AN EFFICIENT ALGORITHM FOR TRAFFIC FLOW EVALUATION ON SMART CITIES BASED ON DEEP LEARNING

by

Adel BAHADDAD[a], Khalid ALMARHABI[b], Mohammed ALSHAHRANI[c*], Mohammed MNZOOL[d], Adil A. M. ELHASSAN[d], Arwa ALZUGHAIBI[e], and Ahmed M. ALGHAMDI[f]

[a] Department of Information System, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
[b] Department of Computer Science, College of Engineering and Computing in Al-Qunfudah, Umm Al-Qura University, Makkah, Saudi Arabia
[c] Department of Computer, Applied College, Najran University, Najran, Saudi Arabia
[d] Department of Civil Engineering, College of Engineering, Taif University, Taif, Saudi Arabia
[e] Applied College, Taibah University, Medina, Saudi Arabia
[f] Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia

Original scientific paper
https://doi.org/10.2298/TSCI2502597B

Following the emergence of the knowledge-based economy, the digital economy, and the intelligent economy, smart cities are poised to represent the next phase in urban development. These cities aim not only to leverage both physical and digital infrastructures for urban advancement but also to harness intellectual and social capital as essential elements of urbanization. Smart cities are recognized as pivotal catalysts for transforming various sectors by integrating multiple municipal systems such as transportation, healthcare, and operational frameworks. The concept of a smart society evolves from smart cities, characterized by a digitally interconnected, knowledge-driven community that actively pursues social, environmental, and economic sustainability. Recently, deep learning has gained traction due to its ability to effectively tackle complex problems across diverse applications using both supervised and unsupervised learning methods. This approach relies on advanced techniques for managing large datasets and multilayer neural networks, which often outperform traditional ANN in processing historical data. This paper introduces a novel algorithm based on deep learning designed to accurately predict traffic flow behavior. The algorithm learns from multivariate sequence data by analyzing spatio-temporal dependencies and non-linear correlations. Simulation results demonstrate that the proposed method surpasses existing algorithms in performance.

Key words: *deep learning, neural networks, traffic flow, machine learning*

Introduction

With the rapid expansion of cities driven by urbanization and a growing global population, smart city services have become increasingly prevalent [1]. A *smart city* leverages information and communication technologies to collect, process, and unify data from essential urban infrastructures, as described in [2]. Among the technological advancements shaping modern

* Corresponding author, e-mail: moaalshahrani@nu.edu.sa

urban environments, the IoT stands out for its transformative influence on various aspects of everyday life [3]. Beyond improving quality of life, IoT also plays a crucial role in fostering economic growth. Consequently, IoT research and solution development receive significant support from technology enterprises and academic organizations [4]. By leveraging a network of interconnected devices-ranging from sensors and actuators to smart gadgets and home automation systems-IoT applications facilitate efficient data exchange and processing [5, 6]. The implementation of IoT technology is fundamental to the realization of smart city functionalities. By leveraging cutting-edge communication advancements, smart cities aim to elevate living standards and optimize services for their inhabitants [7, 8]. This vision incorporates a wide array of interconnected systems [9], such as intelligent housing, modern infrastructure, advanced energy grids, eco-friendly solutions, innovative educational platforms, efficient healthcare systems, optimized transportation networks, and dynamic urban housing, fig. 1. Despite their potential, these technologies introduce considerable complexities in network operations. Each application demands specific performance metrics, including varying needs for bandwidth, latency, packet delivery reliability, jitter, and other service quality parameters. These applications generate varying levels of network traffic and involve numerous connected devices, such as actuators and sensors. For example, video surveillance plays a crucial role in identifying congested areas and enhancing preparedness for traffic incidents and emergencies [10]. This type of application requires stringent network conditions, including substantial bandwidth and minimal jitter, to function effectively. Real-time applications, such as online gaming and voice calls, depend on seamless interactions and exhibit high sensitivity to latency. The diverse nature and rapid expansion of these smart city technologies, each producing distinct traffic patterns, present significant challenges for maintaining quality-of-service (QoS). Addressing these complexities demands thorough and strategic solutions.

