Simulators for system dataset generation in the Edge-to-Cloud Continuum

Nawaz Ali
Department of Informatics, Modeling, Electronics and Systems
University of Calabria
Rende (CS), Italy
nawaz.ali@dimes.unical.it

Gianluca Aloi
Department of Informatics, Modeling, Electronics and Systems
University of Calabria
Rende (CS), Italy
gianluca.aloi@unical.it

Pasquale Pace
Department of Informatics, Modeling, Electronics and Systems
University of Calabria
Rende (CS), Italy
ppace@dimes.unical.it

Michele Gianfelice
Department of Mathematics and Computer Science
University of Calabria
Rende (CS), Italy
gianfelice@mat.unical.it

Francesco Pupo
Department of Informatics, Modeling, Electronics and Systems
University of Calabria
Rende (CS), Italy
francesco.pupo@unical.it

Raffaele Gravina
Department of Informatics, Modeling, Electronics and Systems
University of Calabria
Rende (CS), Italy
r.gravina@dimes.unical.it

Giancarlo Fortino
Department of Informatics, Modeling, Electronics and Systems
University of Calabria
Rende (CS), Italy
g.fortino@unical.it

***Abstract*— In the era of the Edge-to-Cloud Continuum paradigm, effectively managing heterogeneous and distributed resources poses significant challenges. Autonomic system operation, supported by Artificial Intelligence (AI) driven resource management and application deployment mechanisms, offers a promising solution. Machine Learning (ML) models are pivotal for this purpose, necessitating large amounts of high-quality data for training, validation, and evaluation. Simulators play a crucial role by generating vast datasets containing diverse data types, facilitating training, testing, and analyzing ML and AI techniques for autonomic system optimization. This paper aims to review existing simulators and identify a candidate simulator suitable for generating datasets within the Edge-to-Cloud Continuum, supporting the development of efficient ML models.**

Keywords—Artificial Intelligence, Machine Learning, Simulators, Dataset generation, Autonomic System Operation, Edge-to-Cloud Continuum, System Optimization

I. INTRODUCTION

The proliferation of diverse computing devices, from smartphones to Internet of Things (IoT) sensors, and the exponential growth of data from various applications, is leading to an ever-growing demand of ubiquitous connectivity, seamless integration, dynamic adaptation to changing conditions, context awareness, and personalized user experiences. Consequently, there's a transition in data processing from centralized to decentralized and multi-tier computing infrastructures and services. This contributed to the birth of the Edge-to-Cloud Continuum paradigm [1] that encompasses a seamless integration of computing resources and services spanning from terminal devices to cloud infrastructure. This paradigm provides a holistic approach to distributed computing, integrating devices, edge nodes and cloud infrastructure into a unified ecosystem to meet the demands of emerging applications and services in various domains, such as IoT, smart cities, healthcare, and Industry 5.0.

In such a context, the mission of effectively managing heterogeneous and distributed resources and services is becoming ever more challenging and, in such a complex context, thinking about human-in-the-loop management is quite unrealistic. Autonomic system operation in the computing continuum, supported by AI-driven resource management and services deployment/orchestration mechanisms, could be a valuable solution.

Machine Learning (ML) models play a key role in achieving such an autonomic system operation scenario, however, for training, validating, and evaluating ML models, a large amount of high-quality (i.e. relevant, well-balanced, up-to-date, and bias-free) data is needed, whose retrieval comes with various challenges and issues that can significantly impact the final performance. In this context, simulators can play a highly essential role, particularly in edge-to-cloud systems. This is due to the fact that simulators generate enormous datasets that contain a wide variety of data types in situations where it would be difficult to collect data in the real world at the time. The simulation results obtained from the simulator encompass a diverse array of events and scenarios, such as network traffic patterns within cloud systems and data originating from external devices. Additionally, it is essential to provide datasets that accurately represent the complex features of actual data encountered in these systems.

ML and other AI algorithms rely on simulated datasets as part of their training set to ensure accurate generalization, exact prediction, and successful learning. It is critical for edge-to-cloud systems to have high reliability as they are subject to quick testing. Data simulation makes it possible for models to overcome constraints that are imposed by the data from the real world, as a result of this, models may now be tested in a wider variety of conditions, including circumstances that are uncommon but potentially significant including controlled environment of edge to cloud system.

