STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–23.
Published online in International Academic Press (www.IAPress.org)

Enhanced Electricity Demand Forecasting through Metaheuristic Optimization of Model Parameters

Ibtissam Lahjili [1,*], Aziz Lmakri[1], Mustapha Hain[1], Hassan Oukhouya[2]

[1] *AIMCE Laboratory, ENSAM, Hassan II University of Casablanca, Morocco*
[2] *LaMSD, MSASE, FSJES, Mohammed First University of Oujda, Morocco*

Abstract Accurate prediction of the fluctuating nature of electricity demand remains a persistent challenge, primarily due to the complexity of distribution systems. This paper provides metaheuristic optimization to enhance state-of-the-art prediction methods. We conducted a comparative study between SARIMAX, which proved to be effective for trends and seasonality as well as the impact of exogenous variables, and the GRU deep learning model, which captures complicated non-linear dependencies. Both the models were optimized with Genetic Algorithm (GA), a metaheuristic approach for efficient search in solution space. The effect of optimization was also tested by comparing the performance with and without GA. First, the results using the real dataset showed that SARIMAX was better than GRU. In the optimized version, the SARIMAX-GA model's predictive capability and understandable variance improved significantly compared to the GRU-GA model.

Keywords Metaheuristic Optimization, Genetic Algorithms, Electricity Forecasting, Deep Learning, GRU, SARIMAX

DOI: 10.19139/soic-2310-5070-3094

1. Introduction

The electricity sector sustains the economies of modern societies in residential, commercial, and industrial processes [46]. Reliable electricity supply forecasting is essential to guide energy consumption toward optimal utilization. Effective forecasting is important, as it sustains reliable and sustainable energy systems [41]; disruption to power and energy sectors may have devastating outcomes on economic stability and social well-being [50]. Electricity's high volatility makes it difficult to cope with, especially in parts of the world where growth in economy and population brings about greater dissimilarities in climatic conditions [35]. In such frameworks, once anticipation diminishes, the importance of developing optimal modeling techniques grows. For the comprehensive stability of operations in various aspects of the power systems, forecasting electricity demand with a high degree of accuracy is critical to providing smarter ways to supply consumption requests [11], so that resource allocation can be improved even as blackout or overproduction risk is lessened dramatically. Accurate load prediction also provides the data necessary for infrastructure strategy planning in terms of power generation [42], grid design [43], and legislation, contributing to economic viability and environmental accountability. But to achieve such accuracy, a minefield of multiple connected variables (like weather patterns and seasonal trends or sudden changes in demand behavior) is required to be navigated through with one's decisions constantly changing the course of demand dynamics. All these variables are totally interconnected. When one variable changes, it can disrupt the functioning of other variables. That's why if we want to make accurate predictions, we need models that can pick up on these tricky relationships and take into account both the quick shifts and the long-term trends [15]. If we overweigh those rigid approaches or make them too simplistic, we may miss some important connections, and that can screw up how accurate we are with our predictions. All of

[*] Correspondence to: Ibtissam Lahjili (Email: lahjli.ibtissam@ensam-casa.ma). AIMCE Laboratory, ENSAM, Hassan II University of Casablanca, 150 Bd du Nil, Casablanca, Morocco, (20700).

ISSN 2310-5070 (online) ISSN 2311-004X (print)

this is why we need a smarter form of forecasting—one that acknowledges not just that real-world systems can be unpredictable but also how to improve our expectations and maximize flexibility when faced with radical uncertainty. By tapping into the latest techniques, we can boost our predictive performance while keeping things efficient and practical [2].

In this paper, we investigate the proposal of improving classical and deep learning (DL) techniques based on genetic algorithm (GA) for electricity demand forecasting using a Moroccan dataset. In particular, we benchmark the classical method Seasonal AutoRegressive Integrated Moving Average with eXogenous factors (SARIMAX) and compare it to literature on time series forecasting from the past decades, capturing trends, seasonality, and external influences on both traditional and deep learning approaches, such as the Gated Recurrent Unit (GRU), which is effective for handling complex nonlinear relationships over long periods. To make our forecasts even better, we apply GA for optimization, which helps us efficiently search through the options and fine-tune the parameters of both models. This optimization allows us to compare relative model success, with and without the GA adjustments, and a substantial improvement in prediction success becomes clear. The study shows the potential for substantially better forecasting by combining optimization methods. The study also illustrates how these advanced methods provide more accurate forecasts of electricity demand, which provide better decision support for power system operation.

The remainder of the paper is structured as follows: Section 2 offers a critical overview of previous research and work on our topic. Section 3 illustrates the methodologies used to analyze and predict the time series of national electricity demand. In section 4, we present a brief description of the dataset used and the training process of the chosen models. Section 5 describes their empirical results and discussions comparing the models with and without GA optimization. Ultimately, section 6 ended the paper with such implications and recommendations for further studies.

2. Related Work

Electricity demand forecasting has attracted much attention, with methods evolving from traditional statistical methods to sophisticated machine learning [17] and ensemble methods. Previous studies indicate that the complementarity between preprocessing (decomposition, variable selection), adaptive models, and hyperparameter optimization is necessary to improve the accuracy, robustness, and transferability of forecasting models. Several recent contributions illustrate the effectiveness of hybrid and optimized approaches. Li et al. propose a framework integrating CatBoost and XGBoost coupled with modern optimizers (Salp Swarm Algorithm (SSA), Parallel Particle Swarm Optimization (PPSO), and Grey Wolf Optimizer (GWO)) and demonstrate that the XGBoost–SSA hybrid achieves outstanding performance for net load forecasting in Turkey ($R^2 \approx 0.998$, very low MAPE), confirming that the ensemble model + metaheuristic combination can finely capture consumption fluctuations and peaks [30]. Consistently, comparative and review studies show a clear trend toward hybrid solutions: Ugbehe et al. provide a comprehensive overview of methodologies (statistical, ML/AI, and hybrid) and identify key challenges, including data quality, rigorous validation, and operational integration of forecasts [49]. Methodologically, the use of decomposition and multi-scale feature extraction techniques is recurrent. For instance, Hu & Zheng introduce a Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN)–Fuzzy Entropy (FE)–Bidirectional Gated Recurrent Unit (BiGRU)–Attention pipeline where CEEMDAN decomposition and fuzzy entropy measurement reconstruct multi-scale sequences before training in an attention-based BiGRU; gains in MAPE and RMSE are substantial on their test sets, showing the value of well-identified frequency segments to capture nonlinear dynamics [24]. Similarly, Gao et al. use SSA-optimized Improved Variational Mode Decomposition (IVMD) coupled with Incremental Extreme Learning Machine (IELM)-Adaboost for medium-term forecasting, demonstrating the added value of decomposition to reduce noise and produce relevant confidence intervals [19]. The static/dynamic integration of linear and nonlinear components is a central theme: Nontapa et al. show that a hybrid decomposition–Seasonal Autoregressive Integrated Moving Average with eXogenous regressors (SARIMAX)–Artificial Neural Network (ANN) model can combine the robustness of linear components (SARIMAX) Similarly, Baek & Seo apply a SARIMAX–Long Short-Term Memory (LSTM) framework for a university campus, showing that the inclusion of academic predictors (study days, calendar) significantly enhances accuracy and the operational value of monthly forecasts [4]. Another significant contribution is represented by metaheuristics for hyperparameter tuning. Bacanin et al. propose a backtracking search algorithm–sine cosine algorithm approach to automatically optimize LSTM and GRU on multivariate series, such as solar, wind, and load, with measurable gains from MSE and SHapley Additive exPlanations-based interpretability analysis in identifying dominant variables [3]. Chen et al. combine an Enhanced Inception-V4 architecture with an improved Osprey Optimizer, applied over 2003–2023, showing that besides a better predictive performance, demand is markedly sensitive to climatic variables, highlighting the importance of selecting relevant exogenous inputs [11]. Several hybrid comparative studies confirm that hybrid metaheuristics reduce the tendency toward local minima and improve the convergence of optimization algorithms [48, 36, 47]. On the application side, contributions span diverse contexts: residential, commercial, university, and national. Botman et al. present a global probabilistic approach for household forecasting based on similar series selection and empirical quantile estimation. In particular, this method is well suited for environments with concept drift and requires few historical data—only 7 days—and therefore can be deployed at a large scale by grid operators [6]. Pallonetto et al. examine building-level cases and show how LSTM and Support Vector Machine may serve operational purposes—such as demand response—given data availability and granularity [39]. Leme et al. show a national-level case in Brazil where ensemble approaches applied to a unified public dataset deliver robust daily and monthly forecasts. Given such consolidated datasets, generalization is possible, [29]. More fundamental works appear to compare architectures and trade-offs: Son et al. compare DNN and LSTM and show how DNN can outperform LSTM on specific consumer profiles in terms of accuracy and computational cost [45], while Manowska illustrates the ability of LSTM to provide robust long-term forecasts—up to the horizon of 2040—when rich datasets are available. Jiang et al. and Aguilar Madrid & Antonio present composite frameworks (optimized SVR, XGBoost) useful for noise reduction and short-term forecasting, showing that non-deep approaches remain competitive in certain contexts [26, 1]. Finally, a number of limitations emerge from the literature: (i) dependency on multi-year data for validating geographic and temporal generalizability; (ii) high computational cost of metaheuristics for performing the tuning on large multivariate datasets; (iii) variability of results with respect to exogenous variable choice and validation procedures (rolling origin vs. simple split); (iv) the necessity for uncertainty and interpretability analyses (e.g., SHAP) to foster operational adoption [3, 33, 49]. These observations undoubtedly motivate studies that combine rigorous optimization (ablation/algorithm comparisons), robust temporal validation, and the provision of reproducible artifacts (codes, datasets, editable figures)—objectives that our study specifically aims to address by providing a comparative evaluation of a statistical model and a deep learning model, both optimized by using GA, and systematically documenting parameter choices and statistical tests.

Building on the insights and advancements from these researches, our research activities seek to undertake a comparative study between the traditional approach, such as SARIMAX, and sophisticated deep learning structures—GRU—with the usage of metaheuristics of the kind of optimization as involving the usage of the Genetic Algorithm with the aim of boosting the level of the performance as well as the level of accuracy for the forecasting of electrical demand in Morocco. The following part depicts the methods used in this research as well as the research design and the procedure in steps.

3. Methodology

This section presents our methodology for analyzing and forecasting data on national electricity demand over time.

3.1. Deep Learning and Classical methods

3.1.1 Gated Recurrent Unit

Before defining the GRU, it is important to understand the concepts of recurrent neural networks (RNN) and LSTM. RNNs are super popular for handling sequential data. They're used a lot in things like time series forecasting and natural language processing [31]. In RNNs, each temporal step is linked to the previous one, which means that a history of information can be preserved. However, for long sequences, traditional RNNs suffer from a sophisticated 'vanishing gradient problem' [22]. During training, the gradients that were calculated to update the architecture weights end up being tiny. That makes it difficult to fine-tune the weights in the first layers, and they are therefore unable to store long-term information, making it impossible to optimize the processing of long signal dependencies.