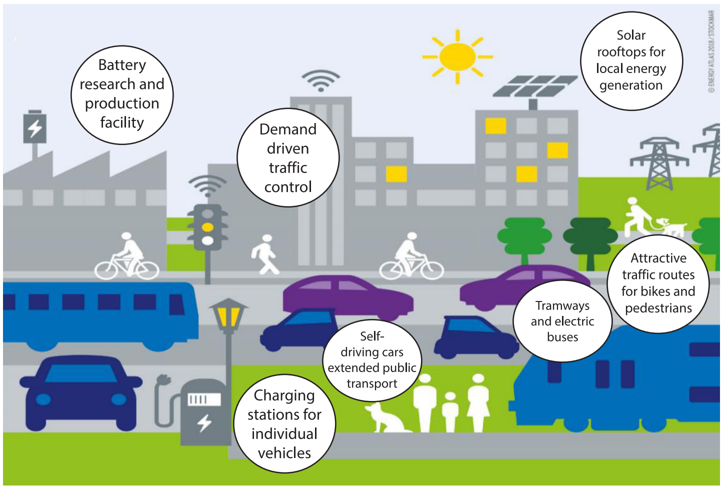

Figure 1. The typical configuration of a network for a smart city

The internet's original architecture was not designed to accommodate QoS requirements; instead, it prioritized best-effort data delivery. Over time, various approaches have been introduced to address QoS challenges, including the development of integrated services, which take a distinct approach [9]. Integrated services aim to ensure QoS for individual data flows by allocating sufficient network resources along the entire transmission path. This approach requires routers to maintain detailed states for each flow, enabling support for both multicast and unicast applications. However, this design increases the complexity of routers, making them more prone to failures. Furthermore, the need to track flow states at every network node limits scalability, particularly as the number of flows increases [9, 11, 12]. To overcome these limitations, differentiated services were developed as an alternative. This model simplifies scalability issues by grouping traffic into QoS classes rather than managing individual flows. Network traffic classification (TC) relies on with IPv4 and IPv6 headers' differentiated services codepoint (DSCP) field. However, these techniques have seen limited implementation in large-scale networks [9, 13].

Traffic categorization is essential for numerous network functions, including monitoring, QoS optimization, and security enforcement [14]. It underpins service differentiation systems, supporting applications like streaming and Voice over IP (VoIP) [15]. By identifying traffic types, resources such as bandwidth and latency are able to be distributed according to project-specific demands to maintain the required QoS. Various techniques allow traffic sorting without modifying the TCP/IP header, with port-based classification being a widely used method that identifies traffic types based on designated port numbers [16]. However, the rise of modern applications employing dynamic ports and tunnelling has rendered this method largely ineffective. Alternatively, deep packet inspection (DPI) examines packet content, comparing it against predefined signatures to determine traffic types [17]. While DPI can be effective, it is resource-intensive, struggles with encrypted traffic, and raises significant privacy concerns [18-20].

The potential of machine learning (ML) algorithms to deliver high accuracy and efficiency in TC has sparked considerable research interest. In supervised learning, the classification process involves several key stages. First, traffic features that represent flow characteristics, such as packet length, are identified and extracted. Next, a ML model is developed based on these features. Finally, the classifier is trained to map specific features to their corresponding categories.

Data sources

Table 1 presents an overview of fixed-sensor technologies, based on [16, 21]. Inductive loop sensors function by monitoring variations in the magnetic field when conductive materials are present. Magnetic sensors detect ferrous metal objects through shifts in Earth's magnetic field. In contrast, video image processors rely on analyzing camera footage to extract data on traffic flow.