Simulators have the capability to generate a wide variety of data, including telemetry for autonomous vehicles, network traffic, and reports from sensors that are connected to the IoT. These are just some of the many forms of data that simulators may generate. The dataset generated through simulation results can be utilized to improve the ML models in controlled environment, allowing assessment of varied results by means of different parameters. This is essential for improving performance from the edge to the cloud, as models must adapt to different processing resources, latency, and network conditions.

To address the above issues, the purpose of this paper is to achieve a dual objective. The first is to carry out a short review of simulators currently available in the research landscape of Edge-to-Cloud Continuum. The second is to identify a candidate simulator that can potentially be actually useful for generating datasets in the context of the Edge-to-Cloud Continuum and with the aim of being able to support the development of efficient ML models.

The rest of the paper is organized as follows. Section II provides the state of the art of existing simulators. In Section III, we outline the main characteristics of some selected simulators, while the most performing and suitable edge-cloud simulator selection is presented in Section IV. In Section V we introduce some extensions needed to improve the selected simulator performances and finally, in Section VII, we conclude the paper.

II. SIMULATORS STATE-OF-THE-ART ANALYSIS

In the recent but extensive literature of simulators, if we consider the specific domain of the autonomic operation in the Edge-to-Cloud Continuum, several tools are available to replicate the behavior and performance of IoT and of edge-to-cloud systems. Some simulators are better suited for replicating and analyzing issues more closely related to networking (e.g. throughput improvement, resource management, load balancing, congestion prevention, etc.), while others are more suitable for investigating the behavior of platforms as distributed computing systems (e.g. in terms of process virtualization, task allocation, distributed control, scalability, etc.). However, if we needed specific datasets to facilitate the design and training of ML tools to be applied in the Edge-to-Cloud Continuum, there are no simulators specifically designed for this purpose. Therefore, we want to investigate if widely used simulators may be successfully exploited to support also dataset generation.

A first category of simulators, typically used for network simulations, which includes NS-3 and OMNeT++, has been widely recognized in the academia and therefore used and extended.

OMNeT++ is an event driven network simulator [2] that can either incorporate ML models into its simulations or train them with its generated data and system traces. From the version 6.0, OMNeT++ can produce data in *.vec* and *.sca* file formats and be integrated with TensorFlow API based on C++, with the python-based framework OpenAI Gym, or with software as Pandas or database as SQLite3. For example, in [3][4] the network data generated by using OMNeT++ have been employed to categorize user activities and network traffic for testing intrusion detection systems or for estimating QoE, and to predict metrics in the network context such bandwidth utilization and network latency.

Based on OMNeT++, Simu5G is an end-to-end simulator specifically tailored for 5G networks [5]: for this purpose, some key features including user mobility and handover have been added with different time/space granularity. In such a way, Simu5G is able to generate datasets in CSV or XML formats and to work as a “synthetic traffic generator" [6] for training and testing ML models for real-world applications such as voice over internet protocol (VoIP).

MimicNet is another simulator built upon OMNeT++ with the addition of custom C++ modules that can integrate ML models and allow estimating the performance of massive data center networks or collecting system traces [7] for third-parties software of data analysis like Pandas.

NS-3 can be considered instead the competitor of OMNeT++’s family simulators. Written in C++ and Python and distributed as open-source, NS-3 is another premier discrete-event network simulator that supports a wide range of wireless technologies, such as Wi-Fi, WiMAX, LTE, 5G, and Bluetooth [8]. It also provides modules for routing, mobility, energy, and security as well as it can output simulation data into specific formats that are compliant with ML’s frameworks and workflows per network analysis purposes such as traffic classification and predictions, anomalies detection, packets management optimization [9]. However, the greater performance, scalability and integrability of OMNeT++ made it as the reference for the network simulators.

Differently from network simulators, another category is represented by platform simulators that adopt a full-fledged approach to the simulations by considering both computing infrastructures and applications. A well-known simulator of this type is CloudSim [10] and several others have emerged from it for increasingly specific purposes.

CloudSim definitively represents a reference for this category, simulating cloud-based scenarios to observe heterogeneous but complementary metrics such as the utilization of CPU, memory, battery and bandwidth and the service performance metrics such as response time, throughput, and availability. CloudSim simulation output and system traces can be exported as CSV files and utilized to train ML models for the purpose of optimizing and managing cloud computing resources [11]. To enhance its usability with a GUI and to provide additional modelling capabilities for the application workloads (e.g., complex usage patterns, real geographic coordinates of both cloud servers and users), CloudAnalyst [12] has been developed on top of CloudSim and SimJava [13] and then tested especially in extreme-scale applications like social networks.

