Tunnel Load Simulator: A Monte Carlo Framework for Synthetic Electrical Demand Modeling of Road Tunnels

Haytham El-Houari[1] & **Cyril Voyant**[2]

[1] *Université Sidi Mohamed Ben Abdellah (USMBA), Fès, Morocco*

[2] *Mines Paris - PSL, OIE Laboratory (Centre Observation, Impacts, Énergie), France*

Abstract

Road tunnels are energy-intensive infrastructures (lighting, ventilation, auxiliary systems), but the scarcity of operational datasets limits data-driven energy management research. This paper introduces Tunnel Load Simulator, an open-source Python package that generates synthetic electrical load profiles for road tunnels. It integrates traffic-dependent demand modeling, lighting and ventilation estimation, stochastic event generation, and Monte Carlo simulation in an efficient framework, configurable via GUI or simulation engine. Outputs include time-resolved demand profiles, KPIs, seasonal analyses, and probabilistic uncertainty envelopes. Validation confirms computational consistency, parameter sensitivity, and stable probabilistic outputs. Released under MIT License, the tool supports energy assessment, uncertainty quantification, ML dataset generation, digital twin prototyping, and educational use.

Keywords: Road tunnels; Energy consumption; Synthetic data generation; Monte Carlo simulation; Load modeling; Infrastructure energy systems; Python software.

Abbreviations

Abbreviation	Definition
DOI	Digital Object Identifier
GUI	Graphical User Interface
KPI	Key Performance Indicator
MC	Monte Carlo
TLS	Tunnel Load Simulator
SCADA	Supervisory Control and Data Acquisition

1. Motivation and Significance

Road tunnels are important components of modern transportation networks and require the continuous operation of energy intensive systems such as lighting, ventilation, drainage, communication, and safety equipment. The electrical consumption of these infrastructures depends on a complex interaction between tunnel geometry, traffic intensity, environmental conditions, and operational events (PIARC, 2024 ; Menéndez et al., 2024; Riess, 2022). The transition toward smarter transportation infrastructures has increased the need for energy-management and digital-twin applications (Fuller et al., 2020; Jones, 2021). However, publicly available tunnel-energy datasets remain extremely scarce because operational data are rarely released due to security and confidentiality constraints (Nikolenko, 2021). Several software platforms are available for modeling transportation and energy systems, including EnergyPlus for building-energy simulation (Crawley et al., 2001), SUMO (Lopez et al., 2018), MATSim (W Axhausen et al., 2016), and Aimsun (Barceló, 2010)for traffic simulation. Nevertheless, these tools are not specifically designed to represent the coupled interactions between traffic demand, lighting operation, ventilation requirements, and stochastic disturbances that characterize road-tunnel electrical consumption.

Most existing studies focus on individual subsystems, particularly lighting (Peña-García, 2022; Xu et al., 2024) or ventilation (Menéndez et al., 2024; Riess, 2022). Although annual energy consumption may range from approximately 551 to 1306 MWh depending on operating conditions (Menéndez et al., 2024), integrated approaches simultaneously modeling the principal tunnel electrical loads remain uncommon. Furthermore, the associated datasets are rarely released publicly, limiting reproducibility. To the best of our knowledge, no open-source software currently provides a dedicated framework for generating synthetic tunnel electrical-load datasets that combine traffic demand, lighting, ventilation, stochastic events, and Monte Carlo uncertainty quantification.

To address the scarcity of publicly available tunnel-energy datasets, we developed Tunnel Load Simulator (TLS), an open-source Python package for generating synthetic electrical-demand profiles of road tunnels. TLS combines simplified models of lighting, ventilation, and auxiliary systems with stochastic representations of traffic variability and operational events. Through Monte Carlo simulation, the software produces synthetic datasets, probabilistic demand envelopes, and performance indicators suitable for benchmarking, uncertainty analysis, machine-learning workflows, and digital-twin prototyping.

2. Software Description

TLS provides a reproducible framework for generating synthetic electrical-load profiles that capture the combined effects of tunnel geometry, traffic conditions, environmental influences, and stochastic operational disturbances (Nikolenko, 2021).

As Table 1 shows, available transportation and energy tools each leave at least one of these needs unmet; TLS v1.0 is designed to cover them jointly.

Table 1. Functional positioning of Tunnel Load Simulator (TLS) against representative transportation and energy-system software.

