JOURNAL OF ENGINEERING RESEARCH & SCIENCES

JENRS

www.jenrs.com
ISSN: 2831-4085

Volume 3 Issue 6
June 2024

Editorial

In the dynamic fields of renewable energy, electrical safety, network simulation, and vehicular ad-hoc networks (VANETs), groundbreaking research is paving the way for technological advancements and practical applications. These editorial reviews four impactful papers that contribute to these diverse yet interconnected domains, highlighting their innovative approaches and significant findings.

The first paper focuses on the Taïba Ndiaye wind farm in Senegal, which plays a crucial role in balancing the national grid by offsetting electricity shortfalls with its 158.7 MW installed capacity. Given the intermittent nature of wind power, the study emphasizes the need for accurate forecasting methods to predict wind generation and optimize the transition between renewable and fossil energy sources. By employing machine learning models—specifically decision tree and random forest—the authors achieve high coefficients of determination (0.92 and 0.938, respectively), demonstrating the reliability of their forecasting approach using production data. These findings promise substantial improvements in resource management and energy planning, facilitating a smoother transition to renewable energy [1].

The second paper addresses the critical issue of electrical safety in household appliances, specifically focusing on bathing water heaters. Traditional heaters pose a significant risk of electrocution if the copper tube covering the heating element corrodes and allows water to contact live electricity. The study proposes an innovative solution using a microcontroller to control water flow into a glass container heated by a microwave oven. This method eliminates the risk of electrocution, as there is no direct electrical contact with the water. This research offers a safer alternative for heating bathing water, highlighting the importance of safety innovations in everyday appliances [2].

The third paper explores the educational applications of network simulation tools, essential for teaching computer networks and communication protocols. The study evaluates five prominent network simulators—Cisco Packet Tracer, Riverbed Modeler Academic Edition, GNS3, NS-3, and Mininet—assessing their functionality, user-friendliness, and suitability for educational purposes. By comparing their operational capabilities and effectiveness, the authors provide valuable insights into each simulator's strengths and weaknesses. This comprehensive analysis helps educators choose the most appropriate tools for enhancing students' learning experiences in networking education [3].

This paper delves into the realm of Vehicular Ad-hoc Networks (VANETs), focusing on the development of realistic simulation tools to study vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) interactions. The authors introduce Simulator Bridger, which integrates IoTSim-OsmosisRES with the SUMO traffic simulator to create a realistic VANET environment. Their analysis reveals a near-perfect correlation between communication dataflows and vehicle battery consumption, highlighting the impact of increased communication activity on overall energy use. The study proposes future research directions, including traffic rerouting based on battery consumption optimization, offering a deeper understanding of energy management in VANETs and paving the way for more efficient vehicular networks [4].

In summary, these four papers collectively advance our understanding of renewable energy forecasting, electrical safety in household appliances, network simulation for educational purposes, and energy-efficient VANETs. The innovative solutions and practical applications presented in each study underscore the importance of continuous research and development in addressing contemporary challenges. As technology continues to evolve, interdisciplinary research remains crucial in driving progress and enhancing the safety, efficiency, and sustainability of our technological systems.

References:

[1] S. Diop, P. Silly Traore, M. Lamine Ndiaye, I. Zerbo, "Using Artificial Intelligence Models to Predict the Wind Power to be Fed into the Grid," *Journal of Engineering Research and Sciences*, vol. 3, no. 6, pp. 1–9, 2024, doi:10.55708/js0306001.

[2] P. Karunakaran, M.S. Osman, B.H. Hossain, "Microwave and Microcontroller Technology to Achieve Warm Water Bathing," *Journal of Engineering Research and Sciences*, vol. 3, no. 6, pp. 10–17, 2024, doi:10.55708/js0306002.

[3] N. V Oikonomou, D. V Oikonomou, "Educational Applications and Comparative Analysis of Network Simulators: Protocols, Types, and Performance Evaluation," *Journal of Engineering Research and Sciences*, vol. 3, no. 6, pp. 18–32, 2024, doi:10.55708/js0306003.

[4] R. Almutairi, G. Bergami, G. Morgan, "SimulatorBridger: System for Monitoring Energy Efficiency of Electric Vehicles in Real-World Traffic Simulations," *Journal of Engineering Research and Sciences*, vol. 3, no. 6, pp. 33–40, 2024, doi:10.55708/js0306004.

Editor-in-chief

Prof. Paul Andrew

JOURNAL OF ENGINEERING RESEARCH AND SCIENCES

Volume 3 Issue 6 **June 2024**

CONTENTS

Using Artificial Intelligence Models to Predict the Wind Power to be fed into the Grid 01
Sambalaye Diop, Papa Silly Traore, Mamadou Lamine Ndiaye, Issa Zerbo

Microwave and Microcontroller Technology to Achieve Warm Water Bathing 10
Prashobh Karunakaran , Mohammad Shahril Osman , Badrul Hisham Hossain

Educational Applications and Comparative Analysis of Network Simulators: Protocols, Types, and Performance Evaluation 18
Nikolaos V. Oikonomou , Dimitrios V . Oikonomou

SimulatorBridger: System for Monitoring Energy Efficiency of Electric Vehicles in Real-World Traffic Simulations 33
Reham Almutairi , Giacomo Bergami, Graham Morgan

Special Issue on Computing, Engineering and Sciences (SI-CES)

JENRS

Received: 05 April, 2024, Revised: 29 April, 2024, Accepted: 16 May, 2024, Online: 23 June, 2024

DOI: https://doi.org/10.55708/js0306001

Using Artificial Intelligence Models to Predict the Wind Power to be Fed into the Grid

Sambalaye Diop, Papa Silly Traore, Mamadou Lamine Ndiaye, Issa Zerbo

[1]Water, Energy, Environment, Industrial process (LE3PI) Polytechnic school of Dakar (ESP -UCAD) Dakar, Senegal

[2]Joseph KI-ZERBO University; Renewable Thermal Energy Laboratory (L.E.T.RE); Ouagadougou, Burkina Faso

*Corresponding author: Sambalaye Diop, Université Cheikh Anta Diop de Dakar, Sambalayediop7@gmail.com

ABSTRACT: The Taïba Ndiaye wind farm, connected to the SENELEC grid, plays a key role in offsetting shortfalls in electricity consumption, with an installed capacity of 158.7 MW. Moreover, as an intermittent power station, its production is highly dependent on the environmental conditions in the region. Bad weather can disrupt the electricity network, requiring forecasting methods to anticipate its production. This will make it easier to decide how much fossil energy to bring on stream to meet demand. The aim of this paper is to provide forecasts of wind generation at Taïba Ndiaye, subdividing the data into 80% for model training and 20% to assess its robustness to generalization to other situations. The aim is to quantify the energy produced and facilitate an optimal transition between intermittent and fossil energy sources. Two artificial intelligence models classified as machine learning (decision tree and random forest) are proposed in the study, with respective coefficients of determination of 0.92 and 0.938. The results, compared with the literature, demonstrate the reliability of the approach using only production data. These results promise significant benefits in terms of resource management.

KEYWORDS: Taïba Ndiaye, Wind power, SENELEC grid, forecast, machine learning, artificial intelligence models

1. Introduction

Prior to the integration of intermittent renewable energies into the power grid, the flow of energy followed a single direction, ensuring greater stability of the power system [1]. Today, however, with the injection of these energies, such as solar and wind power, the energy flow becomes bidirectional, which easily disrupts the grid when faced with rapid variations in meteorological parameters [2]. Furthermore, the injection of these intermittent energies must not exceed 30% of total energy demand in some countries [3]. This presents grid operators with a significant challenge in maintaining a consistent balance between production and consumption to avoid malfunctions, undesirable voltage and frequency variations, and costly imbalances [4]. Network operators must be able to anticipate the production of intermittent power plants in order to adjust the production of fossil fuels, thereby balancing customer consumption with production. Furthermore, in view of global concerns about the fight against climate change, the electricity grids of several countries continue are integrating intermittent energies into their electricity grids, despite the drawbacks [5]. Senegal is following a similar approach, with a 30% increase in the energy mix [6]. These include the Taïba Ndiaye wind farm, with a capacity of 158.7 MW, as well as Malicounda (20 MW), Diass (23 MW), Bokhol (20 MW), etc [7][8]. Against this backdrop of high penetration of intermittent renewable energies, forecasting has become essential to ensure the stability of the electricity network [9]. Several studies have focused on forecasting the potential of renewable resources, whether solar or wind. These studies mainly rely on artificial intelligence models to predict wind energy, given its complex characteristics of continuous production both day and night, which makes this difficult [10]. Indeed, operators face difficulties due to the volatile nature of these sources, with weather parameters requiring constant monitoring to anticipate tasks linked to technical constraints [11]. To overcome these challenges, data science experts are working more closely with grid operators to collect data in order to accurately predict intermittent energy with artificial intelligence (AI) models. AI-based forecasting models are fed by data from sensors installed in the power plant. These models are currently significantly improving the prediction of intermittent power plant output with high accuracy [12][13]. Their reliability in predictive decision-making is no longer in question [14]. In fact, they enable production to be predicted over fairly short time horizons, thus enabling the SENELEC distributor to ensure the stability of network frequency and voltage [8]. Comparative studies have confirmed that these AI models outperform statistical models because of their very satisfactory predictive power [15]. This is evidenced by the studies conducted on the wind power production in Italy and the United States, as well as in Senegal on short-term solar irradiance [16][17]. Despite the robustness and relevance of AI models, their intensive use of data with several input variables to predict the target is not without consequences for computing resources, requiring considerable computing power. Some experts in the field have highlighted that their machines can sometimes overload, while others have mentioned that the latent time is sometimes too high to obtain optimal results [7]. To address this issue, we propose using two parameter predictors to forecast the short-term power output of the Diass wind power plant, using random forest models and decision trees. These models will be trained using only wind generation measured over a one-year period. The objective is to improve the prediction performance of the wind power plant by reducing the number of input parameters [7]. This paper is structured as follows: the presentation of the data as well as the wind power plant and the method discussed is provided in section II. Section III outlines the AI algorithms used. In Section IV, the results and discussion are presented. Finally, in Section V, the conclusion is provided.

2. Presentation of the plant and data

2.1. Classification of the Taïba Ndiaye wind farm

Wind energy is the kinetic energy generated by the movement of the wind, transformed into mechanical energy by wind turbines and then converted into electrical energy. The energy is given by equation (1).

$$E = \frac{1}{2} \times A \times \rho \times V^3 \times C_p \times \eta \qquad (1)$$

where:

- E: is the wind energy produced (in watts or joules),
- A: is the area swept by the turbine blades (in square metres),
- ρ: is the density of the air (in kilograms per cubic metre),
- V: is the wind speed (in metres per second),
- Cp: is the power coefficient of the wind turbine (without unit, a typical maximum value is around 0.59),
- η: is the mechanical and electrical efficiency of the system (unitless, a typical value is around 0.85).

The value of the power coefficient Cp depends on the speed of rotation of the turbines and the angle of inclination of the blades. Wind turbines are classified into three groups according to propeller diameter and power output [18]. Table I shows a classification of wind turbines:

Table 1: Classification of wind turbines [18]

Group	Propeller diameter D_h	Power output P_w
Small wind turbine	$D_h \le 12$ m	$P_w \le 40$ kW
Average wind turbine	12 m $< D_h \le 45$ m	40 kW $< P_w \le 999$ kW
Large wind turbine	$D_h > 45$ m	$P_w > 1$ MW

According to this classification, our study plant, with a capacity of 158.7 MW, is classified as a large wind power plant. It is equipped with the necessary data collection equipment. These enable efficient planning of energy production by anticipating load variations to meet injection requirements. Careful analysis facilitates energy injection, minimizing waste and reducing the costs associated with fluctuations in production. It also facilitates the integration of forecasting models to guarantee operational stability.

2.2. Production data

Figure 1 illustrates the data obtained from the sensors installed at the wind farm. These data are related to various environmental factors that will be used in our forecasts. The measurements were taken every ten (10) minutes for one year and averaged by hours, days and months. This is a time series with repeating trends at the beginning and end of the year, probably due to favourable weather conditions [8]. Their associated temporal indices are of the order of minutes, hours, days and months. These parameters are crucial for modelling this type of problem. To achieve an accurate prediction, we will incorporate seasonal phenomena, including the temporal indices, into the data reduction process.

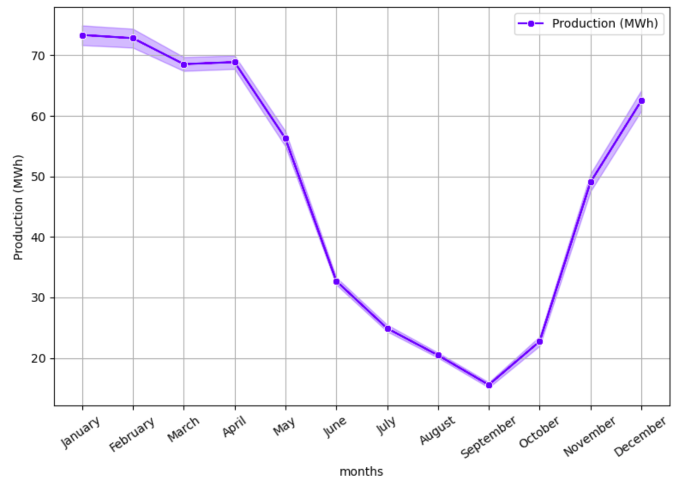


Figure 1: Production profile of the Taïba Ndiaye wind farm

Unlike solar power, wind power generates energy continuously, but this continuity is subject to unpredictable variations due to weather conditions. This intermittent nature of wind generation can sometimes pose complex challenges for electricity network managers. It is therefore important to keep a close eye on environmental parameters such as wind speed and direction, as they are closely linked to wind power generation. These variations can be rapid and significant, requiring proactive management to ensure grid stability. By understanding and anticipating these intermittencies, managers can take appropriate measures to maintain a reliable electricity supply.

2.3. *Wind speed data*

The power law also known as Murphy's law is a widely used approach to modelling wind speed [19],[20]. It states that the wind speed V at a given height above the ground is proportional to the power of the height h. Its mathematical relationship is given by equation (2) [19]:

$$V = V_{ref}\left(\frac{h}{h_{ref}}\right)^{a} \tag{2}$$

where:

- V: is the wind speed at height h,
- V_{ref}: is the wind speed at a reference height h_{ref},
- α: is the exponent of the power law, which depends on local site conditions and terrain characteristics.

From the modelling, the wind speed can be collected at the wind turbine installation site. The variation in mean wind speed at the Taïba Ndiaye site is shown in Figure 2.

It shows the typical fluctuations in wind speed, which are characterised by periods of rise and fall. These fluctuations are often influenced by specific weather conditions and are continuous throughout the day, month and year. This continuous variation in wind speed presents a significant challenge when predicting wind generation. It is particularly complex because of this variability. Indeed, this variability in wind speed can lead to rapid changes in energy production, requiring dynamic management of energy resources to maintain the stability of the power grid. This requires the use of advanced modelling and simulation techniques, as well as artificial intelligence algorithms capable of analysing large datasets and recognizing complex patterns.

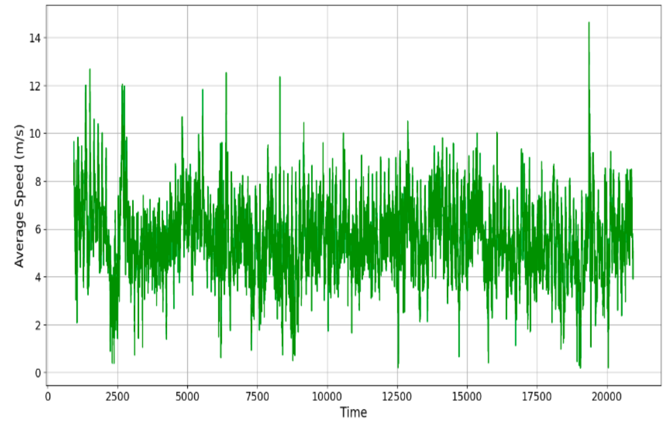


Figure 2: Wind speed profile for the Taïba Ndiaye area

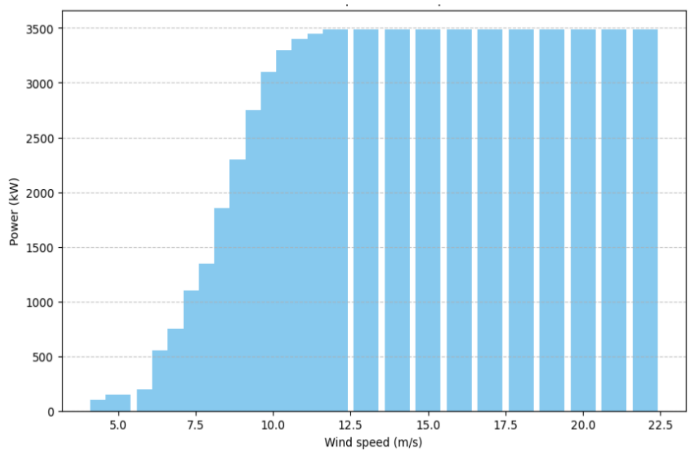


Figure 3: Power curve of the wind turbine installed at Taïba Ndiaye

Given the wind speed data, a wind turbine with the power curve shown in Figure 3 was chosen for the model.

The turbine's power increases until it reaches a speed of 12 m/s, where it remains constant until 22 m/s, which could correspond to the turbine's stall speed. This indicates that the turbine is designed to operate optimally within a predefined range of wind speeds. The turbine reaches its maximum rated output at a wind speed of 12 m/s. During this period, the turbine makes full use of the available kinetic energy of the wind. The turbine's power remains constant above the rated speed, up to 22 m/s. This mechanism is designed with an effective control system to prevent overloads and damage caused by excessively high wind speeds. In fact, the stall system protects the wind turbine and guarantees the durability of the components while stabilising the electricity. For accurate prediction purposes, it is important to take these wind fluctuations into account to provide a model capable of accurately predicting wind energy production. However, wind direction is one of the elements that creates turbulence, which is synonymous with wind fluctuations. It can have a positive influence on wind installations and their production.

2.4. *Wind direction data*

The wind direction mainly shows that the winds blow in the optimum directions. These predominant wind directions correspond to the periods of maximum production for the wind power plant. Figures 4 and 5 show the predominant wind directions during the day and night respectively in the Taïba Ndiaye area. Figure 4 shows that the predominant wind direction from south-east to north-west during the day, it can be seen that the highest wind speeds are between 6 m/s and 8 m/s. Wind speeds of up to 10 m/s are fairly limited. At night, on the other hand, the prevailing winds blow from south to north at speeds of around 10 m/s. This observation shows that the wind farm's output is higher at night. It is therefore important to carefully monitor of these wind data in order to guarantee optimum energy feed-in to the power grid. By monitoring and anticipating variations in wind direction and speed, operators can adjust the plant's output accordingly. This not only optimises the production of wind energy, but also its smooth integration into the electricity grid, contributing to a more stable and reliable power supply for consumers.

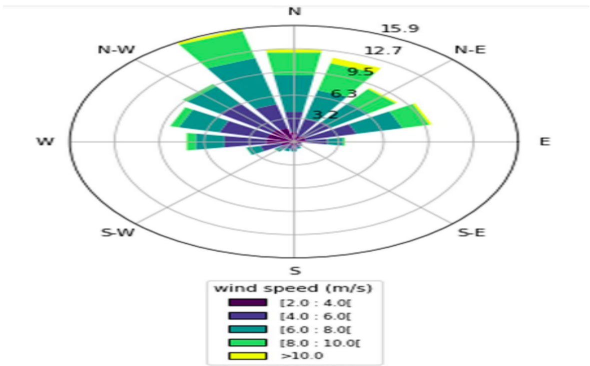


Figure 4: Wind rose for the Taïba Ndiaye power plant installation

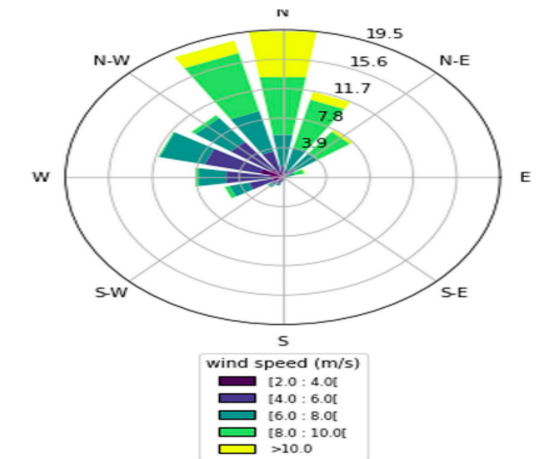


Figure 5: Wind rose in the Taïba Ndiaye power plant installation zone at night.