Other and more recent well-known extensions of CloudSim are iFogSim [14], IoTSim-Edge [15] and EdgeCloudSim [16]: they all exploit CloudSim simulation engine but encompasses additional mechanisms and features for modeling and simulating IoT, Edge and Fog Computing Environments.

Although both network simulators and platform simulators could potentially play the role of data set generators, the former, unlike the latter, are not very flexible and not very suitable for moving away from the classic world of network simulations. The latter, however, are better suited to emulating the entire panorama of applications and services that can populate a context that extends from the edge to the cloud.

For this reason, a set of platform simulators that are best suited for emulating both computing/network infrastructures and applications/services, are analyzed in more detail from the perspective of their usability as dataset generators in the Edge-to-Cloud ontinuum.

A. CloudSim

CloudSim [10], as already said, serves as a prominent platform for simulating cloud computing infrastructures and applications, offering flexible output formats tailored for study and analysis purposes. It predominantly outputs text files (.txt) for simple data logging, CSV files (.csv) for structured data suitable for spreadsheet importation and data analysis tools, and XML files (.xml) for handling more complex structured data, thereby ensuring versatility in simulation results handling.

The datasets and system traces generated by CloudSim encompass various controlled conditions suitable for a range of simulations. These include:

1) *Resource Utilization Data:* This data category covers the utilization of CPU, memory, and bandwidth within the simulated cloud computing infrastructure.

2) *Cloud Service Metrics:* Encompassing performance metrics such as response time, throughput, and availability, this data provides insights into the efficiency and effectiveness of cloud services.

3) *Energy Consumption Data:* Crucial for simulating energy efficiency in data centers, this data category offers insights into the energy consumption patterns within the simulated cloud environment.

4) *Network Traffic Data:* This category includes details regarding network utilization, such as data transmission speeds and latency, providing a comprehensive understanding of network behavior within the simulated cloud infrastructure.

CloudSim datasets and system traces can be leveraged to train machine learning models aimed at optimizing and managing cloud computing resources effectively. Researchers can utilize these datasets for various purposes, including but not limited to:

- Predictive maintenance and resource allocation optimization.
- Anomaly detection and fault prediction to enhance system reliability.
- Dynamic workload scheduling and resource provisioning to improve performance and efficiency.
- Energy-aware resource management strategies to minimize energy consumption and carbon footprint.

By integrating machine learning techniques with CloudSim-generated datasets, researchers can advance the state-of-the-art in cloud computing optimization and management, addressing emerging challenges and maximizing the benefits of cloud-based services.

Cloud based scheduling process can be improved in terms of its efficiency through regression models. For example, a regression model for enhancing the scheduling efficiency validated through CloudSim traces has been presented in [17]. Another study conducted using performance analysis method for evaluating the efficacy of ML models with reference to computing environment has been proposed in [18]. The Google Cluster trace datasets are used to validate their model, which are similar to CloudSim's datasets to monitor the effectiveness of models based on resource management. In [19], CloudSim is used to model resource management approach for cloud computing.

A similar point was brought up by Ahamed et al. (2023) [20] in their discussion of the application of deep learning in cloud computing. The focus of discussion is to use deep learning models to train the generated data. In [21] different types datasets are used including Google Cluster Trace dataset and Plant Lab dataset. To determine task failure, CloudSim data can be used to prepare, clean and model dataset. Resource management in cloud data centers using ML models has been investigated in [22]. To set up the experiments, virtual machine traces that resembled CloudSim were employed.

B. iFogSim

The iFogSim is a simulation tool that is used for resource management of IoT applications in the context of edge and fog computing environment. It is an extension of ClouSim simulator that is compatible with IoT devices, it is a scalable simulator to deal resource management as a performance metrics. Under controlled conditions, iFogSim generates customizable datasets and system traces. The simulation can be executed in edge, IoT and fog computing applications, producing outcomes that facilitate the assessment of the simulated scenarios' efficacy [23] [24]. The result is contingent upon the configuration-specific parameters and may vary across scenarios. More specifically, iFogSim has the ability to generate the subsequent data sets and system traces during the simulations.

