UDC: 519.217.2:519.245]:656.18.021.2
Professional paper

MATHEMATICAL ANALYSIS OF STOCHASTIC TRANSITION MATRICES IN MARKOV MODELS USING MONTE CARLO AND MCMC SIMULATION

Vesna KNIGHTS[1], Olivera PETROVSKA[2], Marija PRCHKOVSKA[3]

[1]*Faculty of Technology and Technical Sciences, University" St. Kliment Ohridski" -Bitola, Republic of North Macedonia*
[2] *Faculty of Technical Science, Mother Teresa University, 1000, Skopje, Republic of North Macedonia;*
[3] *Faculty of Computer Science, Mother Teresa University, 1000, Skopje, Republic of North Macedonia;*
[*]*Corresponding author e-mail: vesna.knights@uklo.edu.mk*

Abstract

Road network control remains a mathematically and practically complex challenge in optimizing urban traffic systems. Traffic congestion, particularly at unsignalized intersections, continues to be a critical issue in densely populated cities. These intersections often allow vehicles to navigate without external control, leading to randomized vehicle behavior and increased risks of conflicts or accidents. Traditional methods have addressed intersection congestion through various control strategies, yet few have formally modeled these dynamics using rigorous mathematical structures.

In this paper, we construct a finite-state Markov chain model to describe the stochastic evolution of traffic states at an unsignalized intersection. The system is represented by a row-stochastic transition matrix, where each state corresponds to a discrete traffic condition (e.g., low flow, moderate flow, queued, and congested). The stationary distribution vector is derived under the condition that its multiplication with the transition matrix leaves it unchanged, capturing the stable, long-term distribution of traffic across states. A spectral analysis of the transition matrix is conducted, with emphasis on the subdominant eigenvalue and the spectral radius, to evaluate the rate of convergence toward equilibrium and assess intersection stability.

The results demonstrate that the spectral properties of the transition matrix serve as robust indicators of system performance and congestion potential. Intersections with slowly decaying eigenvalues exhibit persistent traffic buildup, whereas rapidly converging systems suggest smoother flow. These findings establish a formal mathematical foundation for diagnosing and potentially optimizing traffic behavior at critical road network points.

Keywords: Markov chains, Intersection, Mathematical Modeling.

1. Introduction

Unsignalized intersections represent a critical element of modern traffic networks, particularly in urban and suburban areas where full signalization is either economically unfeasible or unnecessary due to moderate traffic volumes. At these intersections, right-of-way is typically determined by signage (e.g., stop signs, yield signs) or informal rules of priority, requiring drivers to make real-time decisions based on their perception of gaps in opposing traffic flows. These decisions are inherently stochastic and are influenced by driver behavior, vehicle characteristics, and environmental conditions [1-2].

Unlike signalized intersections, where traffic movements are governed by deterministic light cycles, unsignalized intersections rely heavily on individual judgment and interaction between road users. This introduces considerable variability and uncertainty into the traffic system. Traditional deterministic models often fail to capture this randomness, leading to limited predictive accuracy when assessing performance metrics such as average delay, queue length, or conflict probability [3,4]. Therefore, there is a pressing need for probabilistic models that more accurately reflect the inherent randomness of vehicle interactions at these locations [5].