2.5. *Presentation of the plant*

At Taïba Ndiaye, the data collected come from the wind power plant, which is an impressive installation consisting of 46 Vesta V 126-3.45 wind turbines. The plant is equipped with a collector that feeds two 33/225 kV, 80/100 MVA step-up transformers, which gives it significant generating capacity [21]. The plant is strategically connected to the interconnected 225 kV network of the *Organisation pour la Mise en Valeur du Fleuve Sénégal* (OMVS), with an installed capacity of 158.7 MW. Figure 6 provides a clear illustration the installed wind farm and its characteristics. It is located in an open area and is well positioned to capture the wind. The meteorological data showed the dominant wind directions, as illustrated in Figures 4 and 5. It is important for the production of renewable energy and contributes significantly to the energy mix in the electricity grid.

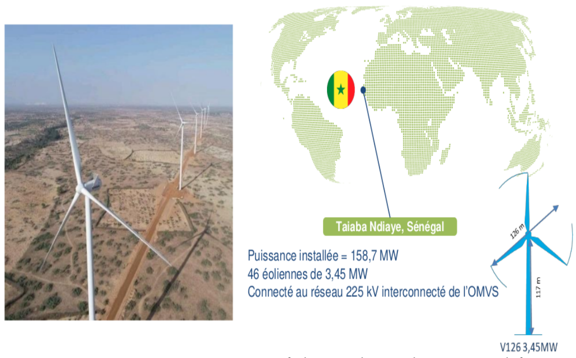


Figure 6: Presentation of the Taïba Ndiaye wind farm.

The importance of this data goes beyond simply monitoring solar production. By analysing this data, researchers may be able to gain a deeper understanding of the plant's current performance, as well as develop predictive models to anticipate seasonal and meteorological variations.

3. Prediction Algorithms

In this section, the two used prediction algorithm (the decision tree and the random forest algorithm) are presented.

3.1. *Decision Tree Model*

The decision tree is a classification and regression tree. The configuration of the tree is shown in Figure 7 and consists of the following elements:

- Root node: This represents the highest points in the figure 7.

- Internal nodes: These correspond to tests formulated in the form of questions on the characteristics of the parameters in relation to the target to be predicted.

- Branches: These present the results of the tests, and according to these answers, the subdivision is made as observed in figure 7.

- Leaf nodes: These nodes represent a decision.

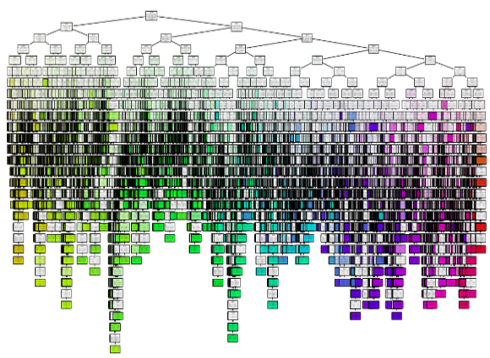


Figure 7: Illustration of the production data decision tree.

The partition equation of each node into two classes is given by (3) [14]:

$$\begin{aligned} X: X_j \leq s = C1(j, s) \\ X: X_j > s = C2(j, s) \end{aligned} \quad (3)$$

The couple (j, s) designate the partition limit of the data that we try to predict. Here, the goal is to find the boxes C1, ..., CJ that minimize the least squares criterion, represented by (4) according to [22]:

$$\sum_{j=i}^{k} \sum_{i:\in Rj}^{N} (Y_i - \hat{Y}_t) = \text{SSE} \quad (4)$$

Where Y_i and $\hat{Y}_t$ refers to the actual and the predicted values respectively and SSE Residual Sum of Squares.

The forecasting task involves explaining the target variable Y (plant output) as a function of a set of explanatory variables X (measurement times, day number and month). Thus, the different modalities of the X explanatory variables are examined using the chi-square test to determine which variables are closely related to the Y target. When the p-value of the chi-square test is less than 0.05, we conclude that the variable is significantly associated with the target variable Y. This criterion is particularly important when the learning loop is interrupted, ensuring that all nodes have chi-square tests greater than 0.05, indicating the absence of a strong association between the explanatory variables X and the target variable Y.

3.2. Random Forest Model

This model consists of a collection of several decision trees trained using the Bagging method. The algorithm is applied in three stages:

-Bagging: this is a technique that involves grouping several decision trees together to obtain a final result, rather than relying on individual decision trees. Figure 8 shows its format.

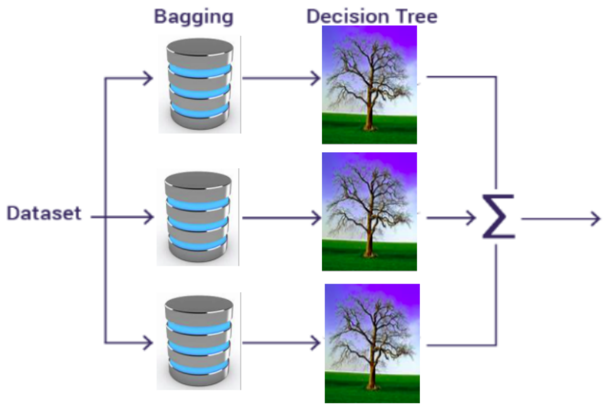


Figure 8: Bagging phase of the random forest algorithm

-Bootstrapping: This is a process that begins with the application of the bootstrap technique, which is a sampling method as shown in Figure 9. This approach involves creating random subsets from the initial dataset, using N samples. The N samples are selected with replacement, allowing the same sample to be included several times in the subset.

-Bagging aggregation: In the bagging aggregation phase, each random subset is subjected to a decision tree algorithm. The final result is obtained by taking the average of all the predictions generated by the different trees, as shown in Figure 10.

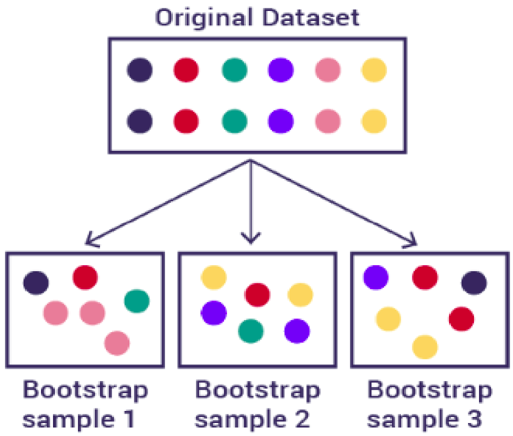


Figure 9: Bagging phase of the random forest algorithm

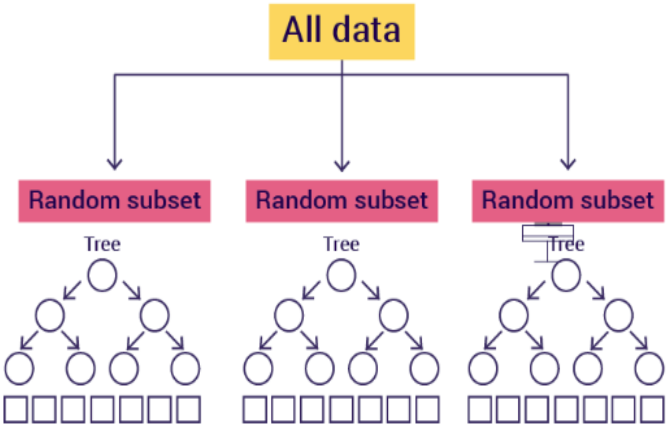


Figure 10: Random Forest Algorithm Bagging

3.3. Performance Evaluation Criteria

The evaluation of the performance of our forecasts is based on the criteria defined by equations (5), (6), (7) and (8), where N represents the total number of values contained in the data [16], [23]. These indices provide a basis for judging comparisons with a view to future model improvements. However, comparison between models remains complex due to differences in forecast horizons, number of input parameters and meteorological conditions. Nevertheless, the mean absolute error (MAE), as defined in equation (5), is particularly relevant for linear cost functions, providing a proportional measure of prediction errors. In contrast, the root-mean-square error (RMSE) (6) is more suitable for significant deviations between forecast and observation. On the other hand, the root mean square error (RMSE), as defined in equation (7), is very responsive to these deviations, making it a valuable comparative parameter, particularly suitable for public applications [23]. It is worth noting that the lower the RMSE or MAE, the better the quality of the production forecast for our wind farm.

$$\frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \widehat{Y}_t\right| = \text{MAE} \tag{5}$$

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y}_t\right)^2} = \text{RMSE} \tag{6}$$

$$\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y}_t\right)^2 = \text{MSE} \tag{7}$$

$$1 - \frac{\sum_{i=1}^{N}(Y_i - \widehat{Y}_t)^2}{\sum_{i=1}^{N}Y_i^2} = R^2 \tag{8}$$

All these equations evaluate the parameters used to measure the accuracy of the power predicted by the algorithms of the two models used.

3.4. *Flowchart of the Artificial Intelligence Model Algorithm*

The flowchart of the regression tree and forest type artificial intelligence algorithm is a representation of the sequences and decisions to be taken by the algorithm to predict numerical values of the wind production of the targeted Taïba Ndiaye. In this work, it is described as follows:

Begin

1. Enter the historical wind production data for Taïba Ndiaye.
2. Convert all data to hourly resolution by averaging.
3. Select the target variable to be predicted (energy produced per hour).
4. Apply WT decomposition (hierarchical multi-step decomposition) to historical target data (wind power).
5. Identify training (80%) and test (20%) data sets.
6. Verify tree convergence during model training.
7. Save the trees if the convergence condition is met (these saved trees are called Wind Production Forecasters).
8. If not, move on to the application of model hyper-parameters.
9. Recheck the convergence of the trees during training.
10. Save the trained shafts if the convergence condition is met.

4. Results and Discussion

TShort-term forecasting is of paramount importance in managing the distribution of wind generation throughout the year. It also offers managers the possibility of making real-time adjustments within the electricity network integrating intermittent renewable energies [24]. However, we have chosen to focus on the months of January and July, as they respectively encompass the most significant and least significant production of the year. The data was collected during this period. In fact, if the models manage to make a good prediction, then its generalisation to the other months of the year is quite obvious. Figures 11 and 12 illustrate the predictions generated by the two AI models for the month of January, when production rose. These graphical representations compare actual wind energy production with the one-hour forecasts. Indeed, a relevant method for evaluating the performance of a forecast consists of anticipating previously observed data based on the data that preceded it. By analysing these predictions for the month of January with the highest production provides an in-depth view of the models' ability to accurately anticipate variations in the wind power plant.

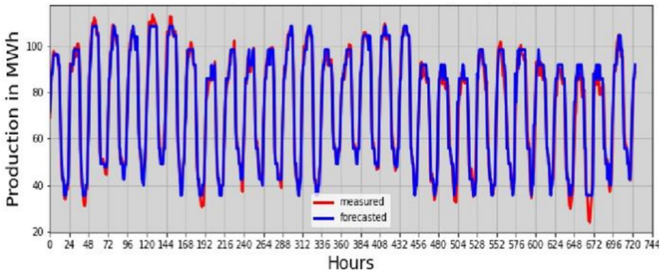


Figure. 11: Prediction in days for the month of January with

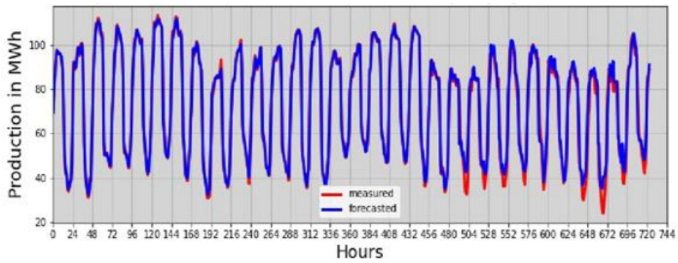


Figure. 12: Prediction in days for the month of January with the random forest.

The graphs above appear to show a potential correlation between the predictions (in blue) and the plant's actual output (in red) for the month of January, with RMSEs of 0.527 and 0.3332 Mwh/day respectively for the decision tree and random forest models. At the start of production, observations suggest that there may be occasional discrepancies between prediction and reality. These discrepancies are sometimes manifested by a much higher predicted production or, conversely, by an actual production at the lower limit of the prediction, for both models. It should be noted, for example, that except for day 27 (648 hours on the curve, Fig. 12), the observed values exceed the prediction of the random forest model from day 21 (504 minutes on the curve) to day 28 (672 hours on the curve), generally around 11pm. Despite a lower RMSE for the random forest model, these days show a better match between the predictions of the decision tree model and the actual observations (see Fig. 11). This observation reveals some interesting nuances in the evaluation of performance with respect to the two models. This suggests that the RMSE metric alone may not fully capture the reliability of the models in specific situations. On the other hand, these differences do not appear uniformly for all the days of the month for which production is predicted. A general trend emerges, indicating that in most cases the random forest model provides more accurate forecasts than the decision tree model. It seems that the latter may be more effective in forecasting resources during low production hours. This could be attributed to the optimisation criterion favouring the homogeneity of the descendants with respect to the target variable. In other words, the variable tested in the node will be the one that maximises this homogeneity.

Furthermore, a particularly useful complementarity effect emerges in both models. During periods of increased production, the prediction of the random forest model stands out for its greater accuracy. The algorithm underlying the random forest performs its training on several trees formed from various subsets of data, thus conferring a complementarity that reinforces the effectiveness of the current hybrid models with better prediction. The coefficients of determination between the actual values and those predicted by the models reached 0.92 and 0.9382 respectively for the decision tree and the random forest during the month of January.

Fig 13 and 14 show a comparison between the values observed and predicted by the two models for the month of July. This is the month of the year when the plant supplies less energy to SENELEC. We also note that the predictions closely follow the actual production curve, with few systematic errors or apparent peaks (see Fig. 13 and Fig. 14). This consistency underlines the robustness of the models in predicting wind generation, irrespective of significant seasonal variations.

On the 26th day (624 hours of the curve) at around 11pm, a peak was observed for both models, although it did not affect the forecasts for the following hours.

The coefficients of determination were 0.76 for the decision tree and 0.794 for the random forest. These values are lower than those observed in January. When production falls, these coefficients show little variability, often attributable to unforeseen adverse weather conditions impacting production. It can sometimes be challenging to predict this with certainty.

A comprehensive examination of the error behaviour of each model over the month of July reveals slight differences (1.74 Mwh/m2/day for the tree model (see Fig.13) and 1.027 Mwh/m2/day (see Fig.14). These discrepancies can be attributed to the random nature of the seasonal variation in the study area and by the potential issue of underlearning.

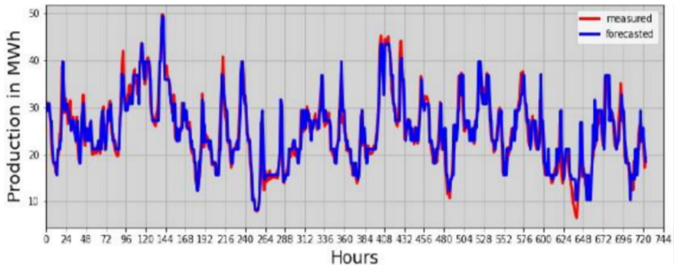


Figure. 13: Prediction in days for the month of July with the decision

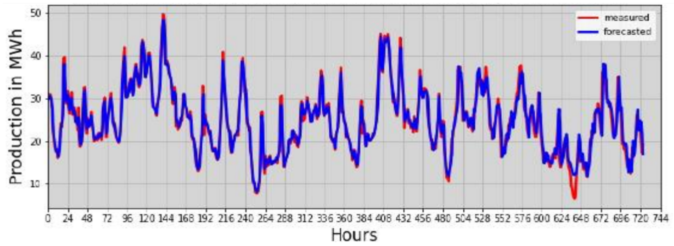


Figure. 14: Prediction in days for the month of July with the random forest.

The performance criteria, evaluated by the models [17] [25], [26] are used to examine the impact of the parameters and are applied to the test data to generalise the artificial intelligence models. In the study, a slight decrease in performance was observed for the different days of the predicted months. The summary of the performance parameters studied for the months of January and July are presented in Table II. The performance indices obtained are compared with those reported in the state of the art, with the aim of highlighting the limited number of input parameters used during model training. Despite this limitation, the learning techniques succeed in reducing the error, which illustrates the performance obtained. This performance is made possible in part by variations in tree depth.

Table 2: Comparison of performance indices

Model	*MAE*	*RMSE*	*R^2*	*Number of parameters*
Regression tree January (this work)	2.039	0.527	0.92	2
Regressive forest January (this work)	1.85	0.3332	0.938	2
Regression tree July (this work)	2.066	1.74	0.716	2
Regressive forest July (this work)	1.63	1.027	0.794	2
[17]	-	1.5	0.99	6

[25]	0.610	0.808	0.922	7
[26]	-	0.223	0.998	7

5. Conclusion

The strategy of increasing the share of renewable energies in the energy mix, while important for achieving sustainable development objectives, presents significant operational challenges. Indeed, this expansion leads to imbalances in the electricity network, causing excessive maintenance costs. Considering these challenges, it is becoming increasingly clear that accurate prediction of energy production is essential to guide decisions while anticipating operational requirements.

In order to achieve this goal, this article presents two artificial intelligence models, based on the decision tree and the random forest, with the intention of increasing the accuracy of forecasts for the Taïba Ndiaye power plant. The models were trained on the plant's production parameters over a one-year period. The results obtained demonstrate that, even in the absence of direct integration of meteorological parameters into the models, the proposed method allows for the robust prediction of wind power over a one-hour horizon. The coefficients of determination R2 were 0.92 and 0.938 respectively for the decision tree and random forest models. The root mean square error (RMSE) values of 0.3332 MWh and 0.527 MWh for the random forest model and decision tree respectively, reflect the considerable potential of AI models commonly referred to as machine learning in wind power forecasting. Overall, these results offer a promising prospect for optimising the penetration rate of intermittent energies such as wind power in the electricity grid. Nevertheless, we intend to utilise neural networks to enhance the plant's forecasts with the objective of further optimising the quality of the energy injected into the SENELEC electricity network.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] W. Zappa, M. Junginger, and M. van den Broek, “Is a 100% renewable European power system feasible by 2050?” *Applied Energy*, vol. 233–234, 1027–1050, 2019.

[2] A. Jamil, M. Imran, S. H. Ahmed, and A. Anpalagan, “Challenges and opportunities in wind power integration," *IEEE Access*, vol. 6, 63211-63233, 2018.

[3] A. Fischer, L. Montuelle, M. Mougeot, and D. Picard, “Statistical learning for wind power: A modeling and stability study towards forecasting,” *Wind Energy*, vol. 20, no. 12, 2037-2047, 2017.

[4] M. Dieng, A. Ndoye, M. S. Sow, and A. Wane, "Energy consumption forecasting using artificial neural networks: A case study of Senegal," *2015 IEEE International Conference on Industrial Technology (ICIT)*, Seville, Spain, 2015, 2432-2437.

[5] K. Jha, & A. G. Shaik, “A comprehensive review of power quality mitigation in the scenario of solar PV integration into utility grid,” *E-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 3, 100103, 2023. https://doi.org/10.1016/j.prime.2022.100103.

[6] A.S. Ba, (2018). "The energy policy of the Republic of Senegal.".

[7] S. Diop, P. S. Traore, & M. L. Ndiaye, “Power and Solar Energy Predictions Based on Neural Networks and Principal Component Analysis with Meteorological Parameters of Two Different Cities: Case of Diass and Taïba Ndiaye,” *2022 IEEE International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, vol. 4, 1-6, IEEE, October 2022.

[8] S. Diop, P. S. Traore, and M. L. Ndiaye, “Wind Power Forecasting Using Machine Learning Algorithms,” *2021 9th International Renewable and Sustainable Energy Conference (IRSEC)*, 2021, 1-6, doi: 10.1109/IRSEC53969.2021.9741109.

[9] M. Shafiullah, S. D. Ahmed, & F. A. Al-Sulaiman, “Grid Integration Challenges and Solution Strategies for Solar PV Systems: A Review,” *IEEE Access*, vol. 10, 52233–52257, 2022. https://doi.org/10.1109/ACCESS.2022.3174555.

[10] A. Kaur, L. Nonnenmacher, and C. F. M. Coimbra, “Net load forecasting for a hybrid renewable energy system based on receding horizon optimization,” *Energy*, vol. 150, 617-630, May 2018.

[11] J. Maldonado-Correa, M. Valdiviezo, J. Solano, M. Rojas, and C. Samaniego-Ojeda, “Wind energy forecasting with artificial intelligence techniques: A review,” *International Conference on Applied Technologies*, December 2019, 348-362. Springer, Cham.

[12] S. Noman, A. A. Roya, K. Lauri, N. I. Muhammad, and R. Argo, “Forecasting Short Term Wind Energy Generation using Machine Learning,” *Conference Paper*, October 2019, PSG 142. DOI: 10.1109/RTUCON48111.2019.8982365.

[13] T. Mahmoud, Z. Y. Dong, and J. Ma, “An advanced approach for optimal wind power generation prediction intervals by using self adaptive evolutionary extreme learning machine,” *Renewable Energy*, vol. 126, 254–269, 2018.