Comprehensive data regarding resource usage, data transfer, task completion, system statistics, and other metrics can be produced using iFogSim [13]. Analyzing system traces requires an understanding of how different components interact and operate in the simulated environment.

- Even Logs: Event logs contain all simulation occurrences such as system connection and disconnection, task assignment, and completion.
- Error Logs: The simulation records system errors and exceptions for troubleshooting and system improvement.
- Performance metrics: Energy consumption, latency, system resource utilization, throughput, and cost can be measured.
- Custom Data: Specific configurations for IoT device behaviors, network traffic patterns, and user-defined measures can also be used to make custom data.

ML models can be trained using iFogSim simulation output.

C. IoTSim-Edge

In edge computing contexts, IoTSim-Edge is a tool that represents the distribution and processing of data as it comes from devices connected to the IoT. In order to train ML models, this simulator provides simulated data in the form of log files. The log files contain the information about IoT devices and some factors about measurement of network metrics such edge time processing and bandwidth consumption. ML models can be trained based on these log files in IoT and edge computing environments. These ML techniques can be used to predict system behavior as well as perform optimization task for resource allocation and network management in dynamic IoT and edge computing [14].

D. PureEdgeSim

PureEdgeSim [25] shares some foundational concepts with CloudSim [10], it is a simulator that facilitates the simulation of resource management strategies and the performance evaluation of various computing environments, including Cloud, Fog, and Pure Edge Computing. It supports scalability by enabling the simulation of thousands of devices and accommodates edge device heterogeneity, considering factors such as mobility, energy source, and latency requirements. The simulator incorporates an improved network model and includes a tasks orchestrator module responsible for managing resources, balancing workload, and facilitating multi-tier scenarios. PureEdgeSim simulator can potentially be used as a dataset generator through the use of its log files that can reflect the complexities and characteristics of real-world edge computing scenarios. These logs can include information such as device heterogeneity, network behavior, resource utilization, and task orchestration.

E. EdgeCloudSim

EdgeCloudSim is a discrete event simulator providing an environment for performance evaluation of edge computing systems [15]. It is an evolution of CloudSim [10] that allows simulating systems comprising (mobile) devices, edge servers and cloud servers, which are individually modeled in terms of computer processing (CPUs) and memory space (RAM and ROM). Devices can execute or offload computing Tasks (modeled in terms of Million Instructions MI) in the continuum according to policies that considers the dynamic resource availability and the position of the devices themselves with respect to the edge/cloud servers.

Indeed, EdgeCloudSim is designed with modular architecture in order to implement key functionalities (such as wireless LAN and WAN transmission, device mobility, realistic and tunable load generation, etc.) separately within classes which are loosely coupled. In particular, EdgeCloudSim has five main modules available: Core Simulation, Networking, Load Generator, Mobility and Edge Orchestrator. To ease fast prototyping efforts, each module contains a default implementation that can be easily extended. Goal of the simulation is to assess the QoS performance of a given task/set of tasks: the QoS is evaluated in terms of service time (with the possibility of distinguishing between computation and networking time), success ratio, service failures (which can be due to lack of network or computation resources or to mobility) and resource utilization. The metrics that EdgeCloudSim allows inspecting include the service failure and the service time (for both it is possible to distinguish the contributions imputable to the lack of computation power, of network bandwidth or coverage due to mobility reasons) as well as the bandwidth, energy and the hardware utilization.

The analysis of the simulations output supports the engineering or re-engineering of the target scenario by providing insights on infrastructural (e.g. is the network bandwidth sufficient for uploading my data?), architectural (e.g. is convenient to offload task towards the Cloud in terms of service failure?) or algorithmic aspects (e.g. is a heuristic convenient enough with respect to another?), thus comprehensively supporting the decision-making processes.

In ensuring accessibility and versatility, EdgeCloudSim supports multiple output formats. These formats support different needs: text files for basic data logging, CSV files for structured data analysis, and XML files for handling more intricate data structures. This flexibility allows researchers to seamlessly integrate simulation log and results into their preferred dataset format.

III. A SELECTION AMONG SIMULATORS

Simulators designed for comparative analysis of computer systems across the continuum are receiving increased attention in research. They are being surveyed and classified [26][27] based on the Quality/Metrics Characteristics outlined in ISO/IEC 25010[28]/25023[29] standards. These characteristics include functional suitability, performance efficiency, compatibility, reliability, maintainability, portability, among others. Additionally, these simulators are evaluated based on their modeling features, which encompass infrastructure modeling, application modeling, resource management modeling, scalability, and mobility models. By categorizing and assessing simulators according to these criteria, researchers can better understand their capabilities and limitations, as well as their suitability for different use cases and scenarios in computer systems analysis and evaluation. This approach enables researchers to make informed decisions when selecting simulators for their specific research objectives and contributes to the advancement of simulation-based research in computer systems.

