5], Dr. K. B. V. BRAHMA RAO[6]

[1]Professor, Shri Vishnu Engineering College for Women, Department of Computer Science and Engineering, Bhimavaram, India
[2]Asst. Professor, Shri Vishnu Engineering College for Women, Department of Computer Science and Engineering, Bhimavaram, India
[3]Associate Professor, S.R.K.R. Engineering College, Department of Computer Science & Engineering, Bhimavaram, India
[4]Assistant Professor, S.R.K.R. Engineering College, Department of Computer Science & Engineering, Bhimavaram, India
[5]Professor, Amrita Sai Institute of Science and Technology, Department of Computer Science & Engineering, Vijayawada, India
[6]Professor, Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vijayawada, India

E-mail: [1]mahesh_vvr@yahoo.com, [2]nrusimhadri.silpa@gmail.com, [3]mahesh.cse.srkr@gmail.com, [4]shiva.shankar591@gmail.com, [5]gvk.vijay73@gmail.com, [6]brahmaraokbv@kluniversity.in

ABSTRACT

The abundance of user usage data has gained exponential dimensions as a result of the ongoing expansion and spread of Web applications and Web-based systems. Web user usage extraction is used to analyze the browsing data and investigate the web user's visiting interests or patterns. To enhance operational performance, web miners must employ predictive machine learning techniques integrated with reinforced Markov decision process. Especially, Higher order Markov decision frameworks promise the stronger predictive performance and penetration than single-order Markov decision, but they have a large state computational complexity. As a result, a selective Markov decision framework is formulated to considerably increase operating efficiency and prediction accuracy. Towards that the researchers introduced An Optimal Machine Learning Model Integrating with Selective Reinforced Markov Decision process (MLSRM). To efficiently collect and store web browsing data, the MLSRM makes use of the distributed HDFS-Spark parallel computing architecture. It then goes through the necessary pre-processing procedures to get the data ready for the Markov decision process. Later, MLSRM developed a reinforcement strategy to derive actionable knowledge so as to understand online user browsing habits with reduced state complexity and improved forecasting performance by intelligently selecting and integrating several Markov decision processes. Despite of compromising the overall accuracy and proposed model integrity, the suggested methodology eliminates lower Markov support states, examines the awarded probability, and quantifies the error at each Markov state. The proposed prediction Markov decision process was put to several tests, and the results are reported in this article.
Keywords: *Machine Learning; Reinforce Learning; Markov Models; Web Mining; Web User Behavior; Spark; Hadoop; Prediction*

1. INTRODUCTION

The scope of user activities on ubiquitous environment of WWW, particularly users' engagement and communication on the Web, has increased dramatically. As a result, the amount of data collected in form of weblogs has significantly grown. Weblog mining is an emerging approach in data discovery that aims to understand key patterns, trends, correlations in large amounts of data, find hidden information and to forecast user behavior. Estimating user purchasing activities, recognizing spammers and criminals, trying to improve online user satisfaction, foretelling user social media postings activities, inferring user complaint activities, provide smart business decisions on top-level strategies are just a few of the practical applications of predicting user behavior on the web. However, the rapid pace of digitization in the mentioned applications is creating the extreme pressure on web scientists to provide faster and right decisions. To deal with these complicated issues, web researchers quickly turned to the Machine Learning paradigm to propose potential solutions for modelling user web usage data.

Reinforcement Learning (RL) is a Machine Learning technique in which a web scientist learns the behavior of a web user in an interactive web environment by using user actions and rewards for its actions. The web user, also known as the agent, discovers which actions yield the greatest reward by exploiting and exploring them. The goal of reinforcement learning is to strike the correct balance between new-environment exploration and existing-knowledge utilization. Figure 1 depicts the reinforcement learning environment while interacting with agent through actions and rewards.

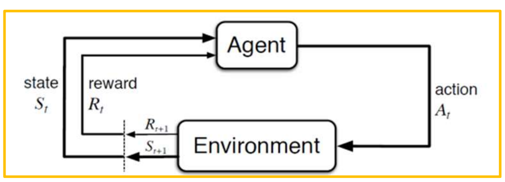


Figure. 1: Agent and Environment in RL

The Reinforcement Learning environment is usually formulated mathematically using Markov Decision Process as it is highly suited for modelling user web navigation data. Markov decision process is more concise, easy to understand and interpret, powerful, and based on well-established concepts. The Markov model's summarization ability is vital for identifying user browsing patterns, and its prediction ability is important for anticipating a user's next hyperlink option after following a known trail.

To simulate a series of web user usage sessions, first-order Markov decision models have been extensively used as they can effectively deal with the conversion probability from first state to second state. In this case, each web page on the site relates to a state of an entity in the Markov model, and each of two web pages seen in order relates to a state of an entity conversion in Markov model. The conversion probability is calculated by dividing the number of times the entity state conversion occurred by the number of times the 1st state inside the set has been travelled. Virtual states are generally attached to every navigation session to indicate the session's begin and termination. A sample first order Markov decision model is presented in Figure 2.

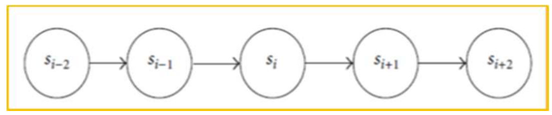


Figure 2: A sample first order Markov decision model

In a similar spirit, a second-order Markov decision models as depicted in Figure 3 are well suited to predict that the state of an entity at a given point in a web navigational sequence is determined by the states of two entities at the two positions before it.

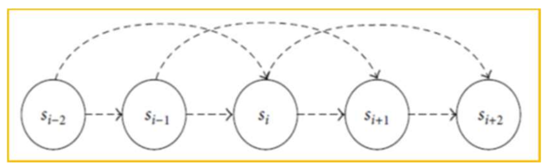


Figure 3: A sample second order Markov decision model

The consistent success rate of first and second order Markov models in a variety of real-world applications prompted researchers to look at higher order Markov models in the context of web usage mining.

When implementing higher order Markov model, it can be very challenging to scale to large amount of web log data. Hadoop and Spark unified computing environment allows web researchers for the distributed data storage and parallelization efficiency. Spark Streaming is perfectly adapted for real-time extraction of information, since it allows for scalable, high-throughput, and fault-tolerant handling of web log data streams to generate qualified web browsing patterns. With this aim, an Optimal Machine Learning Model Integrating with Selective Reinforced Markov Decision process (MLSRM) is presented by using HDFS-Spark to efficiently gather and store online surfing data.

The remaining of the present article is organized as follows. The literature survey in the context of Markov process is presented in section 2. Later, the proposed MLSRM work is demonstrated in detail in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

section 3. Further, the details of experimental setup and results are showcased in section 4. Finally conclusion and future work are made in section 5.