Software	Traffic	Tunnel Energy	Monte Carlo	Open Source
EnergyPlus	No	Partial	No	Yes
SUMO	Yes	No	No	Yes
MATSim	Yes	No	Limited	Yes
Tunnel Load Simulator	Yes	Yes	Yes	Yes

Tunnel Load Simulator is designed as a synthetic-data generator rather than a high-fidelity engineering simulator, aiming to produce statistically and physically plausible electrical-demand scenarios suitable for benchmarking, sensitivity analysis, uncertainty quantification, and data-driven research. To account for the inherent variability of tunnel operations, TLS relies on Monte Carlo simulation, a widely adopted methodology for uncertainty propagation and risk assessment in engineering systems (Robert et al., 2004; Zhang, 2021). Generating multiple independent realizations from which probabilistic demand envelopes, confidence intervals, and KPI distributions are derived. This approach is especially relevant for infrastructure systems, where operational conditions are inherently uncertain and highly variable.

TLS integrates several functionalities designed to support both research and practical exploratory studies.

It generates sub-hourly electrical demand time series over user defined horizons, decomposed into lighting, ventilation, and auxiliary loads. Runs are fully reproducible through deterministic seed management. The main features are:

- Configurable tunnels. Geometry (length, tubes, lanes, altitude, depth, gradient), operating context (urban, peri-urban, rural), lighting and ventilation technology, all set without editing the source.
- Traffic-driven demand. Parameterized daily profiles with morning/evening peaks, seasonal and weekday/weekend effects, driving both lighting and ventilation.
- Stochastic events. Pollution episodes and accidents generated probabilistically (occurrence, duration, timing) to reproduce realistic operational variability.
- Monte Carlo uncertainty. Independent seeded realizations yielding probabilistic load envelopes, percentile profiles, and KPI distributions.

- Outputs and access. Interactive visualization, CSV export, and direct integration of the simulation engine into Python workflows.

Full equations and default parameter values are provided in the repository documentation, while Figure 1 summarizes the overall workflow.

The architecture of TLS is designed around three core principles: modularity, reproducibility, and reusability. The simulation engine is decoupled from the graphical user interface, allowing users to execute simulations programmatically through Python scripts or to explore scenarios interactively through a web-based interface.