To resolve this limitation, LSTM networks have been proposed in [23]. Through memory cells and gates (input, output, and forgetting) that better filter the flow of information and better manage temporal dependencies over long sequences [20], they are very effective in solving the problem of the vanishing gradient. However, the model faces complexity challenges from its increased architectural complexity and its increased need for parameters that lead to learning delays and resource requirements.

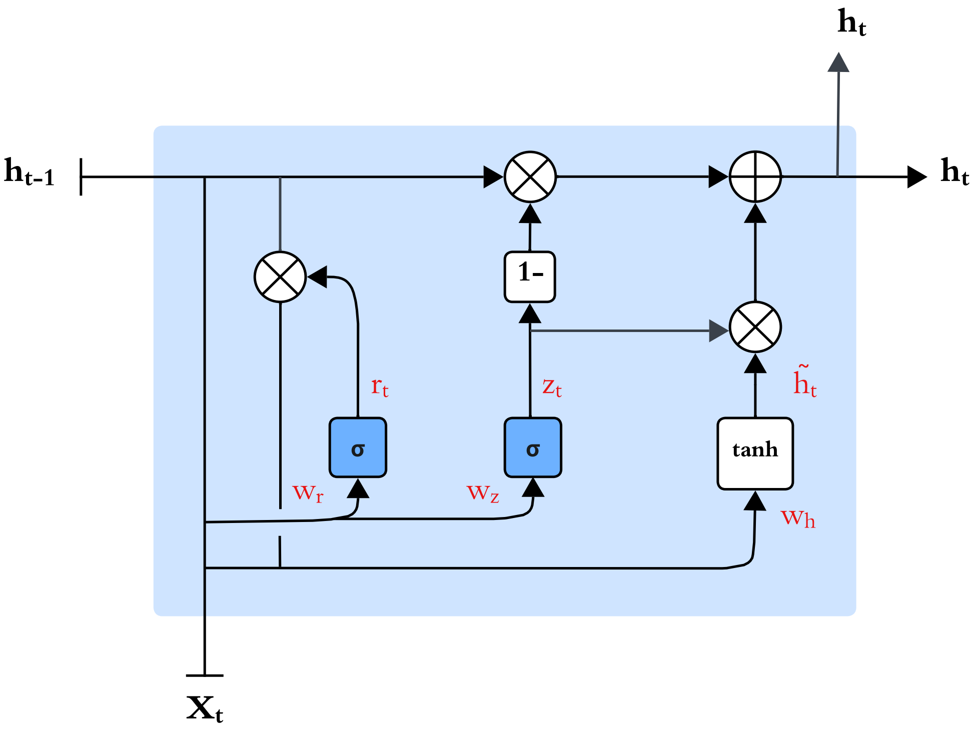

Figure 1. Structure of a GRU.

The GRU [12], a gate system in RNNs, has been suggested as a simplified alternative to LSTMs with the same ability to capture long-term dependencies. The GRU architecture also features a gated mechanism to control information flow. On the other hand, it does not have separate memory cells, which helps to reduce the total number of parameters and therefore optimize efficiency [13].

The GRU unit consists of three essential components that include both an update gate and a reset gate together with the current memory content space [16]. These gates enable the GRU to selectively update and use information from past time steps; they will enable capturing long-term dependencies in sequences. Figure 1 represents the GRU unit architecture, which resembles the design from [21].

The update gate is defined as follows:

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z), \tag{1}$$

where σ is the sigmoid activation function, W_z is the weight matrix, and b_z is the bias. The reset gate is calculated as:

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r), \tag{2}$$

The current memory content is computed as:

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t]), \tag{3}$$

Finally, the final memory state h_t is determined by the combination of the previous hidden state and the candidate activation:

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t. \tag{4}$$

In addition, the GRU-type model captures the long-term dependencies of the series well while being less complex than other models. Given its current performance and simplicity, the GRU-type model is an appropriate choice for forecasting.

3.1.2 SARIMAX Models

The SARIMA model extends ARIMA models to handle univariate time series data that contains seasonal characteristics [8]. The model adds new hyperparameters that specify autoregression (AR) parameters and differentiate (I) and moving average (MA) characteristics of the seasonal element alongside a parameter to determine seasonal period. Consequently, the SARIMA$[p, d, q, P, D, Q, s]$ model is expressed as follows [14]: With p, d, and q representing the orders of the non-seasonal autoregressive (AR), differencing (I), and moving average (MA) components, respectively; P, D, and Q denoting the orders of the seasonal AR, I, and MA components; and s corresponding to the seasonal period.

$$Y_t = \alpha + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \sum_{i=1}^{P} \Phi_i Y_{t-is} + \sum_{j=1}^{Q} \Theta_j \varepsilon_{t-js} + \varepsilon_t \tag{5}$$

where Y_t is the value of the observation at time t, α is a constant, (ϕ_i and θ_j) are the coefficients of the non-seasonal AR and MA terms, (Φ_i and Θ_j) are the coefficients of the seasonal AR and MA terms, and ε_t is white noise with zero mean and finite variance.

Building on this, the SARIMAX model is an extension of the SARIMA model that incorporates external variables, also known as exogenous variables, into the modeling process. This enhancement allows for the consideration of additional variables that may impact the time series. Thus, a SARIMAX model is defined by the same parameters as the $SARIMA[p, d, q, P, D, Q, s]$ model, with the addition of exogenous regressors X_t. The general equation for the SARIMAX [9] model is given as follows:

$$Y_t = \alpha + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \sum_{i=1}^{P} \Phi_i Y_{t-is} + \sum_{j=1}^{Q} \Theta_j \varepsilon_{t-js} + \sum_{l=1}^{k} \gamma_l X_{t,l} + \varepsilon_t \tag{6}$$

where $X_{t,i}$ are the values of the exogenous regressors at period t, and γ_i are the coefficients associated with these regressors.

The selection of the SARIMAX model is achieved by taking the appropriate Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plotting as the first step to decide on the model building stage, followed by unit root testing of the data using Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS). If the time series is not stationary, differencing is used to convert it into a stationary series. Then, model parameters $[p, d, q, P, D, Q, s]$ are determined, and an Ordinary Least Squares (OLS) parameter estimation is performed. Model comparison is accomplished via different methods, such as Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) [7]. The acceptability of the model can then be verified by proper datasets that further analyze the metrics: This is the practical process of use of the SARIMAX model in Figure 2.

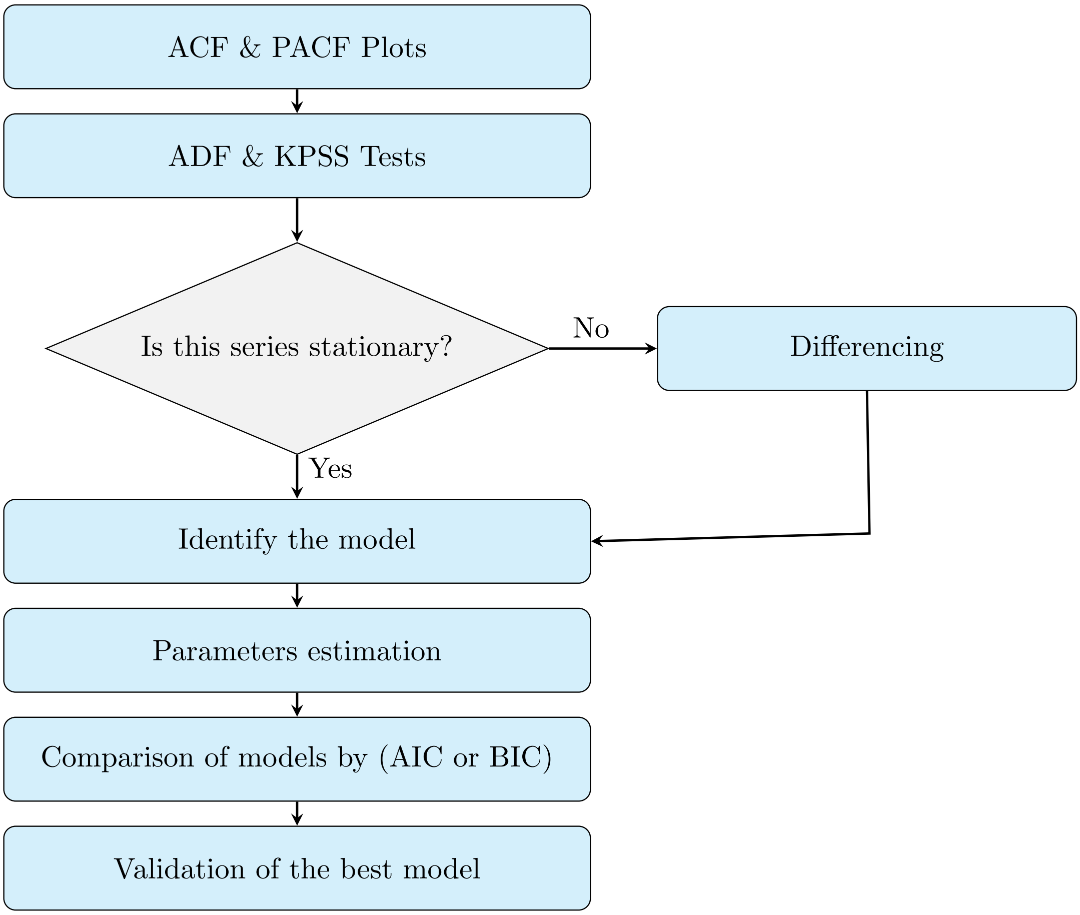
ACF & PACF Plots
ADF & KPSS Tests
Is this series stationary?
No
Differencing
Yes
Identify the model
Parameters estimation
Comparison of models by (AIC or BIC)
Validation of the best model

Figure 2. Process flow of SARIMAX model application.

The SARIMAX model is one of the most robust forecasts, which is an essential step in the modeling process. At the end of this phase, it is possible to assess the performance of the other models.

3.2. Metaheuristic Optimization

3.2.1 Genetic Algorithm

GA [34] is a metaheuristic optimization method based on some principles of natural selection. It uses a population-based search method whereby the ”survival of the fittest” theory is employed. An iterative application of genetic operators on individuals of the current population produces a new generation of population. The principal ingredients of GA are chromosome representation, selection, crossover, mutation, and fitness function evaluation.

