Proceedings of the 3rd International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/100/2025.21986

Applications of Markov Chain in the Field of Computer Science

Xinyi Zhou

Faculty of Science and Technology, Beijing Normal-Hong Kong Baptist University, Zhuhai, China
u430005107@mail.uic.edu.cn

Abstract: In the era of digital transformation, dealing with big data that alters over time is necessary. Markov chain is a fundamental concept in the field of stochastic processes for modeling systems that evolve probabilistically over time, and especially it can be used in computer science for data analysis. This paper focused on analyzing the applications of Markov chain in predicting cloud service trusted state and network traffic. The main problem addressed is how to integrate Markov chains into the complicated computation systems. By employing discrete Markov processes, hidden Markov chains, and fuzzy Markov fields, one can use the transition probability matrix and the stationary distribution of Markov chains to ascertain the stable state of the systems, predict the future state, and then conduct optimizations. The results indicated that if the stationary distribution of Markov process exists, then the future state can be predicted. With the training of some extra parameters, the optimized scheme can be achieved. This research is significant as it provides practical guidance for educators and institutions to utilize Markov chain under digital era.

Keywords: Markov chain, transition matrix, stationary distribution, cloud service trusted state, network traffic

1. Introduction

The Markov chain, first proposed by the Russian mathematician Markov in 1906, was initially used for the study of linguistic statistics. Markov described the probability of the occurrence of words by using a sequence of random variables. Subsequently, Kolmogorov laid the theoretical foundation for it. Later, Metropolis proposed Markov Chain Monte Carlo, and Baum and other researchers proposed hidden Markov Model. Concepts such as the Markov decision process appeared successively. The current hot issues of Markov process concentrate on firstly, the integration of Markov chains with cutting-edge technologies such as deep learning to extract more value from data. Secondly, in high-dimensional and complex data scenarios, the further optimization of Markov chain algorithms to improve computational efficiency and model generalization ability. Thirdly, the expansive applications of Markov chains in emerging fields, such as quantum computing and brain science, providing new methods and ideas for interdisciplinary research.

Previous studies and literatures have shown that Markov chain can be used extensively in the field of computer science. For instance, Qian et al. use improved Markov chain to construct efficient domain name generation algorithm [1]. Chelly et al. utilize the Markov chain-based generation flow network to achieve consistent amortized clustering [2]. Li et al. have researched the combination of deep computer technique and Markov theorem for deep computerized adaptive testing to

DOI: 10.54254/2753-8818/100/2025.21986

revolutionize contemporary assessment practices in education and behavioral health by dynamically adapting test material to meet individual examinees' needs during the evaluation process [3]. Further, Chen et al. have researched for using Markov chain to select important nodes in random network by creating efficient sampling procedure [4]. The importance of exploring the applications of Markov chain lies in improving data manipulation in digital age.

This paper is structured as follows. Section 2 will introduce some basic method and theory with the inclusion of the definition of Markov process and some properties of Markov chain. Section 3 will present some background information about the importance of using Markov chains and introduce two famous applications of Markov chain in the transformation of cloud service trusted state and the prediction of network traffic. Section 4 will conclude the paper with summaries of the findings and the expectations for future researches.

2. Method and Theory

2.1. Definition of Markov Process

Markov process is a probabilistic process that is used to model systems where the state that will occur in the future is contingent solely upon the present state and has no connection or reliance on the state that preceded it. Given the present state in order to predict the future state.

The two main types of Markov process are discrete Markov process and continuous Markov process. For a discrete Markov process, the parameter takes discrete values which are usually represented by the set of integers ($n = 0,1,2,\cdots,$) and the state space is usually discrete. It is clear that discrete time points are used to observe the change of system's state. It is suitable for the analysis of web browsing behavior and text generation. For a continuous Markov process, the time parameter takes continuous values which are usually represented by the set of non-negative real numbers ($t \geq 0$), and the state space is usually continuous. It means that the system state can be observed at any moment. It is often used in field such as queuing theory and the dynamics of biological populations.