2. LITERATURE REVIEW AND MOTIVATIONS

In web engineering, models [2, 3, 4, 5] are frequently used to represent and forecast crucial processes on the WWW, such as social networks, web content, and online behavior. In recent years Markov models [1, 6, 9] are frequently used in pattern recognition of Web users to demonstrate the stochastic and deterministic features of web usage processes. Thus, e-marketing successfully builds relationships with clients and offers them a customized purchasing experience to increase marketing effectiveness.

Furthermore, a significant body of literature on the Markov technique puts up potential solutions for many diverse applications including power management [10, 23], surgical training [8,] Big Data Analytics [7, 24, 28], etc. While these applications help Markov models work more efficiently, prediction-based methods are still advised. With this as their guiding principle, the authors of the current study conduct a literature review on various aspects of Markov, with a focus on assessing web user navigation patterns.

Mei Fang, Shanling Dong, and Zheng-Guang Wu [11] focused their research in recent years on nonhomogeneous Markov jump systems. They did, however, mainly focus on a particular filtering design issue for homogeneous Markov systems.

In terms of prediction accuracy, space complexity, and processing time, a different group of scientists [12, 15] compared the advantages and disadvantages of lower order Markov models with higher order Markov models. They suggested a method based on variable order models to address the drawbacks of fixed order utilizing the kernel smoothing method. The authors unequivocally supported the necessity for future study on the dimensionality curse of sequencing data.

Researchers have focused on hidden models as the other Markov dimension [13, 14]. They have actually decided to tackle the difficult and complex problem of nonlinear Markov jump systems, which characterize the nonsynchronous regulator as well as the stochastic quantization occurrence. Well, using the fuzzy method in conjunction with other creative techniques, they were able to solve the convex optimization problem to its best ability. it makes sense to look at the asynchronous challenge for more complicated systems with certain potential constraints at the same time.

In a nutshell, deep learning models were used in the research study conducted by the authors [16, 17, 30, 32, 33, 35] to generate numerous substantial contributions to extract human movement patterns depending on the transition probability. To reach a greater level of prediction accuracy, they smartly combined the LSTM model with Markov framework. The authors immediately highlighted that thorough analyses with other prediction models and the integration of more features to increase model robustness are potential directions for future investigation for the present researchers.

A rich set of research [18, 19, 20, 22, 25, 26, 29, 31, 34, 36] has focused on reinforcement learning methods in general, and specifically on the combination of various Markov models. For example, repairing missing data, reformulating feature selection, best sampling, supervised learning, extracting rules, etc. are crucial steps of predictive models.

Additionally, the passionate scientists have demonstrated how reinforcement-based Markov chains work like enchantment to tackle real-world issues. For example, the authors [27, 28] tested Markov models that included machine learning approaches to control traffic lights and congestion. The Markov models are thought to be successful at learning from their prior experiences based on the experimental findings.

In a related vein IoT based Smart Irrigation, the researchers [21] also used reinforcement-based Markov to the efficient use of water in drip irrigation, as it is of vital relevance for smart farming, food production, and growth of the economy. This is particularly true in light of the expanding world's population, the changes in climate, and the struggle for water among diverse sectors. They thanked Markov for proving the worth of their proposed work.

The success of the Markov attempts made by the aforementioned eminent researchers inspired the current authors to develop a method for tracking web user navigational patterns within the framework of reinforced selective Markov models.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. PROPOSED MACHINE LEARNING MODEL INTEGRATING WITH SELECTIVE REINFORCED MARKOV DECISION

The Machine Learning model integrating with Selective Reinforced Markov decision process (MLSRM) is proposed to analyze and comprehend stochastic, and it is rightly suitable fit for modelling and forecasting user browsing behavior in the Weblog Scenario. The authors design MLSRM intelligently by concentrating all the key stages as presented in Figure 4. Initially it pays focus on collection of domain specific both structured and unstructured data. Later, the model chooses the Spark-Hadoop environment to take the benefit of parallel processing and distributed storage. Following that, MLSRM carries a series of phases to pre-process the web data and prepare it for further stages. Finally MLSRM integrates with various Markov models to lead minimal state-space complexity, enhanced accuracy rate, and coverage comparable in the process of estimating probabilities.