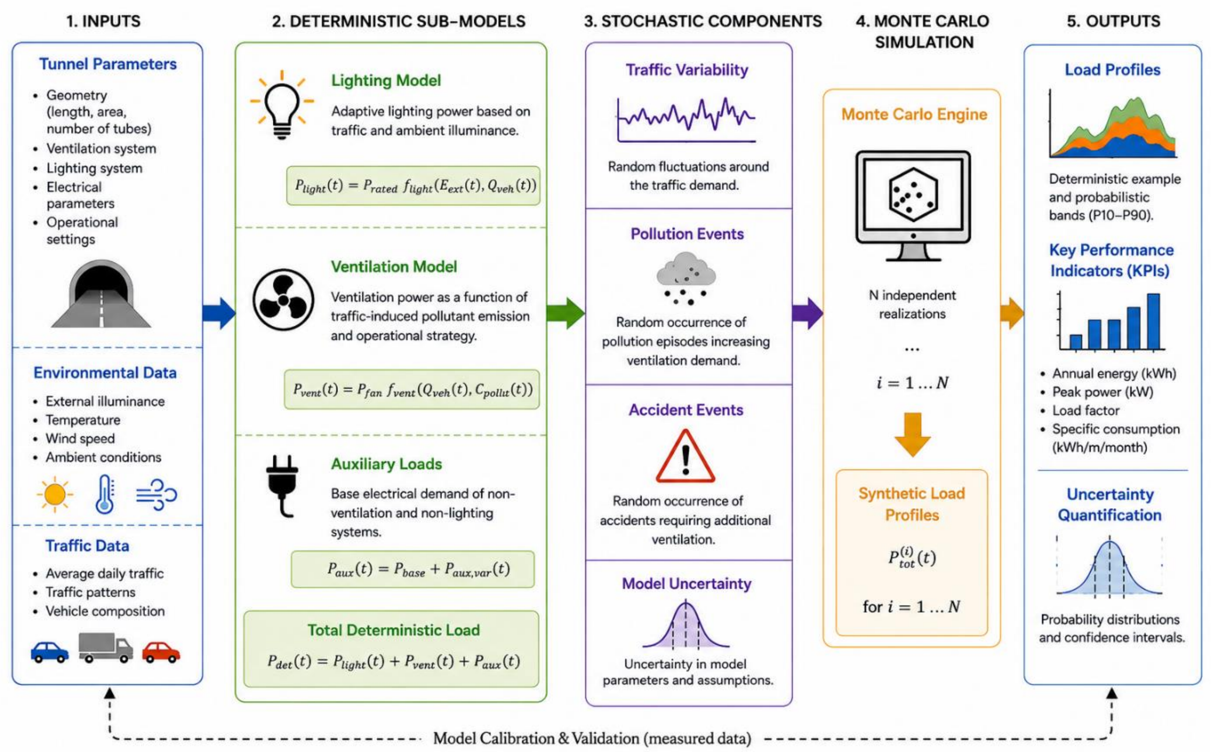

Figure 1. General architecture and computational workflow of Tunnel Load Simulator, illustrating the relationships between user-defined inputs, deterministic and stochastic modeling components, the Monte Carlo simulation engine, and output generation modules.

3 Deterministic and Stochastic Modeling Components

The deterministic layer computes baseline electrical demand from simplified models of lighting, ventilation, and auxiliary systems. A stochastic layer is then superimposed to represent traffic fluctuations, pollution episodes, accident events, and measurement and operational noise.

A vectorized Monte Carlo engine combines deterministic and stochastic components across multiple independent realizations, each initialized with a deterministic seed to ensure reproducibility while preserving statistical independence. Each realization produces a full electrical demand time series at the chosen temporal resolution; the ensemble is then aggregated into uncertainty indicators and performance metrics, moving the output beyond a single deterministic prediction.

$$seed_{\_i} = base_{\_seed} + i$$

Simulation outputs include synthetic load time series and probabilistic performance indicators that can be visualized through the graphical interface or exported for further analysis.

4. Software Structure

The software is organized around four main modules:

1. **Configuration management**, implemented through the Tunnel Config dataclass, which stores all tunnel geometry, operational parameters, traffic characteristics, and stochastic settings.
2. **Traffic and environmental modeling**, which generates synthetic traffic profiles and daylight conditions using vectorized numerical operations.
3. **Load simulation engine**, responsible for computing lighting, ventilation, auxiliary loads, and total electrical demand.
4. **Monte Carlo analysis module**, which performs repeated stochastic simulations and computes probabilistic performance indicators.

The separation of these components facilitates maintenance, testing, and future extension of the software.

5. Vectorized Numerical Computation

Computational efficiency is achieved through NumPy vectorized operations, allowing traffic profiles, environmental variables, stochastic disturbances, and electrical loads to be evaluated simultaneously over complete arrays. This approach enables long-term simulations to be executed efficiently on standard desktop hardware (Harris et al., 2020).

Time-series management and data manipulation are handled using Pandas, which provides efficient indexing, resampling, aggregation, and statistical analysis functionalities

5.1 Data Outputs

Simulation outputs are returned as Pandas DataFrames to facilitate integration with scientific-computing and machine-learning workflows. Generated outputs include representative realizations, KPI summaries, probabilistic demand envelopes, and seasonal demand profiles.

5.2 Computational Efficiency and Reproducibility

Reproducibility is ensured through deterministic seed management and transparent parameterization, enabling identical results to be reproduced from the same configuration and random seed.

On a standard laptop (Intel i7, 16 GB RAM), a one-year simulation with 50 Monte Carlo realizations requires approximately 5 seconds.

5.3 Software Dependencies

The software relies on widely adopted open-source Python libraries, including NumPy for numerical computation, Pandas for data handling, Streamlit for the graphical user interface, and Plotly for interactive visualization. These dependencies contribute to portability, maintainability, and ease of installation across operating systems.

6. Outputs and Usage

Tunnel Load Simulator produces time-resolved electrical-demand profiles together with aggregated performance indicators and probabilistic metrics derived from Monte Carlo simulations. The outputs are designed to support energy assessment, uncertainty quantification, benchmarking studies, and the generation of synthetic datasets for research applications.

6.1 Input Configuration

TLS provides a configurable parameterization framework covering tunnel geometry, operating conditions, traffic demand, stochastic events, and Monte Carlo settings. The main input categories are summarized in Table 2.

Table 2. Main input categories available in Tunnel Load Simulator.

Category	**Main parameters**
Time horizon	Start date, simulation duration, temporal resolution
Geometry	Tunnel length, number of tubes, lanes per tube, altitude, depth, gradient
Operating context	Lighting technology, ventilation system, auxiliary loads
Traffic pattern	Traffic level, morning and evening peak hours, peak width

Stochastic effects	Gaussian noise, pollution-event probability, accident-event probability
Monte Carlo settings	Number of realizations, random seed

This parameterization enables users to investigate a wide range of tunnel configurations and operational scenarios without modifying the source code.