From both surveys and after a careful state-of-the-art analysis summarized in Table 1, EdgeCloudSim has emerged as one of the most comprehensive edge–cloud simulators and with more reliable simulation results: indeed, it seamlessly covers the whole simulation space and simultaneously considers infrastructural, computational and network aspects which concurrently impact on real IoT systems. Hence, EdgeCloudSim has been already used in several research works of heterogeneous Smart-domains, aiming to preliminary assess the performance of an IoT services in many configuration settings before its actual deployment [30][31][32]. Indeed, its versatility and usability (granted by its modular architecture and open-source code, developed according to every good software engineering practice) fostered its applications both on already existing and on IoT systems to-be, but also its inclusion in application-agnostic methodology as in [33].

TABLE I. SIMULATORS COMPARISON

Features	**Simulators**				
	Edge Cloud Sim	***Cloud Sim***	***iFog Sim***	***IoTsim Edge***	***Pure Edge Sim***
Documentation	Tutorials and community	Official website	Tutorials	Poor documentation	Papers available
Setup	Easy	Easy	Easy	Difficult	Easy
Ease of use	Easy	Moderate	Moderate	Moderate	Easy
Mobility	✓		✓	✓	✓
Scalability	Large-scale edge to cloud environment	Large-scale cloud environment	Medium-scale fog computing scenario	Large-scale IoT edge environments	Large scale of edge environments
Orchestration	✓		✓		✓
Networking	✓	✓	✓	✓	✓
Virtualization	✓	✓			
Containerization	✓	✓			
Event Tracking	✓				
Microservices				✓	

Other IoT simulators have been proposed in the last 4 years but none of them combine these integrated properties with aspects of continuum computing or the ease of customization offered by EdgeCloudSim, which is why we consider this simulator a good choice to achieve the purposes discussed so far.

IV. EDGECLOUDSIM INSIGHT

The EdgeCloudSim simulator allows simulating different workloads, infrastructures and offloading policies in the continuum, simultaneously generating datasets and system traces about important optimization metrics like system, network and resource utilization: the system logs (i.e, the events triggering the simulation and that represent "screenshots" of the system) can be gathered and processed so as to implement the data generation task. Of particular relevance is the additional opportunity of defining and simulating ad-hoc ML-driven offloading policies as described in [34] and assessing their performance: this feature of EdgeCloudSim might be used to test and validate the ML models we intend to develop so as to validate them.

However, there are some simplifications which limit the usability and reliability of EdgeCloudSim. First, the task modelling considers exclusively the computational burden and not the subject of the computation; this means that the calculation of a task complexity (in terms of MI) is not straightforward and a preliminary emulation phase is needed to estimate it. Then, tasks are considered independent with each other, so the simulation of synchronous applications is not enabled by default. Finally, it simulates all properties of tasks before the start of simulation and schedules their generations by adding them into the event queue; in this way, when a large-scale system needs to be simulated, all the events (even several thousands) are immediately added to the event queue, which is actually not necessary, and can take a large amount of time or out of memory errors [35]. In order to provide reliability in the simulation, some interventions might be implemented with respect to the current version (4.0) of EdgeCloudSim:

- Novel mechanisms should be introduced and implemented to trace events online and not when the simulation ends. This intervention would simplify the monitoring of the data generation process. It would enable the logging of some intermediate events (i.e., step-by-step device movements) which are not currently tracked.
- Default energy models of edge nodes, edge- and cloud-servers are quite simple, and they might be improved since to the energy awareness has a great relevance for autonomic system operation optimization purposes. Then, additional computing resources can be introduced: indeed, by default, EdgeCloudSim exclusively deals with CPU, but GPU and other accelerators are key target for optimization. According to these extensions, obviously, the default yet simple cost model might also be improved.
- EdgeCloudSim provides simple models based on random mobility patterns. Given the relevance of the mobility aspect in the computing continuum, these models might be extended, or more advanced ones introduced, coming, for example, from the integration with specific-purpose simulators such as SUMO [36].
- Finally, additional features might be inserted so to make the network modelling more realistic: by default, the Network model provided by EdgeCloudSim handles the transmission delay in the WLAN and WAN by considering both upload and download data, available bandwidth and number of connected devices, based on a single server queue model. Additional features should be considered, especially to model the communication channel better and be compliant with advanced technology and protocols such as 5G and beyond.