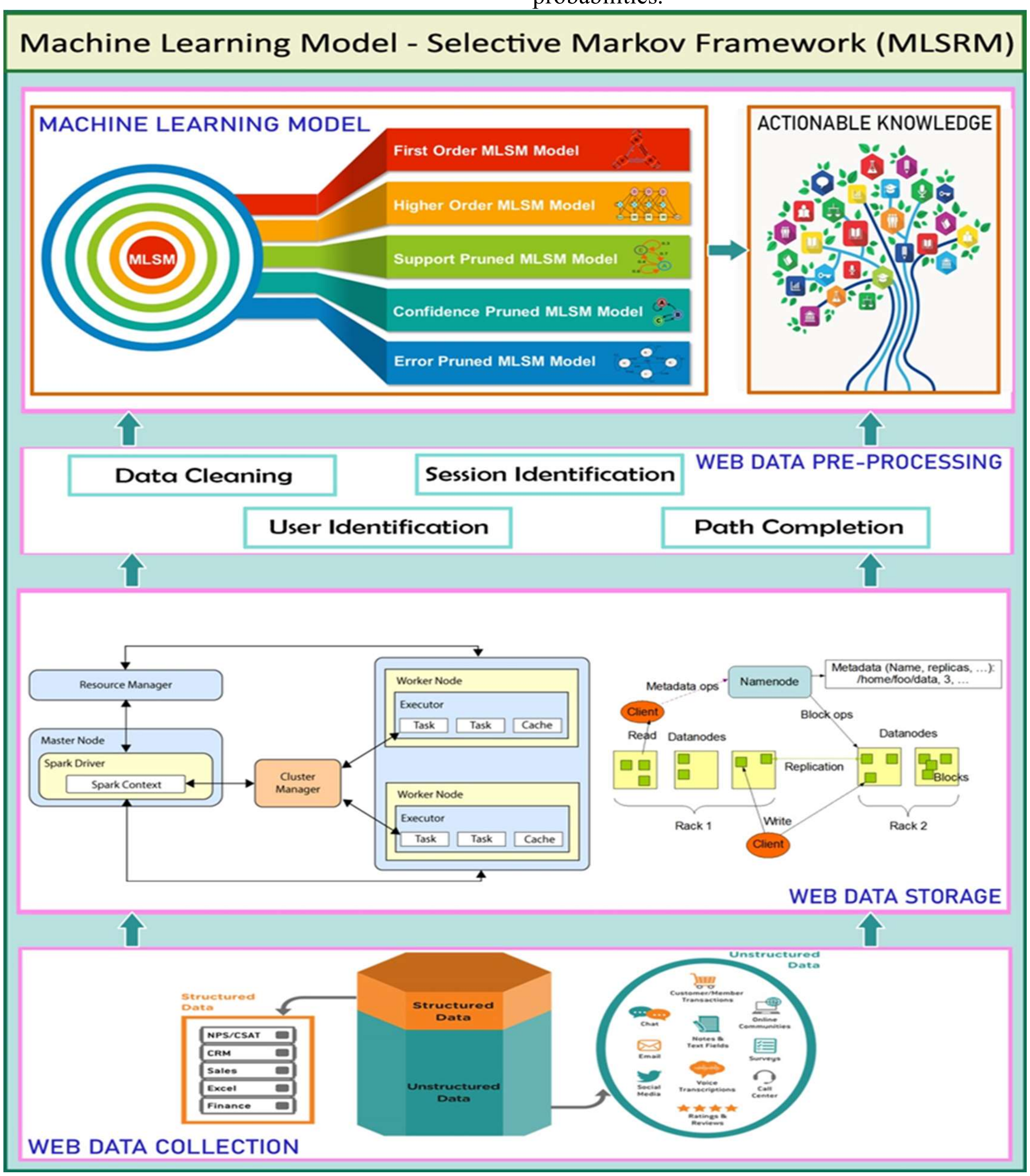


Figure. 4: Architecture of proposed MLSRM Model

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.1. Web Data Pre-Processing

At first the proposed MLSRM focusses on collection of web users' browsing data stored in terms of web log entries that are in simple and complex format. The attributes like IP address, URL, Visiting path, Path Traversed, Time stamp, Page last visited, Success rate, Agent type, Request type, etc. are considered for the present study. These log files are collected at web proxy server as it maintains a separate copy of original server log files. The data in the log is stored in plain text "ASCII" format and multiple access log format is chosen by MLSRM for its next stage.

3.2. Web Data Storage

The Spark environment was chosen for the proposed MLSRM to store the collected web log data. The streaming framework of Spark enables the MLSRM towards in-memory data distributed storage and real-time parallel processing, as well as the ability to execute many times quicker than most other big data technologies.

3.3. Web Data Pre-Processing

A realistic web log is likely to have noise, incompleteness, and/or be unfit at all machine learning systems. Preprocessing of web log data is a requisite for fining the data and to make it allowable to ML Systems, it upgrades the system's performance as well as reliability. With that aim, MLSRM carries the following sequence of pre-processing stages to make ready the web log data for better training the reinforced Markov decision models.

3.3.1. **Cleaning of Weblog Data:** Data cleaning is an activity of recognizing, selecting, and removing unwanted or irrelevant properties from raw weblog data. Because there are so many properties in the weblog file, just the necessary ones are selected; the rest are eliminated.

First, entries for access to .jpeg, .gif, Scripts, and other audio/video files should be eliminated from the proposed model since they are transferred without the web user's request. Later, the proposed system focuses on dismissing web user requests for web pages that are not present on the webserver; these entries are labeled with various error codes. Finally, the work focus on eliminating the visits performed by web crawler by employing intelligent techniques proposed in [2, 3] because those accesses do not reflect how real visitors navigate the site. Furthermore, the system considers data to remove that is too infrequent or too frequent will not contribute to the formation of any essential information from it.

3.3.2. **Identification of Web User:** At this level, the proposed work focuses on identifying a unique user. When web page requests are routed through a proxy server, they will register the same IP address in the weblog, so, making it harder to identify individuals uniquely. In addition, caching at multiple levels, as well as saved online sites, makes it difficult to recognize and detect user individuality. Here, the authors consider a heuristic method to detect the unique user using based on variation in operating system and web browser. Whether any of the criteria for records with the identical IP address vary, it specifies the presence of a separate web user. When a web user does this intentionally, though, it may confuse.

3.3.3. **Identification of Sessionization:** Sessionization is a challenging task in the process of web usage identification to build meaningful web user sessions from a large amount of weblog data since the HTTP protocol is stateless. In this context, a session means a set of accesses finished by an individual web user with a distinctive IP address over duration of time. A web server can provide each request made by the web user in the session from a client machine or proxy servers. To perform sessionization from such very big and incremental weblogs, the suggested solution is a unique distributed architecture that leverages the Hadoop and Spark paradigm.

The suggested solution removes the requirement for backward web browser operations while still maintaining the timestamp of visited pages. The following are the two primary phases of the proposed solution:

- **Phase 1:** Web users' web page visits are segmented into smaller webpage demand sequences termed candidate sessions by employing the parameters: session duration time and web page-stay duration.
- **Phase 2:** Candidate sessions are further subdivided into the maximum number of

sub-sessions possible, with a connection from the before next webpage pair in the sequence. The web page-stay duration criteria for consecutive web pages are also met at the same time. It is necessary to verify the entire session length time again because it was verified in the step1.

This framework outperforms in terms of effective sessionization, and due to its scalable spark architecture, it can manage any amount of online consumption data.

3.3.4. **Path Completion:** The series of web pages or pathways viewed by web users as they explore a website is stored as clickstream data. Information about a web user's aims, expertise, and preferences could be contained in path data. The process of path analysis adds a new dimension to anticipating customer behavior that hasn't been examined by analysts studying only on recorded data.

Thus, the proposed work considers the referrer field and website topology to fix the issue of the incomplete path and detect missing page sequences to improve the quality of weblog entries.

3.4. Prediction Model Integrating with Selective Reinforced Markov Decision Process

The processed visited of web pages by the web user is taken to indicate a web user's browsing experience. To ascertain concealed relationship among frequently accessed web usage patterns one has to use its sequence length and corresponding transition probabilities. For this context the researchers innovatively propose the prediction model which is intelligently integrating the potentiality of Markov framework. The intrinsic nature of Markov feature is especially useful in reinforcement learning environment to generate qualified decisions and accurate transition values. In this approach, the current state description must be informative in order to improve the effectiveness and accuracy in generating future possible states as are claimed on a function of the current state. To control the number of states in the complex Markov framework and achieve maximum conditional probability, an appropriate level of Markov order with the right strategy must be chosen. With an aim of predicting web user behavior, MLSRM design an approach for selecting following suitable Markov models based on browsed web page sequences.

3.4.1. First-Order MLSRM Model (1-O Model):

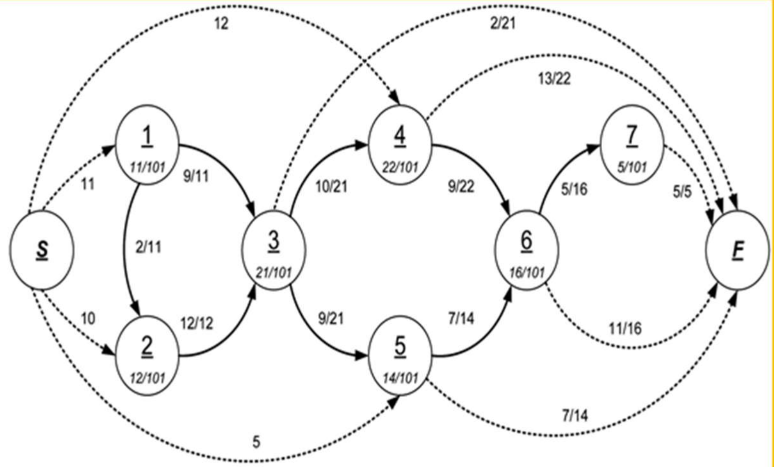
The 1-O Models are widely used to simulate a succession of web user usage experiences. All visited web pages are represented as states in this case, and each pair of visited web pages corresponds to a model's state transformation. The transformation probability is calculated by dividing the no. of occasions the transformation was crossed by the no. of occasions the first Markov state in the visited pair. Imaginary Markov states are frequently generated and attached to the beginning and end of each browsing session.