6.2 Simulation Outputs

Each simulation generates a time series describing tunnel operation and electrical demand. Exported variables include traffic indicators, subsystem loads, event markers, and energy-consumption metrics. The principal output variables are listed in Table 3; Figure 2 shows a representative example illustrating the decomposition of total demand into lighting, ventilation, and auxiliary loads.

Table 3. summarizes the principal variables generated by the simulator.

Variable	**Description**	**Unit**
traffic_index	Normalized traffic demand index	-
lighting_kw	Lighting electrical load	kW
ventilation_kw	Ventilation electrical load	kW
auxiliary_kw	Auxiliary electrical load	kW
power_kw	Total electrical demand	kW
energy_kwh	Energy consumption per timestep	kWh
pollution_event	Pollution-event indicator	Binary
accident_event	Accident-event indicator	Binary

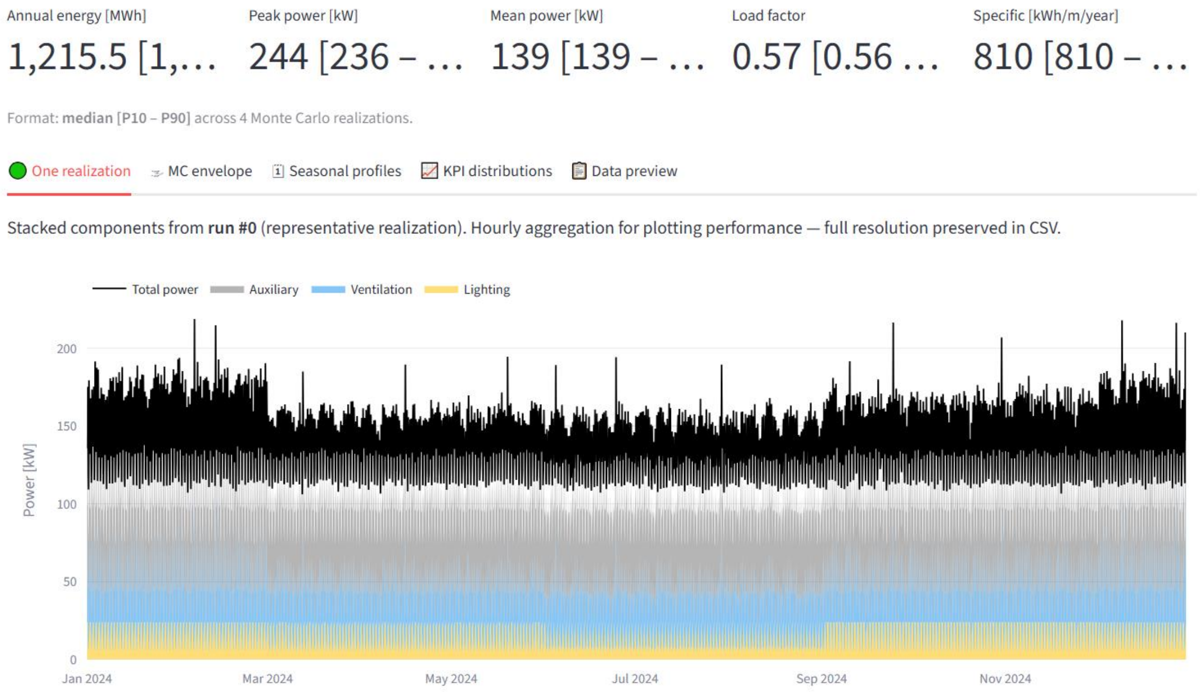

Figure 2. Representative electrical-load decomposition generated by TLS.

6.3 Key Performance Indicators

TLS automatically computes key performance indicators (KPIs) describing tunnel energy performance.

The generated indicators include:

- total energy consumption;
- annualized energy consumption;
- peak electrical demand;
- average electrical demand;
- load factor;
- specific annual energy consumption;
- number of pollution events;
- number of accident events.

Table 4 illustrates a representative KPI summary obtained from a Monte Carlo simulation.

Table 4. Example of key performance indicators generated by Tunnel Load Simulator.