V. CONCLUSIONS

In the context of developing ML and other AI methods, simulators play a highly essential role, particularly in edge-to-cloud continuum scenarios. This is due to the fact that simulators can generate enormous datasets that contain a wide variety of data types in situations where it would be difficult to collect them in the real world. All this data could be extremely useful for training, testing, and analyzing ML and other AI techniques for autonomic system operation purposes. Therefore, in this paper, we have analyzed the most popular simulators, the various features and functionalities, along with their advantages and disadvantages and the result of our study makes us understand that there is not a single simulator capable of generating every type of data set, in every type of context. As an output of this study, the EdgeCloudSim simulator, although it needs some non-trivial extensions, seems to be a very promising and versatile solution. Future works, in fact, will be centered on some interventions to extend its functionalities with respect to the current version with a particular focus on the enhancement of events tracing and on the improvement of mobility and network/channel models.

ACKNOWLEDGMENT

This work has been funded by the Horizon Europe research and innovation programme of the European Union, under grant agreement no 101092912, project MLSysOps.

REFERENCES

[1] Rosendo, D., Costan, A., et al. (2022). Distributed intelligence on the Edge-to-Cloud Continuum: A systematic literature review. Journal of Parallel and Distributed Computing, 166, 71-94.

[2] Varga, A., & Hornig, R. (2010, May). An overview of the OMNeT++ simulation environment. In 1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems.

[3] El–Hajj, W., Al–Tamimi, M., & Aloul, F. (2015). Real traffic logs creation for testing intrusion detection systems. Wireless Communications and Mobile Computing, 15(14), 1851-1864.

[4] Schwarzmann, S., et al. (2019, October). Estimating video streaming QoE in the 5G architecture using machine learning. In Proceedings of the 4th Internet-QoE Workshop on QoE-based Analysis and Management of Data Communication Networks (pp. 7-12).

[5] Nardini, G., Sabella, D., Stea, G., Thakkar, P., & Virdis, A. (2020). Simu5g–an omnet++ library for end-to-end performance evaluation of 5g networks. IEEE Access, 8, 181176-181191.

[6] Nardini, G., Noferi, A., Ducange, P., & Stea, G. (2023). Exploiting Simu5G for generating datasets for training and testing AI models for 5G/6G network applications. SoftwareX, 21, 101320.

[7] Zhang, Q., Ng, K. K., et al. (2021, August). MimicNet: Fast performance estimates for data center networks with machine learning. In Proceedings of the 2021 ACM SIGCOMM 2021 Conference (pp. 287-304).

[8] Campanile, L., Gribaudo, M., Iacono, M., Marulli, F., & Mastroianni, M. (2020). Computer network simulation with ns-3: A systematic literature review. Electronics, 9(2), 272.

[9] Gawłowicz, P., et al. (2019, November). Ns-3 meets openai gym: The playground for machine learning in networking research. In Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (pp. 113-120).

[10] Buyya, R., Ranjan, R., et al. (2009, June). Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In 2009 international conference on high performance computing & simulation (pp. 1-11). IEEE.

[11] Jawaddi, S. N. A., & Ismail, A. (2024). Integrating OpenAI Gym and CloudSim Plus: A simulation environment for DRL Agent training in energy-driven cloud scaling. Simulation Modelling Practice and Theory, 130, 102858.

[12] Buyya, R. (2009). CloudAnalyst: A CloudSim-based tool for modelling and analysis of large scale cloud computing environments. Distrib. Comput. Proj. Csse Dept. Univ. Melb, 433-659.

[13] Howell, F., McNab, R. (1998). SimJava: A discrete event simulation library for java. Simulation Series, 30, 51-56.

[14] Gupta, H., Vahid Dastjerdi, A., et al. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Software: Practice and Experience, 47(9), 1275-1296.

[15] Jha, D. N., Alwasel, K., et al. (2020). IoTSim‐Edge: a simulation framework for modeling the behavior of Internet of Things and edge computing environments. Software: Practice and Experience, 50(6), 844-867.