Table 1. Traffic sensor capabilities [22, 23]

Sensor technology	Vehicle classification	Multiple detected zone	Capital cost
Inductive loop			45391
Magnetic sensor			0.5-3
Video image processor		✓	45521
Microwave radar	✓	✓	45516
Laser radar sensor	✓	✓	6.5-8
Active infrared	✓	✓	6.5-8
Passive infrared			0.8-1
Audio sensor	✓		3.5 - 7

Microwave radar technology emits and receives electromagnetic waves to detect objects. Active infrared sensors utilize reflected infrared light from laser diodes to identify vehicles. In contrast, passive infrared sensors detect energy reflected from either vehicles or the surrounding environment. Laser radar sensors project near-infrared beams across traffic lanes to monitor vehicle movement. Audio sensors estimate traffic volume or density by processing audio signals. Moreover, the widespread adoption of GPS-enabled smartphones and vehicles has introduced a new data stream that can complement traditional sensors or even serve as an alternative when they are unavailable.

Advancements in estimating traffic flow for advanced traffic management systems (ATMS) and intelligent transportation systems (ITS) have been made possible through GPS trajectory data. By utilizing mobile crowd-sensing methods, these trajectories can also be collected from 95 smartphones, whether from pedestrians or vehicles [24], providing valuable insights into road traffic patterns for both motorists and pedestrians.

In this section, a comparison is made between fixed-location sensors and mobile sensors, based on the criteria listed are:
– Positives and negatives.
– Commonly used types of measured data.
– A list of datasets that are freely accessible.

Information from stationary sensors

Traditional fixed-position sensors are primarily based on presence detection technologies, which are installed at specific locations, referred to as point p. This set-up guarantees that measurements are consistently taken at the exact same spot on the roadway. Depending on the type of sensor deployed, it may be capable of monitoring one or more lanes. The data gathered from these fixed sensors can be represented as a sequence of structured measurements, labelled as m_p, recorded at position p while monitoring traffic flow in a specific direction along a road segment (such as when using an inductive loop detector):

$$\overline{m}_p = m_{p,t}, \ t = 1,2,3,\ldots T \tag{1}$$

where $m_{p,t}$ is the measurement's estimate at time t and position p.

The measurements provided by a sensor are contingent upon its functionality. Basic sensors may only track traffic volume by counting vehicles, whereas more advanced sensors can assess factors such as speed and flow density. The most sophisticated sensors even have the ability to classify vehicles, offering a more comprehensive view of traffic dynamics.

Stationary fixed sensors offer a key benefit over mobile sensors in terms of reliability, as they continuously capture data from every vehicle that passes. On the other hand, GPS sensors are limited to tracking only one vehicle at a time. This makes fixed sensors more suitable for generating aggregate data, like total vehicle counts or traffic density.

Flow estimation accuracy is influenced by the number of mobile sensors present in the area. A major drawback of traditional fixed-position sensors is their inability to monitor the movement paths of objects in motion. As a result, connecting different road segments becomes challenging, and any conclusions drawn must rely on inferred data, such as through spatial correlation analysis. Additionally, setting up and maintaining an extensive sensor network can be highly costly.

Automated fare collection (AFC) systems, while primarily focused onll management, also capture valuable data from fixed points. The data gathered through smart tickets can offer significant insights into patterns of urban mobility [25].

Each transaction is linked to the cardholder's identification, enabling the system to capture both the boarding (entrance) and arrival (exit) locations. As a result, even though data is collected at fixed points, it can reveal spatial correlations and provide additional insights. Research on smart ticketing card data has been discussed in over ten studies [26-29], but none of these used publicly available datasets.

Most datasets are restricted due to legal constraints, with only a few available for research use. One frequently utilized dataset in various studies is the Caltrans performance measurement system (PeMS) dataset [30-32]. It includes 135 datasets from the california department of transportation, collected from over 39000 different detectors, such as inductive loop sensors, magnetic sensors, and microwave radar sensors, all providing real-time data.

The freeway network in major California urban areas is monitored by these sensors. the archived data user service holds more than ten years of historical data, including PeMS data, and offers extensive information such as traffic statistics, vehicle classifications, incident details, and census traffic counts. This data is collected from Caltrans and a variety of local agency systems.

The traffic information service provides vital data for major roads in England [33]. Since April 2015, it has offered statistics on average travel times, speeds, and traffic flow every 15 minutes for the Strategic Road Network, which includes motorways and A roads managed by Highways England. These travel time and speed estimates are based on information from 145 fixed sensors within the system.