https://doi.org/10.62792/ut.jnsm.v10.i19-20.p3095

Markov chains, a class of stochastic processes characterized by memoryless transitions between discrete states, offer a powerful mathematical framework for modeling systems that evolve probabilistically over time. When applied to traffic at unsignalized intersections, Markov models can capture the dynamic state transitions governed by vehicle arrivals, departures, and gap acceptance behavior. Each state can represent a snapshot of the intersection's condition, such as the number of vehicles queued on the minor road, the availability of acceptable gaps on the major road, or the progression of individual driver decisions [7-8].
This paper presents mathematical modeling using Markov chains, with application in traffic (intersections without traffic lights). The results should show whether the system has stable and predictable behavior. This provides a basis for future proposals for the introduction of traffic lights or roundabouts based on mathematical indicators [9-10].
Central to any Markov model is the stochastic transition matrix, which encapsulates the probabilities of moving from one traffic state to another in a single time step. Analyzing the structure and properties of this matrix allows researchers to understand long-run system behavior, identify equilibrium conditions, and explore how sensitive these outcomes are to variations in traffic flow parameters [11,12]. Despite the utility of Markov models in traffic engineering, the literature reveals a gap in rigorous mathematical analysis of transition matrices tailored to unsignalized intersection dynamics, particularly when considering realistic traffic flow distributions and stochastic driver behavior[13-15].
This study aims to fill that gap by providing a detailed mathematical treatment of stochastic transition matrices in the context of Markov models for unsignalized intersections. We focus on both theoretical aspects, such as matrix stochasticity, irreducibility, ergodicity, and limiting behavior, and applied outcomes, including the estimation of steady-state probabilities and their implications for intersection performance [16,17].
By linking mathematical theory with practical traffic applications, this research offers a dual contribution: it advances the theoretical understanding of Markov processes in transportation systems and provides a foundation for data-driven analysis and design of safer, more efficient unsignalized intersections [18-21].

2. Markov Chain

Markov chains are named after the Russian mathematician Markov Andrei, who first formulated these processes and studied their properties of these processes. Markov chains have wide application in various fields of science, such as: physics, technology, transport, economics, and others.
A given system will have the property of a Markov chain if we have its current state, and the future state of the system does not depend on past states, i.e., something that happened in the past does not affect and does not give a forecast for the future.
From the state space X (discrete or continuous) and from the space T of the argument t (discrete or continuous) Markov processes can be:

- Discrete Markov processes with discrete time, which are called a Markov chain,
- Continuous Markov processes with discrete time, which are called a Markov sequence,
- Discrete Markov processes with continuous time,
- Continuous Markov processes with continuous time.

Definition: A random process $X(t)$ is a Markov process if for arbitrary n moments in time $t_1<t_2<\ldots<t_n$ in the segment $[0,T]$, the random distribution function of the last value $X(t_n)$, for fixed values $X(t_1), X(t_2),\ldots,X(t_{n-1})$, depends only on $X(t_{n-1})$, i.e. for the given values $x_1, x_2, \ldots, x_n$ the relation is satisfied

$$Pr\{(X(t_n) \leq x_n \,/\, X(t_1) = x_1, \ldots, X(t_{n-1}) = x_{n-1}\} = Pr\{(X(t_n) \leq x_n \,/\, X(t_{n-1}) = x_{n-1}\}$$
[1]

Types of Markov Chains:

- Discrete-time or continuous-time
- Finite or infinite state space
- Ergodic, irreducible, aperiodic, and absorbing chains (depending on specific properties)

The transition probabilities between states and are determined based on:

- Arrival Rates: Modeled using Poisson distributions to capture vehicle arrivals on both major and minor roads.
- Gap Acceptance Behavior: Represented by a probability distribution function describing the likelihood that a driver will accept a gap of duration.
- Departure Rules: Vehicles are allowed to depart if the gap is acceptable, otherwise, they remain in the queue.

Unregulated intersections, due to their unpredictable nature, display stochastic behavior of vehicle crossings, which can lead to congestion and conflicts.

3. Methodology of Research

A stochastic transition matrix is constructed that represents the changes in four-time intervals during the day: morning, forenoon, afternoon, and evening. To provide probabilities of transition from one state to another, the raw data from the observation are normalized in rows. This procedure provides the necessary structure for the application of the Markov model.
The following values are analyzed: fixed vector of the system, spectral radius, and the speed of stabilization of the traffic through eigenvalues. A comparison of the obtained results with the Monte Carlo method is made. Also, a Machine Learning method and parameter inference are applied to the obtained parameters.

