Evaluating Traffic Classification Techniques for Smart City Networks

Dr. Anupriya[1]
[1]*Pt. C.L.S. Govt. College Karnal, Haryana, India*

***Abstract*—Smart city networks support a wide range of applications, each imposing specific Quality of Service (QoS) requirements, making network management particularly challenging. Despite the need, large-scale deployments of QoS-supporting solutions remain limited. Traffic classification plays a key role in managing network aspects, including QoS assurance. However, traditional traffic classification methods, such as port-based approaches, are increasingly inefficient due to their inability to handle dynamic port allocations and encrypted traffic. Recently, machine learning (ML) has emerged as a promising alternative for traffic classification, offering intelligence that enhances network management. In this study, we apply four supervised ML algorithms—Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), and Decision Tree (DT)—to predict and classify network traffic.**

***Index Terms*—machine learning; traffic classification; smart city; quality of service; Internet of things; supervised learning.**

I. INTRODUCTION

The Internet of Things (IoT) represents a technological revolution that has gained increasing significance over time. Its profound impact on numerous aspects of daily life underscores the strength and transformative potential of IoT technologies [1]. Furthermore, IoT plays a crucial role in driving economic growth, prompting significant investments from technology companies and research centers in the development and advancement of IoT solutions [2].

IoT applications interconnect a wide array of objects—including actuators, sensors, smart devices, and home appliances—with the Internet to enable seamless data transmission and processing [3,4]. As a foundational technology, IoT is pivotal in enabling smart city services. A smart city leverages the latest advancements in communication technologies to enhance the quality of services provided to residents, ultimately aiming to improve their quality of life [5,6].

Smart cities encompass a variety of innovative solutions [7], such as smart buildings, smart education, smart healthcare, smart transportation, smart grids, smart environments, and smart homes (Figure 1). These solutions collectively deliver numerous benefits to citizens by facilitating intelligent and efficient services. However, the diverse and specific requirements of these applications introduce significant challenges and complexity to network management.

Different applications within smart city environments involve numerous interconnected objects, such as sensors and actuators, which collectively generate massive volumes of network traffic. These applications also present diverse Quality of Service (QoS) requirements, including bandwidth, packet loss, delay, jitter (variation in delay), and best-effort service options [8]. For example, video surveillance systems play a critical role in enhancing traffic management by supporting rapid response to emergencies and accidents, as well as aiding in the identification of congested roadways [9]. This application has stringent requirements, including high bandwidth and low jitter, for the network traffic to reach its destination.

Real-time applications, such as online gaming and telephony, demand highly reliable interactions and are extremely sensitive to network delays. The rapid expansion of smart city applications—each generating different types of traffic—intensifies the challenges associated with maintaining consistent QoS support. The diversity in traffic patterns and requirements makes efficient network management increasingly complex. Therefore, these challenges must be properly addressed to ensure the reliability and performance of smart city services. Traffic classification based on machine learning (ML) algorithms has garnered significant research interest due to its potential for high accuracy and efficiency. In the context of supervised learning, the application of ML to traffic classification typically involves several key steps. First, relevant traffic features that characterize flow attributes—such as packet length, inter-arrival time, and flow duration—are extracted. These features serve as the input for the learning process. Next, a machine learning model is constructed and trained using labeled datasets, enabling the model to learn the patterns and behaviors associated with different types of network traffic. Third, the classifier is trained to associate specific features with known traffic classes. Finally, the model is applied to classify data traffic, predicting the classes in traffic flow.

II. RELATED WORK

This section presents a review of related studies from the literature, highlighting their different approaches to traffic classification and Quality of Service (QoS) support. Various studies have conducted comparative analyses of traffic classification using machine learning techniques across different datasets, such as backbone networks, while others have investigated QoS support for smart city applications at various network layers, including the data link and transport layers.