Journal of Theoretical and Applied Information Technology
31st January 2023. Vol.101. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

session	freq.
S,1,2,3,4,F	2
S,1,3,5,F	7
S,1,3,4,F	2
S,2,3,5,6,F	2
S,2,3,4,F	3
S,4,6,7,F	1
S,4,6,F	5
S,4,F	6
S,5,6,7,F	4
S,5,6,F	1
S,2,3,4,6,F	3
S,2,3,F	2

(a)



(b)

Figure 5: Sample of Web Browsing Session and 1-O Model

The MLSRM model captures variable length past data, so it is possible to forecast the association among the web pages and accurately characterize web user browsing behavior. In the following example demonstration, a state comprises a set of metadata words that describe Web page. So that, it is feasible for finding high-probability trails made up of web pages related to a specific context. With the help of an example, the proposed model demonstrates the series of increasing order Markov models.

The graphic shown in Figure 5 (a) is a sample of a web browsing session. The web session starts and finishes in a simulated state. Frequency defines how many times the matching sequence of web pages has been browsed. Figure 5 (b) depicts the 1-O model for all web browsing sessions. Each Web page has its state, and every two web pages seen in order are linked via a connection.

The 1-O model maintains the following information corresponding to a webpage; (i) Unique ID of webpage (ii) Number of times the webpage is visited-Wn (iii) Total Number of webpage views-Wt and (iv) the relation between Wn and Wt. The 1-O Model assesses the associated webpages based on the rewarded reinforcement value from the entire collection of webpages in the visited web-categories.

For instance, out of the fourteen visits to fifth webpage, fifth page came at the beginning of an online session. In seven of the fourteen instances following a visit to a fifth webpage, the user moves on to sixth webpage before closing the browser window.

A trail's rewarded probability is determined by multiplying its beginning state's value by the values of the connections it has travelled.

$$RP\ (3,4) = \frac{21}{101} * \frac{10}{21} = 0.099$$

$$RP\ (1,3,5) = \frac{11}{101} * \frac{9}{12} * \frac{9}{21} = 0.035$$

The 1-O model is a simple manner of describing a set of web sessions, however, its accuracy is typically inadequate, necessitating upgrades to higher-order models.

The 1-O model is a simple manner of describing a set of web sessions, however, its accuracy is typically inadequate, necessitating upgrades to higher-order models.

3.4.2. Higher Order MLSRM Model (H-O Model): The 1-O model could not correctly reflect the rewarded conditional probabilities of 2-O model. Towards that, the MLSRM in H-O model calculates conditional probability based on fixed sequence of webpages of a state and transformation among the states. Giving the input, for instance, the webuser usage sequence (1, 3) was repeated 9 times, resulting in #(1, 3) = 9, while the web usage sequence (1, 3, 4) was repeated twice, resulting in #(1, 3, 4) = 2. As a result, RP(4|1,3) = #(1, 3, 4) / #(1, 3) = 9 is the rewarded probability value for reading web page 4 after visiting web page 1 and 3 in order.

The 1-O Model could not get high accuracy in capturing 2-O rewarded probabilities, for example, at the state 3,

$$RP(4|1,3) - P(4|3) = |2/9 - 10/21| = 0.254$$

$$RP(4|2,3) - P(4|3) = |8/12 - 10/21| = 0.190$$

From the above, it is clear that state 3 could not reflect conditional probabilities of 2-O model properly. The accuracy of transition probabilities from a state is improved by separating the in-paths that correspond to different conditional probabilities. Towards that, the MLSRM, duplicates the state 3 and mentioned as 31 as well as diverting (2, 3) link to 31. The model adjusts weights of 3 and 31 out links based on how many times the three-state sequence has been followed.

For example, at state 3, the weight of the path-link (3, 4) = 2 and the weight of the path-link (3, 5) = 7. The out-links from duplicate state 31 are rationalized using the same technique as before. Subsequently replicating 4 states to appropriately denote all 2-O rewarded conditional probabilities, the resultant is publicized in Figure 6.

As the extended model demonstrated in Figure 6, all of the out links depict valid 2-O probability estimations. The trail (1, 3, 5) now has a rewarded probability value of 11/101 * 9/11 * 7/9 = 0.069. Trail (3, 4) has a probability estimate of (9/101 * 2/9) + (12/101 * 8/12) = 0.099, which is the same as the first-order value. As a result, the second-order model properly represents conditional 2-O probability values while maintaining accurate values at 1-O Model.

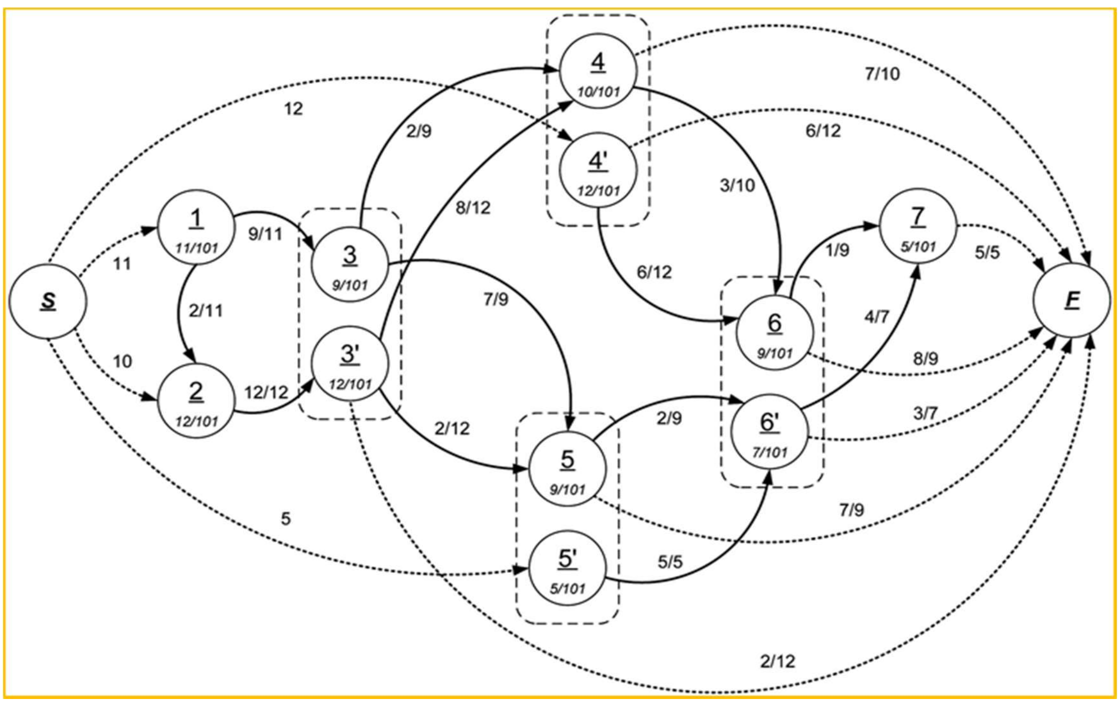


Figure 6: Sample 2-O Model of MLSRM.