3.1. Markov model for intersection: At an intersection without traffic lights, the movement of vehicles is monitored during the day in four time intervals. Each row in the matrix represents the state in a certain time interval:

- First row → Morning (S_1)
- Second row → Forenoon (S_2)
- Third row → Afternoon (S_3)
- Fourth row → Evening (S_4)

For each time period, the change in the load on the intersection (the number of vehicles) is analyzed, during which transitions from one state to another are observed. This forms a stochastic transition matrix, in which each row represents the probability of transitioning to the next state (from S_1 to S_4). In the matrix below, m and n are unknown coefficients that need to be determined so that each row (state) represents a correct probability vector (sum = 1).
Definition: The vector $p=(p_1, p_2, \ldots, p_n)$ is called a probability vector if its components are non-negative numbers and their sum is 1.
The probability of transition from S_1 to the others is obtained by dividing the number of vehicles in each period by the total number of vehicles.
To determine the transition probabilities from each state, we use observational data. For example, during the **Morning interval (S_1)**, the following transitions were observed:

- 30 vehicles transitioned to Forenoon (S_2),
- 40 vehicles transitioned to Afternoon (S_3),
- 20 vehicles transitioned to Evening (S_4), and
- 30 vehicles remained in the Morning state (S_1 to S_1).

This gives a total of 120 vehicles observed during the Morning interval. To compute the transition probabilities, we divide each observed frequency by the total number of vehicles:

From S_1 to →	Forenoon (S_2)	Afternoon (S_3)	Evening (S_4)	Morning (S_1)
Probability	30/120=1/4	40/120=1/3	20/120=1/6	30/120=1/4

[1]

These values form the first row of the transition matrix. A similar process is applied to compute the probabilities for the other time intervals (S_2, S_3, S_4), ensuring that each row sums to 1 and reflects realistic traffic transitions during the day.

Morning(S_1)

$$A = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} \\ \frac{5}{18} & \frac{1}{3} & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \quad [2]$$

Fixed vector of the system

Definition: A stochastic matrix **A is regular** if all elements of some degree A^n are positive.

Theorem: Let A be a regular stochastic matrix. Then:
For the matrix A there exists a unique fixed vector t, for which all components are positive:
$t \cdot A = t$ [3]

We are looking for the vector t=(x, y, x, w) such that:
$x + y + z + w = 1$ [4]

$$t = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \quad [5]$$

$$A = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} \\ \frac{5}{18} & \frac{1}{3} & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \qquad [6]$$

$$\begin{aligned} x &= \frac{1}{4}x + \frac{1}{8}y + \frac{1}{8}z + \frac{5}{18}w \\ y &= \frac{1}{4}x + \frac{1}{2}y + \frac{1}{8}z + \frac{1}{3}w \\ z &= \frac{1}{3}x + \frac{1}{4}y + \frac{1}{4}z + \frac{1}{6}w \\ w &= \frac{1}{6}x + \frac{1}{8}y + \frac{1}{2}z + \frac{1}{3}w \end{aligned} \qquad [7]$$

Fixed vector of the system by time intervals

For the fixed vector of the system over time intervals, we obtain:

$$t \approx \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \approx \begin{bmatrix} 0.1815 \\ 0.3144 \\ 0.2352 \\ 0.2686 \end{bmatrix} \qquad [8]$$

These values represent the long-term (stable) probability that the system will be in each of the weather states:

- 18.15% morning (S_1),
- 31.44% forenoon (S_2),
- 23.52% afternoon (S_3),
- 26.86% evening (S_4).

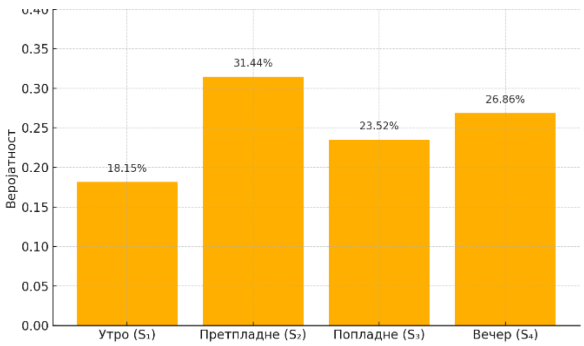

Figure1. Fixed vector of the system by time intervals

Diagram of states and values from a stationary distribution

$$t \approx \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \approx \begin{bmatrix} 0.1815 \\ 0.3144 \\ 0.2352 \\ 0.2686 \end{bmatrix} \quad [9]$$