For instance, Aureli et al. [11] proposed a dynamic classification method known as learning-based Differentiated Services. Their approach focused on discovering traffic characteristics and dynamically assigning service classes to IP packets. Similarly, Zhongsheng et al. [12] applied Support Vector Machine (SVM) algorithms for traffic classification in campus backbone networks. Their methodology involved traffic data collection and feature generation before applying SVM for classification.

Continuing, Perera et al. [13] conducted a comparative study of six supervised learning algorithms for traffic classification: Naive Bayes, Bayesian networks, Random Forest (RF), Decision Tree (DT), Naive Bayes Tree, and Multilayer Perceptron. Rahman et al. [14] proposed a cloud robotics framework tailored for smart city applications. In their framework, robotic agents leverage cloud services through task offloading to enhance QoS and system performance. They formulated an optimization problem based on a directed acyclic graph and employed a genetic algorithm to determine the optimal offloading decisions. In summary, machine learning algorithms have been extensively studied for evaluating the performance of supervised classifiers, and deep learning techniques have also been explored for QoS improvement in smart city networks. However, unlike existing studies, we present a comprehensive analysis of supervised classification algorithms—specifically SVM, RF, KNN, and DT—for network traffic classification based on statistical features.

III. TRAFFIC CLASSIFICATION METHODS

We adopt a four-step methodology for traffic classification using machine learning, as illustrated in Figure 1. These steps include: (1) Data Gathering and Feature Selection, (2) Preprocessing, (3) Construction of Machine Learning Models, and (4) Result Analysis and Visualization.

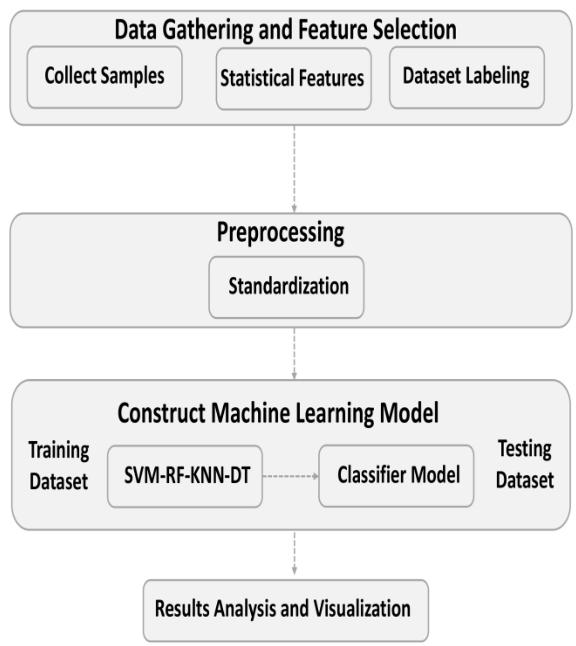

Figure 1. Steps to build and evaluate proposed machine learning algorithms.

3.1. Dataset

In this study, we utilize the dataset constructed by Moore and Zuev [15] to apply machine learning algorithms and the port-based method for traffic classification. This dataset is specifically designed to capture the characteristics of smart city data traffic, including the diversity of data sources, the high volume of traffic samples, and the variety of data types. The dataset consists of 10 different sub-datasets, each monitored at different times of the day from a single Internet website. This website hosted approximately 1000 users connected to the Internet through a full-duplex gigabit link. The dataset includes a total of 248 features, such as flow duration, TCP port number, and the mean and variance of packet inter-arrival times. These features are critical for accurately classifying different types of network traffic. The dataset contains approximately 377,000 samples, which are based on TCP traffic flows. Each traffic flow is labeled according to its associated application. The classes and their corresponding applications are listed in Table 1.

Table 1. Applications for traffic classification

Classification	Application
Bulk	ftp
Database	postgres, sqlnet oracle, ingres
Interactive	ssh, klogin, rlogin, telnet
Mail	imap, pop2/3, smtp
Services	X11, dns, ident, ldap, ntp
WWW	www
P2P	KaZaA, BitTorrent, GnuTella
Attack	Internet worm and virus attacks
Games	Half-Life
Multimedia	Windows Media Player, Real

To build a suitable dataset for our experiments, we collect random samples from each class while ensuring a balanced representation. We also remove certain features that represent non-statistical information, as they are irrelevant for the classification task. Additionally, the **Games** class, which contains a limited number of samples, is excluded from the dataset. As a result, the final dataset consists of 11 classes, which are listed in Table 2.