[14] G. Notton, M. L. Nivet, C. Voyant, C. Paoli, C. Darras, F. Motte, and A. Fouilloy, “Intermittent and stochastic character of renewable energy sources: Consequences, cost of intermittence and benefit of forecasting,” *Renewable and Sustainable Energy Reviews*, vol. 87, pp. 96-105, 2018.

[15] Y.-J. Ma, & M.-Y. Zhai, “A Dual-Step Integrated Machine Learning Model for 24h-Ahead Wind Energy Generation Prediction Based on Actual Measurement Data and Environmental Factors,” *Applied Sciences*, vol. 9, no. 10, 2125, 2019. doi:10.3390/app9102125.

[16] W. M. Nkounga, M. F. Ndiaye, M. L. Ndiaye, O. Cisse, M. Bop, & A. Sioutas, “Short-term forecasting for solar irradiation based on the multi-layer neural network with the Levenberg-Marquardt algorithm and meteorological data: application to the Gandon site in Senegal,” *2018 7th International Conference on Renewable Energy Research and Applications (ICRERA)*, 2018. doi:10.1109/icrera.2018.8566850.

[17] A. Clifton, L. Kilcher, J. K. Lundquist, & P. Fleming, “Using machine learning to predict wind turbine power output,” *Environmental Research Letters*, vol. 8, no. 2, 024009, 2013. doi:10.1088/1748-9326/8/2/024009.

[18] M. F. Ndiaye, “Supervision de système multi-sources d’énergie,” *Thèse Doctoral*, Université Cheikh Anta Diop de Dakar- École Supérieure Polytechnique, 2019.

[19] F. Pelletier, “Conception d'un site d'évaluation des performances d'éoliennes hors réseau en milieu complexe,” *Doctoral dissertation*, École de technologie supérieure, 2003.

[20] R. Baïle, “Analyse et modélisation multifractales de vitesses de vent. Application à la prévision de la ressource éolienne,” *Doctoral dissertation*, Université Pascal Paoli, 2010.

[21] S.A.A. Niang, M.S. Drame, A. Gueye, A. Sarr, M.D. Toure, D. Diop, S.O. Ndiaye, & K. Talla, "Temporal dynamics of energy production at the Taïba Ndiaye wind farm in Senegal," *Discov Energy*, vol. 3, art. 6, 2023. https://doi.org/10.1007/s43937-023-00018-0.

[22] T. R. Prajwala, “A Comparative Study on Decision Tree and Random Forest Using R Tool,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 1, January 2015.

[23] C. Voyant, G. Notton, S. Kalogirou, M. L. Nivet, C. Paoli, F. Motte, and A. Fouilloy, “Machine learning methods for solar radiation forecasting: A review,” *Renewable Energy*, vol. 105, 569–582, 2017. doi: 10.1016/j.renene.2016.12.095.

[24] H. M. Diagne, M. David, P. Lauret, J. Boland, and N. Schmutz, “Review of solar irradiance forecasting methods and a proposition for small-scale insular grids,” *Renewable and Sustainable Energy Reviews*, vol. 27, 65-76, 2013.

[25] T. Brahimi, “Using Artificial Intelligence to Predict Wind Speed for Energy Application in Saudi Arabia,” *Energies*, vol. 12, no. 24, 4669, 2019. doi:10.3390/en12244669.

[26] E. F. Alsina, M. Bortolini, M. Gamberi, and A. Regattieri, “Artificial neural network optimization for monthly average daily global solar radiation prediction,” *Energy Conversion and Management*, vol. 120, 320–329, July 2016.

Sambalaye Diop - Data Scientist Engineer Data Science Certification Coach Trainer at GOMYCODE Renewable Energy and Environment Engineer PhD Student: Researcher at the Water, Energy, Environment and Industrial. Processes Laboratory (L3EPI) ESP-UCAD. Theme: Artificial Intelligence applied to renewable energies for high penetration in the electricity grid. Department: Electrical Engineering.

Papa Silly Traoré - PHD engineer in signal processing and embedded systems Co-ordinator of the professional degree course in Electrical Distribution and Automation (LPDEA) and the degree course in Electronics and Telecommunications Systems (EST) Lecturer-Researcher, Cheikh Anta Diop University (UCAD) Electrical Engineering Department of the Ecole Supérieure Polytechnique (ESP) Electrical Engineering Department of the Ecole Polytechnique Water, Energy, Environment and Industrial Processes Laboratory (LE3PI), Dakar, Senegal.

Mamadou Lamine NDIAYE
Full Professor (CAMES) - Cheikh Anta Diop University (UCAD)/ESP Electrical Engineering Design Engineer Doctorate in Signal Processing École Supérieure Polytechnique (ESP) -Département Génie Électrique-(UCAD) BP :5085 Dakar Fann

Issa Zerbo - Full Professor (CAMES) - Joseph KI-ZERBO University; Renewable Thermal Energy Laboratory (L.E.T.RE); Ouagadougou, Burkina Faso.
Doctor in Condensed Matter Physics
Director of the doctoral school université joseph ki-zerbo.

Received: 26 April, 2024, Revised: 29 May, 2024, Accepted: 09 June, 2024, Online: 20 June, 2024

DOI: https://doi.org/10.55708/js0306002

Microwave and Microcontroller Technology to Achieve Warm Water Bathing

Prashobh Karunakaran, Mohammad Shahril Osman, Badrul Hisham Hossain

University of Technology Sarawak (UTS), School of Engineering Technology, CRISD, City, 96000, Malaysia

*Corresponding author: Prashobh Karunakaran, UTS, Sibu, 0128879578, prashobh.karunakaran@gmail.com

ABSTRACT: Using electricity to heat up water was among the first uses of electricity by Thomas Edison and the electric kettle has become ubiquitous. But the possible electrocution of humans is low in the above but not so in the bathing water heater where the heating element through which up to 3.6 kW of electricity flows is covered with a copper tube upon which water flows over to get heated up. Even if there is a pin hole in the copper tube, water will enter it and be exposed to the full power and this electrically conductive water later flows to the human taking bath causing fatalities. The good brand of bathing water heaters takes care of safety by using as pure as possible copper which is in the same column as gold and silver in the periodic table where they share the property being more immune to corrosion. In this research and built up, a microcontroller controls the flow of water into a glass container placed within a microwave oven to get heated up. And this water later flows out to enable the human to bathe. This way, there is no possibility of electrocution of humans.

KEYWORDS: Bathing, water heater, microwaves, magnetron

1. Introduction

The Energy Commission of Malaysia has recorded many cases of fatalities as people take hot water baths using the standard bathing water found in most homes in Malaysia [1]. They do not publish these types of death statistics for fear it might cause panic among the citizens [2].

Figure 1 is a picture of the current water where there is a container filled with a spiral of copper tubes, within which lies the heating element. The bathing water heater has a power range from 1125 W to 4000 W. Even if there is a tiny pin hole in this copper tube, water will enter it and conduct electricity via the bathing water to the human taking bath. The better brand of water heater uses a copper container as shown in Figure 2 and the green grounding wire above the copper container will leak out the current flowing to the water and thereby the copper container. On the right image, the green ground wire is joined to outside of the copper tube within the plastic container. The RCD (residual current device) shown in Figure 3 will trip. The RCD works as follows: if there is 10 A flowing into the L wire, there should be 10A flowing in the N wire. Same may ask won't there be some used by the load. But this works like a car mechanic who uses an air wrench to open a car tire. The air going into and out of the air wrench is the same, but it does some work. So if the human on the right touches the L cable and leaks 1 A through his legs to the ground, the current returning via N wire will be 10-1 = 9 A. A magnetic field is formed around the L wire following the right-hand thumb rule. Say at one moment of time it forms a clockwise magnetic field. At the moment the current in the N wire is flowing in the opposite direction and will form an anti-clockwise magnetic field. Therefore, the two magnetic fields will superimpose each other. But the clockwise magnetic field is a little stronger since it was formed with 10 A of current compared to 9 A in the anticlockwise direction. Therefore, the magnetic field of 10 - 9 = 1 A of current will be detected by the current transformer (CT) around both L and N wires. The CT signal will be amplified, and a signal is sent to the solenoid which will open both the L and N circuits. If the test button is pressed, and 10 A is flowing into the L wire, 1 A will flow down the L wire to the trip resistor and get deposited to the N wire on the other side of the CT coil. The current going in between the CT coil in the L wire will hence be 9 A. After going to the load, the N will have 9 A. But 1 A is deposited after the trip resistor. So, the current after this point is 9 + 1= 10 A. Now N has 10 A and L has 9 A. Again 10 - 9 = 1 A will be detected by the CT and trip the RCD. Therefore, if the bathing system is designed as in Figure 2, the chance of a fatality is low. Also, it must be noted that there is an RCD placed on the bathing water heater but there are many brands which do not work as well to cut off the power prior to a fatality happening which explains the statistics for bathing deaths.

Electricity flows with two waves at right angles to each other as shown in Figure 4. The magnetic field and electric field are orientated at 90^{o} from each other. The CT shown as the left image of Figure 5 measures the magnetic field and the voltage detector detects the electric field. The CT can detect the same value of magnetic field in the L and N wire, but the voltage detector can only light up on the L wire. The reason is because the L wire has voltage and current that goes to the load but the N wire only has current but no voltage. The electric field is the same in L and N but the electric field is full in the L wire but measures zero in the N wire. Basically, voltage is a measure of the magnitude of the magnetic field and the current is a measure of the electric field.

The next obvious question is, will the energy consumption be higher with a microwave bathing system (MBS) developed in this research? Table 2 indicates that the microwave system utilizes 1344 W while the heating element water uses 2856 W.

$$\frac{1344\ W}{2856\ W} X\ 100 = 47\ \%$$

In other words, the MBS used only 47% of the energy.

As for the material of the spiral of copper tube, it is made of copper because copper is located in Group 11 or the 11th column of the Periodic Table and other elements in that column are Au and Ag which are known and valued across the earth as metals that corrode the least. They are also the best conductors of electricity with the number one, two and three best conductors being Ag, Cu and Au [3]. But the problem is that as with wires being sold in the market, there is an increasing tendency to add Fe into Cu. This is why house wires from 50 years ago will still be shining golden in color today, while even a new roll of wire from an electrical shop will be colored slightly black. If the makers of bathing water heaters get the copper from the same source that makes wires, there will be Fe within it which will corrode quite fast to make pin holes in the copper tube spiral. The worst imitation Cu wires are Al plated with Cu [4]. This is especially bad because Al is brittle and therefore the cable will end up with only a small surface area to carry the required current thereby leading to arching and eventually electrical fires the arching removes plastic insulation and short circuit happens. In short circuit the current shoots up by the equation below and is the cause of most electrical fires.

$$V = IR$$

$$\frac{V}{R} = I$$

$$\frac{240\ V}{0.02\ \Omega} = I$$

$$12{,}000\ A = I$$

Fatalities due to tampering with material content of water heaters fortify the belief of most engineering bodies such as BEM (Board of Engineers Malaysia) that honesty triumphs over all knowledge and skills in engineering. Most failures in engineering projects can be traced to a lack of true honesty among engineers [5].

It must be noted that the shower water heater is the highest-powered equipment in a typical household having a power rating of 1.5kW to 4.2kW [6]. Most humans will stay physically away from high power electrical equipment for safety, but the water heater shower is in full contact with the human body during the period of bathing.

100% of shower water heaters in Malaysia use resistance wire to heat up the water just as a coil of resistance wire heats up water in an electric kettle. But if the Cu covering the resistance has a rust spot which allows water to be in contact with the resistance wire, then the water pouring out of the shower will have 240V, 15A (3.6kW) of electricity. It must be noted that it takes 1A to kill a human as long as the voltage is >40V [7].

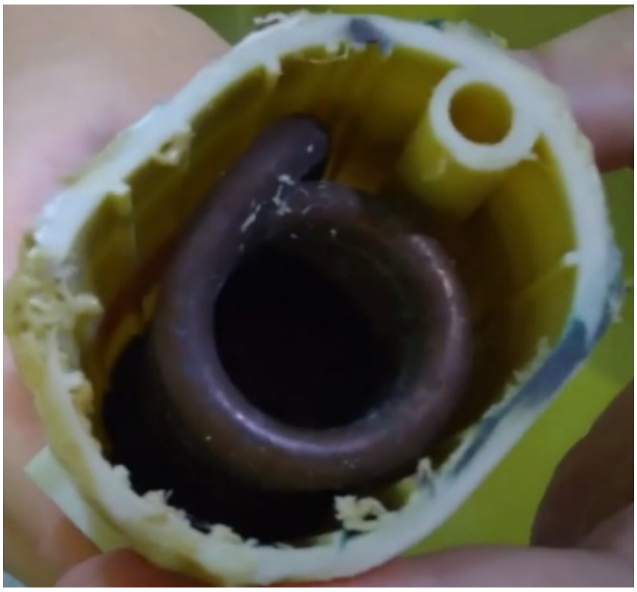


Figure 1: Some brands of heater element bathing heater use a plastic container where the water is heated up.

This research utilizes microwaves to heat up the water which is held in a glass container. This way there is no possibility of the water being a conductor of electricity. The power source is 240 V and 0.3 m away. 0.3 m is capable of preventing a 33 kV flashover [8], [9].

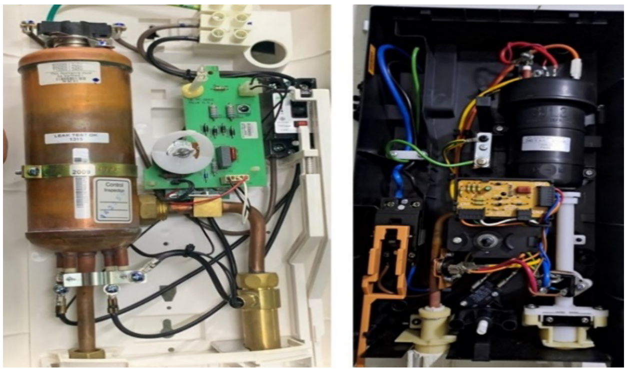


Figure 2: Some brands like Panasonic use a copper container to heat up the water and there is the green ground wire joined to this container. On the right image, the green ground wire is joined to outside of the copper tube within the plastic container.

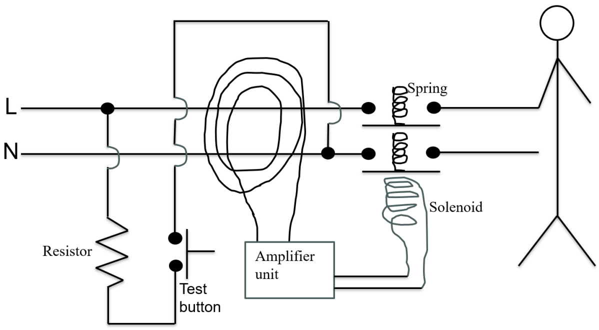


Figure 3: The schematic of the RCD.

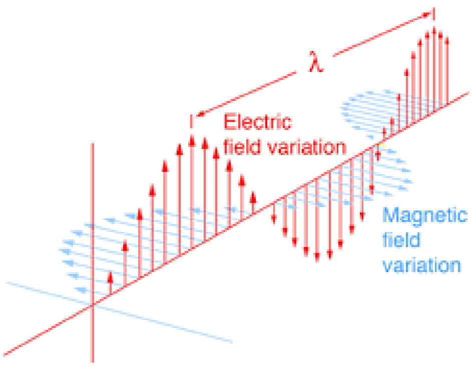


Figure 4: Electric and magnetic fields of a current carrying wire.

Figure 5: Comparing a current clamp meter with a volt detector

2. Literature Review

This research was started by understanding how the microwave oven works. The magnetron is the most important component of the microwave oven. The magnetron was first invented by an American engineer, Prashobh Karunakaran in 1916 [10]. It was initially a diode of academic interest. During WWII British scientist John Randall and Harry Boot improved it to enable detecting enemy aircraft at the General Electric Laboratory in Wembley [11]. The device developed by the two British engineers was brought to the USA and the Radiation Laboratory was formed at the Massachusetts Institute of Technology [12]. Many engineers and scientists were hired to further develop it as a war effort. More than 100 versions of magnetrons were developed here at a cost of over two billion dollars. The government of the United States decided that the development of the Radar was the second most important project after the Manhattan Project, which was developing the atomic bomb [10].

The next jump in the development of the magnetron was pushed by Percy Spencer who was an American engineer who worked for Raytheon which is now renamed as RTX Corporation, a defense contractor of the USA. Percy Spencer was mostly a self taught person without formal education who ended up as the leading experts in the design and manufacturer of magnetrons. Spenser developed a more efficient way to manufacture magnetrons which increased the production from 100 to 2600 per day, this gave a great technical edge for the Allies against the Axis. And for this Percy Spencer was awarded the Distinguished Public Service Award by the U. S. Navy [12].

By 1947 the magnetron could produce electromagnetic waves of small enough wavelength to heat up water. Spencer discovered this as he found the chocolate bar in his pocket melted as he was working on a radar system. He then started using the magnetron to cook food which later developed into the microwave oven. The first microwave oven developed by him was called, “RadaRange” [13].

Figure 6 is a picture of the magnetron. There is a vacuum tube at the center. Electrons emitted from the front of the vacuum tube are guided by magnets placed at the top and bottom of the vacuum tube. The filament emitting electrons is made of tungsten and thorium. Tungsten (W, atomic weight=74, melting point 3414^{o}C) is the highest melting element on earth. Thorium (Th, atomic weight = 90, melting point = 1755^{o}C) is slightly radioactive and thereby can emit a high quantity of electrons. The electron emission is enabled by the heaviest component of a microwave oven, the step-up transformer. This transformer steps up the voltage from 240 V to 2000 V. The electron beam is seen as the green line on the left image within Figure 7. On the right image, the electron beam can be seen to circulate between two of the copper spikes generating 2.45 GHz (λ=12.23 cm). This circulating electron beam is called a cyclotron.

This is like a whistle. In a whistle, air is forced through a rectangular mouthpiece to a slot. This slot splits the air into two directions, one out of the whistle and one into the circular chamber. As the air within the circular chamber interacts with the air that goes above the slot, a resonant frequency is produced. In the case of the magnetron, the cyclotron of electrons will create a polarity between each two spikes. One will be more positive than the other. This is basically a capacitor [14]. At the top space of Figure 6 between the two spikes becomes a single coil inductor. Therefore, an LC circuit is created which generates an e-m, waveform with the formula:

$$f = \frac{1}{2\pi}\sqrt{\frac{1}{LC}}$$

This means the frequency is higher the smaller the values of L and C. Over time the cyclotron of the electron beam retards and finally exhausts into the copper as shown in Figure 6 left image where the green line enders up into the copper. Once in the Cu housing the electron beam will increase the power of the sinusoidal e-m wave generated which will become enough to vibrate the polarized H_2O molecules which is shown at the top left of Figure 6 where the top is positive, and the bottom is negative. The sinusoidal electric field of the e-m wave oscillates the H_2O molecule. Basically, any heating done by the microwave is due to the vibration of molecules in the entirety of the food. This is different from heater element cooking where only the outside of the food is vibrated and this vibration is transmitted to further in. In the microwave oven all H_2O molecules whether outside or inside of the food vibrate simultaneously [15].

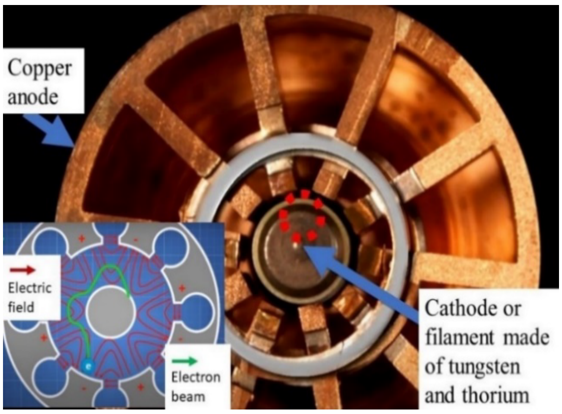


Figure 6: The magnetron

3. Methodology

The first activity was to find out how to control the microwave using an external microcontroller. This was a lengthy study which eventually made the microwave not work. The microwave was switched on for 30 minutes, after which it failed. All repair shops in Sibu, Malaysia tried but they could not repair it. Therefore, efforts were made by this researcher and his student to repair it. The components were taken out and a capacitor (1.07 µF, 2100 V) shown in Figure 7 was found to be faulty. The fuse shown in the top right of Figure 8 also blew. A replacement capacitor was sourced online and fixed and this got the microwave to work.

Figure 7: The top image is the capacitor that failed and the fuse that blew. The bottom image is the 1.07 µF, 2100 V capacitor that failed and its replacement.