$$t \cdot A = t$$

$$\begin{aligned} x &= \frac{1}{4}x + \frac{1}{8}y + \frac{1}{8}z + \frac{5}{18}w \\ y &= \frac{1}{4}x + \frac{1}{2}y + \frac{1}{8}z + \frac{1}{3}w \\ z &= \frac{1}{3}x + \frac{1}{4}y + \frac{1}{4}z + \frac{1}{6}w \\ w &= \frac{1}{6}x + \frac{1}{8}y + \frac{1}{2}z + \frac{1}{3}w \end{aligned} \quad [10]$$

Figure 2 shows a diagram of the states and values from the stationary distribution, and each node is named according to the time of day, and its stationary probability is included.

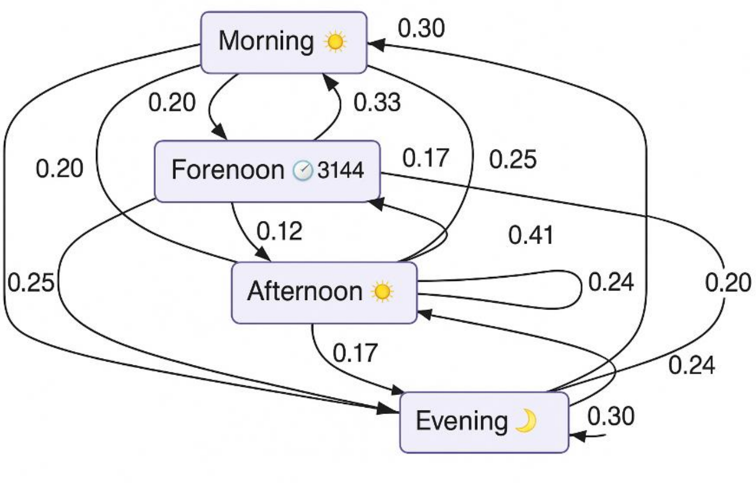

Figure 2. Diagram of states and values from a stationary distribution

This model can be used to predict traffic intensity at different times of the day. Traffic lights, roundabouts, or intelligent signaling can be planned, especially for the S_2 interval, where there is the highest intensity. For this intersection with moderate and predictable traffic, a roundabout is a better solution than traffic lights.

4. Comparison with the Monte Carlo model

Aim: To simulate traffic movement across 4-time intervals of the day (morning, forenoon, afternoon, evening) based on a stochastic matrix, and after many simulations, to obtain a realistic estimate of how often each condition occurs.

1. Initial definitions

We have 4 weather conditions:

$$S = \{S_1 = Morning, S_2 = Forenoon, S_3 = Afternoon, S_4 = Evening\}$$

Stochastic transition matrix:

$$P = \begin{bmatrix} 0.25 & 0.25 & 0.333 & 0.167 \\ 0.125 & 0.5 & 0.25 & 0.125 \\ 0.125 & 0.125 & 0.25 & 0.5 \\ 0.278 & 0.333 & 0.167 & 0.222 \end{bmatrix} \qquad [11]$$

2. Step simulation

At each step, a new state is calculated as a random selection from the distribution given in the order belonging to the current state.

Equation:
If we are in state S_i, then:
Next state = Sample from categorical distribution with probabilities P_{i1}, P_{i2}, P_{i3}, P_{i4}.
If the current state is S_1, then we have state:

$$P_1 = [0.25, 0.25, 0.333, 0.167] \quad [12]$$

We generate a random number $r \in [0,\ 1]$

We look at which interval r falls into (Table 1):

Table 1: interval of random number r

Interval	Next state
[0.00,0.25)	S_1
[0.25,0.50)	S_2
[0.50,0.833)	S_3
[0.833, 1.00)	S_4

$$S_{t+1} = \begin{cases} S_1 \quad if \quad r < P_{i1} \\ S_2 \quad if \quad P_{i1} \le r < P_{i1} + P_{i2} \\ S_3 \quad if \quad P_{i1} + P_{i2} \le r < P_{i1} + P_{i2} + P_{i3} \\ S_4 \end{cases} \qquad [13]$$