[16] Sonmez, C., Ozgovde, A., & Ersoy, C. (2018). Edgecloudsim: An environment for performance evaluation of edge computing systems. Transactions on Emerging Telecommunications Technologies, 29(11), e3493.

[17] Fernández-Cerero, D., Troyano, J. A., et al. (2022). Machine learning regression to boost scheduling performance in hyper-scale cloud-computing data centres. Journal of King Saud University-Computer and Information Sciences, 34(6), 3191-3203.

[18] Saxena, D., Kumar, J., et al. (2023). Performance analysis of machine learning centered workload prediction models for cloud. IEEE Transactions on Parallel and Distributed Systems, 34(4), 1313-1330.

[19] Barthwal, V., Rauthan, M. M. S., & Varma, R. (2020). A survey on application of machine learning to manage the virtual machines in cloud computing. International Review of Applied Sciences and Engineering, 11(3), 197-208.

[20] Ahamed, Z., Khemakhem, M., Eassa, F., Alsolami, F., & Al-Ghamdi, A. S. A. M. (2023). Technical Study of Deep Learning in Cloud Computing for Accurate Workload Prediction. Electronics, 12(3), 650.

[21] Gollapalli, M., AlMetrik, M. A., AlNajrani, B. S., AlOmari, A. A., AlDawoud, S. H., AlMunsour, Y. Z., ... & Aloup, K. M. (2022). Task Failure Prediction Using Machine Learning Techniques in the Google Cluster Trace Cloud Computing Environment. Mathematical Modelling of Engineering Problems, 9(2).

[22] Khan, T., Tian, W., Zhou, G., Ilager, S., Gong, M., & Buyya, R. (2022). Machine learning (ML)-centric resource management in cloud computing: A review and future directions. Journal of Network and Computer Applications, 204, 103405.

[23] Fahimullah, M., Philippe, G., et al. (2023). Simulation Tools for Fog Computing: A Comparative Analysis. Sensors, 23(7), 3492.

[24] Mahmud, R., Pallewatta, S., Goudarzi, M., & Buyya, R. (2022). Ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments. Journal of Systems and Software, 190, 111351.

[25] Mechalikh, C., et al. (2019, July). PureEdgeSim: A simulation toolkit for performance evaluation of cloud, fog, and pure edge computing environments. In 2019 international conference on high performance computing & simulation (HPCS) (pp. 700-707). IEEE.

[26] Svorobej, S., Takako Endo, et al. (2019). Simulating fog and edge computing scenarios: An overview and research challenges. Future Internet, 11(3), 55.

[27] Ashouri, M., Lorig, F., Davidsson, P., & Spalazzese, R. (2019). Edge computing simulators for iot system design: An analysis of qualities and metrics. Future Internet, 11(11), 235.

[28] ISO/IEC 25010:2023(en) Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Product quality model (https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-2:v1:en)

[29] ISO/IEC 25023:2016 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of system and software product quality (https://www.iso.org/standard/35747.html)

[30] Casadei, R., Fortino, G., et al. (2019). A development approach for collective opportunistic edge-of-things services. Information Sciences, 498, 154-169.

[31] Aloi, G., Fortino, G., et al. (2020). Simulation-driven platform for Edge-based AAL systems. IEEE Journal on Selected Areas in Communications, 39(2), 446-462.

[32] Barbieri, A., Marozzo, F., & Savaglio, C. (2021, October). Iot platforms and services configuration through parameter sweep: a simulation-based approach. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1803-1808). IEEE.

[33] Savaglio, C., & Fortino, G. (2021). A simulation-driven methodology for IoT data mining based on edge computing. ACM Transactions on Internet Technology (TOIT), 21(2), 1-22.

[34] Sonmez, C., Tunca, C., Ozgovde, A., & Ersoy, C. (2020). Machine learning-based workload orchestrator for vehicular edge computing. IEEE Transactions on Intelligent Transportation Systems, 22(4), 2239-2251.

[35] Freymann, R., Shi, J., Chen, J. J., & Chen, K. H. (2021, December). Renovation of EdgeCloudSim: an efficient discrete-event approach. In 2021 Sixth international conference on fog and mobile edge computing (FMEC) (pp. 1-8). IEEE.

[36] Behrisch, M., Bieker, L., Erdmann, J., & Krajzewicz, D. (2011). SUMO–simulation of urban mobility: an overview. In Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation. ThinkMind.