The maximum permissible change between a 1-O model and the corresponding 2-O rewarded probability value is determined by the reinforcement factor γ, which is used by the MLSRM to control the number of new states produced by the approach. If the difference between the matching 1-O rewarded probability and the 2-O rewarded probability is larger than γ, then a state in a 1-O model is replicated. The average difference between the 1-O and 2-O rewarded probability value for a specific condition is γ, a threshold. If the average difference between the 1-O and 2-O rewarded conditional probabilities exceeds, the state γ is then reproduced. Additionally, the associations' criteria may be used to identify in-links with equal conditional probability if > 0 and the state have three or more in-links. When γ is calculating the maximum reinforcement probability value of deviation, indicate it by γ_m, and when it is calculating the average probability value of deviation, indicate by γ_a.

The process remains the same when expanding a model to higher orders. N-order conditional probability values are compared to the corresponding lower-order values in order to distinguish between their n-state length in-paths, and duplication is carried out to the incorrect states. Experimental results show that the model's execution time is roughly linearly related to the model's order. The first and second-order Markov models can be built in the same way as the higher Markov model, according to the previous two instances, but at a cost of significantly more state-space complexity.

All high order Markov models, as was indicated in the part above, have the potential to outperform single-order Markov models in terms of prediction accuracy and coverage, but at the expense of a hefty boost in state-space complexity. Create methods for shrewdly combining various orders of Markov models to produce a model with a low state complexity, improved prediction accuracy, and coverage of all high order Markov models. Based on this finding, we should start with the full high-order Markov model and exclude a large number of the states that are probably not going to provide good predictions. This will permit efficiency to be maintained while reducing the complexity of all states. This pruning stage's main goals are to reduce state complexity and, secondarily, to improve the model's predictive power.

Towards that, the authors propose two distinct methods, each with a higher order of difficulty. The initial strategy basically excludes states with extremely low order of support values. The following strategy makes use of statistical techniques to identify situations where the probabilities of switching between the two most common behaviors are not statistically significant.

3.4.3. Support Pruned MLSRM Mode (SP Model) : The SP Model finds the states with lower support also have lower prediction accuracy. As a consequence, these low-support situations may be removed from the SP model without impacting its complete accuracy or coverage. The awarded frequency threshold φ is a measure in this strategy that controls how much pruning is done. In particular, the model removes all states of the various orders that have fewer than occurrences.

- Firstly, irrespective of the order of the models, the same frequency threshold is applied to all of them.
- Furthermore, because higher-order states have low support, the pruning strategy of SP model, minimizing the state-space complexity.
- Finally, rather than specifying the proportion of instances that each state must support, the frequency threshold parameter φ defines the exact number of instances that each state must support.

The above steps are carried out largely for two reasons:

- The reliability of predicted transition probabilities for a given state is determined by the actual number of events rather than the relative number.
- Because the total number of instances rises exponentially with the order of the Markov model, the same fractional pruning threshold will have a completely different perspective for different Markov models.

3.4.4. Confidence Pruned MLSRM Model(CP Model) : One of the shortcomings of the SP Model method is that it does not take into account all the variables that affect the state's validity. In particular, the rewarded probability distribution of a state's emitted actions is disregarded. Take, for example, a Markov state with 2 outgoing actions, and one of those is far more likely than the other. Due to the obvious evident variation in the outgoing probability, though this state's total support is minimal, the predictions calculated by this state will be extremely trustworthy. In contrast, if the incoming probability values in the case of the former are very close to one another, the variation must be supported by a wide number of occurrences in order to be reliable. Before determining pruning judgments, the pruning strategy should include not just the state's support, but also all the probability distribution of the exciting activities.

The purpose of this discovery is to create the confidence-pruned MLSRM model. This model uses statistical methods to analyze each condition to see whether the likelihood of the operation that is selected the most frequently varies noticeably from the odds of the other operations that could be undertaken by this state. This state is removed if the rewarded probability value fluctuations are not significant because it is unlikely to provide high precision. Nonetheless, the state is maintained if the rewarded probability value of changes is significant.

The method determines if the possible action can be distinguished from the second most probable action by computing the 100(1-α)percent confidence mean time over the possible action and assessing whether the likelihood of the second strike falls within that interval. The state will be reduced if this is the case; else, it will be preserved.

$$\hat{P} - \mathrm{U}\alpha_{/2}\sqrt{\frac{\hat{P}(1-\hat{P})}{n}} \leq P \leq \hat{P} + \mathrm{U}\alpha_{/2}\sqrt{\frac{\hat{P}(1-\hat{P})}{n}} \quad (1)$$

α/2 is an ordinary normal distribution as well as n represents Markov state of MLSRM.

In this strategy, the level of trimming is regulated based on the confidence factor. The confidence interval grows as the value decreases, leading to greater trimming. Also, if a state is connected with a high number of cases, the above equation produces a strong confidence interval. As a consequence, even if the probability variation between the two most likely actions is minimal, the state would also most probably be maintained.

According to the authors, these strategies, which combine intelligently distinct reinforced Markov models, reduce state complexity while increasing prediction accuracy. As a consequence, MLSRM has improved operating performance when it comes to anticipating online user behavior.

4. EXPERIMENTAL SETUP & RESULTS

The data from the server-side weblogs is tested in a spark-Hadoop run-time environment for a year. During the data preparation phase, incorrect and image searches were discarded, and the sessionization duration restriction on requests was enforced, resulting in abnormally lengthy use sessions being split into two or shorter usage sessions. The total size of a single user's 238 queries was 2.14 MB. Requests such as audios, videos, images and others are 1.3 MB in size (61 per cent). As a result, cleaning is an important stage in the pre-processing process, as it limits human user access to the weblog by around 60%.

4.1. Performance of MLSRM over various Markov Orders

The 1-O model was derived from the group of sessions. Following this, the model was tested for rewarded conditional probabilities at both 2-O and H-O models, however if required, a state was replicated to distinguish the in-paths due to conditional probability discrepancies. As previously stated, the parameter determines the amount of tolerance that may be used when describing conditional probabilities. Another option specifies the bare minimum of requests a webpage must receive in order to be chosen for duplication. We used several visits of 30 in these tests. When the order of the model is increased while the accuracy threshold γ is set to 0, Figure 7 demonstrates the fluctuation in the model number of states.