For each step $t=1, 2, ..., N$, we store the state S_t. Finally:

C_j = Number of times the system has been in S_j

Distribution (normalization): $\widehat{\pi_j} = \frac{C_j}{N}$

$\widehat{\pi_j}$= Approximate probability of the condition S_j
C_j= Number of visits to S_j
N = Total number of simulated steps

For the research on the specific case and N = 10 000, we obtain:

C_1 = 1870, C_2 = 3122, C_3 = 2382, C_4 = 2626

$$\widehat{\pi_1} = \frac{1870}{10000} = 0.187 = 18\% \qquad \widehat{\pi_3} = \frac{2382}{10000} = 0.2382 = 23.82\%$$

$$\widehat{\pi_2} = \frac{3122}{10000} = 0.3122 = 31.22\% \qquad \widehat{\pi_4} = \frac{2626}{10000} = 0.2626 = 26.26\% \qquad [14]$$

Comparison: Markov model vs. Monte Carlo
Figure 3 shows the simulation of comparison between the two models according to the obtained values.

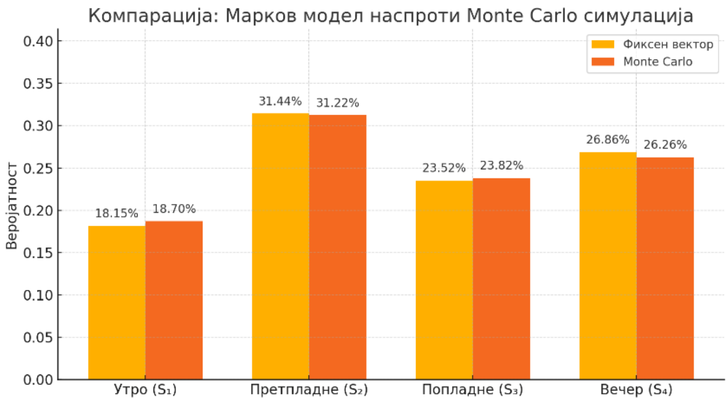

Figure 3. Comparison - Markov model vs. Monte Carlo

The Monte Carlo method with 10,000 simulations very accurately follows the long-term stable distribution obtained with the Markov model.

5. Improving the Markov Chain Model with Markov Chain Monte Carlo (MCMC) Machine Learning

Markov Chain Monte Carlo (MCMC) is a powerful class of algorithms used to sample from complex probability distributions, especially when direct sampling is difficult or impossible. It combines two main ideas:

- Markov Chains — These are random processes where the next state depends only on the current state (memoryless property).
- Monte Carlo Methods — These are statistical techniques that rely on random sampling to perform numerical estimations.

In statistics, we often want to know about a probability distribution, such as the posterior distribution in Bayesian inference. If that distribution is too complicated to describe analytically, MCMC helps us generate samples from it, so we can estimate quantities like means, variances, or probabilities.

The idea is to construct a Markov chain that has the target distribution as its stationary distribution. Then, by simulating the chain for many steps, the samples generated can be used as approximations of the target distribution.
The goal of MCMC is to generate samples from a complex (often unknown) distribution, especially when direct computation is impossible.
How does MCMC work?

- Start with an initial state x_0
- Define a transition rule (usually with "Metropolis-Hastings" or "Gibbs sampling")
- Accept or reject the new state
  based on probability (according to the target distribution)
- Repeat the process many times,
  so that the result is a sequence of states $x_1, x_2, ..., x_n$, which are real samples of the desired distribution

Markov Chain Monte Carlo (MCMC) simulation

Posterior Distribution Histogram: Shows the probability distributions for each parameter obtained from the MCMC simulation, as shown in Figure 4.