The steps of voting [28]: Population initialization first constructs a population (Y) of (n) chromosomes by random or some other method. After constructing the chromosomes in (Y), their fitness must be evaluated. Two chromosomes (or another number if desired) (C_1, C_2) are selected from the population Y on the basis of their fitness for replication: these are taken to be the fittest genomes, and they provide the basis for the next generation. A single-point crossover operator is applied to C_1 and C_2 with a crossover probability (C_p) to generate the offspring, O, which is mutated with uniform mutation probability (M_p) to yield the newly generated offspring, O', which is then added to the next generation. When the next generation is formed, the selection-crossover-mutation process is repeated on the current population until a full generation of new offspring is produced. The process is depicted in Figure 3 and is as follows:

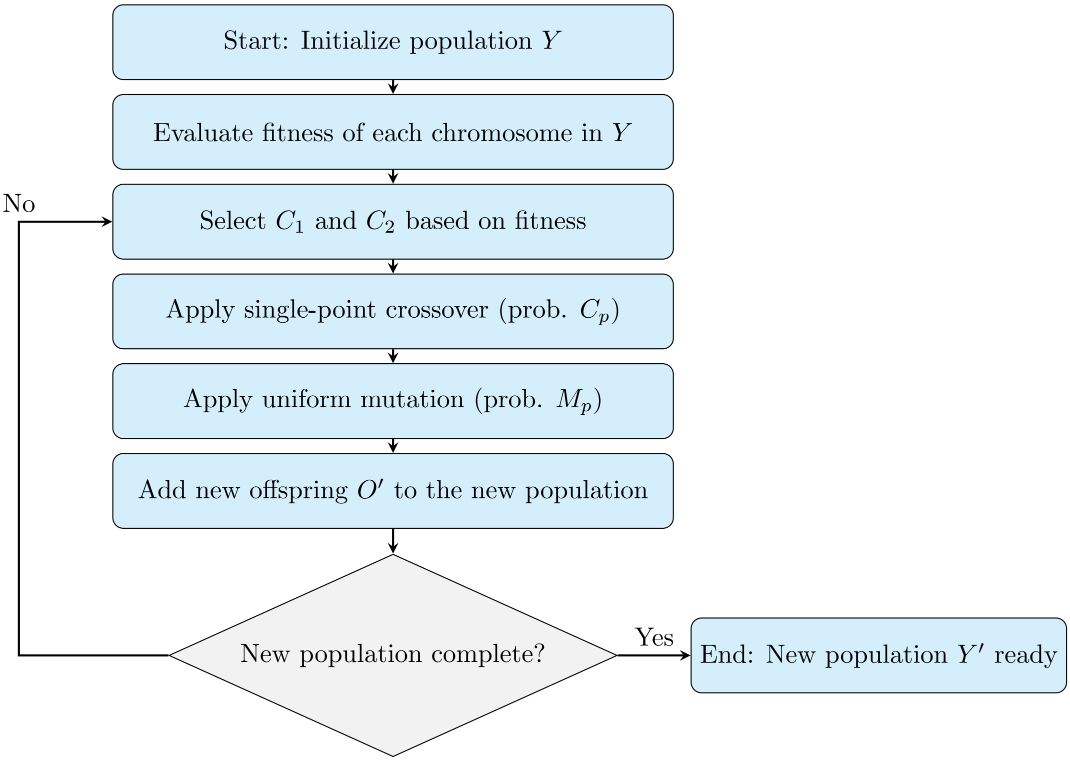
Start: Initialize population Y
Evaluate fitness of each chromosome in Y
Select C_1 and C_2 based on fitness
Apply single-point crossover (prob. C_p)
Apply uniform mutation (prob. M_p)
Add new offspring O' to the new population
New population complete?
No
Yes
End: New population Y' ready

Figure 3. Process flow of GA.

As a metaheuristic optimization method, GA is particularly effective for solving complex problems with large solution spaces, where traditional optimization techniques may struggle [27].

3.2.2 GRU optimized by GA

The optimization of GRU models using GA, called GRU-GA, is an effective method for enhancing the performance of these models. GRU-GA helps identify the best model configurations for specific tasks [25]. The GRU-GA process follows several iterative steps as illustrated in Figure 4:

- **Initialization:** The optimization starts through initial population generation, which contains various GRU model configurations such as layer number, neuron count, activation function, learning rate, batch size, and epoch number. The predefined ranges define limits for selecting parameters at random, which creates a wide exploration of model structures.
- **Evaluation:** The candidate solutions are ranked by how well each one learned the data based on the Mean Squared Error (MSE) as the cost function. The MSE is computed as a means of measuring the discrepancy between the predicted and the actually observed values from the training set. The lower the MSE, the better the model performs. This step thus lets us rank all the solutions based on how well they learned the data.
- **Selection:** After evaluating all the solutions once, the best-performing solutions, which are also known as elites, are chosen for becoming the parents of the new generation. The elites' MSE values reach minimum values, indicating the maximum predictive accuracies from the alternatives. The algorithm performs well for the good models by virtue of the selection process, which promotes well-crafted models into the new generation of iterations.
- **Crossover:** The crossover operation helps diversify genetic material by combining chosen elite solution hyperparameters. A crossover operation uses elite solution hyperparameters to create new candidate solutions by joining these parameters. Through component exchange the algorithm creates new configurations that potentially yield improved performance compared to their existing predecessors.

- **Mutation:** During mutation we add this stage to stop early convergence together with keeping diversity in the population. This phase makes random adjustments to particular hyperparameters in the newly created solutions. Through these modifications the algorithm explores different areas within the search space; thus, it prevents getting stuck at local optimum points.
- **New Generation:** The new population emerges through uniting the superior solutions from earlier generations together with recently made offspring. The iterative design process transforms the GRU model framework through several generational cycles, which gradually enhances its hyperparameters.
- **Stopping Criterion:** The optimization process concludes when the predefined stopping condition gets fulfilled. The optimization process needs to continue until it reaches the predefined number of generations that demonstrate minimal performance enhancement. When the stopping condition meets its requirements, the best-performing GRU solution becomes the final configuration tested for predictive tasks.

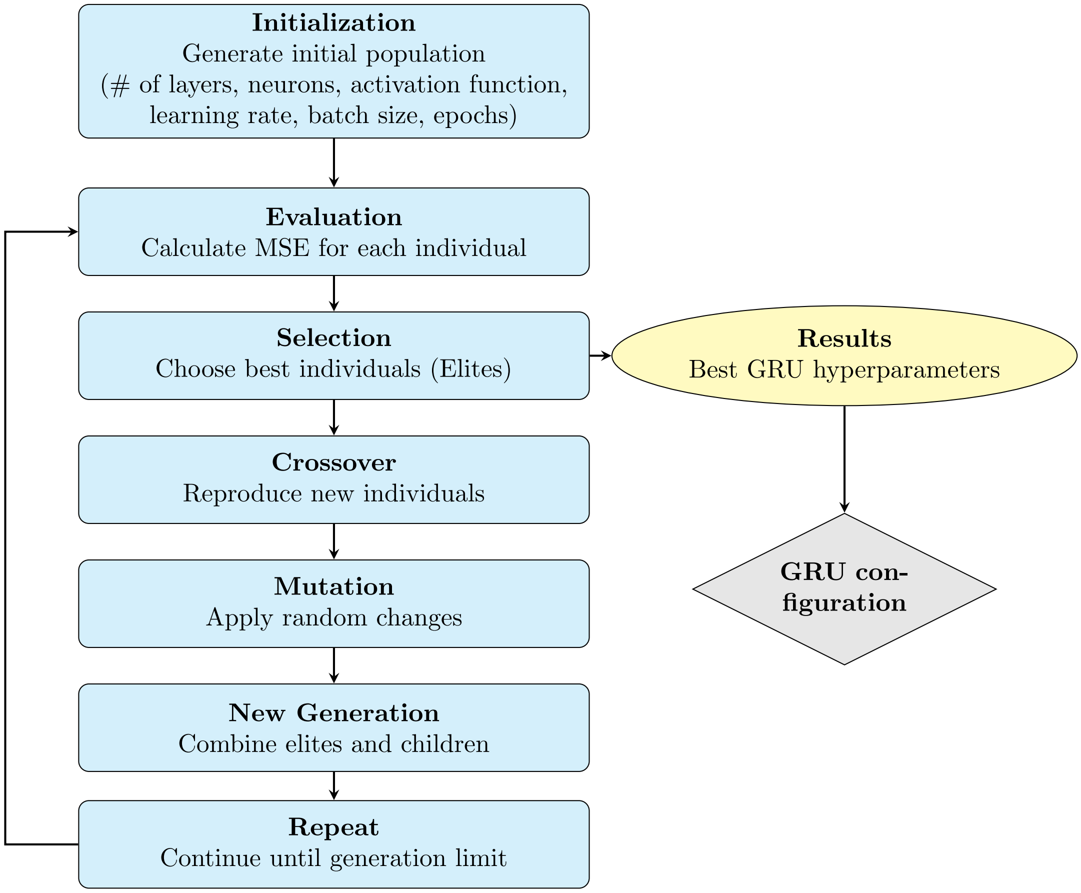
Initialization
Generate initial population
(# of layers, neurons, activation function, learning rate, batch size, epochs)
Evaluation
Calculate MSE for each individual
Selection
Choose best individuals (Elites)
Crossover
Reproduce new individuals
Mutation
Apply random changes
New Generation
Combine elites and children
Repeat
Continue until generation limit
Results
Best GRU hyperparameters
GRU configuration

Figure 4. Diagram of GRU optimization steps by GA.

The GRU-GA framework comes with multiple favorable aspects when applied [37]. By performing worldwide parameter searches, the system raises opportunities to discover the best GRU model configuration. The model performance becomes progressively better throughout every generation because GRU-GA utilizes natural ideas to conduct selection and mating and mutation from optimal solutions. The optimization process uses MSE as a definitive performance metric to guide all model performance operations.