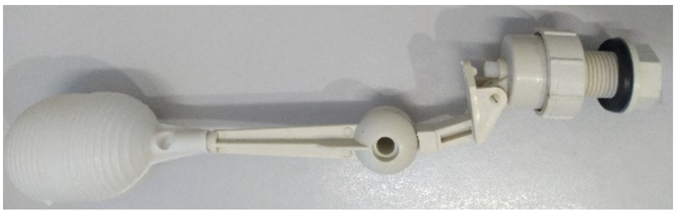


Figure 8: The miniature plastic float valve which was initially purchased for this research.

While waiting for the capacitor to arrive, parts were purchased for subsequent needs for the project. A water container needed to be placed within the microwave. A miniature float valve shown in Figure 9 was purchased to stop the water once the container was full and plastic pipes were purchased. But once the capacitor arrived and the microwave was energized, it was discovered that plastic components cannot be used anywhere inside the microwave oven and the reason is shown in Figure 10; plastic melts when microwaved.

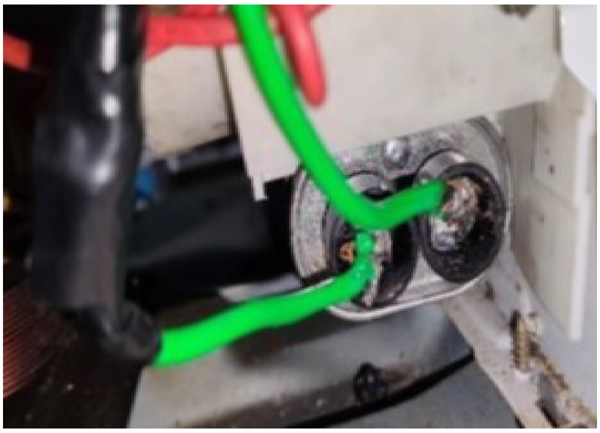


Figure 9: the new capacitor installed

Figure 10: Plastic that melted in the microwave.

Silicon was determined to be suitable since it has a melting point of 1414 °C. Therefore, silicon pipes of Ø = 2 cm were purchased online as shown in Figure 11. A glass container was purchased as shown in Figure 12, which will be used as the holding tank to heat up the water. Two holes were drilled into the top of the microwave, one for a pipe to let water into the holding tank and one to let the water out. A water pump was available but not installed on the outgoing pipe. The five containers of silicone were utilized to seal all holes through which water could come out using the silicon gun shown in Figure 13. Of course, when this system starts being manufactured, it would be optimum to have a glass container where the incoming water is from the top and the outgoing water is from the bottom of the glass container. The glass container purchased for this research could be drilled through using a carbide or diamond-tipped drill bit, but it must be done very slowly and carefully. The glass container must also be periodically cooled upon drilling. But it was decided not to take this chance for this relatively expensive glass container. It was especially hard to reach behind the microwave which required a small hand to reach, and this was achieved by the lady, who had small hands. The silicone had to be applied layer by layer. After one layer dried, it was time to apply the second layer. After applying the silicon for two weeks, the microwave oven was placed on a rooftop to further dry it as shown in Figure 14.

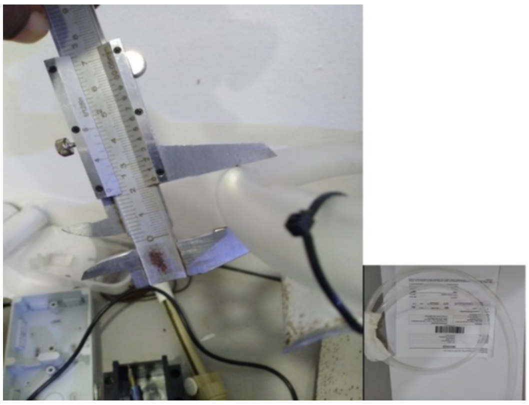


Figure 11: Silicone pipes Ø = 2 cm were purchased

Figure 12: the glass container placed in the microwave oven and the top sealed with silicone

Figure 13: the silicone gun used.

Figure 14: Finally drying the silicone on the rooftop

Simultaneously the controlling of the microwave was worked upon as shown in Figure 15. This took longer than expected with much reference to YouTube videos of this model of Panasonic microwave oven. Table 1 is the discovery made. Pin 3 joined to pin 5 will add 5 seconds to the cooking time. And joining pin 1 to pin 9 will start the microwave oven. The wires were labeled as 3, 1, 9 and 5 as shown in Figure 16. A water valve was purchased as shown in Figure 17 which was controllable using a microcontroller and relay. The valve was rated at 24 V DC so an SMPS (switch mode power supply), shown in Figure 18 was utilized to power it. This is a 62.5 A output capable SMPS which was purchased for an earlier research project which required such high current; this explains why it is larger than a normal SMPS for such low power applications.

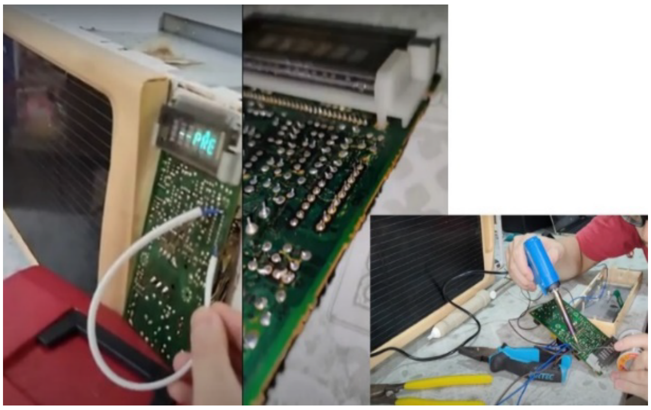


Figure 15: Determining how to control the microwave oven from outside such that a microcontroller can control it.

Table 1: Results

Pin	Pin	Result
3	5	Each contact of pins 3 & 5 will add 5 seconds time of heating
1	9	Start

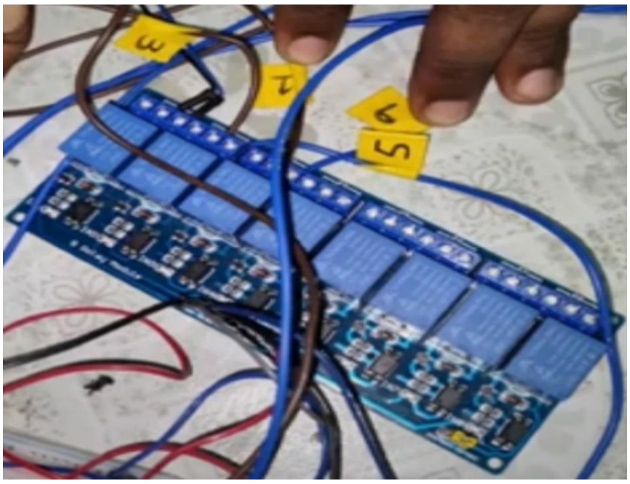


Figure 16: The relays used to trigger the microwave and the whole setup. The relay spec is that it can be triggered by 3.75 - 6 V, with 5 mA. The contacts of the relay can break up to 250 VAC and 30 VDC with a max current of 10 A.

Figure 17: The electrically actuated water valve. Brand: Arita, Model: AW-06, Pressure: 6-7 bar, NC, Orifice: 2.5 mm, Voltage: 24 VDC, Size: ¼" BSP.

Figure 18: The SMPS used to power the electrically actuated valve.

A capacitive sensor shown in Figure 19 was used to replace the float valve. This was installed just above the outgoing pipe as shown in Figure 20. Thus, if there is water at the outgoing pipe, this sensor will be triggered to switch on. The complete circuit was placed in a cardboard box and placed on top of the microwave oven as shown in Figure 20. Table 2 indicates the electrical power consumption using this MBS compared to the conventional heater element bathing water heater. The consumption using this MBS is around 1344 W while the conventional heater element water heater utilizes 2856 W or 47 % less power consumption and is depicted in Figure 21.

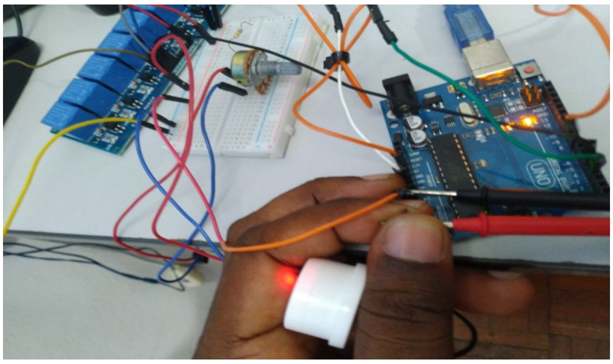


Figure 19: The microcontroller and the white capacitive sensor (held in the hand) to detect water level. The water sensor has the following specs: Input voltage 5~24 V, input current:5 mA, output current: 1~100 mA, operating temperature: 0~105 ºC, Sensitivity: 0~13 mm, Communication: RS485, humidity: 5 %~100 %, material: ABS, ingress protection: IP67.

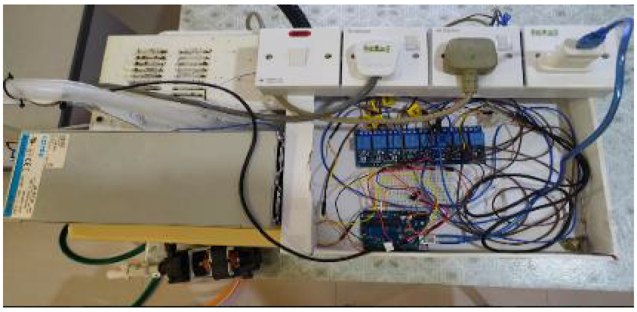


Figure 20: The entire setup. The sensor of Fig. 19 is cable ties to the outgoing pipe.

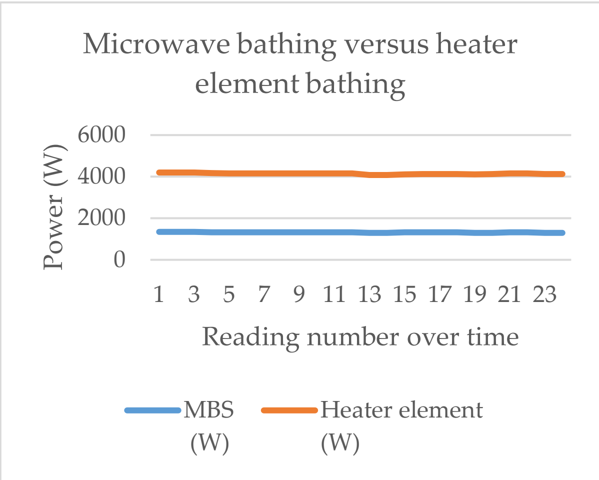


Figure 21: The power consumption of the MBS versus the heater element bathing system used currently

The operation procedure for this MBS is as follows; a person comes to the bathroom. Outside the bathroom are the switch and plug-points which are currently placed on the microwave oven as shown in Figure 20. He / she will switch on the switch which has the neon light above it. This will energize the electrically actuated water valve (1st socket on the right of the switch), the microwave oven (2nd socket) and the microcontroller (3rd socket). The software will start running in the microcontroller and will instruct the microwave to start heating up the water according to a specified time. In the current heater element water heater, the bathing water heat is controlled by turning a knob which is a rheostat to increase the temperature of the water. But for microwave heating, it is controlled by the time period the microwave is switched on. A similar rheostat can be installed in the bathroom which increases voltage to the A/D (analog to digital) pins of the microcontroller which can be translated in the software into the period the microwave is switched on. It would be good to keep this timing in a separate register which the software can pull up. This author previously worked for 14 years at the Western Digital factory where all the process parameters were placed in registers and not directly within the software. The software will look up these Process Parameters to determine how much water to spray, how long to spray or how long a specific motor must turn etc. With this algorithm, the actual software which can be a million lines long need not be disturbed. Note that one accidental typo in a million-line software will render the software inoperable. Such an algorithm is a must in high tech industries where process parameters must continually change to achieve ever increasing data density on a disk platter; the standard in the hard sick industry was to increase the platter data capacity every three months.

Table 2: The power consumption of the microwave bathing water heater compared to the heating element type water heater.

Minutes	Current (A)	Energy Consumed (W)	Current (A)	Energy Consumed (W)
0	0	0	0	0
5	5.6	1344	11.9	2856
10	5.6	1344	11.9	2856
15	5.6	1344	11.9	2856
20	5.5	1320	11.9	2856
25	5.5	1320	11.8	2832
30	5.5	1320	11.8	2832
35	5.5	1320	11.8	2832
40	5.5	1320	11.8	2832
45	5.5	1320	11.8	2832
50	5.5	1320	11.8	2832
55	5.5	1320	11.8	2832
60	5.5	1320	11.8	2832
65	5.4	1296	11.6	2784
70	5.4	1296	11.6	2784
75	5.5	1320	11.6	2784
80	5.5	1320	11.7	2808
85	5.5	1320	11.7	2808
90	5.5	1320	11.7	2808
95	5.4	1296	11.7	2808
100	5.4	1296	11.8	2832
105	5.5	1320	11.8	2832
110	5.5	1320	11.8	2832
115	5.4	1296	11.8	2832
120	5.4	1296	11.8	2832

4. Conclusion

In the current heater element water heater for bathing, there are some brands that use a plastic container (as in Fig. 1) in which the heating element is placed. This can leak electricity into the bathing water. Another problem with the current system is that even if the container is made of copper and grounded as in Fig. 2, the ground may not be properly joined to the ground wire; in most cases home ground wire are just clamped to the ground rod. Corrosion or CuO which is an insulator will form at this point. Only in big installations are the ground wires welded to the ground rod. The RCD installed in each water heater may also not work well. The MBS (microwave bathing system) developed in this research cuts out all the above possibilities. Basically, a microwave oven is used to heat up water which will be used for bathing. Further improvements can be made to the system. The microwave can be made much smaller and the glass container within it can be custom made with one inlet at the top and one outlet at the bottom. Overall, this system saved the user 47% on electrical energy bills and enables electrocution safe bathing.

References

[1] Wakiman, S.U.B, "Implementation of MCOP, Malaysian code of practice on electrical installations of buildings", IET Conference Proceedings, p. 18-24, 2004, DOI: 10.1049/cp:20040770

[2] Staff of the Malaysian Energy Commission, "Interview Conducted by Prashobh Karunakaran," 7 September 2019.

[3] P. Karunakaran, "Electrical Power Simplified," vol. 4, Authorhouse, Indiana, USA, 2014, 238, ISBN-10:1546262466, ISBN-13:978-1546262466.

[4] E. M. Onyekachi and N. B. Nduka, "Empirical Analysis of Core Diameter and Insulation Thickness of House Wiring and Installation Cables in Nigeria," International Journal of Advanced Research in Science, Engineering and Technology, vol. 6, no. 9, 2019.

[5] M. Amanullah, "Engineering Ethics in Islam: An Evaluative And Comparative Study Between Code of Ethics of Institution of Engineers, Bangladesh (IEB) and Code of Professional Conduct of Board of Engineers Malaysia (BEM)," IIUM Engineering Journal, vol. 12, no. 5, 2011, DOI: 10.31436/iiumej.v12i5.193.

[6] P. Karunakaran et al., "Optimizing of Bathing Water Heater using Microwave and Microcontroller Technology," in 2023 IEEE Applied Sensing Conference (APSCON), 1-8, 2023, DOI: 10.1109/APSCON56343.2023.10101161

[7] P. Karunakaran, "Electrical Power Simplified," AuthorHouse, 2018.

[8] A. Sukhnandan, "A theoretical and experimental investigation into fire induced flashover of high voltage transmission lines," Doctoral dissertation, Academaia.edu, 2005.

[9] J. E. Brittain, "Electrical Engineering Hall of Fame: Albert W. Hull [Scanning Our Past]," Proceedings of the IEEE, vol. 98, no. 4, 635-637, 2010, DOI: 10.1109/JPROC.2010.2041837

[10] M. H. F. Wilkins, "John Turton Randall, 23 March 1905-16 June 1984," Royal Society Publishing, 1987.

[11] T. K. Sarkar and M. S. Palma, "A history of the evolution of RADAR," in 2014 44th European Microwave Conference, 734-737, 2014, DOI: 10.1109/EuMC.2014.6986539

[12] J. M. Osepchuk, "The history of the microwave oven: A critical review," in 2009 IEEE MTT-S International Microwave Symposium Digest, 1397-1400, 2009, DOI: 10.1109/MWSYM.2009.5165967

[13] S. Horikoshi et al., "Microwave as a heat source," Microwave Chemical and Materials Processing: A Tutorial, pp. 1-17, 2018.

[14] P. J. Kelly and R. D. Arnell, "Magnetron sputtering: a review of recent developments and applications," Vacuum, vol. 56, no. 3, 159-172, 2000, DOI: 10.1016/S0042-207X(99)00189-X

[15] S. Chandrasekaran, S. Ramanathan, and T. Basak, "Microwave food processing—A review," Food research international, vol. 52, no. 1, 243-261, 2013, DOI: 10.1016/j.foodres.2013.02.033

Received: 30 April, 2024, Revised: 20 June, 2024, Accepted: 21 May, 2024, Online: 21 June, 2024

DOI: https://doi.org/10.55708/js0306003

Educational Applications and Comparative Analysis of Network Simulators: Protocols, Types, and Performance Evaluation

Nikolaos V. Oikonomou [1], **Dimitrios V . Oikonomou** [2]

[1] Department of Informatics & Telecommunications, University of Ioannina, Arta,47150, Greece

[2]Department of Regional & Cross Border Studies, University of Western Macedonia, Kozani,50100, Greece

*Corresponding author: Nikolaos V. Oikonomou, University of Ioannina Department of Informatics & Telecommunications, haikos13@gmail.com

ABSTRACT: This work explores the role of simulation in computer networks, discussing various network types, communication protocols, and the utilization of network simulators, with a focus on educational settings. We specifically analyze and compare five prominent network simulators: Cisco Packet Tracer, Riverbed Modeler Academic Edition, GNS3, NS-3, and Mininet. These tools are examined in terms of their functionality, user-friendliness, and suitability for educational purposes, assessing how they facilitate learning for students and trainees. The comparison extends to their operational capabilities, differences, effectiveness, and overall impact on networking education. The evaluation aims to highlight each simulator's strengths and weaknesses, providing insights into their practical applications in an academic context.

KEYWORDS: Network Architecture, Network Simulation, Network Protocol, Emulator, Educational Technology, Performance Evaluation

1. Introduction

In today's interconnected world, networks encompass nearly every aspect of our global society, forming a vast web that spans the entire Earth. This extensive coverage is not just limited to geographic expansiveness but also permeates various sectors of human activity, ranging from economic operations and governmental infrastructure to personal communication and entertainment. Networks facilitate the seamless flow of information, enabling not just global connectivity but also driving advancements in technology and society. A computer network, by definition, is a system comprising both autonomous and non-autonomous computers or nodes interconnected through various means of communication. These networks are fundamental to the modern digital ecosystem, supporting an extensive array of devices beyond traditional computers, including mobile phones, printers, cameras, televisions, and even more sophisticated IoT (Internet of Things) devices like smart thermostats and security systems. The term "computer" in this context serves a formal role, recognizing any device capable of sending, receiving, and processing data as part of the network infrastructure. As networks have become ubiquitous, the functions and effectiveness of network simulations have similarly evolved to become indispensable tools in network design and management. These simulations are employed extensively across a spectrum of applications—from crafting robust architectures for large organizations and services to developing state defense mechanisms. They play a pivotal role in telecommunications, where they help in optimizing network performance and security under various scenarios without the need to physically alter the network during testing. Moreover, the role of network simulations extends into the realm of education, where they provide a practical learning experience for students and professionals alike. Through simulations, learners can explore complex network dynamics and interactions in a controlled environment, enhancing their understanding of network management, problem-solving, and strategic planning. These simulations are also crucial in the design and implementation of networks. They allow engineers and network designers to experiment with network configurations, simulate loads and attacks, and foresee how a network might behave under stress or failure conditions. The predictive capabilities of network simulations help in preempting problems and designing networks that are both resilient and scalable. In summary, the development of network technology and simulations reflects our growing dependence on digital connectivity and the continuous need for advancements in network reliability, security, and efficiency. As we look towards future innovations—such as 5G networks, enhanced broadband capabilities, and more sophisticated cybersecurity measures—the role of network simulations will only grow in importance, shaping the backbone of our digital world and ensuring that networks not only cover the Earth geographically but also meet the evolving demands of a highly connected future [1].

2. Types of Networks

Networks are divided into wired and wireless in terms of their connectivity. Wired networks are defined as those that communicate with each other through physical means, namely through networking cables. Wireless networks communicate without any transmission medium such as any cable. In terms of their coverage, we have the 3 basic categories of networks:

Local coverage networks that usually connect devices within the same building or nearby building infrastructure and their coverage range does not exceed a radius of 1 km.