In this paper, the author mainly focused on analyzing the Discrete Markov process. The mathematical definition of Discrete Markov process is that concerning a succession of random variables$X_1, X_2, \cdots, X_n$, for all $n \geq 0$, and for all possible values $x_1, x_2, \cdots, x_n$ of the random variables, satisfy the following formula

$$P\left(X_{n+1} = x_{n+1} \middle| X_n = x_n, X_{n-1} = x_{n_1}, \cdots, X_0 = x_0\right) = P\left(X_{n+1} \middle| x_{n+1}\right) \tag{1}$$

A Markov process is often typified by a transition probability matrix P. Each element, which denoted by P_{ij}, in the matrix P denotes the probability by which there is a shift from state i to state j in one state. It satisfies the following relation

$$P_{ij} = P(X_{n+1} = x_j | X_n = x_i) \tag{2}$$

The form of the transition probability matrix is thus $P = \begin{pmatrix} P_{00} & \cdots & P_{0n} \\ \vdots & \ddots & \vdots \\ P_{n0} & \cdots & P_{nn} \end{pmatrix}$.

2.2. Properties of Markov Chain

The irreducibility and stationary distribution are two properties of Markov chain. An irreducible Markov chain means that no matter starting from what states, it is possible to reach the other states within a finite number of steps. The mathematical definition of the irreducibility of Markov chains is that for a Markov chain $\{X_n\}$ with a state space S, if for any $i, j \in S$ there exists a positive integer m,

DOI: 10.54254/2753-8818/100/2025.21986

such that the probability of transitioning from state i to state j in m steps $P_{ij}(m) > 0$, that is, $P(X_n = j|X_0 = i) > 0$, then this Markov chain is irreducible [5].

The stationary distribution of discrete Markov process is that after the Markov chain ran for a long enough time, the probabilities of the system being in various states no longer change with time, reaching a stable state. This stable probability distribution is the steady-state distribution of Markov chain [5]. The existence conditions for the steady-state distribution are that the Markov chain has the properties of irreducibility, aperiodicity and positive recurrence. The mathematical definition of stationary distribution is that for a Markov process $\{X_n, n = 0,1,2,\cdots\}$, if there exists a probability distribution $\pi = (\pi_1, \pi_2, \cdots, \pi_i, \cdots)$, $\sum \pi_i = 1$, such that for all states i and any $n \geq 0$, $P(X_n = i) = \pi_i$. The probability distribution π does not change with the passage of time [5].

There are three calculation methods for stationary distribution. Firstly, one is calculating $\pi P = \pi$ (where P is the transition matrix). Since the value of the input vector of P equals to the output vector, vector π is the stationary distribution [5]. The second one is calculating $\pi^{(k+1)} = \pi^k P$, until π^k converges, it represents the stationary distribution [1]. The third one is calculating P^n (where P is the transition matrix), when reaching the values of all rows tend to be equal and the sum of the values of each row is 1, each row represents the stationary distribution [5].

3. Results and Applications

3.1. Markov Chain-Based Cloud Service Trust Condition

3.1.1. Construction of Markov Model

IT tactics have been completely transformed by cloud computing. However, choosing the best cloud service is still difficult because of the wide range of performance characteristics, the market's competitiveness, and the fact that traditional performance metrics frequently fail to take real-time fluctuations into account, leading to assessments that might not adequately match business requirements [6]. Researchers use a Markov chain model to continuously track and analyze changes in users' responses because assessing aspects like performance, dependability, scalability, and security is crucial [6]. This allows for a more thorough and rapid evaluation than static models. Definition of trustworthiness level of cloud service is that a service is trusted if it consistently evolves in the anticipated direction. In contrast, a service is not trusted if it is unable to continue operating regularly because of a trustworthiness issue [7].

There are five extra parameters that associate with the construction of Markov model. The specific cloud service-relevant reliability characteristic model is shown in Figure 1. For these parameters, β_i represents three classes of the cloud service trustworthiness $\beta_1, \beta_2, \beta_3$, which denote the reliability of the operational and maintenance tasks of service providers, the dependability of service data, and the credibility of the quality standard of service, respectively [7]. α_i represents the 16 significant indicators that impact the reliability of cloud services through the research of the literature and experts' visits [7]. F is the parameter that shows the frequency of issues with trustworthiness in the long-run functioning of cloud services. The frequency of issues with cloud service trustworthiness increases with a greater value of F [7]. The parameter L shows the degree of cloud service failure severity during extended use. The harm brought on by the cloud service trustworthiness issue increases with the value of L [7]. Fuzzy entropy, denoted by $E(A)$, is a quantitative measure reflects the degree of uncertainty of elements belonging to the fuzzy set. When the boundary of a fuzzy set is less clear and the membership degree distribution of elements is more dispersed, the fuzzy entropy is larger [7].