KPI	Median	P10	P90
Annual energy (MWh)	3233	3229	3237
Peak power (kW)	873	792	922
Mean power (kW)	369	369	370
Load factor	0.42	0.40	0.47
Specific consumption (kWh/m/year)	2156	2153	2158

These indicators facilitate rapid comparisons between alternative tunnel designs, equipment configurations, and operating conditions.

6.4 Probabilistic Outputs

Monte Carlo simulations generate probabilistic summaries of electrical demand, including mean, median, P10, and P90 profiles. These statistics provide uncertainty envelopes that quantify the variability associated with traffic fluctuations, pollution episodes, accident events, and stochastic disturbances. Figure 3 illustrates a typical probabilistic envelope generated from multiple Monte Carlo realizations.

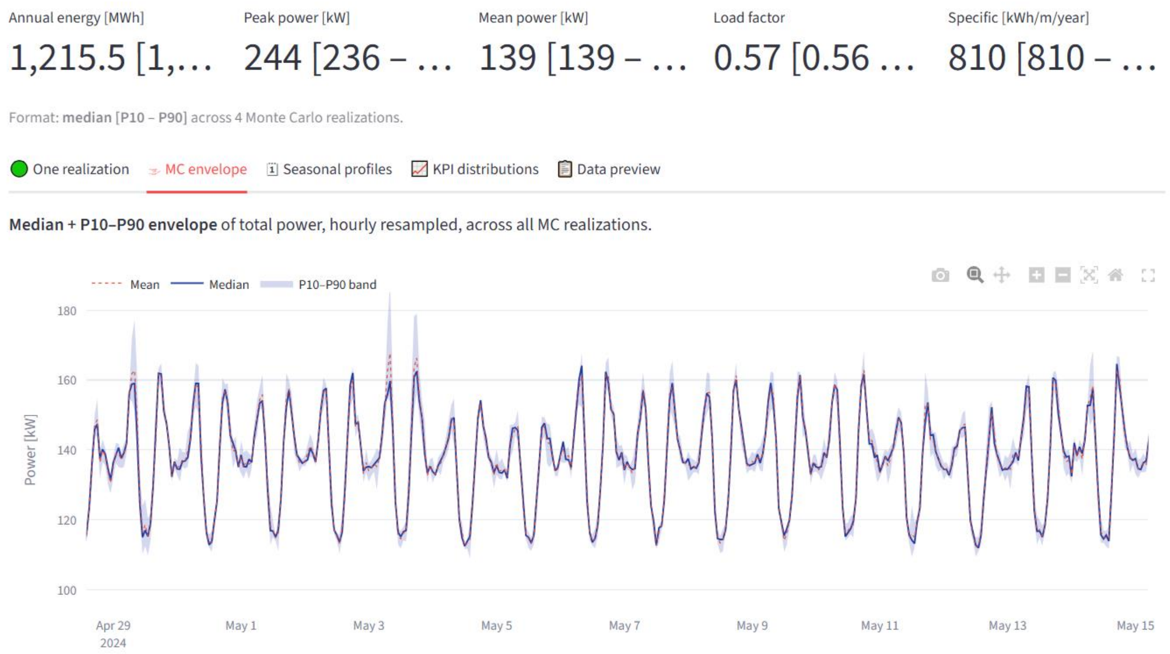

Figure 3. Probabilistic electrical-demand envelope obtained from Monte Carlo simulations.

6.5 Seasonal Profiles

TLS also generates seasonal demand profiles by aggregating hourly electrical consumption according to meteorological seasons, allowing users to identify seasonal variations and compare the relative influence of traffic patterns and daylight conditions throughout the year. An illustrative example is shown in Figure 4.