3.2.3 SARIMAX optimized by GA

The integration of GAs with SARIMAX models provides a powerful approach for enhancing time series forecasts by optimizing model parameters. SARIMAX models are effective in capturing seasonal trends and the effects of exogenous variables [32]. Optimization using a GA unfolds in several steps, illustrated in Figure 5:

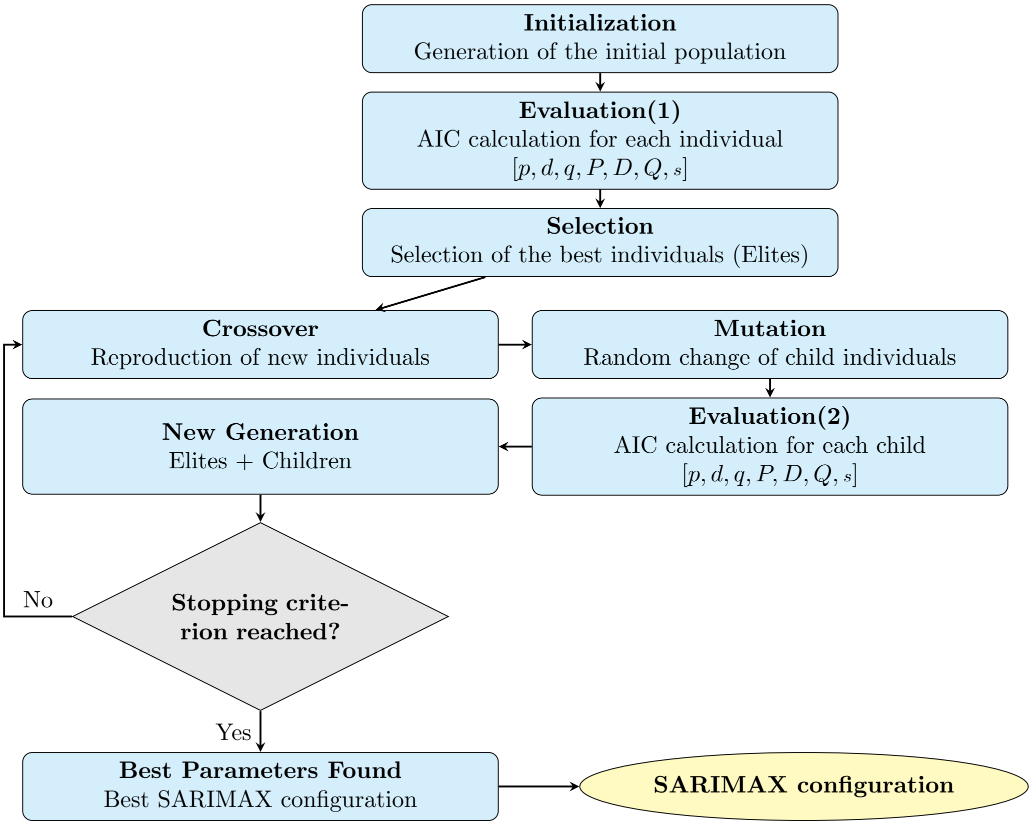
Initialization
Generation of the initial population
Evaluation(1)
AIC calculation for each individual
$[p, d, q, P, D, Q, s]$
Selection
Selection of the best individuals (Elites)
Crossover
Reproduction of new individuals
Mutation
Random change of child individuals
New Generation
Elites + Children
Evaluation(2)
AIC calculation for each child
$[p, d, q, P, D, Q, s]$
No
Stopping criterion reached?
Yes
Best Parameters Found
Best SARIMAX configuration
SARIMAX configuration

Figure 5. Diagram of SARIMAX optimization steps by GA.

- **Initialization:** The optimization procedure starts by creating the first set of candidate solutions as an initial population. The proposed solutions represent different combinations of SARIMAX parameters that are represented in the form of $[p, d, q, P, D, Q, s]$. Random selection of parameters from defined ranges throughout initialization serves to discover optimal models accurately by expanding the search scope.
- **Evaluation (1):** The fitness function evaluates solutions composed of different SARIMAX parameter values $[p, d, q, P, D, Q, s]$. Various models can utilize AIC to serve as an evaluation metric. The AIC quantifies how model simplicity compares with its ability to match data patterns to select better-performing models through lower values of AIC. The evaluation step arrays solutions from best to worst according to their modeling performance of time series data.
- **Selection:** The evaluation step chooses the optimal solutions, referred to as elites, which create the starting components for producing the following generation. The current generation can identify *elite* solutions that demonstrate the best performance levels based on the AIC evaluation. Through the selection process the evolutionary method continues to focus on improving accuracy by transferring the most accurate model components from the current generation to the next generation.

- **Crossover:** During this step selected solutions create new solutions through exchanging different sections of their parameter values. Through parameter combination and solution population diversity enhancement, this process enables the evaluation of new parameter sets.
- **Mutation:** After crossover occurs, the mutation phase makes random changes to the available solutions. The modification process aims to discover fresh solutions while preserving diversity by maintaining options within the population to prevent remaining stuck at poor solutions.
- **Evaluation (2):** Once offspring are generated through crossover and mutation, the evaluation process establishes the performance assessment for new offspring through measurements with AIC. The evaluation process selects the eligible offspring to continue becoming future generational members.
- **New Generation:** The new generation is formed on the basis of the best-selected individuals as well as the offspring produced in the previous steps. This generation is made by mixing the well-performed individuals together with their offspring in such a way as to maintain a proper compromise between exploration and exploitation. Then it takes a new run, which will be evaluated for the next selection and evolution process.
- **Stop Criterion (Number of Generations):** The specified stop criterion sets a predetermined generation number that marks the termination of the execution. Repetition of the selection and crossover, then mutation stages, is required before stopping if the specified criterion has not been achieved. The optimization process completes its execution when the established number of generations reaches completion. The SARIMAX model receives its final configuration from the best solution that emerges from life cycles.

The SARIMAX-GA model is advantageous on various counts. First, it thoroughly investigates hyperparameters, allowing for an extensive search for optimal configurations of the SARIMAX model. With its diverse usefulness, it can parallel execute complex problems for a range of data types, within which can fit time series with quite a variety of intricate patterns. The process of artificial selection, crossover, and mutation performs an iterative refinement that optimizes the SARIMAX hyperparameters for improved forecasting results.

4. Experimental Settings

4.1. Data Description

The dataset used in this study was obtained from the Electricity Division of the ONEE[†] in Morocco. The data included 8,760 hours of national electricity consumption figures for the whole year, from January 1, 2023, 00:00, to December 31, 2023, 23:00, in kilowatt-hours (kWh). The available dataset had different variables, such as the following:

- National Electricity Consumption: The main target variable represents National Electricity Consumption as the total electricity usage across the nation during each hourly period.
- Hour: The specific time of each observation was shown by the Hour category to enable researchers to study daily consumption patterns.
- Day Type: This variable distinguished between weekdays and weekends, assigning numeric values to different days: Monday was assigned 1, Working Day (grouping Tuesday, Wednesday, and Thursday) was assigned 2, Friday was 3, Saturday was 4, and Sunday was 5. This categorization helped capture demand variations based on the day of the week, with working days showing similar consumption patterns.
- Temperature: The hourly average temperature (in °C), as temperature fluctuations significantly impacted electricity demand.

Table 1 and 2 present, respectively, a sample of the dataset and key descriptive statistics. For improved clarity, Figure 6 shows the national electricity demand over the entire study period, while Figure 7 provides a detailed view of one month to illustrate the distribution patterns in national electricity demand.

[†] Office National de l’Électricité et de l’Eau Potable `http://www.one.org.ma/`

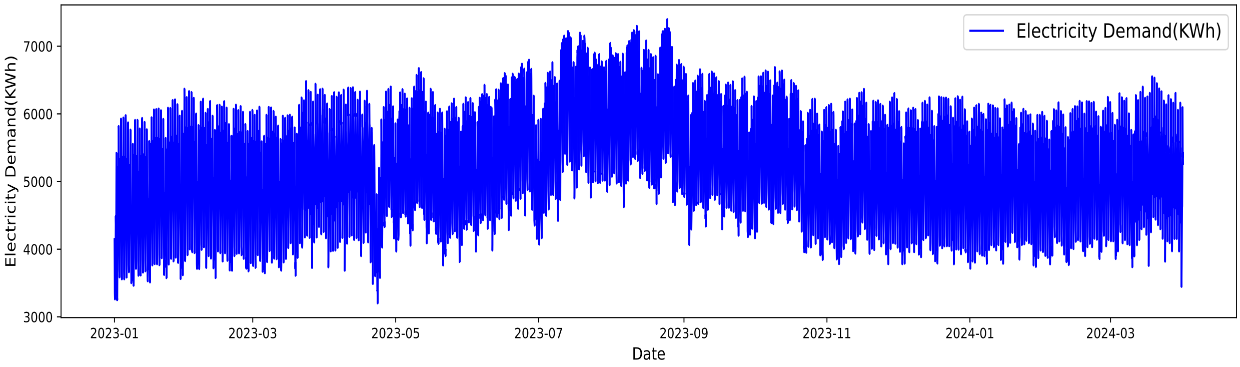

Figure 6. Temporal evolution of the data over the given period.

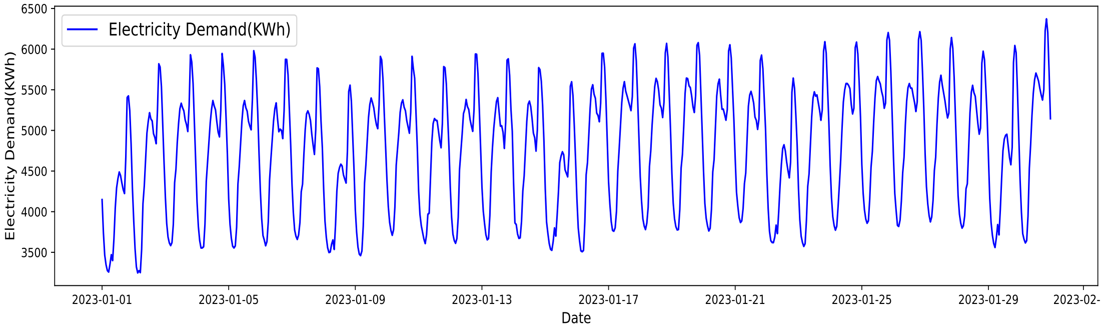

Figure 7. Temporal evolution of data over a January month.

Table 1. Sample of electricity demand and temperature data for January 1, 2023.

Date-time	Hour	Day Type	Electricity Demand (KWh)	Temperature (°C)
2023-01-01 00:00:00	0	5	54148.0	13.3
2023-01-01 01:00:00	1	5	53744.0	13.4
2023-01-01 02:00:00	2	5	53466.0	13.0
2023-01-01 03:00:00	3	5	53347.0	12.8
2023-01-01 04:00:00	4	5	53276.0	12.1
...	...	...	...	...

Table 2. Descriptive statistics of the dataset.

Statistic	Count	Min	Max	Mean	Std
Electricity Demand	8760	3199	7488	5317.2477	794.5542
Temperature (°C)	8760	4.8772	37.9896	19.8480	5.6589
Day Type	8760	1	5	2.7205	1.2816

Overall, the chosen dataset is, therefore, a very suitable base for modeling, as it has relevant patterns and variations. The following subsection will describe how the models were trained with this dataset.

4.1.1 Time Series Characteristics and Stationarity Analysis

To ensure the suitability of the electricity demand series for time series analysis, a stationarity check was performed using the KPSS test. The original series was found to be non-stationary, highlighting the presence of trends and seasonal variations. To stabilize the series, a first-order differencing was applied. The KPSS test on the differenced series yielded a statistic of 0.0106 with a p-value of 0.1, exceeding the 5% significance level. This indicates that the differenced series is stationary. Table 3 summarizes these results.

Table 3. KPSS stationarity test results before and after first-order differencing.

Series	KPSS Statistic	p-value	Stationarity
Original Series	4.6410	0.01	Non-stationary
First-order Differenced Series	0.0106	0.1	Stationary

Following the stationarity analysis, the autocorrelation (ACF) and partial autocorrelation (PACF) functions of the electricity demand series were examined to explore temporal dependencies. To ensure stationarity, first-order differencing was applied as determined in the previous analysis. Figures 8 and 9 present the ACF and PACF plots of the differenced series, which were used to identify candidate non-seasonal and seasonal AR and MA orders for the SARIMAX model.