DOI: 10.54254/2753-8818/100/2025.21986

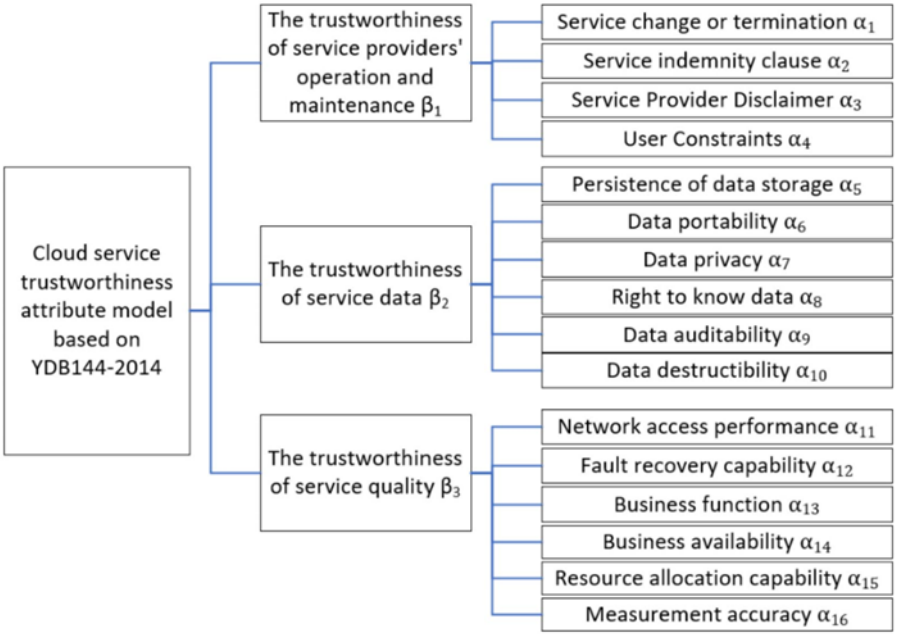

Figure 1: The cloud service-relevant reliability characteristic model [7].

Since Markov process is characterized by its memoryless property, the transitions between four trusted states S_1, S_2, S_3, S_4 can be predicted based on Markov chain principle by modeling a transition probability matrix. They represent a state of utmost trustworthiness, a fundamental assurance state, a crucial assurance state, and a doubted state, respectively [7]. The trusted state matrix is

$$TM = \begin{bmatrix} Psi\left(S_{(1\to1)}\right) & Psi\left(S_{(1\to2)}\right) & Psi\left(S_{(1\to3)}\right) & Psi\left(S_{(1\to4)}\right) \\ Psi\left(S_{(2\to1)}\right) & Psi\left(S_{(2\to2)}\right) & Psi\left(S_{(2\to3)}\right) & Psi\left(S_{(2\to4)}\right) \\ Psi\left(S_{(3\to1)}\right) & Psi\left(S_{(3\to2)}\right) & Psi\left(S_{(3\to3)}\right) & Psi\left(S_{(3\to4)}\right) \\ Psi\left(S_{(4\to1)}\right) & Psi\left(S_{(4\to2)}\right) & Psi\left(S_{(4\to3)}\right) & Psi\left(S_{(4\to4)}\right) \end{bmatrix} \quad (3)$$

where the element $\mathrm{Psi}(S_{m\to n})$ is the probability of the reliability condition transforming from state m to state n, and $\sum_{m=1}^{4} Psi\left(S_{(m\to n)}\right) = 1$.

3.1.2. Method of Computing the Reliable Condition of the Cloud Service

Expert evaluation of underlying indicators is that for each trusted attribute indicator α_j, experts evaluate its frequency level interval $[F_{min}(\alpha_j), F_{max}(\alpha_j)]$, and the severity of loss level interval $[L_{min}(\alpha_j), L_{max}(\alpha_j)]$.

Calculation of the Membership Degree $\mu_{S_n}(\alpha_j)$ is that the membership degree of the indicator α_j belong to state S_n is calculated by the geometric region overlapping method $\mu_{S_n}(\alpha_j) = \frac{\mathrm{Square}(\alpha_j)\cap\mathrm{Square}(S_n)}{\mathrm{Square}(\alpha_j)}$, where $\mathrm{Square}(\alpha_j)$ is the coverage area of α_j in the risk matrix, and $Square(S_n)$ is the geometric region of state S_n [7]. Calculation of Transition Probabilities by Integrating Membership Degrees is that for each trusted category β_i, the element $P(S_n \to S_m, \beta_i)$ of its transition matrix $TM(\beta_i)$ is calculated by formula $Psi(S_n \to S_m, \beta_i) = \sum_{j=1}^{total} \mu_{S_m}(\alpha_j)$ when $\mu_{S_n}(\alpha_j) > 0$ [7]. The trusted state matrix is given by