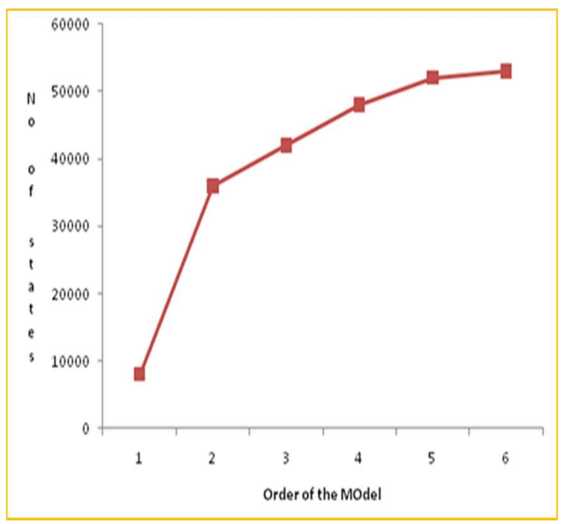


Figure 7: Performance of MLSRM over various orders

4.2. Operational Performance of MLSRM orders at Various Frequency Thresholds

About frequency threshold, the MLSRM was compared against 1st, 2nd, 3rd, and 4th order Markov Models. As demonstrated in Figure 8, the experimental findings reveal that MLSRM outperforms low order Markov Models in terms of operational performance in Spark environment.

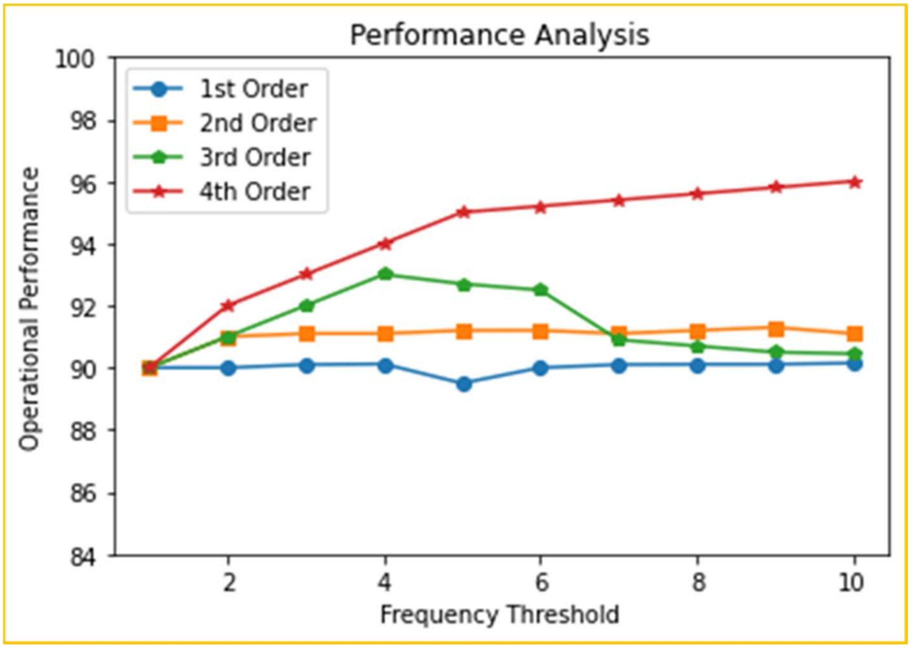


Figure 8: Operational Performance Analysis of MLSRM orders at various Frequency Threshold

4.3. Prediction Accuracy of MLSRM over various Markov Models

The authors continue by carrying out numerous tests to validate the proposed framework in comparison to current models. As shown in Figure 9, the prediction accuracy of MLSRM is superior to that of Mobility Markov Chains (MM), Variable Multi-Order Markov Models (VLMM) and Kernel Variable Length Markov Model, (KVLMM)[12].

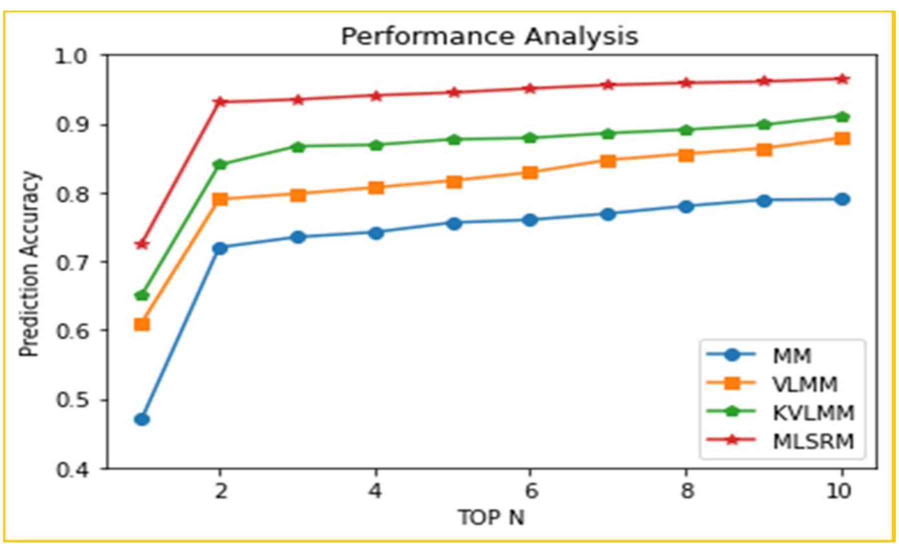


Fig. 9. Prediction Accuracy of MLSRM over various Markov Model

4.4. Prediction Accuracy of MLSRM over various Server side Weblogs

Multiple experiments are run in Spark on different real-time gathered server-side weblogs to

Journal of Theoretical and Applied Information Technology
31st January 2023. Vol.101. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

acquire more precise results, and the prediction accuracy is taken as the experimental result over existing Markov Models. It is obvious that MLSRM is demonstrating improved robustness, and Figure 10 illustrates this.