Table 2. Characteristics of the traffic classification dataset used in this study.

Traffic Class	Samples
Attack	1500
Database	1068
FTP–Control	993
FTP–Data	2019
FTP–Passive	1297
Interactive	110
Mail	2081
Multimedia	43
P2P	542
Services	1921
WWW	1499

3.2. Experimental Setup

All experiments for the machine learning algorithms and the port-based method were conducted using Python. Specifically, we utilized the Scikit-learn library to implement, train, and evaluate the machine learning models. The dataset was divided into training and testing subsets, with 75% of the samples allocated for training and 25% for testing. We also performed data preprocessing, including feature scaling through standardization, to ensure that all feature values exhibited similar properties and to prevent any bias toward specific features. The objective of the standardization technique is to rescale the feature values so that they have a mean of 0 and a standard deviation of 1. The standardized value (z-score) of a sample is calculated using Equation (1):

$$z = \frac{x-\mu}{\sigma} \qquad (1)$$

where x is the sample value to be standardized, μ is the mean of the training samples, and σ represents the standard deviation of training the samples.

In addition, we tuned various parameters specific to each machine learning algorithm to enhance classification accuracy. To evaluate performance and mitigate overfitting, we employed tenfold cross-validation, a method known for providing reliable and robust results when applied to machine learning models. For the port-based method, we compiled a list of popular and well-known port numbers corresponding to the applications represented in the dataset to facilitate its application and evaluation.

IV. RESULT & DISCUSSION

This section presents the evaluation results obtained from the machine learning algorithms and the port-based method for traffic classification. We evaluate the machine learning algorithms based on performance metrics, the impact of the number of classes on classification accuracy, and their training and execution times.

4.1 Evaluation of Machine Learning Algorithms

We implemented and compared four machine learning algorithms: Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), and Decision Tree (DT). Specific model parameters were tuned for each algorithm to enhance accuracy. For the SVM implementation, we employed a linear kernel. SVM performs supervised learning by mapping data samples into a high-dimensional space, where it determines a hyperplane that optimally separates the samples and maximizes the margin between classes using support vectors [27]. The SVM achieved an average accuracy of 97.41%, demonstrating its strong performance (Figure 2). The precision, recall, and F1-score for each traffic class using SVM are summarized in Table 3. The results indicate that the *Interactive* and *Multimedia* classes exhibit the lowest performance among the classification labels. Specifically, the *Interactive* class achieved a precision of 0.82, recall of 0.72, and F1-score of 0.77, while the *Multimedia* class achieved a precision of 0.62, recall of 0.83, and F1-score of 0.71. These results suggest that the number of traffic samples significantly impacts classification performance, as these two classes contained fewer samples compared to the others.

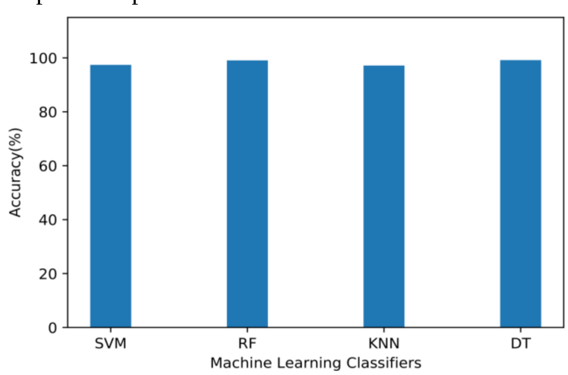

Figure 2. Average accuracy of machine learning algorithms.