DOI: 10.54254/2753-8818/100/2025.21986

$$TM(\beta_i) = \begin{bmatrix} Psi\left(S_{(1\to1)}, \beta_i\right) & Psi\left(S_{(1\to2)}, \beta_i\right) & Psi\left(S_{(1\to3)}, \beta_i\right) & Psi\left(S_{(1\to4)}, \beta_i\right) \\ Psi\left(S_{(2\to1)}, \beta_i\right) & Psi\left(S_{(2\to2)}, \beta_i\right) & Psi\left(S_{(2\to3)}, \beta_i\right) & Psi\left(S_{(2\to4)}, \beta_i\right) \\ Psi\left(S_{(3\to1)}, \beta_i\right) & Psi\left(S_{(3\to2)}, \beta_i\right) & Psi\left(S_{(3\to3)}, \beta_i\right) & Psi\left(S_{(3\to4)}, \beta_i\right) \\ Psi\left(S_{(4\to1)}, \beta_i\right) & Psi\left(S_{(4\to2)}, \beta_i\right) & Psi\left(S_{(4\to3)}, \beta_i\right) & Psi\left(S_{(4\to4)}, \beta_i\right) \end{bmatrix} \quad (4)$$

Markov Chain Prediction: Suppose the probability vector of the trusted state at the current time t is $\mu^t = [\mu_{S_1}^t, \mu_{S_2}^t, \mu_{S_3}^t, \mu_{S_4}^t]$, and the probability vector at the next moment is calculated by the transition matrix: $\mu^{t+1} = \mu^t \cdot TM(\beta_i)$ [7].

In conclusion, this method quantifies uncertainties through fuzzy entropy and dynamically predicts state transitions using the Markov chain. It provides a systematic evaluation framework for the trustworthiness of cloud services from underlying indicators to global states, and is applicable to scenarios that require dynamic monitoring and risk prevention [7].

3.2. Markov Model for Predicting Network Traffic

3.2.1. Establishment of the Hidden Markov Model

In the immediate future, precise network traffic forecasts are essential for effective traffic management, which includes congestion reduction and traffic control [8]. For the foreseeable future, accurate network traffic forecasting will be crucial to the success of different traffic control techniques. However, because transportation systems are inherently unpredictable and chaotic, it can be difficult to achieve accurate forecasting in both free-flow and congested traffic conditions [8].

This paper showed a method that need to construct a hidden Markov model. The Hidden Markov Model, abbreviated as HMM, is a statistical model which is employed to depict a Markov process where there are hidden, unknown parameters [9]. It includes hidden states and observable states. The hidden states satisfy the Markov property, that is, the hidden state at the next moment depends only on the current hidden state and is independent of other historical states. Each hidden state will output an observable state with a certain probability [9]. HMM consists of observation set, which composed of all possible observation values; state transition probability matrix, which describes the probability of transitioning from one hidden step to another hidden step; and observation probability matrix, which represents the probability of generating each observation value under each hidden state [9]. In this case (prediction for network traffic), hidden states represent different levels or patterns of network activity, such as low traffic load, medium traffic load, and high traffic load states [8]. The observable states are the actual measurements of network traffic, which can be the number of packets transmitted per unit time [8]. The HMM attempts to classify the traffic fluctuations' results.

Construct a hidden Markov model with 3 hidden states M_1, M_2, M_3, which represents heavy traffic burden, moderate traffic burden and light traffic burden, respectively. The probability of transitioning between each state is based on the previous researches on specific observation states. As shown in the following matrix, where P stands for the matrix of concealed state transitions and π is the vector of original states probability [8]:

$$P = \begin{bmatrix} P\left(M_{(1\to1)}\right) & P\left(M_{(1\to2)}\right) & P\left(M_{(1\to3)}\right) \\ P\left(M_{(2\to1)}\right) & P\left(M_{(2\to2)}\right) & P\left(M_{(2\to3)}\right) \\ P\left(M_{(3\to1)}\right) & P\left(M_{(3\to2)}\right) & P\left(M_{(3\to3)}\right) \end{bmatrix} \quad (5)$$

DOI: 10.54254/2753-8818/100/2025.21986

$$\pi = \left[p\left(M_1\right), p\left(M_2\right), p\left(M_3\right)\right] \tag{6}$$

If from state i transitions to state j occurs N_{ij} times, then $\alpha_{ij} = N_{ij}/\sum_k N_{ik}$.