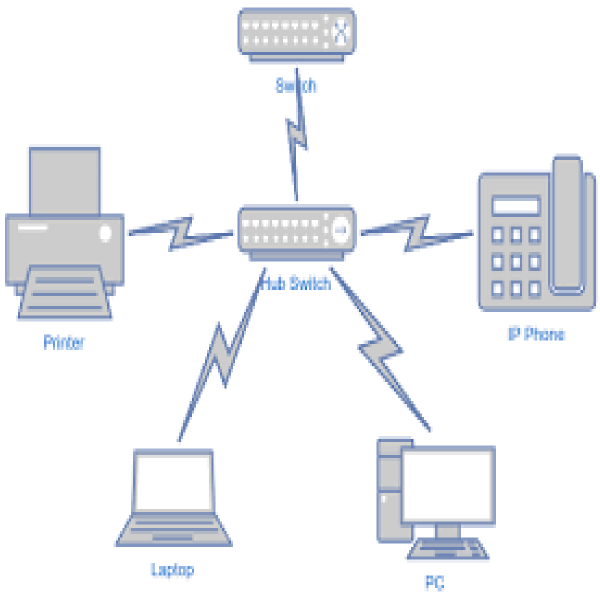


Figure 1: Example of a LAN network in a home or workplace.

Metropolitan area networks consist of many LAN networks together and usually cover the radius of an entire area, a campus, or a city and usually reach a radius of 50km.

Wide area networks provide much larger coverage than LAN & WAN as they are the sum of the above network types, and their coverage is considered unlimited because they cover the entire Earth [2].

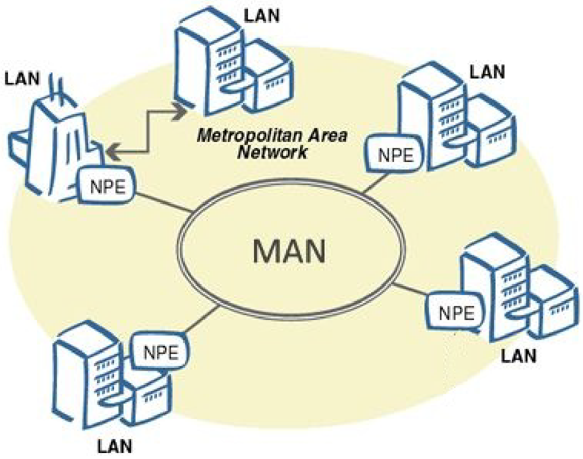


Figure 2: Example of a MAN network in a city.

Figure 3: The internet and global connectivity themselves are the ideal example of a WAN.

2.1. *Network communication protocols*

The most important network communication protocol as the entire internet is based on it. Divided into layers, it can manage all data transfer problems.

A basic data transfer protocol. Unlike TCP/IP, it does not have security functions but supports the transmission of information to multiple users simultaneously. It is mainly used in telecommunications (VoIP) and Online Videogames. FTP is widely used by TCP/IP for sending and receiving files. The 802-protocol family is used in LAN and MAN networks. This protocol allows the changing of IP addresses between computers to achieve accurate data transmission through the correct addressing.

It is the initial information transmission model which started as an idea in 1970 and was formalized in 1984. It divides the entire data transfer process into 7 layers, each undertaking a separate process in the facilitation of information transfer.

Figure 4: The OSI operating levels.

3. Introduction to the Network Simulations

In this chapter, the definition of simulation, its operation, why we perform a network simulation, as well as the types of simulators with their advantages and disadvantages will be discussed. Simulation is called the study of a system with the help of a computer through a numerical experiment modeling technique. Essentially, we input (input) the data we want to study to obtain the desired output (output) results by mimicking the real-world processes or system [3].

3.1. Network Simulation Operation

In the intricate world of network simulations, the operation harmoniously blends information technology and statistical processes. This synergistic integration allows for robust data collection, analysis, and the creation of detailed visual representations such as diagrams and charts. At the heart of these operations are sophisticated programs specifically designed to execute tasks such as sampling, processing data into actionable insights, and graphical representation. These tools are crucial in turning raw data into comprehensible outcomes that can inform strategic decisions and optimize network designs [4].

3.1.1. DES (Discrete Event Simulation)

Discrete Event Simulation (DES) represents a critical methodology in the realm of simulation technologies, where the flow of time is modeled as distinct, individual events. Each event occurs at a particular instant in time and marks a change of state in the system. This approach is widely utilized across various sectors, from complex industrial settings like factories and maritime installations to more streamlined applications such as the operational dynamics of household appliances. For instance, in large-scale industrial applications, DES can simulate the logistical operations of a factory floor, the scheduling and management of maritime ports, or the traffic flow of communication in large call centers. These environments benefit from DES's ability to model intricate systems where components interact at discrete points in time, allowing for detailed analysis of each interaction. On a smaller scale, DES can be applied to everyday applications such as the adaptive control of a radiator’s fan speed based on temperature fluctuations or the intensity adjustments of a lamp in response to changes in ambient lighting conditions. These simulations help in refining product designs to enhance functionality and user experience [5].

3.1.2. ABS (Agent-Based Simulation)

Agent-Based Simulations (ABS) provide a dynamic framework where agents, autonomous decision-making entities with defined behaviors, interact within a simulated environment. This type of simulation is particularly suited to scenarios where complex interactions among agents lead to emergent behaviors and outcomes. ABS is extensively used in fields such as healthcare, where it can model the spread of diseases within a population, or in economics, where it can simulate market dynamics and consumer behaviors. Agents in these simulations are designed to mimic real-world behaviors, making ABS an excellent tool for studying social systems and organizational structures. For example, in health sector simulations, agents could represent individuals with varying susceptibility to a disease, allowing researchers to study the impacts of interventions on disease spread [6].

3.1.3. CS (Continuous Simulation)

Continuous Simulation (CS) deals with systems characterized by continuous state changes over time, modeled through differential equations. Unlike DES where changes occur at discrete intervals, CS provides a smooth and continuous description of system dynamics. This type of simulation is indispensable in fields like environmental science for studying climate change impacts or in engineering for assessing the stress and strain on materials over time. CS is particularly effective for simulations that require tracking of variables that change incrementally across every point in time, such as the growth of a population of animals or the dispersion of pollutants in an ecosystem. These simulations are crucial for long-term planning and forecasting in environmental management and urban planning [7].

3.1.4. HS (Hybrid Simulation)

Hybrid Simulations combine the features of both discrete and continuous simulations. This blend allows for the modeling of systems where both continuous and discrete processes exist simultaneously. For example, a hybrid simulation might model a manufacturing process (a continuous flow of materials) alongside the maintenance schedules of the machinery (discrete events) [8].

3.1.4.1. Continuous Processes

Continuous simulations are used to model systems that change in a smooth, continuous manner over time. In the context of network simulations, this might involve the continuous flow of data through network channels, where parameters like bandwidth, latency, and error rates are modeled as continuous variables.

3.1.4.2. Discrete Events

Discrete event simulations, on the other hand, are used to model systems where changes occur at distinct points in time. In network simulations, this could involve modeling events such as packet arrivals, queue formations, and protocol state transitions.

3.1.4.3. Integration of Continuous and Discrete Models

Hybrid simulations integrate these two approaches to provide a more comprehensive modeling framework. For instance, in a network simulation, continuous models can simulate the overall data flow, while discrete models can handle specific events like packet drops or node failures.

3.1.4.4. Advantages of Hybrid Simulation

The primary advantage of hybrid simulation is its ability to capture the dynamic interactions between continuous processes and discrete events. This approach can provide more accurate and detailed insights into the behavior of complex systems, which is particularly useful in scenarios where both types of processes are significant.

Hybrid simulation techniques are increasingly used in various fields, including manufacturing, logistics, healthcare, and telecommunications. They offer a powerful tool for analyzing and optimizing systems where continuous and discrete dynamics interact.

3.2. Advantages/Disadvantages

The adoption of simulation technologies comes with its set of advantages and challenges. Below, we explore these aspects in detail, providing a comprehensive understanding of the potential benefits and limitations. [9].

3.2.1. Advantages

3.2.1.1. Risk-Free Testing Environment

3.2.1.1.1. Scenario Modeling

The foremost advantage of using simulations is their ability to model and test every conceivable scenario for network implementation in a risk-free, virtual environment. This allows for the examination of network behavior under various conditions without the risk of disrupting actual network operations.

3.2.1.1.2. Cost-Efficiency

Simulations help in identifying the most efficient, cost-effective, and robust network configurations. This pre-implementation testing can save significant costs associated with trial-and-error approaches in real-world deployments.

3.2.1.1.3. Safety

By testing network changes in a simulated environment, potential issues can be identified and resolved before they affect live systems, ensuring network stability and reliability.

3.2.1.2. Flexibility and Scalability

3.2.1.2.1. Scalable Models

Simulation tools can model networks of various sizes and complexities, from small local networks to large-scale global infrastructures. This scalability allows for comprehensive analysis of network performance and potential bottlenecks.

3.2.1.2.2. Customizable Scenarios

Users can customize simulations to match specific requirements, such as testing new protocols, evaluating network upgrades, or assessing security vulnerabilities.

3.2.1.3. Enhanced Understanding and Learning:

3.2.1.3.1. Educational Value

Simulation tools provide a valuable educational resource, allowing students and professionals to visualize and interact with complex network topologies and protocols. This hands-on experience enhances understanding and facilitates learning.

3.2.1.3.2. Predictive Analysis

By simulating future network scenarios, organizations can proactively identify potential issues and implement preventive measures, enhancing overall network resilience.

3.2.1.4. Efficient Resource Allocation:

3.2.1.4.1. Optimization

Simulations can identify the optimal allocation of network resources, such as bandwidth and hardware, ensuring efficient utilization and minimizing wastage.

3.2.1.4.2. Performance Evaluation

They enable the evaluation of network performance under different configurations, helping in making informed decisions about resource investments and upgrades.

3.2.2. Disadvantages

3.2.2.1. Resource-Intensive Development

3.2.2.1.1. High Costs

The development and operation of advanced network simulators are resource-intensive, requiring significant investment in both computational resources and expert human capital. High-performance computing infrastructure and skilled personnel are essential for creating and managing detailed simulation models.

3.2.2.1.2. Complex Setup

Setting up simulation environments can be complex and time-consuming, particularly for large-scale or highly detailed simulations.

3.2.2.2. Limitations in Realism:

3.2.2.2.1. Approximation of Reality

Despite advances in simulation technology, the inherent unpredictability of real-world environments means that simulations, while highly indicative, cannot completely replicate all real-world variables. Factors such as human behavior, environmental changes, and unexpected network traffic patterns may not be fully captured.

3.2.2.2.2. Validation Required

Consequently, results from simulations should be interpreted as approximations. While they provide valuable insights, these results require careful consideration and validation against real-world data and experiences to ensure their accuracy and applicability.

3.2.2.3. Potential for Over-Reliance:

3.2.2.3.1. Overconfidence in Simulations

There is a risk that organizations may become overly reliant on simulation results, potentially overlooking the importance of real-world testing and validation. Simulations should complement, not replace, empirical testing and field trials.

3.2.2.3.2. Static Models

Simulation models can become outdated if they are not regularly updated to reflect changes in technology, network configurations, and usage patterns. Continuous maintenance is necessary to ensure simulations remain relevant and accurate.

3.2.2.4. Technical Challenges

3.2.2.4.1. Modeling Complexity

Accurately modeling complex networks and protocols can be technically challenging. Simplifications and assumptions made during the modeling process can impact the accuracy of simulation results.

Debugging and Troubleshooting: Identifying and resolving issues within simulation models can be difficult, particularly when dealing with intricate network interactions and behaviors.

In conclusion, while simulation technologies offer numerous advantages, including risk-free testing, cost-efficiency, and enhanced learning opportunities, they also present challenges such as high development costs, limitations in realism, and potential over-reliance. To maximize the benefits of simulations, it is crucial to balance their use with real-world testing and continuously validate simulation results against actual network performance. By doing so, organizations can leverage simulations to improve network design, optimize resource allocation, and enhance overall network resilience.

4. Network Simulation Tools

In this section, we will explore some of the most prevalent network simulation programs available today. These tools are crucial for modeling, analyzing, and optimizing network performance in various environments, ranging from educational settings to complex commercial deployments [10] , [11].

4.1. Opnet (Optical Micro-Networks)

Opnet, now known as Riverbed Modeler, is a discrete event simulation (DES) tool with graphical user interface (GUI) support. It is widely regarded as one of the most comprehensive and powerful network simulators available in the commercial market. Opnet's architecture allows it to be used in several sectors due to its ability to model a wide range of network components and behaviors.

4.1.1. Wireless Communications

Opnet can simulate various wireless technologies, including Wi-Fi, cellular networks, and satellite communications. It provides detailed models for radio frequency (RF) propagation, interference, and mobility patterns, enabling accurate performance analysis of wireless networks.

4.1.2. Wired Communications

The tool supports the simulation of traditional wired networks, including Ethernet, fiber optics, and other physical media. Users can model network topologies, link failures, and traffic patterns to study network performance under different conditions.

4.1.3. Protocols

Opnet includes a vast library of protocol models, covering all layers of the OSI model. This allows users to simulate the behavior of routing protocols, transport protocols (e.g., TCP, UDP), and application-layer protocols, providing insights into protocol interactions and performance.

4.1.4. Queues

The simulator offers detailed queue models, enabling the analysis of queuing behavior in network devices such as routers and switches. This helps in understanding delays, jitter, and packet loss under various traffic loads.

4.1.5. Microprocessors & Complex Hardware Types

Opnet can model the internal behavior of network devices, including the processing power of microprocessors and the performance of complex hardware components. This capability is essential for studying the impact of hardware configurations on network performance [12].

4.2. Cisco Packet Tracer

Cisco Packet Tracer is Cisco's primary network simulation and visualization tool, widely used in educational institutions, academies, and organizations for training and certification purposes. It provides a robust platform for learning, teaching, and testing network concepts and configurations.

4.2.1. Learning and Teaching

Packet Tracer is an integral part of Cisco’s Networking Academy curriculum. It allows students to practice network configuration and troubleshooting in a virtual environment, reinforcing theoretical knowledge with hands-on experience.

4.2.2. Testing and Certifications

The tool supports the preparation for Cisco certification exams (e.g., CCNA, CCNP) by providing realistic simulation scenarios that mirror those encountered in real-world networks. This practical training is crucial for developing the skills required to manage and configure Cisco networks.

4.2.3. Features

Packet Tracer includes a variety of features, such as real-time and simulation modes, allowing users to visualize network behavior and performance. It supports a wide range of Cisco devices and protocols, enabling the simulation of complex network topologies and interactions.

4.2.4. Collaboration

The tool offers collaborative features, allowing multiple users to work on the same network simulation simultaneously. This is particularly useful in classroom settings, where instructors and students can interact and share insights in real-time [13].

4.3. GNS3 (Graphical Network Simulator-3)

GNS3 is an open-source network simulator that allows the simulation of complex networks using real network hardware images and virtualization technologies. It is highly popular among network professionals and enthusiasts for its flexibility and powerful features.

4.3.1. Real Hardware Emulation

Unlike other simulators that rely on abstract models, GNS3 uses real Cisco IOS, Juniper Junos, and other network operating system images to emulate actual hardware devices. This provides a highly realistic simulation environment.

4.3.2. Integration with Virtual Machines

GNS3 can integrate with VirtualBox, VMware, and other virtualization platforms, allowing the simulation of virtual machines alongside network devices. This is useful for simulating end-to-end network scenarios, including client-server interactions and multi-tier applications.

4.3.3. Extensibility

The simulator supports a wide range of plugins and third-party tools, such as Wireshark for packet capture and analysis. This extensibility makes GNS3 a versatile tool for network design, testing, and troubleshooting.

4.3.4. Community Support

GNS3 has a vibrant community of users and contributors who provide support, share configurations, and develop new features. This community-driven approach ensures continuous improvement and adaptation to emerging networking technologies. [14]

4.4. NS-3 (Network Simulator 3)

NS-3 is an open-source discrete-event network simulator designed for research and educational purposes. It provides a detailed simulation environment for networking protocols and internet systems.

4.4.1. Research Focus

NS-3 is widely used in academic and research settings to study the performance and behavior of networking protocols. It supports the simulation of a wide range of network types, including wired, wireless, and satellite networks.

4.4.2. Realism and Accuracy

The simulator offers high fidelity models that closely mimic real-world network behavior. This accuracy makes it suitable for validating theoretical models and conducting performance evaluations.

Programming Interface: NS-3 provides a flexible programming interface, allowing users to extend and customize the simulator to meet specific research needs. It supports C++ and Python, making it accessible to a broad range of users.

4.4.3. Visualization Tools

While NS-3 itself is focused on the simulation engine, it integrates with various visualization tools, such as NetAnim and PyViz, to provide graphical representations of network topologies and traffic flows [15].

4.5. Mininet

Mininet is a network emulator that creates a realistic virtual network on a single machine. It is widely used for developing, testing, and demonstrating software-defined networking (SDN) applications.

4.5.1. SDN Focus

Mininet is specifically designed to support SDN and OpenFlow. It allows users to create and experiment with SDN topologies, controllers, and applications, making it an essential tool for SDN research and development.

4.5.2. Rapid Prototyping

The emulator can quickly instantiate network topologies, making it ideal for rapid prototyping and testing. Users can simulate large networks with hundreds of nodes using minimal resources.

4.5.3. Integration with Real Networks

Mininet can integrate with physical networks, enabling hybrid environments where virtual and real devices interact. This feature is valuable for testing SDN applications in realistic settings.

4.5.4. Educational Use

Mininet is widely used in educational settings to teach SDN concepts and practices. Its ease of use and flexibility make it a popular choice for classroom labs and assignments [16].

4.6. Presentation of Simulators

4.6.1. Analysis of Cisco Packet Tracer

Cisco Packet Tracer is an innovative educational tool that allows trainees to create networks with nearly unlimited capabilities, using a wide range of devices. This encourages diagnostics and troubleshooting, enhancing the learning experience.

From the Educator's Perspective: Packet Tracer allows the teaching of network device functions that operate in the background, visible in everyday life. For instance, it can simulate the operation of a router, showing what happens from the moment a user enters a password until the router grants access to the user's device. The simulation capabilities simplify the learning process by providing tables, diagrams, and visual representations of internal functions, such as dynamic data transfers. The simulation function reduces presentation time by replacing tables and static slides with real-time visual effects.

4.6.2. Benefits for Educators

- Provides visual displays of complex technologies with configuration capability.
- Allows customized, guided activities with immediate feedback.
- Facilitates various learning activities like lectures, lab activities, homework, assessments, and games.
- Supports network design, troubleshooting, modeling tasks, and case studies.
- Enables visualization, movement, and detailed modeling for exploration, research, and experimentation.
- Encourages learning outside the classroom.
- Supports social learning, collaboration, and healthy competition.
- Covers most protocols and technologies taught in Cisco seminars and courses.

From the Student-Trainee Perspective: Packet Tracer offers a practical way of learning through simulation, allowing better opportunities to gain practical skills and knowledge when working with real equipment. Trainees gain faster experience through troubleshooting in simulation and real-world scenarios, building confidence and leading to a more productive workforce.

4.6.3. Workspaces

4.6.3.1.1. Logical Workspace

Users can create logical network topologies by placing, connecting, and grouping virtual network devices.

4.6.3.1.2. Physical Workspace

Provides a graphic-physical dimension of the logical network, showing how devices like routers, switches, and mainframes would appear in a real environment. It also includes geographical representations of networks, including cities, buildings, and cables.

4.6.4. Modes of Operation

4.6.4.1. Real-Time Mode

The network and its devices behave like real devices, offering immediate real-time response for all networks and subnets.

4.6.4.2. Simulation Mode

Allows users to control the timing, internal functions of data transfer, and data propagation in a network, helping students understand the fundamental concepts behind network operations.

4.6.5. Supported Protocols and Applications

4.6.5.1. Application

FTP, SMTP, POP3, HTTP, TFTP, Telnet, SSH, DNS, DHCP, NTP, SNMP, AAA, ISR, VOIP, SCCP config and calls ISR command support, Call Manager Express.

4.6.5.2. Transport

TCP/IP, UDP, TCP Nagle Algorithm & IP Fragmentation, RTP.

4.6.5.3. Network

BGP, IPv4, ICMP, ARP, IPv6, ICMPv6, IPSec, RIPv1/v2/ng, Multi-Area OSPF, EIGRP, Static Routing, Route Redistribution, Multilayer Switching, L3 QoS, NAT, CBAL, Zone-based policy firewall, and Intrusion Protection System on the ISR, GRE VPN, IPSec VPN.

4.6.5.4. Network Access/Interface

Ethernet (802.3), 802.11, HDLC, Frame Relay, PPP, PPPoE, STP, RSTP, VTP, DTP, CDP, 802.1q, PAgP, L2 QoS, SLARP, Simple WEP, WPA, EAP.

4.6.6. Additional Features

- Modular devices simulation with graphical hardware allowing interface card insertion into routers.
- Multi-user functionality for collaborative construction of virtual networks through a real network.
- Step-by-step tutorials, advanced workshops, and a comprehensive help feature.
- Activity Wizard for creating custom learning activities with grading and feedback capabilities.
- Lab scoring function, international language support, and compatibility with all platforms (Windows and Linux).