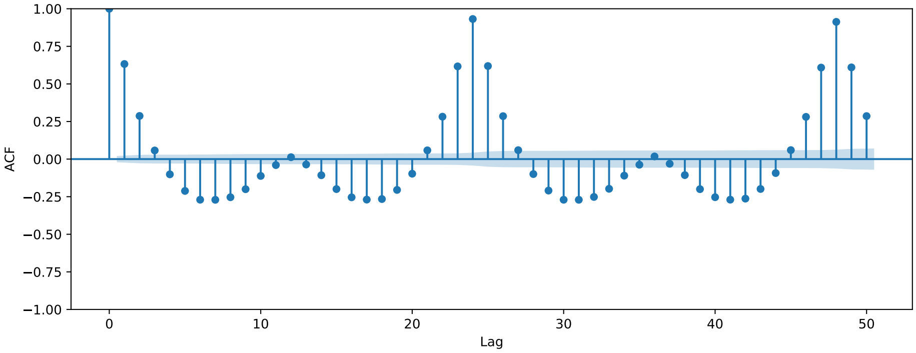

Figure 8. Autocorrelation function (ACF) of the differenced electricity demand series.

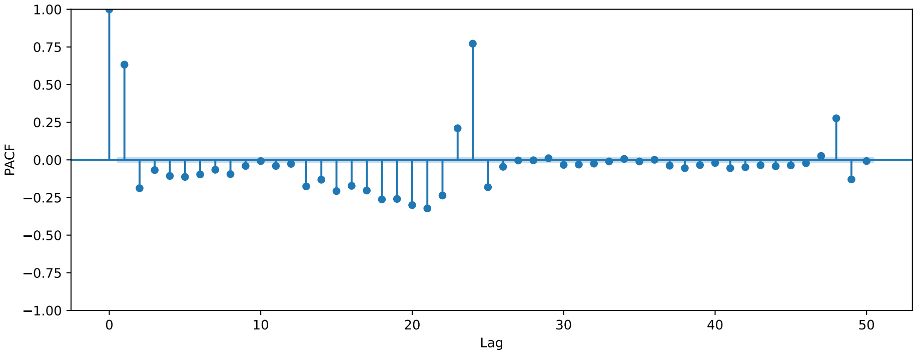

Figure 9. Partial autocorrelation function (PACF) of the differenced electricity demand series.

The ACF plot exhibits a perfect correlation at lag 0, followed by a series of significant positive peaks at short lags (0–4), which indicates a strong short-term inertia in the electricity demand series. Pronounced spikes are also observed at lags 24 and 48, reflecting a dominant daily periodicity that persists within the broader weekly cycle ($s = 168$). Similarly, the PACF displays a dominant peak at lag 1 after the initial lag 0, confirming the presence of essential non-seasonal autoregressive effects. This is immediately followed by significant negative partial autocorrelations at lags 2 and 3, suggesting a short-term corrective mechanism or a ”return-to-the-mean” behavior in demand fluctuations. Furthermore, the PACF shows significant seasonal spikes at lags 24 and 48, highlighting a deep temporal memory where current demand is strongly influenced by the previous two days.

Overall, the coexistence of these significant peaks at both short-term and seasonal scales (24 and 48 hours) confirms the complex seasonal structure of the data, justifying the consideration of a weekly period in the subsequent analysis.

4.2. Models Training and Forecasting Process

Firstly, in general, training the models requires following the steps illustrated in Figure 10.

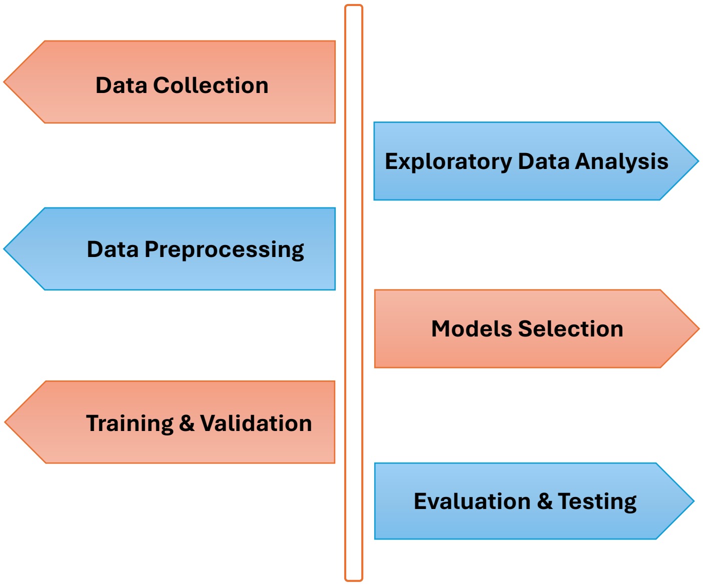

Figure 10. Model training process.

For our study, we collected a dataset provided by ONEE in Morocco, which comprises historical national electricity consumption and other variables influencing this consumption. We preprocessed this data as part of preliminary processing by removing duplicates and outlying points and by appropriate treatment of missing values. We obtained an initial understanding of this dataset through an exploratory data analysis to analyze its distribution, trends, and relationships. We tested different time periods and performed multiple experiments to define the training data based on these observations. The period from January 1, 2023, 00:00 to November 30, 2023, 23:00, was selected by cross-validation in order to have a stable train-test split and to cover seasonal variations and production shifts. The data from December 1 to December 31, 2023, was retained as a test set to validate models on new data. We then moved to model selection to ensure those selected models would efficiently extract intricacies in the dataset. We selected GRU and SARIMAX, as those models are most capable of extracting seasonality, exogenous variables, and time dependencies. These models were fitted to the dataset after it had been prepared. These The hyperparameters were also optimized to ensure peak performance through the genetic algorithm. Lastly, to find out the reliability and accuracy of the predictions, models were evaluated through MSE, MAE, MAPE, and R², as they provide a summary measure of performance based on electricity consumption prediction.

On the other hand, this section also presents the architectures of the models that are used for this study, designed to forecast electricity demand. These architectures are carefully selected to capture the data complexity and seasonality.

4.2.1 GRU Architecture

The architecture and hyperparameters of the GRU model were selected following a structured, step−by−step Grid Search procedure. First, a predefined search space was defined for each key hyperparameter based on the recommendations of Fahd et al. [18] and preliminary experiments. This search space included the number of GRU layers (1−3), activation functions (ReLU, tanh, sigmoid), number of neurons per layer (10–100), learning rate (10^{-5}–10^{-2}) , batch size (10–100), number of training epochs (10–100), and optimizer (Adam, RMSprop).

Subsequently, all possible combinations within these ranges were evaluated by training the GRU model on the training set and assessing its performance on the validation set. Model performance during Grid Search was evaluated on the validation set using the Mean Square Error (MSE). The optimal configuration was selected based on the minimization of this validation error. The final GRU configuration obtained through this Grid Search is summarized in Table 4.

Table 4. GRU model configurations selected via Grid Search.

Parameter	**Range Explored**	**Selected Value**
Number of layers	1-3	2
Activation function	ReLU, tanh, sigmoid	ReLU
Neurons per layer	10-100	62 (1st layer), 84 (2nd layer)
Learning rate	$10^{-5} – 10^{-2}$	0.00927
Batch size	10-100	69
Number of epochs	10-100	48
Optimizer	Adam, RMSprop	Adam

The selected GRU architecture consists of 2 stacked layers, using the ReLU (Rectified Linear Unit) activation function in all the layers, which enables effective modeling of nonlinear temporal relationships. The first and second layers contain 62 and 84 neurons, respectively. The learning rate was set to 0.00927, with a batch size of 69, and the model is trained for 48 epochs using the 'adam' solver, learning the weights by adaptively varying the learning rate depending on the gradient estimates to improve the rate of convergence and performance.

4.2.2 SARIMAX Architecture

The selection of the SARIMAX model parameters was based on a comprehensive statistical analysis of the time series for electricity demand. This analysis depended on the stationarity tests and the study of the autocorrelation (ACF) and partial autocorrelation (PACF) functions, which are shown in Section 4.1.1. To address the non-stationary behavior seen in the original series, a first-order differencing ($d = 1$) was applied, effectively removing the underlying trend component. In addition, a seasonal differencing term ($D = 1$) with a seasonal period of 168 hours (which is the same as a weekly cycle of 7 × 24 hours) was added to the data to show the strong weekly seasonality that was found.

Building on the insights provided by the ACF and PACF plots in Section 4.1.1, the orders of the SARIMAX model were subsequently specified as follows:

- Non-seasonal AR(p) and MA(q) terms were selected by examining the PACF and ACF plots. Candidate values for the AR order were $p \in \{0, 2, 3, 4\}$, based on PACF peaks at lags 0, 1, 2 (negative), 3, and 4, while candidate values for the MA order were $q \in \{0, 1, 2, 3, 4\}$, based on ACF peaks at lags 1–6. For simplicity and to avoid overfitting, the final non-seasonal orders were selected based on validation performance and residual diagnostics.

- Seasonal AR (P) and MA (Q) orders were initially inferred from the seasonal PACF and ACF plots. Candidate values were $P \in \{0, 1\}$ and $Q \in \{0, 1\}$, with a seasonal period $s = 7 \times 24$ (weekly data). The final seasonal orders were chosen after validation and residual diagnostics to ensure the model captures weekly seasonality without unnecessary complexity.

Table 5. SARIMAX model configurations.

Parameter	**Value**
p (Auto-regressive order)	2
q (Moving average order)	0
d (Differencing)	1
P (Seasonal auto-regressive order)	0
Q (Seasonal moving average order)	0
D (Seasonal differencing)	1
s (Seasonal period)	168(7 × 24)

The final model architecture was determined to be SARIMAX $[p = 2, d = 1, q = 0, P = 0, D = 1, Q = 0, s = 168]$, as shown in Table 5. These parameters were empirically determined through a systematic diagnostic process, guided by the visual analysis of the ACF and PACF plots. This selection by experts makes sure that the model is simple but still captures both the short-term autoregressive dynamics (the dependencies found at lags 1 and 2) and the prominent weekly seasonal cycle ($s = 168$). Furthermore, the inclusion of both first-order (d=1) and seasonal (D=1) differencing ensures the stationarity of the series, providing a robust foundation for accurate forecasting.

4.2.3 GRU-GA Architecture

To optimize the GRU, we configured the GA with the hyperparameters presented in Table 6.

Table 6. GA hyperparameters for GRU.