Moving sensor data

Mobile crowd-sensing techniques gather valuable data from devices in motion, such as smartphones and GPS-enabled vehicles (*e.g.*, taxis and bicycles), to track the routes of both vehicles and pedestrians. A key area of focus for many of these initiatives is monitoring urban transit systems, including buses, trams, and subways, as well as mapping road conditions and alerting authorities to issues like bumps that require intervention [34].

Additionally, call detail records (CDR), which are collected by network operators from mobile devices, provide another rich source of data. Though CDR do not include communication content, they capture essential metadata, such as location information, that helps define the transaction. With extensive geographic coverage, large datasets, and accurate location tracking, CDR are particularly effective for studying commuter patterns. This data enables insights into the movement behaviors of mobile phone users, offering valuable perspectives on how residents of smart cities navigate their environments, with implications for both economic and political analysis [35].

In contrast to fixed-location sensors, these sensors are dynamic, moving with their owner. From a technical perspective, GPS sensor data can be interpreted as a series of measurements, with each GPS co-ordinate linked to a precise timestamp:

$$\overline{\boldsymbol{p}} = \{p1 \to p2 \to ... \to pt \to ...\},\ \ t = 1, 2, ...\ T \tag{2}$$

Each data point, denoted as pt, includes latitude, longitude, and a timestamp. This data can also be interpreted as a sequence of time-stamped points, each containing a set of latitude and longitude co-ordinates. Due to this structure, the data from mobile sensors is often referred to as spatio-temporal data, where the temporal aspect corresponds to the timestamp, and the spatial aspect refers to the GPS co-ordinates.

Moving sensor data provides a deeper level of insight, allowing us to track the exact routes taken by vehicles and pedestrians, recognize various movement patterns, and establish links between different road segments. These sensors, which can move freely throughout the area, make it easier to identify the actual paths used by cyclists and pedestrians. Additionally, since the sensors are already integrated into vehicles or smartphones, they offer a much lower infrastructure cost compared to fixed-location sensors, as the focus is solely on data collection and processing, rather than installation or maintenance. This enables monitoring of roadways that were previously unobserved by stationary sensors.

Unlike stationary sensors, obtaining aggregated data from moving sensors requires several sensors traveling in the same direction. One significant challenge with spatio-temporal data is ensuring the completeness of the recorded trajectories. Without contributions from all vehicles or pedestrians in the flow, the data may not accurately reflect the full picture. To achieve reliability, a large portion of participants must provide data from their sensors. Additionally, GPS data from moving sensors carries inherent uncertainty [36]. Necessitating a map-matching process [37] to correct discrepancies between GPS readings and the road network during data preprocessing. In 2018, new GPS chips in smartphones [37, 38], provided up to 1 m accuracy by combining L5 and L1 signals. However, like fixed sensors, most of the relevant datasets are not publicly available.

The traffic classification technique utilising machine learning

The TC with ML follows a structured four-step process, fig. 2, consisting of feature selection, data collection, preprocessing, model development, and result analysis with visualization. In the initial stages of data collection and feature choice, samples of traffic flows are gathered and organized into a dataset for subsequent analysis. A detailed overview of the collected dataset is provided in section *Results and Discussion*. The following step involves preparing training and test datasets by eliminating irrelevant attributes and personally associating each specimen with its respective category. During preprocessing, the samples are standardized to ensure the features are properly scaled. The TC model is built using ML algorithms applied to the training dataset. Four commonly used supervised learning methods were compared: SVM, RF, KNN, and DT. Once the models are trained, they are assessed using the test dataset. The performance of each algorithm is evaluated using four metrics: accuracy, precision, recall, and F1-score. Additionally, k-fold cross-validation is utilized to provide a more robust evaluation of the classification performance.