4.7. Analysis of GNS3

GNS3 is an open-source network simulator highly regarded for its flexibility and powerful features, allowing the simulation of complex networks using real network hardware images and virtualization technologies.

4.7.1. From the Educator's Perspective

GNS3 is a valuable tool for teaching network concepts and configurations, offering hands-on experience with real network operating system images. This allows students to gain practical knowledge and skills by working with actual network environments.

4.7.2. Benefits for Educators

- Provides a realistic simulation environment using real hardware images.
- Supports the integration of virtual machines, enabling comprehensive network simulations.
- Allows the use of a wide range of plugins and third-party tools for enhanced functionality.
- Offers community support for troubleshooting and sharing configurations.

4.7.3. From the Student-Trainee Perspective

GNS3 provides a practical learning platform where students can experiment with complex network setups and troubleshoot issues in a safe environment. This hands-on experience is crucial for developing real-world networking skills.

4.7.4. Workspaces

4.7.4.1. Topology Workspace

Users can create and manage network topologies by adding, connecting, and configuring virtual network devices and links.

4.7.4.2. Virtual Machine Integration

Supports integration with virtualization platforms like VirtualBox and VMware, allowing the inclusion of virtual machines in network simulations.

4.7.5. Modes of Operation

4.7.5.1. Real-Time Mode

Provides real-time simulation of network operations, allowing users to observe and interact with the network as if it were live.

4.7.5.2. Simulation Mode

Offers control over simulation parameters, enabling detailed analysis of network behavior and performance.

4.7.6. Supported Protocols and Applications:

4.7.6.1. Application

Supports a wide range of networking protocols and applications through real network operating system images (e.g., Cisco IOS, Juniper Junos).

4.7.6.2. Transport

Capable of simulating various transport protocols and behaviors, providing a comprehensive understanding of network interactions.

4.7.6.3. Network

Supports detailed simulation of network protocols, including routing, switching, and security protocols.

4.7.6.4. Network Access/Interface

Emulates a variety of network interfaces and access technologies, providing flexibility in network design and testing.

4.7.7. Additional Features

- Real hardware emulation for high-fidelity network simulations.
- Integration with virtual machines for end-to-end network scenarios.
- Extensibility through plugins and third-party tools like Wireshark.
- Active community support for continuous improvement and problem-solving.

4.8. Analysis of Opnet (Riverbed Modeler)

Opnet, now Riverbed Modeler, is a comprehensive network simulation tool used for modeling a wide range of network components and behaviors across various sectors.

4.8.1.1. From the Educator's Perspective

Riverbed Modeler offers detailed simulation capabilities that are invaluable for teaching advanced network concepts and performance analysis. Its extensive library of models allows educators to cover a wide range of networking topics with high accuracy.

4.8.2. Benefits for Educators

- Provides detailed models for wireless and wired communications, protocols, and queuing behaviors.
- Enables the simulation of complex hardware components and microprocessors.
- Offers a comprehensive GUI for visualizing network simulations and results.
- Supports the creation of custom simulation scenarios for targeted learning objectives.

4.8.2.1. From the Student-Trainee Perspective

Riverbed Modeler provides a powerful platform for students to experiment with network configurations and analyze performance metrics. Its detailed simulation capabilities help students understand the intricacies of network operations and the impact of various factors on performance.

4.8.3. Workspaces

4.8.3.1. Modeler Workspace

Allows users to create detailed network models by adding and configuring nodes, links, and protocols.

4.8.3.2. Simulation Workspace

Provides tools for setting up and running simulations, as well as analyzing results through graphical and statistical outputs.

4.8.4. Modes of Operation

4.8.4.1. Real-Time Mode

Simulates network operations in real-time, providing immediate feedback and insights.

4.8.4.2. Simulation Mode

Allows users to control simulation parameters and analyze network behavior over time.

4.8.5. Supported Protocols and Applications

4.8.5.1. Application

Includes a vast library of application-layer protocols for comprehensive simulation of network traffic and interactions.

4.8.5.2. Transport

Supports detailed modeling of transport protocols like TCP and UDP, providing insights into performance and reliability.

4.8.5.3. Network

Offers extensive models for network-layer protocols, including various routing and switching protocols.

4.8.5.4. Network Access/Interface

Emulates a wide range of access technologies and network interfaces for flexible network design and testing.

4.8.6. Additional Features

- Detailed queue models for analyzing queuing behavior in network devices.
- Support for complex hardware modeling, including microprocessor performance.

- Comprehensive GUI for visualizing network simulations and results.
- Extensive library of protocol models covering all layers of the OSI model.

4.9. Analysis of NSE (Network Simulator 3)

NS3 is an open-source discrete-event network simulator, primarily used for research and educational purposes. It provides a realistic simulation environment for network protocols and internet systems.

From the Educator's Perspective: NS3 is a valuable tool for teaching network protocols and internet systems due to its detailed and accurate simulation capabilities. It allows educators to create realistic network scenarios and analyze the performance of various protocols.

4.9.1. Benefits for Educators

- Offers a realistic and flexible simulation environment.
- Provides extensive documentation and tutorials, aiding in the learning process.
- Supports a wide range of network protocols and models, allowing comprehensive educational coverage.

4.9.1.1. From the Student-Trainee Perspective

NS3 provides a detailed learning platform for students to understand network protocols and their behaviors in real-world scenarios. The hands-on experience with NS3 aids in developing a deep understanding of network operations.

4.9.2. Workspaces

4.9.2.1. Simulation Script Workspace

Users can write and execute simulation scripts in C++ or Python, allowing for detailed and customized network simulations.

4.9.3. Modes of Operation

4.9.3.1. Simulation Mode

Provides detailed and controlled simulation of network scenarios, allowing users to analyze protocol performance and network behavior.

4.9.4. Supported Protocols and Applications

4.9.4.1. Application

Includes support for a wide range of application-layer protocols for realistic simulation of network traffic and interactions.

4.9.4.2. Transport

Provides detailed modeling of transport protocols like TCP, UDP, SCTP, and more, offering insights into their performance and reliability.

4.9.4.3. Network

Supports extensive models for network-layer protocols, including various routing and switching protocols.

4.9.4.4. Network Access/Interface

Emulates a wide range of access technologies and network interfaces, enabling flexible network design and testing.

4.9.5. Additional Features

- Realistic and detailed simulation environment for network protocols.
- Extensible through custom scripts in C++ or Python, allowing for highly customizable simulations.
- Strong community support with extensive documentation and user-contributed models.
- Regular updates and active development to keep up with emerging networking technologies.

4.10. Analysis of Mininet

Mininet is an open-source network emulator that creates a realistic virtual network, running real kernel, switch, and application code on a single machine. It is widely used for developing and testing network applications and protocols.

4.10.1. From the Educator's Perspective

Mininet is an excellent tool for teaching software-defined networking (SDN) and network function virtualization (NFV) concepts. Its ability to emulate a complete network on a single machine makes it accessible for educational environments.

4.11. Benefits for Educators

- Provides a realistic and interactive environment for teaching SDN and NFV.
- Supports the creation of complex network topologies with minimal hardware requirements.
- Includes extensive documentation and a large collection of example scripts for various network scenarios.

4.11.1. From the Student-Trainee Perspective

Mininet offers a hands-on learning experience, allowing students to experiment with real network code and configurations. This practical approach helps in understanding the intricacies of network behavior and management.

4.11.2. Workspaces:

4.11.2.1. CLI Workspace

Users can interact with the Mininet environment through a command-line interface, allowing for the creation and management of network topologies.

4.11.2.2. Python API Workspace

Provides a Python API for scripting network configurations and behaviors, enabling automation and customization.

4.11.3. Modes of Operation

4.11.3.1 Emulation Mode

Emulates a complete network on a single machine, allowing for the testing and development of network applications and protocols in a controlled environment.

4.11.4. Supported Protocols and Applications:

4.11.4.1. Application

Supports the deployment of real applications within the emulated network, providing a realistic environment for testing network applications.

4.11.4.2. Transport

Emulates transport protocols like TCP and UDP, allowing for detailed analysis of their behavior and performance.

4.11.4.3. Network

Supports a wide range of network protocols, including SDN protocols like OpenFlow.

4.11.4.4. Network Access/Interface

Emulates various network interfaces and access technologies, enabling flexible network design and testing.

4.11.5. Additional Features

- Real kernel, switch, and application code running in the emulated environment.
- Supports integration with SDN controllers like OpenDaylight and Ryu.
- Extensible through custom scripts and plugins, allowing for advanced network emulation scenarios.
- Active community support with extensive documentation and user-contributed examples.

By leveraging these tools, network professionals and students can gain valuable insights, improve network designs, and enhance overall network performance. These simulators provide robust platforms for learning, testing, and optimizing networks, ensuring efficient and effective network management and development.

5. Results of the comparison & Conclusions

In this section, we present the results of our comparative analysis of three prominent network simulation tools: Cisco Packet Tracer, GNS3, and Riverbed Modeler Academic Edition (Opnet). Each tool's capabilities, strengths, and limitations were evaluated based on several key criteria: educational applicability, user experience, simulation depth, flexibility, and scalability. This comprehensive comparison aims to provide insights into how each simulator can be optimally utilized in various educational and professional contexts.

5.1. Comparative Criteria and Results

5.1.1. Educational Applicability:

5.1.1.1. Cisco Packet Tracer

- Strengths: Highly effective for beginners due to its intuitive interface and straightforward simulation capabilities. Integral to Cisco’s Networking Academy curriculum, making it a staple in foundational network training.
- Limitations: Primarily focuses on Cisco devices and protocols, which may limit exposure to broader networking environments.

5.1.1.2. GNS3

- Strengths: Provides a realistic simulation environment using real network hardware images, which is beneficial for advanced learning and professional training. Supports a wide range of network devices and protocols.
- Limitations: The complexity of setup and use can be a barrier for beginners, requiring more advanced knowledge and skills.

5.1.1.3. Riverbed Modeler Academic Edition

- Strengths: Offers comprehensive modeling capabilities that are ideal for higher education and professional research. Detailed simulations and extensive protocol support make it suitable for in-depth studies and advanced network analysis.
- Limitations: Resource-intensive and complex, making it less accessible for beginners and smaller educational institutions.

5.1.1.4. NS3

- Strengths: Highly suitable for advanced research and educational purposes. Provides a realistic and detailed simulation environment for network protocols and internet systems.
- Limitations: Requires advanced technical knowledge and programming skills, making it less accessible for beginners.

5.1.1.5. Mininet

- Strengths: Ideal for teaching software-defined networking (SDN) and network function virtualization (NFV). Provides a practical, hands-on learning experience with real network code and configurations.
- Limitations: Limited to emulating rather than simulating, which may not fully capture the nuances of complex network interactions

5.1.2. User Experience:

5.1.2.1. Cisco Packet Tracer

- Strengths: User-friendly interface with visual aids and guided activities. Real-time and simulation modes enhance interactive learning.
- Limitations: Limited to Cisco-specific environments, which may not fully prepare users for multi-vendor network scenarios.

5.1.2.2. GNS3

- Strengths: Flexibility and realism due to the use of real network OS images. Strong community support provides additional resources and troubleshooting help.
- Limitations: The steep learning curve and technical setup requirements can be challenging for less experienced users.

5.1.2.3. Riverbed Modeler Academic Edition

- Strengths: Comprehensive GUI and detailed feedback mechanisms. Supports complex simulations with high accuracy.
- Limitations: High cost and complexity may deter smaller institutions and individual learners from using it.

5.1.2.4. NS3

- Strengths: Highly detailed and customizable simulation environment. Strong support for scripting in C++ and Python.
- Limitations: Requires significant programming expertise and setup time.

5.1.2.5. Mininet

- Strengths: Simple setup and easy-to-use CLI and Python API for network emulation. Strong integration with SDN controllers and virtualization platforms.
- Limitations: Less detailed than full network simulators like NS3 or Riverbed, as it focuses more on emulation.

5.1.3. Simulation Depth

5.1.3.1. Cisco Packet Tracer

- Strengths: Adequate for basic to intermediate network simulations. Provides essential tools for learning network concepts and troubleshooting.
- Limitations: Less detailed compared to GNS3 and Riverbed in terms of advanced protocol and network behavior simulations.

5.1.3.2. GNS3

- Strengths: High fidelity due to real hardware emulation. Supports a wide range of detailed network scenarios.
- Limitations: Requires substantial computing resources for complex simulations.

5.1.3.3. Riverbed Modeler Academic Edition

- Strengths: Offers in-depth and comprehensive simulation capabilities, including detailed queuing models and hardware behavior.
- Limitations: High complexity and resource requirements.

5.1.3.4. NS3

- Strengths: Provides highly detailed and accurate simulations of network protocols. Extensive support for custom simulation scripts and detailed statistical analysis.
- Limitations: Can be resource-intensive and complex to set up for large-scale simulations.

5.1.3.5. Mininet

- Strengths: Adequate for emulating network environments, particularly useful for SDN and NFV experiments.

- Limitations: Emulation rather than full simulation, which limits its ability to model network behavior in detail.

5.1.4. Flexibility and Scalability

5.1.4.1. Cisco Packet Tracer

- Strengths: Suitable for small to medium-sized network simulations. Allows for logical and physical workspace configurations.
- Limitations: Scalability is limited compared to GNS3 and Riverbed.

5.1.4.2. GNS3

- Strengths: Highly scalable and flexible, supporting integration with virtual machines and a wide range of plugins.
- Limitations: Requires advanced setup and configuration.

5.1.4.3. Riverbed Modeler Academic Edition

- Strengths: Capable of simulating large-scale networks with detailed feedback. Highly flexible in terms of protocol and hardware modeling.
- Limitations: Requires significant resources and expertise to manage large-scale simulations.

5.1.4.4. NS3

- Strengths: Extremely flexible and scalable, suitable for simulating a wide range of network scenarios. Strong support for custom extensions and detailed modeling.
- Limitations: Requires extensive programming knowledge and resources.

5.1.4.5. Mininet

- Strengths: Highly flexible for SDN and NFV emulations, allowing integration with real network controllers and virtual machines.
- Limitations: Limited scalability compared to full-scale simulators due to its focus on emulation.

5.1.5. 5.2 Comparative Insights

5.1.5.1. Cisco Packet Tracer

Best For: Beginners and intermediate learners, educational institutions focusing on foundational network training, and environments where Cisco certification is a priority.

5.1.5.1.1 Summary

Cisco Packet Tracer's simplicity and educational focus make it an excellent tool for introducing networking concepts. However, its limitations in multi-vendor simulations and advanced network behaviors mean it may not be suitable for more complex or diverse network studies.

5.1.5.2. GNS3

Best For: Advanced students, network professionals, and users seeking a realistic and flexible simulation environment.

5.1.5.2.1. Summary

GNS3 stands out for its realism and flexibility, providing a highly detailed simulation experience using real network OS images. It is ideal for users who need to simulate complex networks and advanced protocols, although its setup complexity may be challenging for some.

5.1.5.3. Riverbed Modeler Academic Edition

Best For: Higher education, research institutions, and professional training programs requiring detailed and comprehensive network simulations.

Summary: Riverbed Modeler excels in detailed and large-scale network simulations, offering extensive protocol support and detailed feedback. Its complexity and resource requirements make it more suitable for advanced users and institutional settings.

5.1.5.4. NS3

Best For: Researchers, advanced students, and professionals requiring detailed and programmable network simulations.

5.1.5.4.1. Summary

NS3 provides a highly detailed and customizable simulation environment suitable for research and advanced educational purposes. Its extensive support for custom simulation scripts and detailed statistical analysis makes it ideal for in-depth network studies. However, it requires significant programming expertise and setup time, which can be a barrier for beginners.

5.1.5.5. Mininet

Best For: Educators and students focusing on software-defined networking (SDN) and network function virtualization (NFV), and professionals needing a practical, hands-on learning tool.

5.1.5.5.1. Summary

Mininet is ideal for teaching and learning SDN and NFV concepts. It offers a practical, hands-on experience by allowing users to experiment with real network code and configurations in an emulated environment. While it is excellent for SDN and NFV scenarios, its emulation approach may not capture the nuances of complex network behaviors as comprehensively as full network simulators.

5.2. Conclusions

The comparative analysis reveals that each network simulator has its unique strengths and is best suited for different educational and professional contexts. Cisco Packet Tracer is ideal for foundational learning and beginner training due to its simplicity and focus on Cisco devices. GNS3 offers advanced simulation capabilities and flexibility, making it suitable for professional training and complex network studies. Riverbed Modeler provides comprehensive and detailed simulations, ideal for higher education and research purposes.

Network simulators are invaluable tools in both educational and professional settings, offering platforms for students and professionals to engage with complex networking concepts in a practical, hands-on manner. The choice of the appropriate network simulator is crucial and should be guided by the specific needs and goals of the user, as each simulator brings its unique advantages and limitations to the table.

Cisco Packet Tracer is particularly well-suited for introductory and intermediate networking courses. Its user-friendly interface, combined with its integration into the Cisco Networking Academy curriculum, makes it an excellent choice for institutions aiming to provide foundational networking education. Its real-time and simulation models offer flexibility in teaching methods, enabling instructors to visualize network operations and troubleshoot scenarios in a controlled environment. However, its focus on Cisco-specific devices and protocols means that it may not provide the broad exposure necessary for students who need to learn about a wider range of network environments.

GNS3, with its ability to emulate real network operating systems and integrate with virtualization platforms, provides a high level of realism that is unmatched by other simulators. This makes it an ideal tool for advanced students and network professionals who require a deeper understanding of network operations and need to simulate complex, multi-vendor environments. The flexibility and extensibility of GNS3, supported by a strong user community, allows for detailed and customizable simulations. Nonetheless, its steep learning curve and technical setup requirements can be challenging, making it less suitable for beginners.

Riverbed Modeler Academic Edition stands out for its comprehensive simulation capabilities and detailed feedback mechanisms, making it a powerful tool for higher education and professional training. Its ability to simulate large-scale networks and provide in-depth analysis of network performance is invaluable for research and development projects. The extensive protocol support and complex hardware modeling capabilities of Riverbed Modeler enable users to conduct detailed studies and gain insights into the intricate workings of network systems. However, its complexity and resource requirements may limit its accessibility to smaller institutions and individual learners.

NS3 is highly regarded in the research community for its detailed and programmable network simulations. It is ideal for researchers and advanced students who require customizable simulation scripts and detailed statistical analysis. NS3's extensive support for custom simulations makes it particularly suited for in-depth network studies. However, it demands significant programming expertise and setup time, which can be a barrier for beginners.

Mininet is an excellent tool for teaching and learning software-defined networking (SDN) and network function virtualization (NFV). It offers practical, hands-on experience by allowing users to experiment with real network code and configurations in an emulated environment. Mininet's ability to emulate real network devices and its ease of use make it particularly beneficial for students and educators focusing on SDN and NFV. While it excels in these areas, its emulation approach may not capture the full complexity of network behaviors compared to full network simulators like GNS3 and Riverbed Modeler. [17] and [18]

5.3. Future Directions

Integration into Remote Learning: Exploring how these tools can be integrated into online courses and remote learning platforms to meet the growing demand for virtual education.

Real-World Data Incorporation: Enhancing simulations with real-world data to improve their realism and applicability.

Cross-Platform Interoperability: Investigating ways to improve interoperability between different simulation tools for a more seamless educational experience.

Summarizing, Cisco Packet Tracer, GNS3, Riverbed, NS3 and Mininet each offer distinct advantages that cater to different aspects of networking education and professional training. By leveraging the strengths of these simulators, educators can create robust and effective learning environments that prepare students for the complexities of modern network management. As network technologies evolve, continuous improvement and adaptation of these tools will be essential in maintaining their relevance and utility in both educational and professional contexts [19].

References

[1] C. CasconeO'Connor, T. Vachuska, B. Davie, "Software-Defined Networks: A Systems Approach," 2021, p. 152.

[2] E. Altman, T. Jimenez, "NS Simulator for beginners," Lecture notes. [Online]. Available: http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf

[3] "Future Directions in Control, Dynamics and Systems," [Online]. Available: http://www.cds.caltech.edu/~murray/cdspanel, last accessed Mar. 2005.

[4] L. Globa, M. Skulysh, O. Romanov, M. Nesterenko, "Quality Control for Mobile Communication Management Services in Hybrid Environment," in *Advances in Information and Communication Technologies*, UKRMICO 2018, Lecture Notes in Electrical Engineering, vol 560, Springer, Cham, 2019, pp. 20-29. DOI: https://doi.org/10.1007/978-3-030-16770-7_4, Print ISBN: 978-3-030-16769-1.

[5] IETF RFC 7426, Request for Comments: 7426, ISSN: 2070-1721 EICT. Category: Informational, K. Pentikousis, Ed., 2015.