The calculating ways to predict the future states are shown in the follow. Using the transition matrix to iteratively compute future state distributions. For single-step prediction: the current state is π_t, then the next state distribution satisfies $\pi_{t+1} = \pi_t \cdot P$ [8]. For multi-step prediction: the future state π_{t+k} satisfies that $\pi_{t+k} = \pi_t \cdot P^k$ [8]. For example, if the observation state of the current state of the network traffic is "medium traffic load", that is, $\pi = [0,1,0]$. The transition matrix is $P = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.4 & 0.5 \end{bmatrix}$. Therefore, the next state distribution is $\pi P = [0.2,0.5,0.3]$, that is, have 20% probability of "high traffic load", 50% probability of "medium traffic load", and 30% probability of "low traffic load". For multi-step prediction, calculating $\pi_{t+k} = \pi_t \cdot P^k$, as a result, π_t converges to $[0.283,0.431,0.304]$, satisfies the stationary distribution. Therefore, no matter what the conditions are, the probability of "high traffic burden" is 0.283, the probability of "medium traffic burden" is 0.431, and the probability of "low traffic burden" is 0.304.

3.2.2. Optimization: Training and Parameter Estimation of HMM

The Baum-Welch Algorithm (EM Algorithm) is shown as the follow. In E-step need to compute forward probability $\alpha_t(i) = P(x_1, x_2, \cdots, x_t, \cdots, z_t = q_i)$, which represents the joint probability that at time t, the state is q_i and the sequence $x_1, \cdots, x_t$ has been observed. The backward probability $\beta_t(i) = P(x_{t+1}, \cdots, x_T | z_t = q_i)$, which represents the probability of observing the sequence $x_{t+1}, \cdots, x_T$ given that the state at time t is q_i [8]. M-step is used to update the parameter P (the state-transition matrix), B (the observation probability matrix), and π (the original state probability vector) to maximize the likelihood function $P(X) = \pi_{Z_1} \cdot \prod_{t=1}^{T-1} p_{z_t z_{t+1}} \cdot \prod_{t=1}^{T} b_{z_t}(x_t)$, which describes the probability of the observed sequence $X = (x_1, \cdots, x_T)$ occurring when given the parameters π, P, and B. By continuously adjusting π, P, and B, $P(X)$ can increase [8].

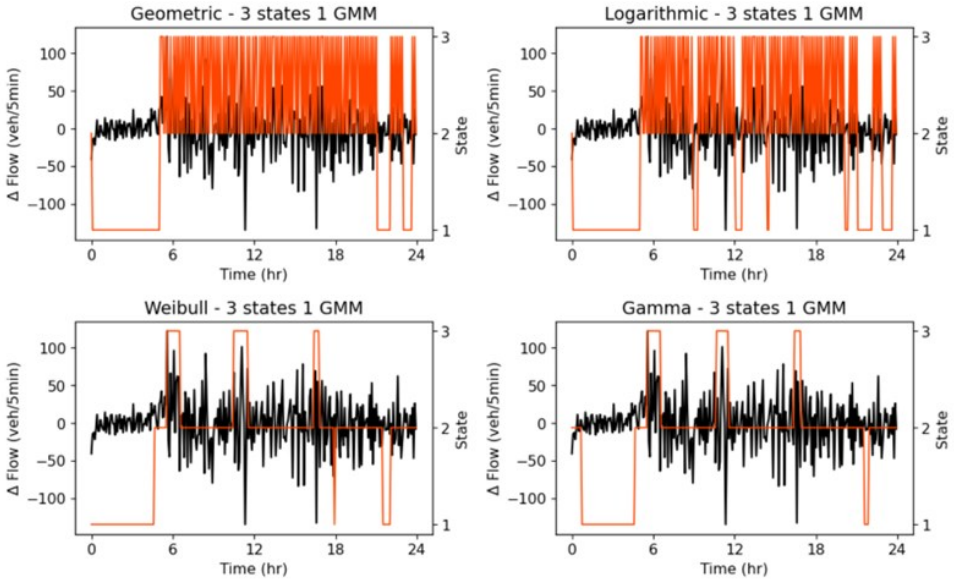

Figure 2: Determined hidden states based on varying sojourn densities, utilizing a model with 3 hidden states and incorporating 1 Gaussian mixture component [8].