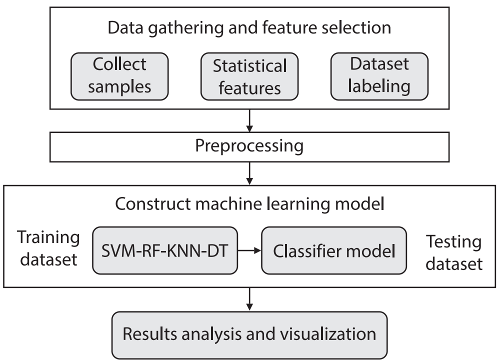

Figure 2. Procedures for constructing and assessing recommended ML algorithms

Assessment of the traffic classification

This section describes the approach used to evaluate the study. It starts by examining the traffic patterns in the dataset. Then, the performance metrics for assessing the model are introduced. Finally, the set-up for the experiment for testing the port-based technique and ML algorithms is presented.

Dataset

For TC, ML techniques combined with a port-based method were applied to the dataset provided by Moore and Zuev [39]. This dataset, sourced from a computer lab at Cambridge

University, includes diverse smart city traffic data with multiple origins, extensive traffic sampling, and various data types. It was collected over a full-duplex gigabit link, capturing activity from approximately 1000 users accessing a single website throughout the day. The dataset contains 248 features, including statistics on packet inter-arrival times, flow duration, and TCP port numbers, with around 377000 samples from TCP traffic. Table 1 outlines the classes and their associated applications. To refine the dataset, we eliminate non-statistical features and randomly sample from each class. The *Games* class is excluded due to insufficient data, resulting in a final set of 11 classes, as detailed in tab. 2.

Table 2. Applications for TC

Application	Cataloguing
ftp	Bulk
Postgres salnet oracle ingres	Database
ssh, klogin, rlogin, telnet	Interactive
imap, pop2/3, smtp	Mail
X11, dns, ident, ldap, ntp	Services
WWW	WWW
KaZaA, BitTorrent, GnuTella	P2P
Internet worm and virus attacks	Attack
Half-life	Games
Windows media player, real	Multimedia

Results and discussion

The outcomes of assessing both port-based TC and machine learning techniques are shown in this section. We evaluate the algorithms' performance by looking at things like training and execution times, as well as how class size affects accuracy. We then use the port-based categorization method to compare the algorithms' performance.

The machine learning algorithm evaluation

We implemented and evaluated four ML algorithms: SVM, RF, KNN, and DT, optimizing their model parameters to enhance accuracy. For the SVM application, a linear kernel was used. The SVM, a supervised learning method, maps data into a higher-dimensional space where support vectors define a hyperplane that best separates the classes and maximizes the margin between them [40]. The SVM achieved an impressive average accuracy of 97.14%, as shown in fig. 3. outlines the precision, recall, and F1-score for each traffic class with the SVM model. The results highlight that the Collaborative and Multimedia classes performed the weakest. The Interactive class recorded a precision of 0.85, recall of 0.75, and F1-score of 0.81, while the Multimedia class had a precision of 0.63, recall of 0.84, and F1-score of 0.71. These outcomes indicate that the smaller sample sizes of these categories, coupled with the variability in the data traffic, affected the classification accuracy.

In the RF approach, decision trees are trained on data streams, with their predictions combined to determine the final class. To maximize both accuracy and efficiency, the model was configured with 50 trees, using entropy for optimal performance. The RF method achieved an outstanding average classification accuracy of 98.08%, fig. 4, showcasing its exceptional effectiveness. Remarkably, the Interactive and Multimedia categories delivered the highest