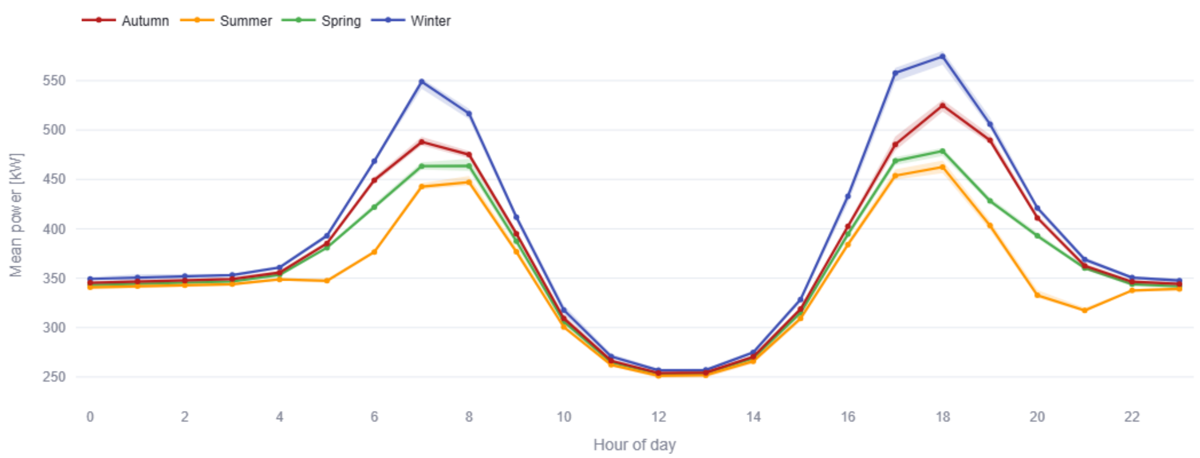

Figure 4. Average seasonal electrical-demand profiles generated by Tunnel Load Simulator.

6.6 Usage Scenarios

TLS can support a broad range of applications, including synthetic-data generation for machine learning, forecasting, and anomaly-detection studies, uncertainty quantification, energy-performance assessment (annual consumption, peak demand, load factors), sensitivity analysis across geometry, traffic, and equipment parameters, and digital-twin development. The generated datasets can also be integrated into optimization, predictive-maintenance, and decision-support workflows for transportation infrastructures.

7. Validation

The objective of Tunnel Load Simulator is not to reproduce the behavior of a specific tunnel but to generate physically plausible synthetic electrical-load profiles. Accordingly, validation focuses on numerical consistency, parameter sensitivity, stochastic robustness, and agreement with engineering trends reported in the literature.

7.1 Computational Verification

Numerical verification was performed to ensure physically consistent outputs. Electrical demand is constrained to non-negative values, energy consumption is derived directly from power demand and timestep duration, and deterministic seed management guarantees reproducible Monte Carlo simulations (Robert et al., 2004; Wilson et al., 2017). Repeated executions using identical input parameters and random seeds produce identical outputs, whereas different seeds generate statistically independent realizations.

The resulting datasets were systematically inspected to verify:

- absence of negative power values;
- consistency between power and energy calculations;
- correct activation and duration of stochastic events;
- reproducibility of simulation outputs under fixed seeds.

These checks confirmed the numerical consistency and stability of the implementation.

7.2 Sensitivity Analysis

Sensitivity analysis was conducted to evaluate the response of the simulator to variations in key input parameters, following common validation practices for infrastructure-energy models (Menéndez et al., 2024; Riess, 2022).

The performed tests showed that:

- increasing traffic intensity produces higher lighting and ventilation loads;
- increasing tunnel length leads to larger total energy consumption;
- more energy-intensive ventilation technologies result in higher electrical demand;
- higher probabilities of pollution or accident events increase ventilation-related consumption.

These trends are consistent with the expected behavior of tunnel electrical systems and support the physical plausibility of the implemented relationships.

Figure 5. Example sensitivity analysis showing the influence of traffic intensity on annual energy consumption.