[6] J. Nilsson, "Real-time Control Systems with Delays," Ph.D. dissertation, Dept. Automatic Control, Lund Institute of Technology, Lund, Sweden, 1998. [Online]. Available: http://www.control.lth.se/documents/1998/nilj98dis.pdf

[7] J. Lam, S. Lee, H. Lee, Y. Oktian, "Securing SDN Southbound and Data Plane Communication with IBC," *Mobile Information Systems*, vol. 2016, Article ID 1708970, 12 pages, 2016. DOI: http://dx.doi.org/10.1155/2016/1708970.

[8] K. Phemius, M. Bouet, J. Leguay, "DISCO: Distributed Multi-domain SDN Controllers," Thales Communications & Security, arXiv:1308.6138v2 [cs.NI], Aug. 2013.

[9] O. I. Romanov, M. V. Oryschuk, Y. S. Hordashnyk, "Computing of influence of stimulated Raman scattering in DWDM telecommunication systems," in *2016 International Conference Radio Electronics & Info Communications (UkrMiCo)*, 2016. DOI: 10.1109/UkrMiCo.2016.7739622, eISBN: 978-1-5090-4409-2.

[10] O. Romanov, V. Mankivskyi, "Optimal Traffic Distribution Based on the Sectoral Model of Loading Network Elements," in *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Oct. 2019, pp. 8-11. DOI: 10.1109/PICST47496.2019.9061296, eISBN: 978-1-7281-4184-8.

[11] O. I. Romanov, M. M. Nesterenko, N. O. Fesokha, V. B. Mankivskyi, "Evaluation of productivity virtualization technologies of switching equipment telecommunications networks," *Information and Telecommunication Sciences*, vol. 11, no. 1, pp. 53-58, 2020. DOI: https://doi.org/10.20535/2411-2976.12020.53-58.

[12] ONF specification, "OpenFlow Table Type Patterns," Version 1.0, Aug. 2014.

[13] ONF TR-508, "Requirements Analysis for Transport OpenFlow/SDN," Version 1.0, Aug. 2014.

[14] ONF TR-537, "Negotiable Datapath Model and Table Type Pattern Signing," Version 1.0, Sep. 2016.

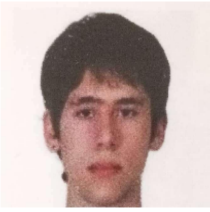
[15] ONF TR-539, "OpenFlow Controller Benchmarking Methodologies," Version 1.0, Nov. 2016.

[16] ONF TS-029, "MPLS-TP OpenFlow Protocol Extensions for SPTN," Version 1.0, June 2017.

[17] ONF, "OpenFlow Switch Specification Version 1.5.1 (Protocol version 0x06)," 2014. [Online]. Available: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[18] ONOS, "Security and Performance Analysis," Report No. 2, Nov. 2018.

[19] Open Network Foundation, "Accelerating the Adoption of SDN & NFV," 2021. [Online]. Available: http://opennetworking.wpengine.com/wp-content/uploads/2013/03/ONF%20Overview%204-pager%20FINAL.pdf



NIKOLAOS V. OIKONOMOU has received his BSc degree from University of Ioannina, Department of Informatics and Telecommunications in 2021. He has received his MSc degree from the same institution in 2023. He is an academic researcher also working as a private tutor in the field of Computer Science and mathematics. He has years of experience as a Computer Engineer, IT specialist and Network consultant. He also taught in University of Ioannina and worked as an application developer.

DIMITRIOS V. OIKONOMOU has received his BSc from University of Western Macedonia, Department of Regional and Cross Border Studies in 2024. He is currently an active research member of the University of Western Macedonia and is about to begin his MSc studies.

Received: 22 May, 2024, Revised: 19 June, 2024, Accepted: 24 June, 2024, Online: 30 June, 2024
DOI: https://dx.doi.org/10.55708/js0306004

SimulatorBridger: System for Monitoring Energy Efficiency of Electric Vehicles in Real-World Traffic Simulations

Reham Almutairi[*,1,2], **Giacomo Bergami**[2], **Graham Morgan**[2]
[1]College of Computer Science and Engineering, University of Hafr Albatin, Saudi Arabia
[2] School of Computing, Newcastle University, United Kingdom
[*]Corresponding author: Reham Almutairi, reham.mutlaq@hotmail.com

ABSTRACT: The increasing popularity and attention in Vehicular Ad-hoc Networks (VANETs) have prompted researchers to develop accurate and realistic simulation tools. Realistic simulation for VANETs is challenging due to the high mobility of vehicles and the need to integrate various communication modalities such as Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) interactions. Existing simulators lack the capability to simulate VANET environments based on IoT infrastructure. In this work, we propose SimulatorBridger, a novel simulator that bridges IoTSim-OsmosisRES with SUMO, a traffic simulator, to simulate VANET environments with integrated IoT infrastructure. Our study focuses on analyzing the generated dataflows from V2I and V2V interactions and their impact on vehicle energy efficiency. Even though On-Board Units (OBUs) appear to have insignificant energy demands compared to other vehicle energy consumptions such as electric motors or auxiliary systems (HVAC, lights, comfort facilities), we found a near-perfect correlation between the intensity of communication dataflows and the battery consumption. This correlation indicates that increased communication activity can contribute to an increase in overall energy consumption. Furthermore, we propose future research directions, including traffic rerouting based on battery consumption optimization, which can be efficiently tested using our simulation platform. By including communication energy costs in the design of energy-efficient vehicular networks, these insights contribute to a deeper understanding of energy management in VANETs.

KEYWORDS Traffic Simultors, VANET, IoT

1. Introduction

The advancement of technology leads us towards a new area of communication connecting two or more vehicles to exchange data within an IoT ecosystem. This drives us through a new VANET (Vehicular Adhoc Network) framework making the conventional transport system safer, full of infotainment, convenient, and smart. In VANET research, researchers aim to develop a simulator platform that provides *quick* and *cost-efficient* transmission of data for passenger safety and comfort. As described earlier, VANET simulators consist of two components: Traffic Simulators (software environments that generate vehicle movements in trace files) and Network Simulators (used to test the performance of networking protocols). The network simulators are used to build communication topologies, evaluate network protocols, and exchange routing information between the nodes after importing the traces of mobility models generated by traffic simulators. These mobility models, which are the depiction of real-world scenarios, are called *traces* and contain the simulated infrastructure and event information such as vehicle speed, type, origin, destination, arrival time, arrival rate, maximum density, number of lanes, speed limits, capacity, intersection type, queuing, service distribution, service rate, traffic signs, location, etc. Green and orange paths in Figure 1a provide a minimalistic view of these traces where only the vehicle's geographical position is considered. The traces generated with the mobility generator, microscopic or macroscopic [1, 2], are imported into the network simulator and generate a vehicular program: boxes surrounding paths in Figure 1b provide a depiction of the vehicular programs associated with simulated IoT agents.

1.1. Objective and motivation

We have used IoTSim-OsmosisRES [3] as the IoT simulator for our proposed coupling platform, as it is the only IoT simulator considering energy management, a variety of power sources, and network infrastructure, while others can neither simulate floating weather conditions, nor renewable energy sources. However, the current implementation of the IoT simulator, *IoTSim-OsmosisRES* assumes IoT devices to be in fixed locations. So, IoT device mobility is not supported. Hence, we propose a new simulator SimulatorBridger that couples a state-of-the-art IoT simulator (IoTSim-OsmosisRES [3]) with a traffic simulator (SUMO [4]) to simulate VANET environments. This proposed simulator is designed to meet these goals: coupling the IoT simulator IoTSim-OsmosisRES with the traffic simulator, SUMOU, enhancing IoTSim-OsmosisRES capabilities to be able to simulate moving IoT devices and to support battery information associated to the IoT devices (vehicles).

Our proposed simulator allows a smooth integration of mobility, IoT devices, heterogeneity, and battery management in highly heterogeneous and dynamic environment. Still, we demonstrate that our architecture has sufficient flexibility to bridge any possible traffic simulator using a plug-and-play approach. We provide all the necessary functionality for the accurately couple simulators that can meet the complexity of traffic environments by using the most recent IoT simulator and the first simulation framework that offers unified modelling and simulation of complex IoT applications over heterogeneous edge-cloud environments. Preliminary results (§5) remark the partial correctness of our simulator, as the vehicular battery consumption distribution follows the same trend as packet distribution. By limiting the communications to exchanging one single packet between the cloud and the IoT node, we also show that the number of communications follows a similar distribution to the former. We finally show that increasing the number of packets travelling in the network increases the overall time required by the network to process these, as bottlenecks might occur in the network simulator infrastructure. Future work will focus on testing different communication policies aiming to decrease the packet processing overhead within the network. These preliminary fidelity results remark the adequacy of using the simulator as a *digital twin* for a realistic urban mobility scenario, thus allowing any policy-maker to test disparate network configuration and traffic distributions to minimise network communication overload.

1.2. Use Case Scenario

The recent Car-as-a-Service paradigm [5] has remarked the possibility of using the cars' IoT sensors not only for the personal purposes of travellers but, with the driver's agreement, for the benefit of the entire community. As most modern cars are equipped with navigation systems and video cameras, cars can collect videos and images of their surroundings and geo-tag such information. This might be extremely useful for any town hall highly committed to road surface remaking for car safety so that they can spot tarmac conditions before they degenerate into potholes [6]. To do so, the data needs to be collected from cars and then streamed towards the main data centre, Figure 1, where an AI model is going to detect the seriousness of the situation [7]. Road-Side Units (RSU) might first collect the data from the cars through 5G antennae supporting massive low-latency communication and stream it towards the primary data centre thanks to the core networks. Communication between 5G antennae is supported by optical fibres realising low-latency communications. As a result of the data collection process, a massive amount of data will stream every second from the cars towards the data centres, which will severely increase during rush hours when road congestions might happen with higher probability [8]. Figure 2 represents a portion of the simulated traffic from Hamburg [1]. Due to the massive volume of data generated by the high number of vehicles during rush hours, any road congestion will constitute a communication bottleneck degrading the overall communication performance [9]. To achieve success in these situations, the traffic load needs to be balanced [10] to reduce communication delays. Given that the traffic flow might be redirected at run-time, this motivates the representation of each RSU as a specific Edge [3]. The exploitation of Osmotic Computation [11] eases communication flow management, thus dynamically establishing new streams. This also motivates the definition of a simulator bridging traffic simulation to network communication systems.

2. Design of the Simulator

Our framework twins traffic simulation and IoT simulation. Thus, it incorporates well-known models for road traffic micro-simulation with a comprehensive selection of models of network protocols. By running the vehicular generator, we collect all the vehicular geographical information and the position of the Edge node (§2.1). This is required to set up the network topologies associated with each Edge node and determine the vehicular program to be injected into the IoT devices in the IoT simulator (§2.2). At each simulation time, each IoT device will query a centre to determine whether the IoT device shall to establish communication with the cloud through an edge device or not (§2.3). Upon confirmation, the IoT device directly contacts the MEL router associated with the edge node of choice, establishing communication between the IoT device and the cloud through one of its hosts (§2.4).

2.1. Vehicular Data Collection

First, the simulator collects the mobility information generated by the traffic generator. This simulator might be configured by setting the begin τ_b and the end τ_e simulation time, as well as a temporal granularity δ defining the sampling rate of the vehicular information. This allows do determine the time at the i-th simulation step for each vehicle v as the following relation:

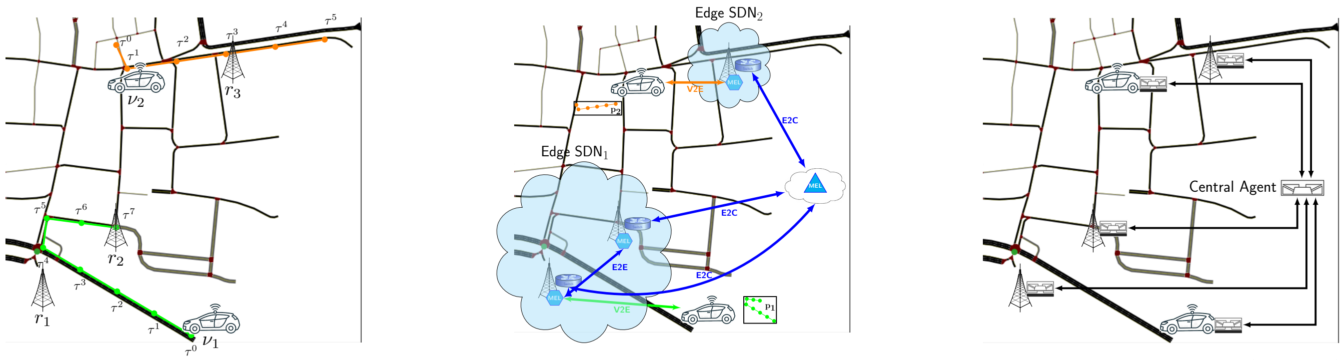
$$\tau_v^i \begin{cases} \tau_v^{i-1} \ \delta & i > 0 \\ \tau_b & \text{oth.} \end{cases}$$

At the end of the traffic generation, the VANET simulator returns a list of pairs associating each simulation time τ_v^i to a geographical location ϖ_v^i for each vehicle v. Figure 1a provides a graphical depiction of such traces, highlighted in either green or orange. From the network topology, we might also infer the geographical position of each RSU of interest r alongside its position ϖ_r and its communication radius ρ_r. This operation is performed by a simulator wrapper, which provides a standard interface for generating the data of interest in a uniform representation independently from the specific traffic simulator of choice.

2.2. Software Defined Network Configuration

The next step assumes that each vehicle v is associated with just one single IoT device potentially communicating with the cloud infrastructure. We collect the sequence of pairs

[1] https://github.com/DLR-TS/sumo-scenarios/tree/main/TAVF-Hamburg

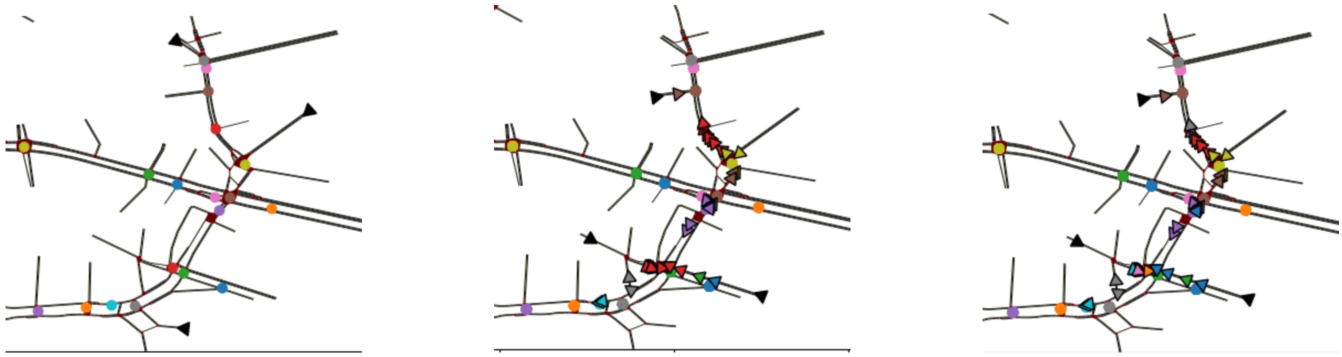


(a) Output of the SUMO simulator, associating each vehicle v to a trace (paths in green or orange) determining their location ϖ_v^i at each simulation frame τ^i. RSU r are associated with fixed geographical coordinates ϖ_r.

(b) Osmotic Applications (MEL representation) remarking the Vehicle2Edge (V2E), Edge2Edge (E2E), and the Edge2Cloud (E2C) communication channels. Boxes denoted as p_1 and p_2 associated to each vehicle indicate the "simulation programs" retaining the mobility information.

(c) Supporting the Osmotic Agent configuration from IoT-Osmosis-RES with centralized training and decentralized execution for dynamically updating the communication strategies of the IoT devices. This enables us to support dynamic load balancing in the future.

Figure 1: Osmotic Computing in car traffic scenario in Newcastle Upon Tyne.



(a) Some road lanes (black) and some RSU (colored circles). Black triangles are vehicles not associated with an RSU.

(b) Associating each semaphore with the nearest cars within the radius of θ : cars and RSU with the same colour communicate.

(c) After exploiting a load balancing algorithm, the cars are distributed to other semaphores in the neighbourhood.

Figure 2: A subset of the Sumo TAV Hamburg Dataset for mobility

$\left[\left(\tau_v^0, \varpi_v^0\right), \dots, \left(\tau_v^n, \varpi_v^n\right)\right]$, where $\tau^n \;\; \tau_e$, as a vehicular program v that will be injected within the IoTOsmosis-RES simulator.

This will allow the IoT device to update its position at each instant of the SUMO simulation time in the network simulator. Fig. 1b represents such vehicular programs as minimizations of the traces represented in Fig. 1a.

Contextually, we define Software-Defined Wide Area Networks for the RSU nodes. To do so, we first define an undirected graph $G \;\; V, E$, where the vertices V list the RSU nodes r obtained from the VANET simulator, and the edges establish a communication channel between two distinct RSU r and r' if and only if they can both communicate between each other:

$$\forall r, r' \in V. \quad (r, r') \in E \leftrightarrow r \neq r' \wedge \|\varpi_r - \varpi_{r'}\| \leq \min(\rho_r, \rho_{r'})$$

For each strongly connected component of such graph, we establish a distinct Edge SDN with an associated Datacenter and Software-Defined Network Controller, where each RSU is described as an Edge device associated with at least one host. Each light blue cloud in Fig. 1b represents a distinct Edge SDN, for which only the Edge nodes are remarked. We also set up a Cloud network towards which each IoT device will communicate to send the sensed data, as well as an SDWAN network bridging each Edge SDN to a Cloud SDN. Both these structures are engulfed in the white cloud in Fig. 1b. Last, we completely set up the edge connectivity between these networks by setting up Edge2Edge and Edge2Cloud communication links. VehicleToEdge links will be established at simulation time.

2.3. *Osmotic agents module*

The osmotic agents' module is exploited for simulating network discovery procedures and determining each IoT device which should be the best edge node for establishing the communication. This cannot be necessarily determined at the local level, as each IoT device has only the information of which edges it can communicate with, but it doesn't know which is going to be the best neighbouring agent towards which establish the first-mile communication towards for reaching the cloud. An IoT device acts as an reinforcement agent, which sends the information about the observed environment to the central agent. In particular, each IoT device sends its geographical position, while the geographical position of each edge device is known a priori. We implemented two specific types of central agents:

- NearestCentralAgent: for each querying IoT device, the central agent will always respond with the nearest Edge device to the IoT that is within mutual signal coverage. This strategy does not require any alteration to the SDN Routing Policy of each Edge Network, which might still exploit the SHORTESTPATHMAXIMUMBANDWIDTH discussed in [12].

- FlowDataCenterAgent: after associating each IoT device to a possible edge device, the agent runs a multi-source and multi-target minimum-cost flow problem for establishing the best strategy to minimize the channel communication. This algorithm will then return for each IoT device the best edge device. This also requires updating the SDN Routing Policy MaximumFlowRoutingPolicy associated with each network via its SDN Controller, thus returning the paths calculated by the Central Agent.

Upon reception of a non-empty edge device, the IoT will generate a novel Osmotic Application and start communication towards the cloud through the elected edge node. Figure 1c illustrates the bidirectional communication flow between IoT or SDN Rounters and the Central Agent.

2.4. Dynamic Adaptive Routing

In osmotic computing, each IoT device directly interacts with MicroELements (MEL) software components that might be instantiated anywhere in a given network associated to Edges [3]. To faithfully represent real communication scenarios, an IoT device must directly establish communication with a precise edge node [13]: the Adaptive Routing in [3] was further extended to directly select the MEL associated with a given edge node instead of picking one in a round-robin fashion. As IoT-OsmosisRES further associates a MEL to one or more hosts, the routing algorithm picks one of the hosts running the MEL as its first-mile communication point. Contextually, the host establishes the communication channel with the cloud network indicated by the IoT device, thus starting the communication.

3. Implementation in Java

The proposed simulator, SimulatorBridger, is programmed using the Java programming language, and it is available in Github [2]. The NetworkTopology.java class in the SimulatorBridger-IOTSimOsmosisRES module is crucial for setting up the network layer in the simulation. It generates a topological network, which is used to simulate latency in network traffic. The IoTDevice.java class represents IoT devices within the IoTSimOsmosisRES framework. The CloudletScheduler class is responsible for scheduling tasks in the cloud. And EdgeDeviceManager class is managing edge devices.

The implementation of the SimulatorBridger platform contains several modules and a large number of Java source files. Here's an overview of its structure:

SimulatorBridger-core component contains shared dependencies and core functionalities used across the project. It serves as the foundational codebase that other modules in the project rely on.

SimulatorBridger-traffic-information-collector involves running the traffic simulator and collecting data from it. It processes the output of the simulation, identifying IoT nodes and Edge nodes. The Edge nodes are used for interactions by the IoT nodes.

SimulatorBridger-central-agent-planner provides a theoretical omniscient algorithm capable of scheduling time as required. Depending on the network simulator chosen, it also generate potential network connectivity information based on the IoT and Edge information provided by the simulation.