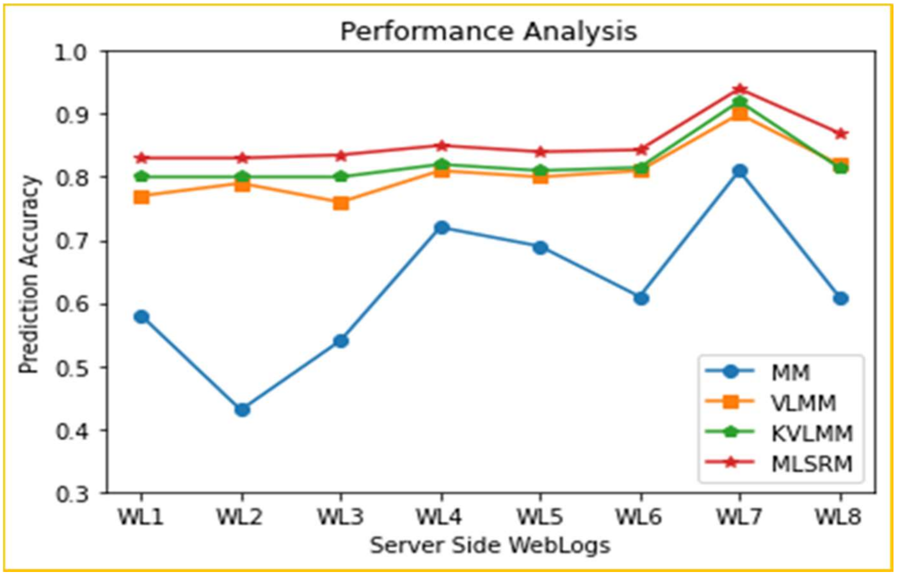


Figure 10: Prediction Accuracy of MLSRM over various Server side Weblogs

4.5. Accuracy of MLSRM over various Prediction Sequence Lengths

In most cases, the forecast is unsuccessful in successfully matching past data, which significantly lowers prediction accuracy. Additionally, there are still some concerns with low coverage because of the impact of dataset size. Experiments with different MLSRM orders are conducted to evaluate the proposed model's performance on the above concerns. As shown in Figure 11, the authors discovered that higher order Markov models consistently outperformed lower order models, even with expanding prediction lengths.

When compared with the results of conventional Markov Models, the experimental results of MLSRM perform significantly better since it makes use of the advantages offered by the Machine Learning framework and the SPARK execution environment.

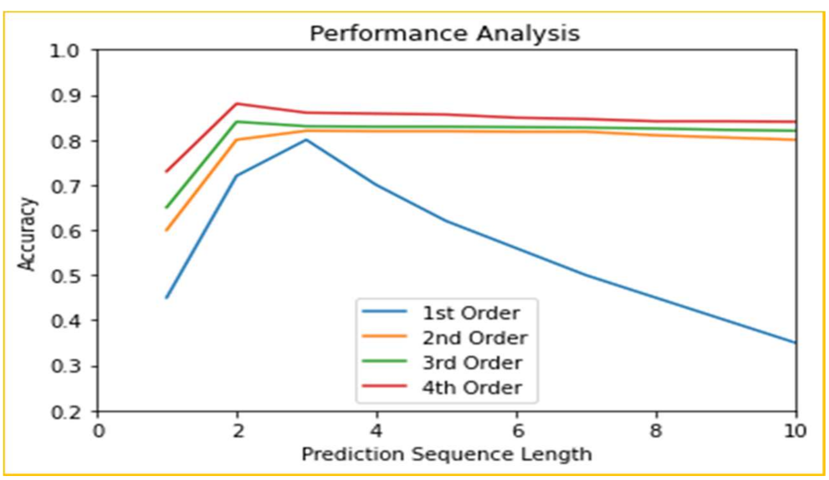


Figure 11: Accuracy of MLSRM over various Prediction Sequence Lengths

5. CONCLUSION

In the current study, the authors suggested an optimal machine learning model (MLSRM) that integrates with a selective reinforcement Markov decision process and equally focuses each step of predicting web browsing patterns. Further, the authors explained how the HDFS-Spark parallel computing architecture incorporates the MLSRM's architectural design, the integration of organic components, and the execution of the system. MLSRM created a reinforcement strategy by intelligently choosing and combining various Markov decision processes to comprehend online user browsing habits with reduced state complexity and increased prediction performance. With the help of experimental data, the present model has demonstrated the integration of Selective Reinforced Markov Decision process for finding high probability, and quantifies the accuracy at each Markov order. It is clearly envisages that this cutting-edge machine learning technique increases prediction accuracy by including selective reinforce Markov decisions. Although deep reinforcement learning is still not fully understood by the scientific community, given how effective it has recently been in predicting webpage patterns, it is likely soon for more exciting advancements in other fields.

REFERENCES:

[1] Huaigu Cao and Venu Govindaraju, "Preprocessing of Low-Quality Handwritten Documents Using Markov Random Fields", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 7, doi: 10.1109/TPAMI.2008.126

[2] V.V.R. Maheswara Rao, Dr. V. Valli Kumari, Dr. KVSVN Raju "Study of Visitor Behavior by Web Usage Mining" International Conference on Recent Trends in Business Administration and Information Processing - BAIP 2010, proceedings by Springer LNCS-CCIS, ISSN: 1865-0929, ISBN: 978-3-642-12213-2, Vol. 70, pp. 181-187, 2010

[3] V.V.R. Maheswara Rao, Dr. V. Valli Kumari, "A Novel Lattice Based Research Frame Work for Identifying Web User's Behavior with Web Usage Mining", International Conference on Advances in Information and Communication Technologies - ICT 2010, proceedings by Springer LNCS-CCIS, ISSN: 1865-0929, ISBN: 978-3-642-15765-3, Vol. 101, Part 1, pp. 90-99, 2010

[4] L.K. Joshila Grace, V.Maheswari and Dhinaharan Nagamalai, "Analysis of Web Logs and Web User in Web Mining", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.1, pp.99-110, 2011, doi: 10.5121/ijnsa.2011.3107

[5] Michal Munka, Jozef Kapustaa and Peter Sveca, "Data Preprocessing Evaluation for Web Log Mining: Reconstruction of Activities of a Web Visitor", International Conference on Computational Science, ICCS 2010, ScienceDirect, Procedia Computer Science 1 (2012), 2012 doi:10.1016/j.procs.2010.04.255

[6] Lu Yu, Jason M. Schwier, Ryan M. Craven, Richard R. Brooks and Christopher Griffin,"Inferring Statistically Significant Hidden Markov Models", IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 7, pp.1548-1558, 2013, doi: 10.1109/TKDE.2012.93

[7] Hyun-Ju Jo and Ji Won Yoon, "A New Countermeasure against Brute-Force Attacks That Use High Performance Computers for Big Data Analysis", International Journal of Distributed Sensor Networks, Vol. 2015, pp.1-7, doi: 10.1155/2015/406915

[8] Qiang Zhang and Baoxin Li, "Relative Markov Models for Video-Based Evaluation of Motion Skills in Surgical Training", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, No. 6, pp.1206-1218, 2015, doi: 10.1109/TPAMI.2014.2361121

[9] Stephen Adams, Peter A. Beling and Randy Cogill, "Feature Selection for Hidden Markov Models and Hidden Semi-Markov Models", IEEE Access, Vol.4, pp.1642-1657, 2016, doi: 0.1109/ACCESS.2016.2552478

[10] Hiroyuki Okamura, Satoshi Miyata and Tadashi Dohi, "A Markov Decision Process Approach to Dynamic Power Management in a Cluster System", IEEE Access, Vol.3, 2015, doi: 10.1109/ACCESS.2015.2508601

[11] M. Fang, S. Dong and Z. G. Wu, "Estimation for piecewise homogeneous Markov jump systems: A dual hidden Markov model approach," 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), 2017, pp. 82-87, doi: 10.1109/SPAC.2017.8304255

[12] Xing Wang, Xinhua Jiang, Lifei Chen and Yi Wu, "KVLMM: A Trajectory Prediction

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Method Based on a Variable-Order Markov Model with Kernel Smoothing", IEEE Access, Vol.6, pp.25200-25208, 2018, doi: 10.1109/ACCESS.2018.2829545

[13] S. Dong, Z. Wu, P. Shi, H. Su and T. Huang, "Quantized Control of Markov Jump Nonlinear Systems Based on Fuzzy Hidden Markov Model," IEEE Transactions on Cybernetics, vol. 49, no. 7, pp. 2420-2430, July 2019, doi: 10.1109/TCYB.2018.2813279.