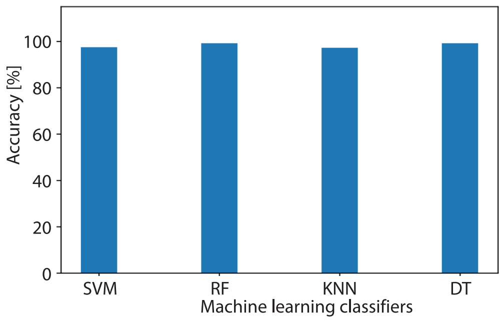

Figure 3. Average accuracy of ML algorithms

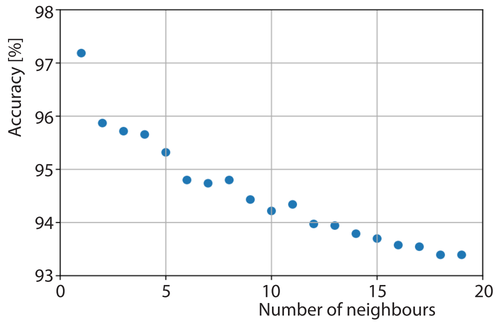

Figure 4. Accuracy based on the number of neighbours in the KNN method

results. The Interactive category achieved perfect precision (1.00), a recall of 0.92, and an F1-score of 0.95. For Multimedia, precision was 0.85, recall was 1.00, and the F1-score reached 0.93. These results underscore how the RF method, leveraging an ensemble of decision trees, significantly boosts classification accuracy across diverse categories.

In the *k*-nearest neighbours (KNN) algorithm, the Manhattan distance metric is used to maximize both accuracy and performance. The KNN works by classifying data based on the proximity of the *k*-nearest data points. The final classification is determined through a majority vote among these neighbors, with the votes weighted according to their distance. The effectiveness of KNN largely depends on the chosen distance metric and the value of *k*. As shown in fig. 5, the ideal *k* value is found by evaluating accuracy for *k*-values ranging from 1-20. The results suggest that accuracy decreases as the number of neighbors increases, with the most accurate classification achieved when $k = 4$. The KNN algorithm attained an average accuracy of 97.17%, which was marginally lower than the other algorithms, fig. 4. Below, we present the precision, recall, and F1-score for each class as evaluated by the KNN method. The DT algorithm constructs a series of nodes and conditions that ultimately lead to leaves, where classifications are predicted. In our application, we utilize the entropy metric to optimize performance. The DT method achieved the highest average accuracy of 99.18% fig. 4, outperforming all other classification algorithms tested.

Figure 4 showcases the results of each ML algorithm based on F1-score, precision, and recall. The decision tree (DT) algorithm led with the highest F1-score of 99.37%, surpassing KNN at 97.16%, random forest (RF) at 98.41%, and SVM at 98.06%. In terms of accuracy, DT also took the top spot with 99.27%, followed by RF at 99.16%, SVM at 98.08%, and KNN at 97.17%. Regarding recall, DT maintained its dominance with a score of 99.27%, while RF scored 98.91%, SVM recorded 98.09%, and KNN had 96.77%. Overall, the DT algorithm excelled in all performance metrics, outshining the other models in every category.

Conclusions

As interest in smart cities continues to grow, the adoption of innovative solutions aims to enhance the comfort, effectiveness, and efficiency of everyday life. These urban environments, with their array of applications, diverse data streams, and varying QoS demands, present significant challenges in traffic management. Efficient TC is vital to address these challenges and ensure smooth operation. Traditional methods, such as port-based techniques and DPI, struggle with dynamic port numbers and encrypted traffic. In contrast, ML algorithms are adept at handling these complexities while supporting QoS control.

In this research, we evaluated four supervised ML algorithms for TC: SVM, RF, KNN, and DT. We also explored the effectiveness of a port-based TC method. The findings reveal that incorporating statistical features greatly improves the accuracy of ML models for TC. Of the algorithms tested, the DT achieved the highest accuracy at 99.18%, whereas the KNN performed the least, with an average accuracy of 97.16%.

Our analysis also revealed that the port-based classification method, which depends on fixed port numbers, is less effective than ML algorithms. In contrast to port-based methods, ML models consider a broader range of factors, leading to better classification outcomes. Moving forward, we plan to incorporate ML-based TC into smart city systems that manage critical data and address routing challenges. Future research will also explore other ML techniques, such as eXtreme Gradient Boosting (XGBoost), to enhance TC in smart city contexts.