Hyperparameter	**Value**
Population size	10
Number of generations	3
Number of elites	3
Mutation rate	0.2
Crossover probability	0.7

The genetic algorithm (GA) optimization process was conducted using a population of 10 individuals developed over three generations, preserving three elite solutions at each iteration to maintain the top-performing candidates. A mutation rate of 0.2 was applied to ensure sufficient exploration of the search space and avoid premature convergence, while a crossover probability of 0.7 facilitated effective recombination of solutions. Several preliminary experiments were also performed using larger populations (20-50 individuals) and longer evolution (5-15 generations). These configurations increased computational time without improving performance. Consequently, the final GA configuration was chosen because it was the best balance between convergence quality and cost. To optimize the GRU model's performance, ranges were defined for its hyperparameters, allowing the GA to search for an optimal configuration. The best hyperparameter values found during the optimization process and a summary of these search ranges are shown in Table 7.

Table 7. GRU hyperparameters and best values.

Hyperparameter	Range/Values	Best Value
Number of GRU layers	1 to 3	3
Neurons per layer	10 to 100	44(1st layer), 62(2nd layer), 88(3rd layer)
Activation function	`ReLu`, `tanh`, `sigmoid`	`ReLu`
Learning rate	10^{-5} to 10^{-2}	0.0086
Batch size	10 to 100	27
Number of epochs	10 to 100	29

The optimization of the GRU model had identified the best configuration for key hyperparameters. The model had been performing optimally with three layers of GRUs, which had been chosen from a set of 1 to 3. The first, second, and third layers had 44, 62, and 88 neurons, respectively, which had been chosen from a set of 10 to 100. The activation function of the form ReLU had been chosen as the optimum of all ReLU, tanh, and sigmoid for determining the nonlinear relations. The learning rate of 0.0086 had ensured proper convergence. The batch size of 27 had maintained a balance between computational efficiency and batch updates for the gradients. Training for 29 epochs, which had been chosen from a set of 10 to 100, had permitted adequate learning without overfitting. These parameters had worked together to improve the model's performance.

Summarily, a GA had been used for the optimization of the GRU model's hyperparameters with the aim of minimizing the loss function (MSE) and maximizing its performance. The process of optimization had included the tweaking of the GRU's settings and the assessment of the loss function, wherein the lowering of the former indicated a rise in the latter's performance. The optimum individual revealed by the genetic algorithm, e.g., [3, 44, 62, 88, 'ReLu', 0.0086, 27, 29], had been chosen for the configuration and training of the final GRU model.

4.2.4 SARIMAX-GA Architecture

To enhance the performance of SARIMAX, we configured the GA with the hyperparameters presented in Table 8.

Table 8. GA hyperparameters for SARIMAX.

Hyperparameter	Value
Population size	10
Number of generations	2
Number of elites	2
Mutation rate	0.1
Crossover probability	0.8

The GA optimization process for the SARIMAX model was conducted using a population of 10 individuals who evolved over 2 generations, with 2 elite solutions preserved at each iteration to retain the best-performing candidates. The population size was chosen to ensure sufficient diversity while avoiding unnecessary computational costs, and two generations were sufficient for convergence. A mutation rate of 0.1 introduced controlled variability to prevent premature convergence, and a crossover probability of 0.8 facilitated effective recombination of parameters between solutions to enhance the search for improved configurations. Preliminary experiments with larger populations and additional generations did not yield performance improvements and increased computational time. Therefore, the selected GA settings represent an optimal balance between exploration, exploitation, and computational efficiency. To guide the GA in identifying the optimal SARIMAX configuration, ranges for the model's hyperparameters were defined, as presented in Table 9. This approach, analogous to that applied for the GRU model, allowed the GA to focus on the most promising parameter settings and improve overall model performance.

Table 9. SARIMAX model hyperparameters and best Values.

Hyperparameter	Range/Values	Best Value
p (Auto-regressive order)	[0, 2, 3, 4]	4
d (Differencing)	[1]	1
q (Moving average order)	[0, 1, 2, 3, 4]	3
P (Seasonal auto-regressive order)	[0, 1]	1
D (Seasonal differencing)	[1]	1
Q (Seasonal moving average order)	[0, 1]	1
s (Seasonal period)	168(weekly data)	168

Table 9 outlined the hyperparameter ranges and their best values for a SARIMAX model. The autoregressive order (p) was tested with values of 0, 2, 3, and 4, and the best value selected was 4. The differencing parameter (d) was limited to [1], with the best value also being 1. The moving average order (q) ranged from 0 to 3, and the optimal value was 3. The seasonal autoregressive order (P) had values 0 and 1, with 1 being the best. Similarly, the seasonal differencing (D) and seasonal moving average order (Q) both had values of [1], and the best value was 1. Finally, the seasonal period(s) was set to 7 × 24, reflecting a weekly data frequency. These values were determined to be the most effective for the model. In summary, to optimize our SARIMAX model, we integrated a GA to determine the most appropriate parameters. The goal is to minimize the AIC, ensuring the best model performance.

The best-performing parameters (those with the lowest AIC), including [4, 1, 3, 1, 1, 1, 7 × 24], were incorporated to refine the final SARIMAX model. This guarantees that our model is adjusted to our data, taking into account seasonality and other external variables (hour, day type, and temperature) for each station.

5. Results and Discussions

To evaluate the performance of our approaches, SARIMAX, GRU, GRU-GA, and SARIMAX-GA were performed on identical training and test datasets. They were assessed by means of error measures: MAE, MSE, MAPE, and R^2. We compared the models with their basic and optimized forms to demonstrate the value of optimizing the model structure. Additionally, descriptive statistics were used to support the observed performance trends, and paired t-tests were applied to assess the statistical significance of the differences between baseline and optimized models.

5.1. *Model Evaluation Criteria*

Various statistical metrics are commonly used for the evaluation of the regression model's predictions [40]. In this study, we focus on the following metrics for assessing deep learning models: Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2). The metrics would allow the presentation of the result's preciseness and the selection of the one performing best. In the following formulas, n is the total number of observations, y_i represents the actual values, $\hat{y}_i$ denotes the predicted values, and $\bar{y}$ is the mean of the actual values.

- **Mean Squared Error (MSE):** Calculates an average value for squared prediction vs actual value discrepancies. The MSE value enables users to determine the general accuracy of their predictive model.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{7}$$

- **Mean Absolute Percentage Error (MAPE):** Expresses the error as a percentage relative to the actual values, offering a more intuitive interpretation of the model's performance.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100. \tag{8}$$

- **Mean Absolute Error (MAE):** Calculates the mean of the absolute differences between predictions and actual observations, providing a measure of the average size of the errors.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{9}$$

- **Coefficient of Determination** (R^2): Measures how well the predictions approximate the actual values. It indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. A R^2 value close to 1 indicates a good fit. The formula is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{10}$$

5.2. *Paired t-tests*

The paired t-test is a statistical method used to compare the means of two related groups to determine whether there is a statistically significant difference between them [44]. In our context, it is applied to the point-wise forecasting errors (MAE or MSE) of two models evaluated on the same dataset. Given two sets of errors, e_1 and e_2, for each observation, the test statistic t is calculated as:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}. \tag{11}$$

where $\bar{d}$ is the mean of the differences $d_i = e_{1,i} - e_{2,i}$, s_d is the standard deviation of these differences, and n is the number of paired observations. The null hypothesis H_0 assumes that the mean difference is zero, i.e., there is no significant difference between the models. A p-value below a predefined significance level (e.g., 0.05) indicates that H_0 can be rejected, confirming that the observed difference is statistically significant.

5.3. *Interpretation*

The results in Table 10 highlight the varying performances of the different forecasting models for electricity demand.

- **Without The GA Optimization**
  - **GRU:** The performance measures suggest that the GRU model has relatively reliable recognition results, but there is much room for improvement. The mean absolute error of the attained 236.9441 shows almost a deviation of 237 units between actual and predicted electricity demand, which implies high predictive errors. Likewise, the relatively large MSE of 77009.5069 indicates a wide gap between errors and that there are higher prediction errors occasionally present. In addition, the MAPE of 4.7085% indicates that the model makes average percentage forecasting errors of about 4.71%. Even though the GRU model provided a decent explanation of the underlying structure in data (as indicated by an R^2 score of 0.8449), it did not illustrate all variability in time series. These findings reveal limitations in the baseline GRU model and motivate the need for further optimization to enhance its predictive accuracy and robustness.
  - **SARIMAX:** In contrast, the SARIMAX model had strong results in terms of forecasting ability in comparison to GRU. The forecasting error MAE is 111.8574, which suggests, on average, predictions of forecasting were away from actual by 111.8574. The error predictions of forecasting were off, i.e., MSE 19081.9046, which is so. The forecasting errors, on average, were about 2.2663 MAPE. The electricity demand time series had an R^2 of 0.9615, which meant about 96.15% of the variance had been explained by the model, which is very satisfactory. These results leave room for improvement. The model accurately predicts certain patterns in the demand time series, as well as sudden fluctuations and outliers, but it also introduces significant predictive errors. In summary, while the SARIMAX model for time series prediction is effective, it should be used with caution, and improving the prediction of more extreme values would be advantageous; on average, having more model predictions would yield better results.

- **With The GA Optimization**
  - **GRU-GA**: The evaluation metrics of GRU-GA increased overall performance compared to that achieved by the baseline GRU model. The model's predictions have an MAE of 181.400 and, as such, miss the actual values by roughly 181 units on average, which represents a significant reduction in forecast error compared to GRU_baseline_original. The MSE of 50464.9416 represents the mean square deviation between actual values and predicted values, indicating a moderate spread of errors. The MAPE of 3.6358% also shows that the average prediction error represents about 3.64% of the actual values, which indicates a better accuracy relative to the test data. Additionally, the R^2 value of 0.9029 indicates that the GRU-GA model accounts for approximately 90.29% of the variability in the data, showcasing its robust capacity to capture underlying patterns and enhance the model's explanatory power. In general, these findings have validated the capability of using metaheuristic optimization for enhanced modeling, which would even secure believable prediction.
  - **SARIMAX-GA**: The performance of the SARIMAX-GA model indicates a significant increase in forecasting accuracy. The value of MAE = 102.6518 means that the forecasts differ from the true values on average for approximately 103 units, which represents a substantial decrease in prediction error compared to the original SARIMAX. The MSE of 16597.5536 indicates a moderate level of error spread, which in turn indicates better precision. MAPE = 2.0215%, which means the average relative error is around 2.02%, indicating a high accuracy of the model. In addition, an R^2 of 0.9665 shows that there is around 96.65% of the trend and temporal variation in the data explained by the SARIMAX-GA model; hence, it has performed remarkably well to capture the trend and temporal variations associated with time-series data. These results further confirm the excellent quality and success of the SARIMAX-GA model in providing precise and meaningful forecasts.

Table 10. Performance evaluation of models before and after optimization by GA.