The Random Forest (RF) algorithm constructs multiple Decision Trees (DTs) trained on subsets of the data flows and then aggregates their results to predict the final class. For the K-Nearest Neighbors (KNN) algorithm, we used the Manhattan distance metric, which provided high accuracy and performance. KNN classifies data samples by identifying the k nearest neighbors and predicting the class based on majority voting among them. The precision, recall, and F1-score for each class obtained using KNN are provided in Table 3.

Table 3. Precision, recall, and F1-score per class for SVM classification.

Traffic Class	Precision	Recall	F1-Score
Attack	0.95	0.95	0.95
Database	1.00	1.00	1.00
FTP–Control	0.98	0.99	0.99
FTP–Data	0.99	1.00	1.00
FTP–Passive	0.98	1.00	0.99
Interactive	0.82	0.72	0.77
Mail	0.99	0.99	0.99
Multimedia	0.62	0.83	0.71
P2P	0.92	0.96	0.94
Services	1.00	0.99	0.99
WWW	0.98	0.95	0.96

Table 4. Precision, recall, and F1-score per class for RF classification.

Traffic Class	Precision	Recall	F1-Score
Attack	0.98	0.97	0.98
Database	1.00	1.00	1.00
FTP–Control	1.00	1.00	1.00
FTP–Data	0.99	1.00	1.00
FTP–Passive	0.98	1.00	0.99
Interactive	1.00	0.92	0.96
Mail	1.00	1.00	1.00
Multimedia	0.86	1.00	0.92
P2P	0.97	0.97	0.97
Services	1.00	0.99	1.00
WWW	0.99	0.97	0.98

The F1-score, precision, and recall results for each machine learning algorithm are presented in Figure 3. The results indicate that, for the F1-score metric, the Decision Tree (DT) algorithm outperformed the others, achieving a score of 99.27%. The K-Nearest Neighbors (KNN) algorithm reached 97.15%, the Random Forest (RF) achieved 99.14%, and the Support Vector Machine (SVM) reached 98.07%. In terms of precision, DT again demonstrated the highest performance with a score of 99.27%, followed by KNN at 97.16%, RF at 99.15%, and SVM at 98.08%. Similarly, for recall, DT achieved the best result at 99.27%, with KNN at 97.16%, RF at 99.14%, and SVM at 98.07%. These findings demonstrate that the DT algorithm outperforms the other machine learning algorithms across all evaluated performance metrics.

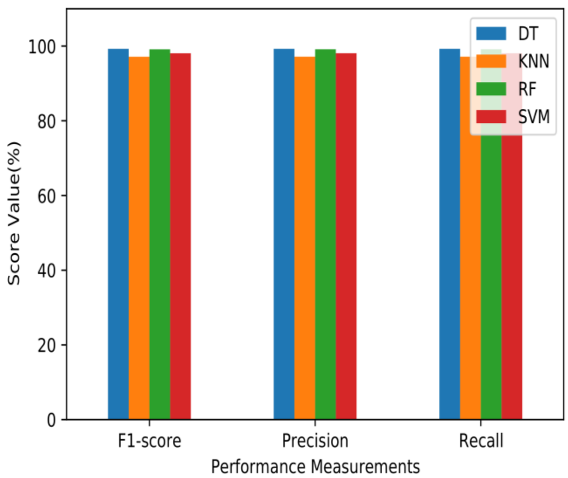

Figure 3. F1-score, precision, and recall performance measurements of machine learning algorithms.

V. CONCLUSION & FUTURE WORK

Smart cities are becoming increasingly prominent, driven by the deployment of smart solutions designed to make daily life more comfortable, productive, and efficient. These environments involve diverse applications, a wide range of data types, and varying Quality of Service (QoS) requirements, presenting significant challenges for traffic management. Traffic classification plays a crucial role in addressing these challenges, particularly for supporting QoS. However, traditional traffic classification methods, such as port-based techniques and deep packet inspection, struggle with encrypted traffic and dynamic port numbers. In contrast, machine learning algorithms offer promising solutions for managing QoS and addressing network complexity.