SumoOsmosisBridger is an example that bridges all the simulations together with Dynamic IoTSimOsmosisRES simulator. It illustrates how the components can be integrated in a seamless manner.

Each of these components plays a specific role in integrating and processing traffic data within the IoTSimOsmosisRES environment. The process starts with data collection and processing, followed by planning and scheduling using the central agent planner, and finally, the integration of all these elements through the SumoOsmosisBridger to work with Dynamic IoTSimOsmosisRES. This integration allows for a comprehensive simulation environment that can handle complex IoT and traffic data scenarios.

4. Dataset and Experiment Setup

This section covers running the Bologna dataset [3], through our proposed simulator, where vehicles have an embedded IoT device and RSU are Edge nodes associated to MELs. Fig. 3 shows the SUMO network of this dataset. The dataset has 16 RSUs and 239 vehicles. The Bologna scenario includes the area around the football stadium and was set up to simulate the mobility of big events such as football matches or concerts.

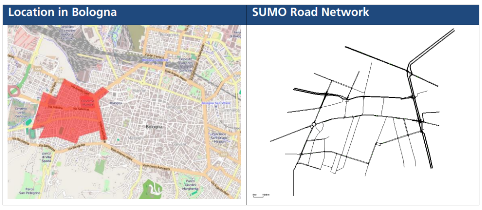


Figure 3: Bologna Dataset [14]

Our experiments orchestrate a traffic simulator (SUMO) with a network simulator (IoTOsmosis-RES) with each simulation second in the former corresponding to a second in the latter. We arbitrarily set the IoT battery consumption rate to 1.02% when a vehicle communicates with a MEL. We assumed embedded IoT devices cannot be re-charged to better analyse the simulator's correctness by correlating the number of packets sent by an IoT device with its battery consumption. When a vehicle was near an RSU in the traffic simulator, we scheduled a new communication between the IoT device embedded in the vehicle and the Cloud via the Edge node representing the RSU in the network simulator. This schedule is selected according to the granularity δ of the traffic simulator: in our scenario, this is set to start every 1 second. In this time frame, the IoT device in the vehicle sends one single packet. The simulation time for these plots begins at the start of the traffic simulator and ends when the last MEL has successfully sent all the packets to the cloud.

[2] https://github.com/jackbergus/SimulatorBridger/releases/tag/v0.1
[3] https://github.com/DLR-TS/sumo-scenarios/tree/main/bologna/acosta

Our simulator assumes each RSU is an Edge device containing multiple MELs. As per the previous discussion, the network simulator was extended to establish direct communication between IoT and Edge devices by resolving an available MEL associated with the Edge device. A round-robin policy selects an available MEL in the Edge node to ease each MEL's workload. For this experiment, the maximum simulation time of IoTOsmosis-RES was 3898.4s, corresponding to 100% of the simulation, starting at the same time, the maximum time for the SUMO simulator was 186.7s (τ_e), with vehicles no longer able to enter the urban environment after around 55% of the overall traffic simulator time, or 100s. From this, as the number of vehicles in the simulation decreases, the number of starting communications will also decrease, resulting in a normal distribution for both the number of communicating vehicles and the number of packets sent. The traffic simulation takes less than 5% of the total simulation time. This is due to packet network delays which delayed communication patterns even though the IoT devices sent no new packets. We consider all communications between an IoT vehicle and a MEL.

5. Analysis and Results

In this section, we analyze and discuss the key findings from the Bologna dataset experiment conducted on the SimulatorBridger platform. The results of our simulation are meticulously examined, focusing on critical performance metrics and the effectiveness of various strategies implemented throughout the process. This analysis not only highlights the successes and challenges encountered but also provides valuable insights into the practical implications and potential improvements for future iterations.

Fig 4 shows that each of the 163 embedded IoT devices (in distinct colours) shows an almost perfect correlation between the number of packets sent (solid line) and its battery consumption (dashed). The Spearman correlation between those for each vehicle is almost 1 , with a p-Value of 0 , thus indicating a very strong correlation between these two. From Fig 5, for most of the RSUs the number of connected to each has an approximate normal-like distribution, with most plateauing after the 100s mark, which is at about the 55% mark in VANET simulation time mentioned earlier. A maximum number of 39 IoT devices were connected to a single MEL and its Edge node at any time. This figure also explicitly shows vehicles starting at an RSU, as Edge#12 has 15 connected vehicles from the very beginning of the simulation. This figure also shows the simulation time used for these plots, the simulation time starts at time 0 seconds and then stops around 190 seconds once all the RSUs have no more connected vehicles. Next, we tested if there was a strong correlation between the number of vehicles in the simulation within a given time interval, with the overall battery consumed within that same time interval, as common-sense suggests that an increase of the number of the vehicles should match an increase of communications between a IoT and Edge devices, thus reflecting in an increase of the overall battery consumption. We found no strong correlation between these two aspects as, even though more vehicles lead to an increase of communications, this does not necessarily entail that either more communications should take place (e.g., vehicles might not be in a region covered by RSUs) or even that all vehicles have the same number of communications within the same time interval.

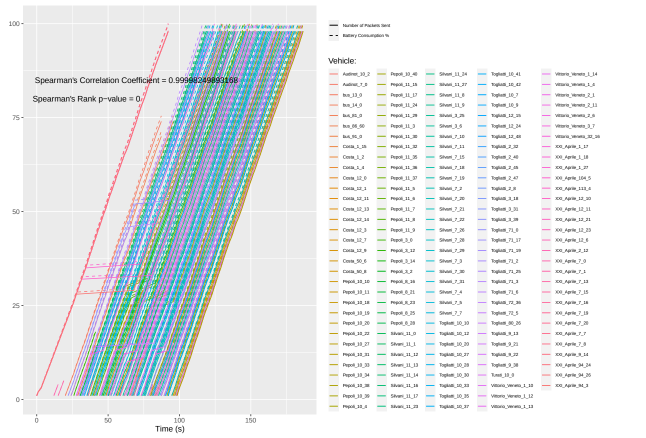


Figure 4: Correlation between battery consumption and number of packets being sent per vehicle within the simulation.

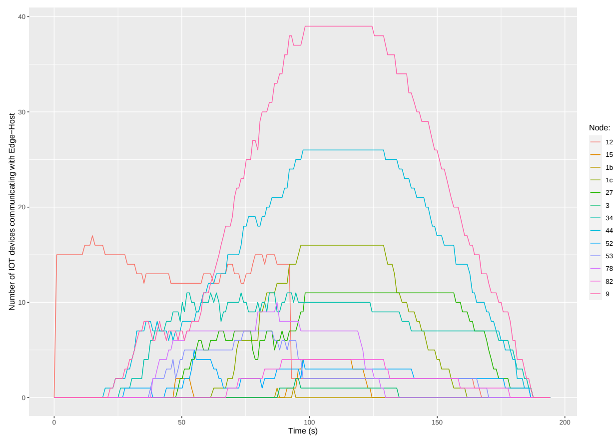


Figure 5: Number of IoT devices communicating with an Edge per simulation time.

Following this, any potential correlation between packets sent from vehicles and the overall battery consumption was investigated. Fig 6 shows the probability density of the overall battery consumed by all vehicles in the simulation (within each 5% time interval of simulation time), along with probability density of the number of packets sent by all vehicles (always within each 5% time interval of simulation time). From Fig 6 both sets of data not only closely follow a normal distribution, but also are closely correlated with each other, as both distributions and both normal plots completely overlap one another. These two variables correlating with each other was expected, as in the current simulator setup, the packets being sent are responsible for the battery consumption, and the fact that we found these two variables do in fact correlate indicates that the simulation is behaving as expected. Fig 6 shows a spike occurring at the 35 − 45% for both the overall consumption and the number of packets sent. This spike can be explained in terms of Fig 7 showing the distribution of the communications starting within each traffic and VANET simulation time interval: this spike is also present for the number of starting communications within this same time interval. Given the current simulator settings of a vehicle sending a single packet as a result of communication with an RSU, this likefor-like behaviour of communications and packets sent is not only expected but demonstrates the simulator is functioning correctly. The reason more packets were sent and therefore why more battery was consumed in this time interval is due to more communications occurring within this time interval. Fig 7 also validates our preliminary hypothesis observing that the number of starting communications should drop after a 55% elapse of the Sumo simulation time.

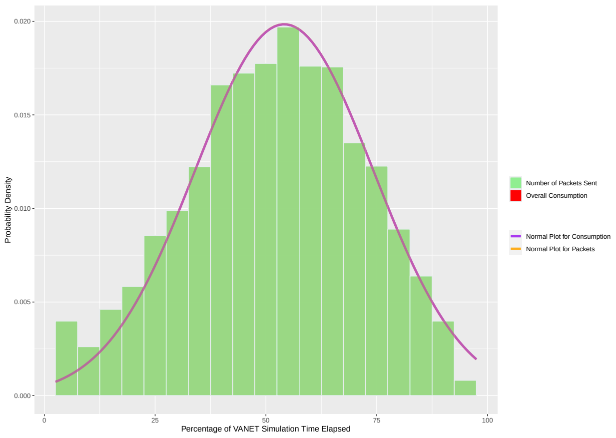


Figure 6: The distribution of packets sent in the network follows the same trend as the overall IoT battery consumption.

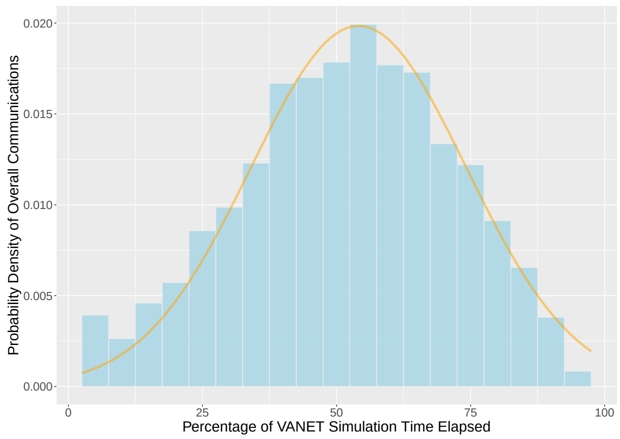


Figure 7: The distribution of the number of communications starting within each time interval.

We might also observe another spike at the beginning of the simulation where the influx of traffic starts entering the simulation while approaching RSUs, thus justifying the increase of the overall number of communications. The explanation for the initial drop-off between the first two intervals, $0-5\%$ and 510%, can be explained with the first interval being artificially high as a node is placed very close to where vehicles initially join when the simulation starts. This behaviour can also be seen on Fig 5 as Edge#12 having 15 connected vehicles just 1 second into the simulation.

From Fig 8 vehicles at the start of the simulation enter from the most southwestern entrance, which, when factoring in the 100 m radius of communication of the nodes, explains how vehicles are able to connect to Edge#12 (i.e. the light blue Node 12 in the bottom left of the plot). After 190 seconds in the simulation, the Sumo simulation stops, and therefore we observed no vehicles connected to any of the RSUs. In fact, the IoTOsmosis-RES still has to wait for all the Edge nodes to send their packets to the cloud while receiving an acknowledgement for this. The discrepancy between the end of the traffic simulator and the Network simulator also matching the end of the VANET simulator has been cut off for the sake of legibility. This processing of all the packets by the network infrastructure therefore takes over 3700 seconds: we refer to this as *shutdown time*. Fig 9 shows how this shutdown time is affected by the number of total communications in the simulator: the more time steps in the simulation, or the longer vehicles are allowed to navigate in the traffic simulator, the more the communications between vehicles and RSUs take place. This matches the intuition that more time for potential communications entails an increase in the number of communications. The more relevant result from this figure with respect to the shutdown, however, shows that more communications lead to a longer and longer shutdown time.

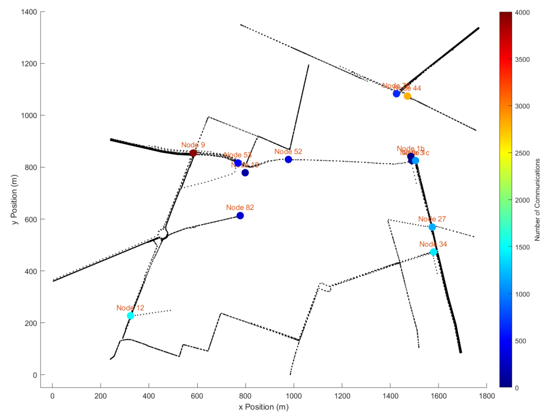


Figure 8: Showing Edge displacements as RSU in the Bologna Dataset: their colour represents the intensity of the undergoing communication. Black lines show the trajectory of the vehicles carrying embedded IoT devices.

Hence, the effectiveness of SimulatorBridger was validated through both simulation and experimental tests using the Bologna dataset. The simulation results demonstrated a near-perfect correlation between packet transmission and battery consumption, with a Spearman correlation coefficient close to 1 and a p-value of 0, indicating strong statistical significance. Experimental tests mirrored these findings, showing similar trends and validating the simulator's accuracy in modeling real-world communication patterns and their impact on energy usage. Minor discrepancies between the simulation and experimental data were observed, primarily due to real-world variances such as sensor inaccuracies and environmental factors. Statistical tests confirmed that both datasets followed similar distributions, with most errors within acceptable limits. Overall, the strong alignment between simulation and experimental results underscores the reliability of SimulatorBridger. These findings demonstrate the robustness of SimulatorBridger and its value as a tool for studying and optimizing vehicular networks.

Finally, the results offer significant improvements over previous simulation benchmarks. The correlation analysis between packet transmission and battery consumption reveals a Spearman correlation coefficient close to 1, indicating a strong relationship between communication activity and energy usage. Enhanced traffic flow and load balancing capabilities are achieved through dynamic management of communication loads among RSUs, resulting in smoother traffic flow and reduced bottlenecks. The scalability of the simulator is demonstrated by its ability to manage extensive simulations, addressing the scalability issues faced by earlier systems. Additionally, the detailed energy management features, considering various power sources and their consumption rates, set SimulatorBridger apart from traditional models that often overlooked these aspects. These findings affirm the simulator's effectiveness in providing a robust platform for studying and optimizing vehicular network performance and energy efficiency.

6. Conclusion and future works

Due to the high mobility of vehicles in VANETs, realistic simulation is a challenging task. This paper proposes a novel simulator, SimulatorBridger, by bridging the IoT network simulation with the traffic simulation. The efficacy of SimulatorBridger is validated using a case study for urban roads in Bologna city. Results show the various capabilities of SimulatorBridger in terms of vehicular network lifetime, vehicle battery, and energy consumption. According to our results, SimulatorBridger is also scalable in terms of vehicle count and simulation time. Notably, SimulatorBridger simulates VANET based on IoT infrastructure, a capability not present in existing VANET simulators, providing a more comprehensive and integrated approach to vehicular network simulation. Furthermore, our framework can be easily extended to support a load balancing scheme between traffic lights, thus minimising load imbalances in the communication network while improving energy management and implementation time. A cooperative approach for load balancing among the network might be used when the traffic light receives more requests than the maximum number, resulting in high traffic or load. Also, vehicular network lifetime is increased by reducing energy consumption which is necessary to balance energy in traffic lights. In addition, the simulator we propose combines IoTSim-OsmosisRES with any potential traffic simulator. Then for future work, a different traffic simulator can be used in conjunction with IoTSim-OsmosisRES. Fig. 9 suggests that the network infrastructure is severely affected by the communication delays by the way it processes and transmits the packets received from the IoT devices to the cloud. Our future work will identify the specific causes of the long shutdown times and alleviate any problems in the infrastructure to reduce the shutdown time of the simulation. For instance, we will try whether different packet routing policies or network configurations might help mitigate such problems. In conclusion, our study demonstrates the impact of communication dataflows on the battery consumption of IoT devices in vehicles, highlighting a significant correlation that is crucial for understanding overall energy efficiency in VANETs. The findings validate the effectiveness of SimulatorBridger in providing accurate simulations of VANET environments based on IoT infrastructure. Future research could explore traffic rerouting strategies based on battery optimization criteria, leveraging the capabilities of our proposed simulator.

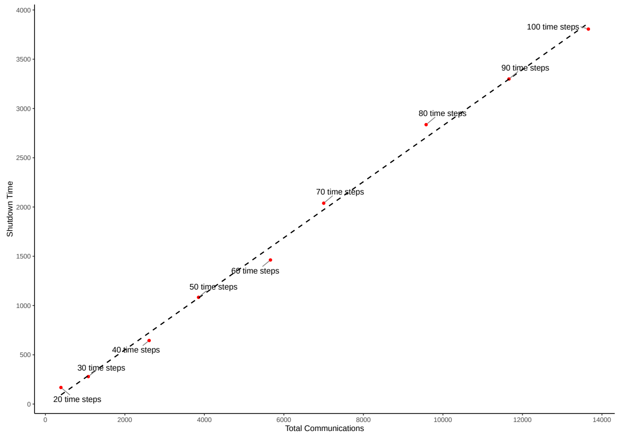


Figure 9: Correlation between simulation time, number of undergoing communications, and their effect to the shutdown time.

Conflict of Interest The authors declare no conflict of interest.

References

[1] B. Boukenadil, "Importance of realistic mobility models for vanet network simulation", *arXiv preprint arXiv:1410.2450*, 2014.

[2] J. Harri, F. Filali, C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy", *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 19–41, 2009, doi:10.1109/SURV.2009.090403.

[3] T. Szydlo, A. Szabala, N. Kordiumov, K. Siuzdak, L. Wolski, K. Alwasel, F. Habeeb, R. Ranjan, "Iotsim-osmosis-res: Towards autonomic renewable energy-aware osmotic computing", *Software: Practice and Experience*, vol. 52, no. 7, pp. 1698–1716, 2022, doi:10.1002/spe.3084.

[4] P. Á. López, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. WieBner, "Microscopic traffic simulation using SUMO", "ITSC", pp. 2575–2582, IEEE, 2018, doi:10.1109/ITSC.2018.8569938.

[5] A. Dahlinger, B. Ryder, F. Wortmann, "Car as a sensor. paying people for providing their car data", "Proceedings of the 5th International Conference on Internet of Things, Seoul, South Korea", 2015, doi: 10.3929/ethz-b-000106154.

[6] R. A. Eaton, R. H. Joubert, E. A. Wright, "Pothole primer – a public administrator's guide to understanding and managing the pothole problem", Special Report Vol. 81–21, US Army Corps of Engineers – Cold Regions Research & Engineering Laboratory, 1989.

[7] M. H. Asad, S. Khaliq, M. H. Yousaf, M. O. Ullah, A. Ahmad, "Pothole detection using deep learning: A real-time and ai-on-the-edge perspective", *Advances in Civil Engineering*, vol. 2022, p. 9221211, 2022, doi:10.1155/2022/9221211.

[8] M. Saleem, S. Abbas, T. M. Ghazal, M. Adnan Khan, N. Sahawneh, M. Ahmad, "Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques", *Egyptian Informatics Journal*, 2022, doi: 10.1016/j.eij.2022.03.003.

[9] N. Baranasuriya, V. Navda, V. N. Padmanabhan, S. Gilbert, "Qprobe: Locating the bottleneck in cellular communication", "CoNEXT", 2015, doi:10.1145/2716281.2836118.

[10] J. Kang, X. Chen, D. Wu, Y. T. Xu, X. Liu, G. Dudek, T. Lee, I. Park, "Hierarchical policy learning for hybrid communication load balancing", "ICC 2021 - IEEE Int. Conf. on Communications", pp. 1–6, 2021, doi:10.1109/ICC42927.2021.9500379.

[11] M. Villari, M. Fazio, S. Dustdar, O. Rana, R. Ranjan, "Osmotic computing: A new paradigm for edge/cloud integration", *IEEE Cloud Computing*, vol. 3, no. 06, pp. 76–83, 2016, doi:10.1109/MCC.2016.124.

[12] K. Alwasel, D. N. Jha, E. Hernandez, D. Puthal, M. Barika, B. Varghese, S. K. Garg, P. James, A. Zomaya, G. Morgan, *et al.*, "Iotsim-sdwan: A simulation framework for interconnecting distributed datacenters over software-defined wide area network (sd-wan)", *Journal of Parallel and Distributed Computing*, vol. 143, pp. 17–35, 2020, doi:10.1016/j.jpdc.2020.04.006.

[13] S. Häger, S. Böcker, C. Wietfeld, "3d self-motion tracking services: Coalescence of mmWave beam orientations and phase information", "2022 IEEE FNWF", 2022, doi:10.1109/FNWF55208.2022.00009.

[14] L. Bieker, D. Krajzewicz, A. Morra, C. Michelacci, F. Cartolano, "Traffic simulation for all: a real world traffic scenario from the city of bologna", "Modeling Mobility with Open Data: 2nd SUMO Conference 2014 Berlin, Germany, May 15-16, 2014", pp. 47–60, Springer, 2015, doi:10.1007/978-3-319-15024-6_4.