	Model	**MAE**	**MSE**	**MAPE**	**R²**
Without GA	GRU [45]	236.9441	77009.5069	4.7085%	0.8449
	SARIMAX [5]	111.8574	19081.9046	2.2663%	0.9615
With GA	GRU-GA [37]	181.400	48186.7294	3.6358%	0.9029
	SARIMAX-GA	102.6518	16597.5536	2.0215%	0.9665

5.4. Models Comparison

Initial forecasting output efficiencies for both the SARIMAX and GRU models are evaluated without optimizing metaheuristics. The test results show that the SARIMAX model does better than the basic GRU model on all evaluation metrics. To cite a few examples, SARIMAX achieves an MAE value of 111.8574, an MSE value of 19081.9046, a MAPE as low as 2.2663%, and an R^2 value of 0.9615. By contrast, the GRU model exhibited increased prediction error: it reached an MAE of 236.9441, an MSE of 77009.5069, an MAPE value greater than 4.7085%, and a lower R^2 equaling 0.8449. These findings suggest that the time-series data for electricity demands can be explained much better by the SARIMAX model than by an unoptimized GRU. The application of the metaheuristics optimization law proved a consistent improvement in performance for both models. The performance of each was then further analyzed to determine the effects of tuning on errors. After optimization, the SARIMAX-GA model performed better than the original SARIMAX model in every single evaluation metric—and even exceeded expectations when it obtained its coefficient of determination, an R^2 value of 0.9665. Similarly, the standardized GRU-GA model surpassed its baseline with a lower MAE of 181.400, reduced MSE at only 48.7294, and fidelity rates increased from 3.6358% in favor over earlier results to now earn an R^2 value as high as 0.8983. These results highlight the effectiveness of metaheuristic optimization in enhancing the predictive accuracy and robustness of both forecasting models.

To statistically validate the improvements obtained after GA optimization, paired t-tests were conducted on the MAE errors between models. The comparisons included (SARIMAX vs SARIMAX-GA) and (GRU vs GRU-GA). Table 11 presents the corresponding t-statistics and p-values.

Table 11. Paired t-test results comparing models on MAE errors.

Model Comparison	t-statistic	p-value
SARIMAX vs SARIMAX-GA	42.824	< 0.001
GRU vs GRU-GA	14.742	< 0.001

As shown in Table 11, the p-values for all model comparisons are well below the 5% significance level, confirming that the performance improvements achieved through GA optimization are statistically significant and not attributable to random variation.

Finally, when comparing the performance of the SARIMAX-GA and GRU-GA models, we found that the SARIMAX-GA model showed superior accuracy, with reduced errors and better explanation of the data variance compared to the GRU-GA model. Therefore, SARIMAX-GA appears to be more suitable for modeling and forecasting our data. To support this comparison, we include the following Figure 11, which shows the R² values for each model, along with a combined Figure 12 that compares all the models.

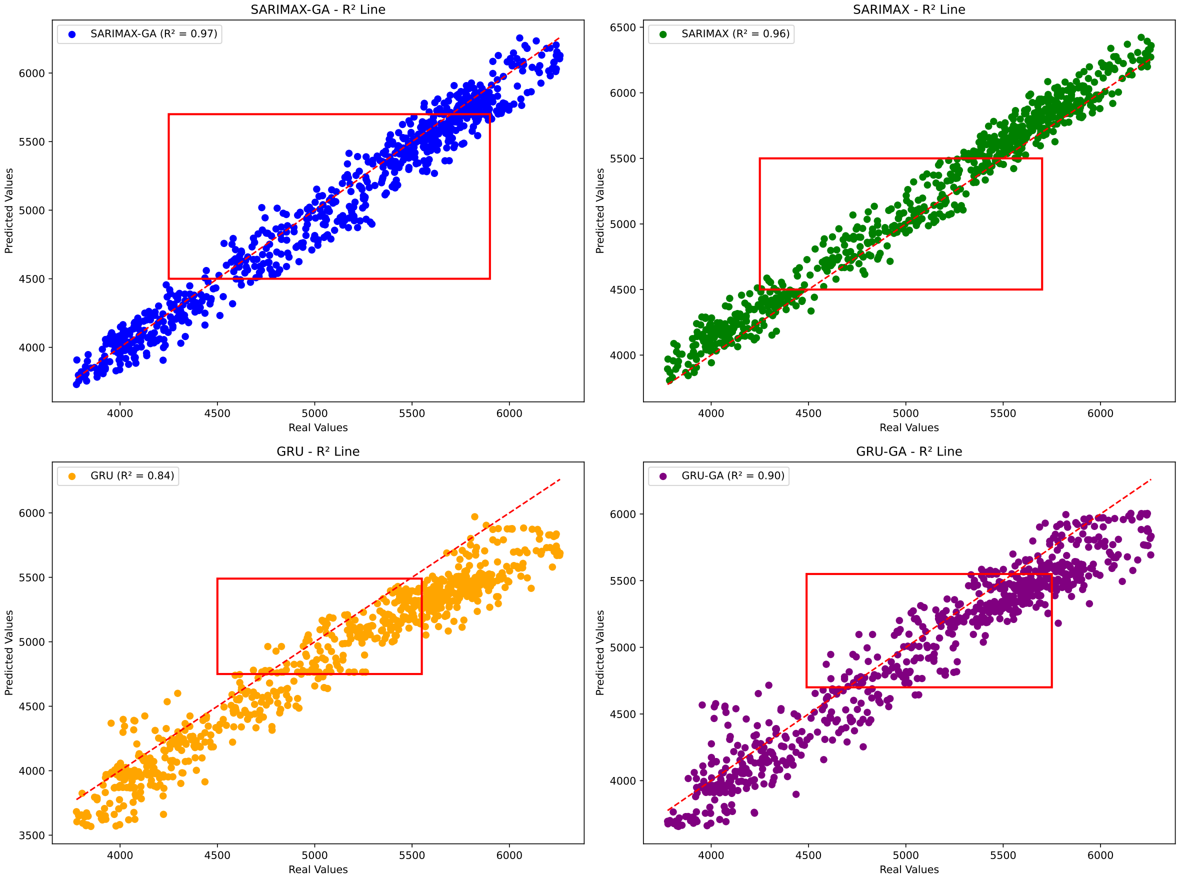

Figure 11. The R² line for each model.

Figure 11 shows in red the areas of maximum performance for each model. These squares encompass the bands of values over which the models show a strong correlation between real and stochastic values, an indicator of solid performance.

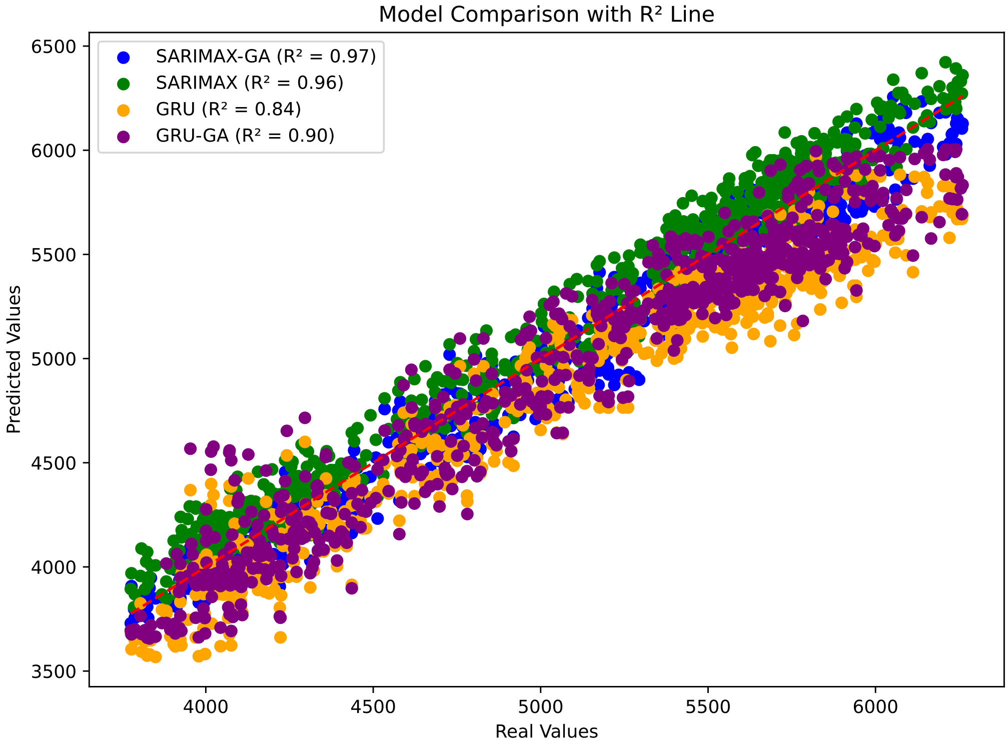

Figure 12. Comparison of all the models using the R² line.

In addition to highlighting the performance differences, this comparison also points to the fundamental significance of hyperparameter optimization using GA. The optimization using GA significantly enhanced the predictive performances of SARIMAX and GRU models by efficiently searching through the search space to identify optimal parameter configurations.

6. Conclusion and Future Work

In this paper, we have presented a work aimed at improving the performance of electricity demand forecasting using the metaheuristic algorithm, namely the GA. The paper compared traditional methods, such as the SARIMAX model, and state-of-the-art DL methods, like GRU. This evaluation is conducted in two runs: without and with optimization via GA, to highlight the impact of tuning hyperparameters. Additionally, real-world data was used, including factors such as temperature, type of day, and hour of the day, which have a significant influence on electricity demand. The resulting findings provided evidence of optimization for enhanced forecasting accuracy, better than that produced by the predictions. This study also presents an inherent limitation because it relied exclusively on hourly data from 2023, as internal confidentiality restrictions at ONEE prevented access to data from earlier years. This limitation hinders a comprehensive analysis of the long-term generalization performance of our models.

Future work will further enhance the accuracy, stability, and practical applicability of the models. We will extend this study by analyzing multi-year data and applying rolling-origin evaluation schemes to assess the temporal stability. The alternative optimization schemes will be investigated to further improve the convergence rate and predictive accuracy. Furthermore, a comprehensive sensitivity analysis regarding the optimization hyperparameters will be developed, the results of which will be reported in the form of tables and contour plots to depict the effects on convergence and forecasting performance. Furthermore, other forecasting models such as those based on machine learning and deep learning will also be investigated and benchmarked for a range of applications. Models will also be extended with additional exogenous features (climatic, economic, social, and infrastructure-related), incorporating state-of-the-art feature selection methods with improved interpretability and robustness. The extension of these models in data-scarce regions could also be considered through cross-region validation and transfer learning. Furthermore, each model will be tested in terms of computational metrics (e.g., required training time per epoch, needed inference time per prediction, and maximum memory demand in terms of GPUs/CPUs). Trade-off curves of accuracy against computational cost will be generated so that a utility-based selection among the EDKs could be performed in terms of their technical features. Collaboration with utilities, a user-friendly interface or API development, and a thorough error analysis, including residual visualization and variable importance evaluation, will further support practical adoption and identify when predictions may be less reliable.

Acknowledgements

The authors' sincere appreciation goes out to all colleagues and academic partners who have helped make this study a reality. The researchers at Mohammed First University of Oujda and those from Casablanca's ENSAM are given special thanks for their helpful input and direction. The authors are also thankful for being given data by Casablanca's National Office of Electricity and Drinking Water (ONEE) for use in research. The help and facilities from the AIMCE lab of Casablanca's ENSAM are finally acknowledged.