In this work, we evaluated four supervised machine learning algorithms—Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), and Decision Tree (DT)—for traffic classification. We also assessed the performance of a port-based method for comparative purposes. The evaluation results demonstrated that utilizing statistical features significantly enhances traffic classification when using machine learning approaches. Among the evaluated algorithms, the DT algorithm achieved the highest average accuracy (99.18%), whereas the KNN algorithm recorded the lowest (97.16%). In contrast, machine learning algorithms leverage multiple features, including but not limited to port numbers, leading to superior classification performance.

For future work, we plan to explore routing challenges by integrating machine learning-based traffic classification into various smart city applications that manage critical data flows. Furthermore, we aim to expand our study by evaluating additional machine learning models, such as eXtreme Gradient Boosting (XGBoost), to further improve classification accuracy and system robustness.

REFERENCES

[1] Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. Comput. Netw. 2010, 54, 2787–2805. [CrossRef]

[2] Imran; Ghaffar, Z.; Alshahrani, A.; Fayaz, M.; Alghamdi, A.M.; Gwak, J. A Topical Review on Machine Learning, Software Defined Networking, Internet of Things Applications: Research Limitations and Challenges. Electronics 2021, 10, 880. [CrossRef]

[3] Gyrard, A.; Zimmermann, A.; Sheth, A. Building IoT-Based Applications for Smart Cities: How Can Ontology Catalogs Help IEEE Internet Things J. 2018, 5, 3978–3990. [CrossRef]

[4] Kirimtat, A.; Krejcar, O.; Kertesz, A.; Tasgetiren, M.F. Future Trends and Current State of Smart City Concepts: A Survey. IEEE Access 2020, 8, 86448–86467. [CrossRef]

[5] Roblek, V.; Meško, M. Smart City Knowledge Management: Holistic Review and the Analysis of the Urban Knowledge Management. In Proceedings of the 21st Annual International Conference on Digital Government Research, Seoul, Korea,15–19 June 2020; pp. 52–60.
[6] Tcholtchev, N.; Schieferdecker, I. Sustainable and Reliable Information and Communication Technology for Resilient Smart Cities.Smart Cities 2021, 4, 156–176. [CrossRef]
[7] Mohanty, S.P.; Choppali, U.; Kougianos, E. Everything you wanted to know about smart cities: The Internet of things is the backbone. IEEE Consum. Electron. Mag. 2016, 5, 60–70. [CrossRef]
[8] Alharbi, F.; Fei, Z. Improving the quality of service for critical flows in Smart Grid using software-defined. In Proceedings of the 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm), Sydney, Australia, 6–9 November 2016; pp. 237–242.
[9] Naphade, M.; Banavar, G.; Harrison, C.; Paraszczak, J.; Morris, R. Smarter Cities and Their Innovation Challenges. Computer 2011, 44, 32–39. [CrossRef]
[10] Nguyen, T.T.T.; Armitage, G. A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. 2008, 10, 56–76. [CrossRef]
[11] Aureli, D.; Cianfrani, A.; Diamanti, A.; Sanchez Vilchez, J.M.; Secci, S. Going Beyond DiffServ in IP Traffic Classification. In Proceedings of the NOMS 2020—2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–6.
[12] Zhongsheng, W.; Jianguo, W.; Sen, Y.; Jiaqiong, G. Traffic identification and traffic analysis based on support vector machine. Concurr. Comput. Pract. Exp. 2020, 32, e5292. [CrossRef]
[13] Perera, P.; Tian, Y.C.; Fidge, C.; Kelly, W. A Comparison of Supervised Machine Learning Algorithms for Classification of Communications Network Traffic. In Neural Information Processing; Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S.M., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 445–454.
[14] Rahman, A.; Jin, J.; Cricenti, A.; Rahman, A.; Yuan, D. A Cloud Robotics Framework of Optimal Task Offloading for Smart City Applications. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–7.
[15] Moore, A.W.; Zuev, D. Internet Traffic Classification Using Bayesian Analysis Techniques. SIGMETRICS Perform. Eval. Rev. 2005,33, 50–60. [CrossRef]