Acknowledgment

This research work was funded by Institutional Fund Projects under Grant No. 0167-611-1443. The authors gratefully acknowledge technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

References

[1] Atzori, L., *et al*., The Internet of Things: A Survey, *Computer Networks*, *54* (2010), 15, pp. 2787-2805
[2] Jiang, D., The Construction of Smart City Information System Based on the Internet of Things and Cloud Computing, *Computer Communications*, *150* (2020), Jan., pp. 158-166
[3] Gyrard, A., *et al*., Building IoT-based Applications for Smart Cities: How Can Onlogy Catalogs Help, *IEEE Internet of Things Journal*, *5* (2018), 5, pp. 3978-3990
[4] Kirimtat, A., *et al*., Future Trends and Current State of Smart City Concepts: A Survey, *IEEE Access*, *8* (2020), May, pp. 86448-86467
[5] Roblek, V., Meško, M., Smart City Knowledge Management: Holistic Review and the Analysis of the Urban Knowledge Management, *Proceedings,* 21st Annual International Conference on Digital Government Research, Seoul, Republic of Korea, 2020
[6] Tcholtchev, N., Schieferdecker, I., Sustainable and Reliable Information and Communication Technology for Resilient Smart Cities, *Smart Cities*, *4* (2021), 1, pp. 156-176
[7] Al-Maqashi, S., *et al*., The Impact of ICTS in the Development of Smart City: Opportunities and Challenges, in: *Smart Cities – Fund. and Perspe.*, In-Tech, Rijeka, Croatia, 2024
[8] Hui, C. X., *et al.*, Greening Smart Cities: An Investigation of the Integration of Urban Natural Resources and Smart City Technologies for Promoting Environmental Sustainability, *Sustainable Cities and Society*, *99* (2023), 104985
[9] Alharbi, F., Fei, Z., Improving the Quality of Service for Critical Flows in Smart Grid Using Software-Defined Networking, *Proceedings*, International Conference on Smart Grid Communications (SmartGridComm), IEEE, Sydney, Australia, 2016
[10] Maha Vishnu, V., M. *et al*., Intelligent Traffic Video Surveillance and Accident Detection System with Dynamic Traffic Signal Control, *Cluster Computing*, *21* (2018), June, pp. 135-147
[11] Binsahaq, A., *et al*., A Survey on Autonomic Provisioning and Management of QoS in SDN Networks, *IEEE Access*, *7* (2019), May, pp. 73384-73435
[12] Braden, R., *et al*., Integrated Services in the Internet Architecture: An Overview, RFC1633, 1994
[13] Roddav, N., *et al.*, On the Usage of DSCP and ECN Codepoints in Internet Backbone Traffic Traces for IPv4 and IPv6, *Proceedings*, International Symposium on Networks, Computers and Communications (ISNCC), IEEE, Istanbul, Turkey, 2019
[14] Tahaei, H., *et al*., The Rise of Traffic Classification in IoT networks: A Survey, *Journal of Network and Computer Applications*, *154* (2020), 102538
[15] Nguyen, T. T., Armitage, G., A Survey of Techniques for Internet Traffic Classification Using Machine Learning, *IEEE Communications Surveys and Tutorials*, *10* (2008), 4, pp. 56-76
[16] Salman, O., *et al*., A Review on Machine Learning-Based Approaches for Internet Traffic Classification, *Annals of Telecommunications*, *75* (2020), 11, pp. 673-710