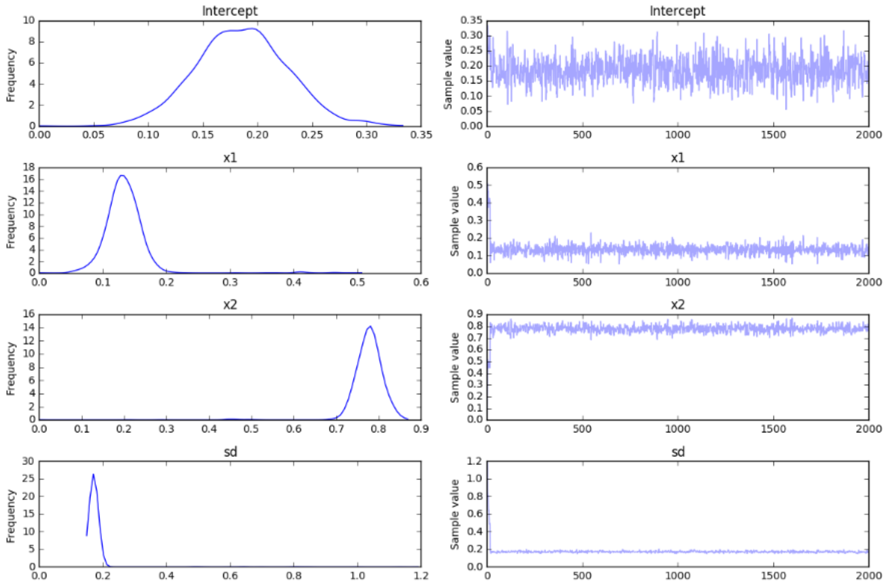

Figure 4. Markov Chain Monte Carlo (MCMC) simulation

Also shown in Figure 4 is a **Trace plot (MCMC Trace)**:
Here, the MCMC algorithm is plotted through values for each parameter over 2000 iterations. Although the line is **chaotic, but stable around the central valu**e, which means that the model has **converged** and the results are stable.

Machine learning results

```
print('Model 1:',mean_absolute_error(true_regression_line,model1))
print('Model 2:', mean_absolute_error(true_regression_line,model2))
print('Average:',mean_absolute_error(true_regression_line,model1*.5+model2*.5))
print('MCMC:',mean_absolute_error(true_regression_line,intercept+x1param*model1+x2param*model2))
```

```
Model 1: 0.382576232379
Model 2: 0.153438347644
Average: 0.20277077251
MCMC: 0.129878529633
```

Figure 5. Python Code and MAE Comparison for Model 1, Model 2, Averaged Model, and MCMC-Based Model

In Figure 5, Python code is presented for calculating the mean absolute error (MAE) of three modeling approaches. The MAE is shown for Model 1, Model 2, the average of the two models, and a model enhanced using the Markov Chain Monte Carlo (MCMC) method.

- Model 1 (with higher) gives the highest error (MAE = 0.38)
- Model 2 (less noise) shows better precision (MAE 0.15)
- Average aggregation of both models gives moderate error (MAE=0.20)
- The best results are obtained with MCMC model (Markov Chain Monte Carlo), where the error is lowest (MAE=0.13)

6. Conclusions

- Instead of classical regression, Markov Chain is used – a process in which the next state depends only on the current state, not on the entire history
- Comparison with Monte Carlo – a technique based on random sampling to approximate solutions.
- Model improvement with machine learning: Markov Chain Monte Carlo (MCMC)
- If used for:
- For long-term stability and structure → Markov model
- For realistic dynamics and fluctuations → Monte Carlo simulation
- For integration of models with noise and reliability → MCMC analysis

References

[1]. Drew, D. R. (1968). Traffic Flow Theory and Control. McGraw-Hill.

[2]. Treiber, M., & Kesting, A. (2013). Traffic Flow Dynamics: Data, Models and Simulation. Springer.

[3]. Dion, F., Rakha, H., & Kang, Y. (2004). Evaluation of the Travel Time Reliability in Transportation Networks Using Dynamic Network Loading. Transportation Research Record, 1876(1), 18–27.

[4]. Papoulis, A., & Pillai, S. U. (2002). Probability, Random Variables, and Stochastic Processes (4th ed.).McGraw-Hill.