Data availability

The data used in this study are available on request from the corresponding author.

Supplementary Materials

To promote transparency and reproducibility, the complete implementation of the proposed methodology, including data preprocessing, model training, and evaluation scripts, is made publicly available in a GitHub repository: https://github.com/lahjiliibtissam/electricity-demand-forecasting

Declarations

The authors declare that they have no conflict of interest.

REFERENCES

1. Aguilar Madrid, E., & Antonio, N. (2021). *Short-term electricity load forecasting with machine learning*. Information, 12(2), 50.
2. Babay, M. A. and Adar, M. and Chebak, A. and Mabrouki, M., (2025) *Forecasting green hydrogen production: An assessment of renewable energy systems using deep learning and statistical methods*, Fuel, 381, 133496.
3. Bacanin, N., Jovanovic, L., Zivkovic, M., Kandasamy, V., Antonijevic, M., Deveci, M., Strumberger, I. (2023). *Multivariate energy forecasting via metaheuristic tuned long short-term memory and gated recurrent unit neural networks*. Information Sciences, 642, 119122.
4. Baek, M., & Seo, Y. (2025). *Hybrid forecasting of university electricity demand using time series and deep learning*. Energy and Buildings, 116400.
5. Bercu, S., Proïa, F. (2013). A SARIMAX coupled modelling applied to individual load curves intraday forecasting. Journal of Applied Statistics, 40(6), 1333–1348. https://doi.org/10.1080/02664763.2013.785496
6. Botman, L., Lago, J., Becker, T., Vanthournout, K., & De Moor, B. (2025). *A global probabilistic approach for short-term forecasting of individual households' electricity consumption*. Applied Energy, 382, 125168.
7. Box, G.E.P., Jenkins, G.M., Reinsel, G.C (2008). *Time series analysis*. Wiley Series in Probability and Statistics.
8. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M. (1978). *Time Series Analysis: Forecasting and Control*. The Statistician, 27, 265-265.

9. Chatfield, C., Xing, H. (2019). *The Analysis of Time Series: An Introduction with R*. (7th ed.), Chapman and Hall/CRC. https://doi.org/10.1201/9781351259446
10. Chen, S., Fang, X., Khayatnezhad, M. (2024). *Forecasting for electricity demand utilizing enhanced inception-V4 using improved Osprey optimization*. Scientific Reports, 14(1), 30832. https://doi.org/10.1038/s41598-024-81487-8
11. Chen, Y., Liu, C., Ge, J., Wu, J., Zhao, X., Gao, Z. (2024). *Deep learning models for forecasting electricity demand in green low-carbon supply chains*. International Journal of Low-Carbon Technologies, 19, 2375-2382. https://doi.org/10.1093/ijlct/ctae186
12. Cho, K., Merrienboer, B.V., Bahdanau, D., Bengio, Y. (2014). *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. arXiv 2014.
13. Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. In Proceedings of the Neural Information Processing Systems Workshop on Deep Learning. arXiv 2014.
14. Cowpertwait, P.S., Metcalfe, A.V. (2009). *Introductory Time Series With R*. Springer.
15. Deb, C., Zhang, F., Yang, J., Lee, S. E., Shah, K. W. (2017). *A review on time series forecasting techniques for building energy consumption*. Renewable and Sustainable Energy Reviews, 74, 902-924.
16. Dutta, A., Kumar, S., Basu, M. (2020). *A Gated Recurrent Unit Approach to Bitcoin Price Prediction*. Journal of Risk and Financial Management, 13(2), 23. https://doi.org/10.3390/JRFM13020023
17. El-Azab, H. A. I., Swief, R. A., El-Amary, N. H., & Temraz, H. K. (2025). *Seasonal forecasting of the hourly electricity demand applying machine and deep learning algorithms impact analysis of different factors*. Scientific Reports, 15(1), 9252.
18. Fahd, K., Badr, H., Eddine, L.S., Mouad, A. (2024). *Forecasting stock market returns using deep learning and time series techniques: a comparative and empirical study using technical indicators*. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-024-20321-9
19. Gao, T., Niu, D., Ji, Z., & Sun, L. (2022). *Mid-term electricity demand forecasting using improved variational mode decomposition and extreme learning machine optimized by sparrow search algorithm*. Energy, 261, 125328.
20. Gers, D. A., Schmidhuber, J., Cummins, F. (2000). *Learning to Forget: Continual Prediction with LSTM*. Neural Computation, 12(10), 2451-2471.
21. Gumaei, A.H., Hassan, M.M., Alelaiwi, A., Alsalman, H. (2019). *A Hybrid Deep Learning Model for Human Activity Recognition Using Multimodal Body Sensing Data*. IEEE Access, 7, 99152-99160. https://doi.org/10.1109/ACCESS.2019.2927134
22. Hochreiter, S., Bengio, Y. (2001). *Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies*.
23. Hochreiter, S., Schmidhuber, J. (1997). *Long Short-Term Memory*. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735
24. Hu, H., & Zheng, B. (2024). *Short-term electricity load forecasting based on CEEMDAN-FE-BiGRU-Attention model*. International Journal of Low-Carbon Technologies, 19, 988-995.
25. Inteha, A., Nahid-Al-Masood (2021). *A GRU-GA Hybrid Model Based Technique for Short Term Electrical Load Forecasting*. 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 515-519.
26. Jiang, P., Li, R., Liu, N., & Gao, Y. (2020). *A novel composite electricity demand forecasting framework by data processing and optimized support vector machine*. Applied Energy, 260, 114243.
27. Katoch, S., Chauhan, S. S., Kumar, V. (2021). *A review on genetic algorithm: past, present, and future*. Multimedia Tools and Applications, 80(5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
28. Konak, A., Coit, D.W., Smith, A.E. (2006). *Multi-objective optimization using genetic algorithms: A tutorial*. Reliab. Eng. Syst. Saf., 91, 992-1007.
29. Leme, J. V., Casaca, W., Colnago, M., & Dias, M. A. (2020). *Towards Assessing the Electricity Demand in Brazil: Data-Driven Analysis and Ensemble Learning Models*. Energies, 13(6), 1407.
30. Li, X., Wang, Z., Yang, C., & Bozkurt, A. (2024). *An advanced framework for net electricity consumption prediction: Incorporating novel machine learning models and optimization algorithms*. Energy, 296, 131259.
31. Lipton, Z.C. (2015). *A Critical Review of Recurrent Neural Networks for Sequence Learning*. ArXiv, abs/1506.00019.
32. Maarof, M.Z., Ismail, Z., Fadzli, M. (2014). *Optimization of SARIMA model using genetic algorithm method in forecasting Singapore tourist arrivals to Malaysia*. Applied Mathematical Sciences, 8, 8481-8491.
33. Manowska, A. (2020). *Using the LSTM Network to Forecast the Demand for Electricity in Poland*. Applied Sciences, 10(23), 8455. https://doi.org/10.3390/app10238455
34. Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag Berlin and Heidelberg
35. Molla, S., Farrok, O., & Alam, M. J. (2024). Electrical energy and the environment: Prospects and upcoming challenges of the World's top leading countries. Renewable and Sustainable Energy Reviews, 191, 114177.
36. Nikou, R., Goli, A., Zackery, A. (2024). *Improving electricity demand forecasting through hybrid neural networks and meta-heuristics: A case study in Iran*. Journal of Dynamics and Games, 0-0.
37. Noh, J.S., Park, H., Kim, J., Hwang, S. (2020). *Gated Recurrent Unit with Genetic Algorithm for Product Demand Forecasting in Supply Chain Management*. Mathematics, 8(4), 565. https://doi.org/10.3390/math8040565
38. Nontapa, C., Kesamoon, C., Kaewhawong, N., Intrapaiboon, P. (2021). *A New Hybrid Forecasting Using Decomposition Method with SARIMAX Model and Artificial Neural Network*. International Journal of Mathematics and Computer Science, 16(4), 1341-1354.
39. Pallonetto, F., Finn, D., Jin, C., Mangina, E. (2022). *Forecast electricity demand in commercial building with machine learning models to enable demand response programs*. Energy and AI, 7, 100121. https://doi.org/10.1016/J
40. Plevris, V., Solorzano, G., Bakas, N., Ben Seghier, M. E. A. (2022). *Investigation of performance metrics in regression analysis and machine learning-based prediction models*. 8th European Congress on Computational Methods in Applied Sciences and Engineering. https://doi.org/10.23967/eccomas.2022.155
41. Peteleaza, D., Matei, A., Sorostinean, R., Gellert, A., Fiore, U., Zamfirescu, B., Palmieri, F. (2024). *Electricity consumption forecasting for sustainable smart cities using machine learning methods*. Internet Things, 27, 101322.
42. Ragupathi, C., Dhanasekaran, S., Vijayalakshmi, N., Salau, A. O. (2024). *Prediction of electricity consumption using an innovative deep energy predictor model for enhanced accuracy and efficiency*. Energy Reports, 12, 5320-5337.

43. Raza, M. Q., Khosravi, A. (2015). *A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings*. Renewable and Sustainable Energy Reviews, 50, 1352-1372. https://doi.org/10.1016/j
44. Montgomery, D. C., & Runger, G. C. (2010). *Applied statistics and probability for engineers*. John wiley & sons.
45. Son, N., Yang, S., Na, J. (2020). *Deep Neural Network and Long Short-Term Memory for Electric Power Load Forecasting*. Applied Sciences, 10(18), 6489. https://doi.org/10.3390/app10186489
46. Strielkowski, W., Civín, L., Tarkhanova, E., Tvaronavičienė, M., Petrenko, Y. (2021). *Renewable Energy in the Sustainable Development of Electrical Power Sector: A Review*. Energies, 14(24), 8240. https://doi.org/10.3390/en14248240
47. Sultana, N., Hossain, S. M. Z., Almuhaini, S. H., Düştegör, D. (2022). *Bayesian Optimization Algorithm-Based Statistical and Machine Learning Approaches for Forecasting Short-Term Electricity Demand*. Energies, 15(9), 3425. https://doi.org/10.3390/en15093425
48. Sharmila, V., Gaikar Vilas, B., Nayan, R., Pavithra, G. (2025). *Hybrid Genetic Optimization and Particle Swarm Optimization for Enhanced Electricity Demand Forecasting Using Artificial Neural Networks*. Sustainable Smart Homes and Buildings with Internet of Things, 253-269. https://doi.org/10.1002/9781394231522
49. Ugbehe, P. O., Diemuodeke, O. E., & Aikhuele, D. O. (2025). *Electricity demand forecasting methodologies and applications: a review*. Sustainable Energy Research, 12(1), 19.
50. Weron, R. (2014). *Electricity price forecasting: A review of the state-of-the-art with a look into the future*. International journal of forecasting, 30(4), 1030-1081.