DOI: 10.54254/2753-8818/100/2025.21986

Construction of Gaussian Mixture Model is $b_j(x) = \sum_{l=1}^{k} c_{jl} N(x|\mu_{jl}, \sum_{jl})$, where k is the number of mixture components, c_{jl} is the weight of the $l-th$ Gaussian component in the $j-th$ state, and $N(X|\mu_{jl}, \sum_{jl})$ is the Gaussian distribution function with mean μ_{jl} and covariance $\sum_{jl}$ [10]. As shown in Figure 2.

The overall goal is that after iteratively optimizing between the E-step and the W-step, the parameter combinations π, P, B and the parameters in Gaussian mixture model (if exist) that enhance the possibility to the likelihood of the observed data can be found to make the probability of the observed sequence occurring under the current parameters the largest [8]. One can then build up a multi-state dynamics model to capturing non-stationary through hidden states. In summary, the applications of hidden Markov model in modeling traffic by defining state and setting probability, estimating traffic by learning parameter, inferencing state and predicting traffic, and detecting anomaly by modeling normal patterns can help to better understand the patterns and dynamic changes of network traffic. They can also predict future traffic trends. As a result, it may enable human being to optimize the allocation of network resources, improve the quality of network services, and enhance network security protection.

4. Conclusion

In this paper, the author has explored the concepts and applications of Markov chains in computer science field. Citing the predictions of cloud service trusted state's transformation and network traffic as two great examples, to show that with the great use of integrating Markov chain into some computation systems, training the systems by adapting the values of some important parameters, which were added into the Markov process, the future state can be predicted and then optimized. However, the limitations of this paper lie in lack of the support from some valid data and some computations of the operation of Markov process to verify the normal functioning of the systems.

In the future, Markov chain can also be used in the following fields of computer science. Such as in strengthening deep reinforcement learning by combing Markov chain with deep neural networks to handle sequential data with long-term dependencies, capture complex modes in time series more effectively, and improve the accuracy of the model's prediction of future states. This can be applied to multimodal tasks such as video analysis and speech recognition. In addition, in modeling the user's interaction behaviors in the virtual environment, such as the order and frequency of the user's interactions with virtual objects. Use Markov chains to generate more natural and user-habit-compliant interaction experiences, providing a basis for the design and optimization of the virtual environment. Overall, this paper may serve as a starting point for further investigations of Markov chain in computer science field.

References

[1] *Qian Zhiye, Li Xue & Li Suogang (2024). Efficient domain name generation algorithm based on improved Markov chain Journal of Communications (S2), 52-58*

[2] *Chelly, I., Uziel, R., Freifeld, O., & Pakman, A. (2025). Consistent Amortized Clustering via Generative Flow Networks. arXiv preprint arXiv:2502.19337.*

[3] *Li, J., Gibbons, R., & Rockova, V. (2025). Deep Computerized Adaptive Testing. arXiv preprint arXiv:2502.19275.*

[4] *Li, H., Xu, X., Peng, Y., & Chen, C. H. (2021). Efficient Sampling for Selecting Important Nodes in Random Network. IEEE Transactions on Automatic Control, 66(3), 1321–28.*

[5] *Ross, S. M. (2014). Introduction to Probability Models. Netherlands: Academic Press.*

[6] *Latifi, F., Nassiri, R., Mohsenzadeh, M., & Mostafaei, H. (2025). A Markov chain-based multi-criteria framework for dynamic cloud service selection using user feedback. The Journal of Supercomputing, 81(1), 89.*

[7] *Yang, M., Jiang, R., Wang, J., Gui, B., & Long, L. (2024). Assessment of cloud service trusted state based on fuzzy entropy and Markov chain. Scientific Reports, 14(1), 1-18.*

DOI: 10.54254/2753-8818/100/2025.21986

[8] *Sengupta, A., Das, A., & Guler, S. I. (2023). Hybrid hidden Markov LSTM for short-term traffic flow prediction. arXiv preprint arXiv:2307.04954.*

[9] *H Weiß, C., & Swidan, O. (2024). Hidden-Markov models for ordinal time series. AStA Advances in Statistical Analysis, 1-23.*

[10] *Saravanakumar, R., TamilSelvi, T., Pandey, D., Pandey, B. K., Mahajan, D. A., & Lelisho, M. E. (2024). Big data processing using hybrid Gaussian mixture model with salp swarm algorithm. Journal of Big Data, 11(1), 167.*