[5]. Zhong, M., & Ziliaskopoulos, A. (2006). A Stochastic Timed Petri Net Model for Urban Traffic Networks. Transportation Research Part B: Methodological, 40(1), 99–122.

[6]. Knights, V., & Petrovska, O. (2024). Dynamic Modeling and Simulation of Mobile Robot Under Disturbances and Obstacles in an Environment. Journal of Applied Mathematics and Computation, 8(1), 59–67.

[7]. Antoska Knights, V., Gacovski, Z., & Deskovski, S. (2018). Guidance and Control System for Platoon of Autonomous Mobile Robots. Journal of Electrical Engineering, 6, 281–288. https://doi.org/10.17265/2328-2223/2018.05.005

[8]. Antoska, V., Jovanović, K., Petrović, V. M., Bašcarević, N., & Stankovski, M. (2013). Balance

Analysis of the Mobile Anthropomimetic Robot Under Disturbances – ZMP Approach. International Journal of Advanced Robotic Systems, 10(4).
[9]. Antoska Knights, V., & Gacovski, Z. (2024). Methods for Detection and Prevention of Vulnerabilities in the IoT (Internet of Things) Systems. IntechOpen.
[10]. Knights, V., Petrovska, O., & Prchkovska, M. (2024). Enhancing Smart Parking Management through Machine Learning and AI Integration in IoT Environments. IntechOpen.
[11]. Knights, V., & Prchkovska, M. (2024). From Equations to Predictions: Understanding the Mathematics and Machine Learning of Multiple Linear Regression. Journal of Mathematical & Computer Applications, 3.
[12]. Knights, V., Petrovska, O., Bunevska-Talevska, J., & Prchkovska, M. (2025). Machine Learning Models and Mathematical Approaches for Predictive IoT Smart Parking. Sensors, 25(7), 2065. https://doi.org/10.3390/s25072065
[13]. Knights, V., Petrovska, O., & Prchkovska, M. (2025). Integrating Machine Learning and AI into IoT-Enabled Smart Parking. In Dhoska, K., & Spaho, E. (Eds.), Bridging Horizons in Artificial Intelligence, Robotics, Cybersecurity, Smart Cities, and Digital Economy. ICITTBT 2024. Sustainable Economy and Ecotechnology. Springer, Cham. https://doi.org/10.1007/978-3-031-72029-1_5
[14]. Theodoridis, S. (2020). Machine Learning: A Bayesian and Optimization Perspective. Elsevier Ltd.
[15]. Praba, R., Darshan, G., Roshanraj, K. T., & Surya Prakash, P. B. (2021). Study On Machine Learning Algorithms. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 7(4), 67–72.
[16]. Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., & Yin, B. (2021). Deep Learning on Traffic Prediction: Methods, Analysis, and Future Directions. IEEE Transactions on Intelligent Transportation Systems, 23(8), 4927–4943. https://doi.org/10.1109/TITS.2021.3049105
[17]. Shafique, M., Naseer, M., Theocharides, T., Kyrkou, C., Mutlu, O., & Orosa, L. et al. (2020). Robust Machine Learning Systems: Challenges, Current Trends, Perspectives, and the Road Ahead. IEEE Design & Test, 37(2), 30–57. https://doi.org/10.1109/MDAT.2020.2971217
[18]. Al-Garadi, M. A., Mohamed, A., Al-Ali, A., Du, X., & Guizani, M. (2020). A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. IEEE Communications Surveys & Tutorials, 22(3), 1646–1685. https://doi.org/10.1109/COMST.2020.2968417
[19]. Jha, S. (2023). Lightning Talk: Trinity - Assured Neuro-symbolic Model Inspired by Hierarchical Predictive Coding. 2023 60th ACM/IEEE Design Automation Conference (DAC), 1–2. https://doi.org/10.1145/3449639.3459291
[20]. Kriebel, F., Rehman, S., Hanif, M. A., Khalid, F., & Shafique, M. (2018). Robustness for Smart Cyber Physical Systems and Internet-of-Things: From Adaptive Robustness Methods to Reliability and Security for Machine Learning. 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 581–586. https://doi.org/10.1109/ISVLSI.2018.00111