[14] F. Li, S. Xu and B. Zhang, "Resilient Asynchronous H_∞ Control for Discrete-Time Markov Jump Singularly Perturbed Systems Based on Hidden Markov Model," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 8, 2020, doi: 10.1109/TSMC.2018.2837888

[15] Sung Youl Oh, Jae Wook Song, Woojin Chang and Minhyuk Lee, "Estimation and Forecasting of Sovereign Credit Rating Migration Based on Regime Switching Markov Chain", IEEE Access, Vol.7, 2019, doi: 10.1109/ACCESS.2019.2934516

[16] P. Wang, H. Wang, H. Zhang, F. Lu and S. Wu, "A Hybrid Markov and LSTM Model for Indoor Location Prediction," IEEE Access, vol. 7, pp. 185928-185940, 2019, doi: 10.1109/ACCESS.2019.2961559

[17] Alexander Luck, Pascal Giehr, Karl Nordstrom, Jorn Walter and Verena Wolf, "Hidden Markov Modelling Reveals Neighborhood Dependence of Dnmt3a and 3b Activity", IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 16, No. 5, pp.1598-1609, doi: 10.1109/TCBB.2019.2910814

[18] W. Wu and M. Liao, "Reinforcement Fuzzy Tree: A Method extracting Rules from Reinforcement Learning Models," 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), 2019, doi: 10.1109/ICIS46139.2019.8940165

[19] N. Yamaguchi, O. Fukuda and H. Okumura, "Model-based reinforcement learning with missing data," 2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW), 2020, pp. 168-171, doi: 10.1109/CANDARW51189.2020.00042

[20] A. Ruan, A. Shi, L. Qin, S. Xu and Y. Zhao, "A Reinforcement Learning-Based Markov-Decision Process (MDP) Implementation for SRAM FPGAs," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 10, pp. 2124-2128, Oct. 2020, doi: 10.1109/TCSII.2019.2943958

[21] L. M. S. Campoverde, M. Tropea and F. De Rango, "An IoT based Smart Irrigation Management System using Reinforcement Learning modeled through a Markov Decision Process," 2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), 2021, pp. 1-4, doi: 10.1109/DS-RT52167.2021.9576130

[22] M. Haklidir and H. Temeltaş, "Guided Soft Actor Critic: A Guided Deep Reinforcement Learning Approach for Partially Observable Markov Decision Processes," IEEE Access, Vol. 9, pp. 159672-159683, 2021, doi: 10.1109/ACCESS.2021.3131772

[23] Oluwafemi Kolade and Ling Cheng, "Markov Model Characterization of a Multicarrier Narrowband Powerline Channel With Memory in an Underground Mining Environment", IEEE Access, Vol. 9, 2021, 10.1109/ACCESS.2021.3072669

[24] Imad Sassi, Samir Anter and Abdelkrim Bekkhoucha, "ParaDist-HMM: A Parallel Distributed Implementation of Hidden Markov Model for Big Data Analytics using Spark", International Journal of Advanced Computer Science and Applications, Vol. 12, No. 4, pp.289-303, 2021.

[25] Z. Wang, Y. Xia, S. Lyu and C. Ling, "Reinforcement Learning-Aided Markov Chain Monte Carlo For Lattice Gaussian Sampling," 2021 IEEE Information Theory Workshop (ITW), 2021, pp. 1-5, doi: 10.1109/ITW48936.2021.9611412.

[26] K. Liu, Y. Fu, L. Wu, X. Li, C. Aggarwal and H. Xiong, "Automated Feature Selection: A Reinforcement Learning Perspective," IEEE Transactions on Knowledge and Data Engineering, 2021, doi: 10.1109/TKDE.2021.3115477.

[27] S. El Hamdani, S. Loudari, S. Novotny, P. Bouchner and N. Benamar, "A Markov Decision Process Model for a Reinforcement Learning-based Autonomous Pedestrian Crossing Protocol," 2021 3rd IEEE Middle East and North Africa COMMunications Conference (MENACOMM), 2021, doi: 10.1109/MENACOMM50742.2021.9678310

[28] Silpa N, and VVR Maheswara Rao. "A Complete Research on Techniques & Technologies of Big Web Data Preparation to Web User Usage Behaviour.", International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-8, Issue-2S11, September 2019.

[29] G. Borrageiro, N. Firoozye and P. Barucca, "The Recurrent Reinforcement Learning Crypto Agent," in IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3166599.

[30] Silpa N, and VVR Maheswara Rao. "Machine Learning-Based Optimal Segmentation System For Web Data Using Genetic Approach." Journal of Theoretical and Applied Information Technology, 100.11, 2022.

[31] L. Lyu, Y. Shen and S. Zhang, "The Advance of Reinforcement Learning and Deep Reinforcement Learning," 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), 2022, pp. 644-648, doi: 10.1109/EEBDA53927.2022.9744760.

[32] Maheswara Rao, V. V. R., et al. "An Enhanced Machine Learning Classification System to Investigate the Status of Micronutrients in Rural Women." Proceedings of International Conference on Recent Trends in Computing. Springer, Singapore, 2022.

[33] Reddy, Shiva Shankar, Mahesh Gadiraju, and V. V. R. Maheswara Rao. "Analyzing Student Reviews on Teacher Performance Using Long Short-Term Memory." Innovative Data Communication Technologies and Application. Springer, Singapore, 2022.

[34] G. Borrageiro, N. Firoozye and P. Barucca, "Reinforcement Learning for Systematic FX Trading," in IEEE Access, vol. 10, doi: 10.1109/ACCESS.2021.3139510.

[35] Reddy, Shiva Shankar, et al. "Developing Preeminent Model Based on Empirical Approach to Prognose Liver Metastasis." Ubiquitous Intelligent Systems, Springer, Singapore, 2022. 665-683.

[36] J. Liu, G. Wang, X. Guo, S. Wang and Q. Fu, "Deep Reinforcement Learning Task Assignment Based on Domain Knowledge," in IEEE Access, vol. 10, pp. 114402-114413, 2022, doi: 10.1109/ACCESS.2022.3217654.