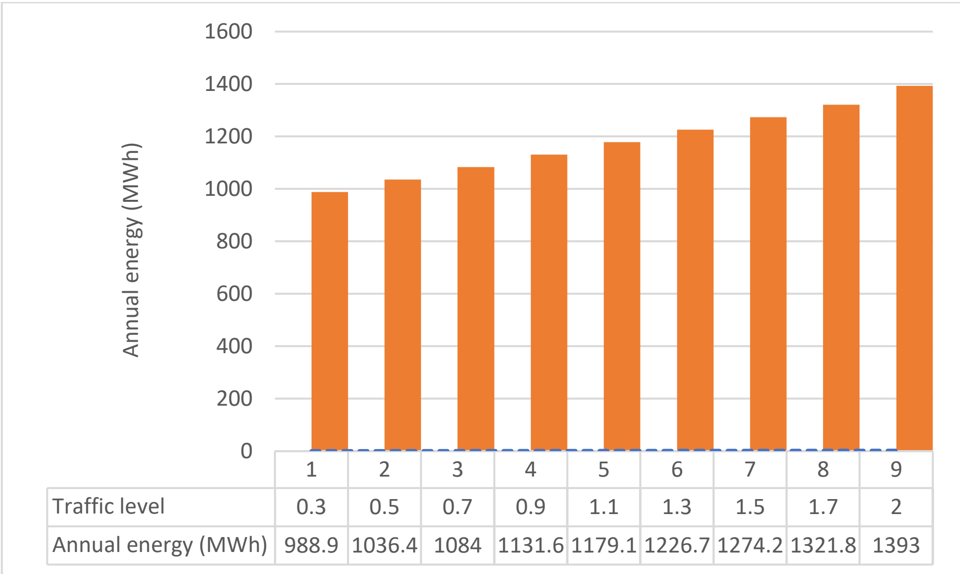

Figure 5. Sensitivity of annual energy consumption to traffic intensity

Annual energy consumption increases from approximately 989 MWh at a traffic level of 0.3 to 1393 MWh at a traffic level of 2.0, corresponding to an increase of about 62%. This trend is consistent with the expected influence of traffic demand on tunnel electrical loads and supports the validity of the implemented modeling assumptions.

7.3 Monte Carlo Robustness

Monte Carlo robustness was assessed through repeated simulations under identical input conditions; uncertainty envelopes and KPI distributions remained stable across runs. For the presented case study, 50 realizations were sufficient to obtain stable estimates of annual energy consumption, peak demand, and load-factor distributions, with median and percentile statistics showing limited variability, indicating adequate convergence for the considered use cases.

The probabilistic outputs generated by the simulator provide useful information regarding the uncertainty associated with traffic fluctuations, pollution events, accident occurrences, and stochastic disturbances. Such uncertainty-aware indicators are particularly relevant for energy planning, scenario analysis, and data-driven studies (Zhang, 2021).

7.4 Consistency with Published Tunnel Energy Characteristics

Although TLS is not calibrated to a specific infrastructure, the generated demand profiles reproduce several characteristics commonly reported in tunnel-energy studies. Lighting and ventilation remain the dominant contributors to electrical consumption, while daily and seasonal demand variations follow expected traffic-driven patterns (Menéndez et al., 2024; PIARC; 2024; Riess, 2022) Furthermore, the probabilistic outputs are consistent with uncertainty-oriented approaches increasingly adopted in infrastructure-energy studies (Zhang, 2021) These observations support the realism of the generated synthetic datasets.

7.5 Limitations

TLS is intended as a first-order synthetic-data generator and does not replace detailed engineering analyses or calibrated operational models.

Several simplifying assumptions are adopted:

- simplified traffic representations based on Gaussian peak profiles;
- empirical lighting and ventilation relationships;
- stochastic event generation based on probability distributions rather than measured incident databases;
- absence of detailed aerodynamic, thermal, or CFD-based tunnel models.

For engineering applications, model parameters should be calibrated using measured traffic, energy-consumption records, equipment specifications, and tunnel-specific operating procedures. Despite its simplifying assumptions, TLS provides a transparent and computationally efficient framework for generating realistic synthetic tunnel-load datasets.

8. Availability and Reuse

8.1. Software Availability

Tunnel Load Simulator (TLS) is an open-source Python software released under the MIT License. The source code is publicly available through GitHub and archived on Zenodo to ensure long-term accessibility and reproducibility. The archived release associated with this publication corresponds to version v1.0.1 and is identified by DOI 10.5281/zenodo.20080043 (El-Houari & Voyant, 2026). The repository includes the simulation engine, graphical user interface, documentation, example configurations, and citation metadata (CITATION.cff). The software is compatible with Python 3.10 or later and can be executed on Windows, Linux, and macOS operating systems.

Table 5. Software availability and distribution information for Tunnel Load Simulator.

Item	Description
Software name	Tunnel Load Simulator (TLS v1.0.1)
Repository	GitHub
Programming language	Python
License	MIT License
Operating systems	Windows, Linux, macOS
Dependencies	NumPy, pandas, Streamlit, Plotly
Documentation	README.md and inline documentation
Citation metadata	CITATION.cff
DOI	10.5281/zenodo.20080043

TLS follows a modular architecture separating the computational engine from the graphical interface, facilitating maintenance, reproducibility, and integration into external scientific workflows.

8.2. Reuse Potential

Tunnel Load Simulator provides a generic and extensible framework for generating synthetic electrical-load datasets for road tunnels. The software can support machine-learning applications, forecasting and anomaly-detection studies, uncertainty quantification, sensitivity analysis, energy-performance assessment, and digital-twin development. Because the simulation engine is implemented as an independent Python module, it can be directly integrated into notebooks, optimization workflows, and larger scientific software ecosystems. Future developments will focus on calibration using measured tunnel data, integration of environmental variables, and more advanced traffic and ventilation models.