[17] Soil, A. C., Standard test Methods for Liquid Limit, Plastic Limit, and Plasticity Index of Soils, ASTM International, D4318-17e1, 2010
[18] Prasad, P., Enhancing Security in Software-Defined Networking (SDN) based IP Multicast Systems: Challenges and Opportunities, M. Sc. thesis, Univ. of Turku, Turku, Finland, 2024
[19] Tongaonkar, A., *et al*., Challenges in Network Application Identification, *Proceedings,* LEET, San Jose, Cal., USA, 2012
[20] Jurcut, A., *et al.*, Security Considerations for Internet of Things: A Survey, *SN Computer Science*, *1* (2020), June, pp. 1-19
[21] Mohamed, N., *et al*., A Cost-Effective Design for Combining Sensing Robots and Fixed Sensors for Fault-Tolerant Linear Wireless Sensor Networks, *International Journal of Distributed Sensor Networks*, *10* (2014), 3, 659356
[22] Alqudah, N., Yaseen, Q., Machine Learning for Traffic Analysis: A Review, *Procedia Computer Science*, *170* (2020), Jan., pp. 911-916
[23] Xie, J., *et al.*, A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges, *IEEE Communications Surveys and Tutorials*, *21* (2018), 1, pp. 393-430
[24] Yao, H., *et al*., Capsule Network Assisted IoT Traffic Classification Mechanism for Smart Cities, *IEEE Internet of Things Journal*, *6* (2019), 5, pp. 7515-7525
[25] Miao, Y., *et al*., Comprehensive Analysis of Network Traffic Data, Concurrency and Computation, *Practice and Experience*, *30* (2018), 5, e4181
[26] Mohamed, K., *et al*., Clustering Smart Card Data for Urban Mobility Analysis, IEEE Transactions on Intelligent Transportation Systems, *18* (2016), 3, pp. 712-728
[27] Chen, R., Zhou, J., Fare Adjustment's Impacts on Travel Patterns and Farebox Revenue: An Empirical Study Based on Longitudinal Smartcard Data, Transportation Research – Part A: *Policy and Practice*, *164* (2022), Oct., pp. 111-133
[28] Chen, E., *et al*., Unraveling Latent Transfer Patterns Between Metro and Bus from Large-Scale Smart Card Data, *IEEE Transactions on Intelligent Transportation Systems*, *23* (2020), 4, pp. 3351-3365
[29] Ma, X., *et al*., Evaluation and Determinants of Metro Users' Regularity: Insights from Transit One-Card Data, *Journal of Transport Geography*, *118* (2024), 103933
[30] Asad, S., *et al.*, Blockchain-Based Decentralized Federated Learning for Privacy-Preserving Traffic Flow Prediction: A Case Study with PeMS-8 Data, *Journal of Computing and Biomedical Informatics*, 7 (2024.), 01, pp. 204-214
[31] Kim, C., *et al.*, Decision Tree Analysis for Developing Weigh-in-Motion Spectra in the Californian Pavement Management System (PaveM), *Transportation Research Record*, *2678* (2024), 3, pp. 546-559
[32] Sun, X., *et al.*, Short-Term Traffic Flow Prediction Model Based on a Shared Weight Gate Recurrent Unit Neural Network. Physica A, *Statistical Mechanics and its Applications*, *618* (2023), 128650
[33] Sarri, P., *et al.*, Incorporating Land Use and Transport Interaction Models to Evaluate Active Mobility Measures and Interventions in Urban Areas: A Case Study in Southampton, UK, *Sustainable Cities and Society*, *105* (2024), 105330
[34] Yao, H., Guizani, M., Intelligent IoT Network Awareness, in: *Intelligent Internet of Things Networks*, Springer, Berlin, Germany, 2023, pp. 37-109
[35] Rodrigues, C., *et al.* Using CDR Data to Understand Post-pandemic Mobility Patterns, in: *EPIA Conference on Artificial Intelligence*, Springer, Berlin, Germany, 2023
[36] Dong, H., *et al.* GPSense: Passive Sensing with Pervasive GPS Signals, *Proceedings*, 30th Annual International Conference on Mobile Computing and Networking, Washington D.C., USA, 2024
[37] Giangualano, M., Implementation and Evaluation of a Map Matching Algorithm, M. Sc. thesis, Politecnico di Milano, Milano, Italy, 2023
[38] Li, Z., *et al.*, Real-Time GNSS Precise Point Positioning with Smartphones for Vehicle Navigation, *Satellite Navigation*, *3* (2022), 1, 19
[39] Moore, A. W., Zuev, D., Internet Traffic Classification Using Bayesian Analysis Techniques, *Proceedings*, ACM SIGMETRICS International Conference on Measurement and Modelling of Computer Systems, New York, USA, 2005
[40] Awad, M., *et al*., Support Vector Machines for Classification, in: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, Springer, New York, USA, 2015, pp. 39-66

Paper submitted: September 20, 2024
Paper revised: November 13, 2024
Paper accepted: December 17, 2024