9. Conclusions

This paper presented Tunnel Load Simulator (TLS), an open-source software package for generating synthetic electrical-load profiles of road-tunnel infrastructures. The software combines traffic-driven demand modeling, lighting and ventilation load estimation, stochastic event generation, and Monte Carlo simulation within a computationally efficient and reproducible framework.

TLS generates realistic time-series datasets, key performance indicators, uncertainty envelopes, and seasonal demand analyses suitable for energy assessment, sensitivity analysis, machine-learning applications, and digital-twin development. Its modular architecture facilitates reuse, extension, and integration into external scientific workflows.

Although based on simplified first-order representations of tunnel operation, the software produces physically plausible demand profiles while remaining fully transparent and reproducible. Future developments will focus on calibration using measured tunnel data and the incorporation of more advanced operational and environmental models.

By providing a freely available and reusable framework, TLS contributes to addressing the scarcity of publicly accessible tunnel-energy datasets and supports future research on tunnel energy systems, infrastructure management, and data-driven decision-making.

Artificial Intelligence (AI) use statement

AI-based tools were used to assist in language editing, clarity improvement, and text condensation. All scientific content, results, and conclusions were generated and validated by the authors.

References

Crawley, D. B., Lawrie, L. K., Winkelmann, F. C., Buhl, W. F., Huang, Y. J., Pedersen, C. O., Strand, R. K., Liesen, R. J., Fisher, D. E., & Witte, M. J. (2001). EnergyPlus: creating a new-generation building energy simulation program. *Energy and Buildings*, *33*(4), 319–331.

El-Houari, H., & Voyant, C. (2026). Tunnel Load Simulator—SmartTunnel project. *Version 1.0.1. Zenodo. Https://Doi.Org/10.5281/Zenodo.20080043.*

Energy consumption | Road Tunnels Manual - PIARC (World Road Association). (n.d.). Retrieved April 27, 2026, from https://tunnelsmanual.piarc.org/en/strategy-and-general-design-sustainability-issues/energy-consumption

Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, *8*, 108952–108971.

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., & Smith, N. J. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362.

Jones, H. (2021). Sustainable infrastructure project evaluation. In *Industry, Innovation and Infrastructure* (pp. 1081–1093). Springer.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wießner, E. (2018). Microscopic traffic simulation using sumo. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2575–2582.

McKinney, W. (2010). Data structures for statistical computing in Python. *Scipy*, *445*(1), 51–56.

Menéndez, J., Sanabria, M., Fernández-Oro, J. M., Galdo-Vega, M., de Prado, L. Á., & Bernardo-Sánchez, A. (2024). Energy consumption and dilution of toxic gases in underground infrastructures: A case study in a railway tunnel under forced ventilation. *Energy*, *307*, 132810.

Nikolenko, S. I. (2021). *Synthetic data for deep learning* (Vol. 174). Springer.

Papageorgiou, M., Papamichail, I., Messmer, A., & Wang, Y. (2010). Traffic simulation with METANET. In *Fundamentals of traffic simulation* (pp. 399–430). Springer.

Peña-García, A., & Cabeza-Lainez, J. (2023). Daylighting of road tunnels through external ground-based light-pipes and complex reflective geometry. *Tunnelling and Underground Space Technology*, *131*, 104788.

PIARC; 2024. (n.d.). Retrieved June 9, 2026, from https://www.piarc.org/fr/PIARC-Association-Routes-Transport-Routier/plan-strategique

Riess, I. (2022). Improving the Energy Efficiency of Road Tunnels. *Tech. Report of Project AGT 2014/005, Appl. Res.*

Robert, C. P., Casella, G., & Casella, G. (2004). *Monte Carlo statistical methods* (Vol. 2). Springer.

W Axhausen, K., Horni, A., & Nagel, K. (2016). *The multi-agent transport simulation MATSim*. Ubiquity Press.

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., & Teal, T. K. (2017). Good enough practices in scientific computing. *PLoS Computational Biology*, *13*(6), e1005510.

Xu, R., Ye, H., Hu, B., Jin, S., Lu, M., Huang, Z., & Chen, L. (2024). Intelligent dimming control and energy consumption monitoring system of tunnel lighting. *Lighting Research & Technology*, *56*(1), 72–86.

Zhang, J. (2021). Modern Monte Carlo methods for efficient uncertainty quantification and propagation: A survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, *13*(5), e1539.