future internet

Article

Fault Prediction and Reconfiguration Optimization in Smart Grids: AI-Driven Approach

David Carrascal †, Paula Bartolomé †, Elisa Rojas *, Diego Lopez-Pajares, Nicolas Manso and Javier Diaz-Fuentes

Universidad de Alcalá, Departamento de Automática, Escuela Politécnica Superior, 28805 Alcalá de Henares, Spain; david.carrascal@uah.es (D.C.); p.bartolome@edu.uah.es (P.B.); diego.lopezp@uah.es (D.L.-P.); miguel.manso@edu.uah.es (N.M.); j.diazf@edu.uah.es (J.D.-F.)

* Correspondence: elisa.rojas@uah.es

† These authors contributed equally to this work.

Abstract: Smart grids (SGs) are essential for the efficient and distributed management of electrical distribution networks. A key task in SG management is fault detection and subsequently, network reconfiguration to minimize power losses and balance loads. This process should minimize power losses while optimizing distribution by balancing loads across the grid. However, the current literature yields a lack of methods for efficient fault prediction and fast reconfiguration. To achieve this goal, this paper builds on DEN2DE, an adaptable routing and reconfiguration solution potentially applicable to SGs, and investigates its potential extension with AI-based fault prediction using real-world datasets and randomly generated topologies based on the IEEE 123 Node Test Feeder. The study applies models based on Machine Learning (ML) and Deep Learning (DL) techniques, specifically evaluating Random Forest (RF) and Support Vector Machine (SVM) as ML methods, and Artificial Neural Network (ANN) as a DL method, evaluating each for accuracy, precision, and recall. Results indicate that the RF model with Recursive Feature Elimination (RFECV) achieves 94.28% precision and 81.05% recall, surpassing SVM (precision 89.32%, recall 6.95%) and ANN (precision 72.17%, recall 13.49%) in fault detection accuracy and reliability.

Keywords: smart grids; AI; fault prediction; machine learning; deep learning

check for updates

Citation: Carrascal, D.; Bartolomé, P.; Rojas, E.; Lopez-Pajares, D.; Manso, N; Diaz-Fuentes, J. Fault Prediction and Reconfiguration Optimization in Smart Grids: AI-Driven Approach. *Future Internet* **2024**, *16*, 428. https://doi.org/10.3390/fi16110428

Academic Editor: Stefano Rinaldi

Received: 15 October 2024
Revised: 12 November 2024
Accepted: 17 November 2024
Published: 20 November 2024

1. Introduction

Smart grids (SGs) represent the future solution for electricity transmission and distribution networks [1–3], as their main objective is to achieve full network monitoring, allowing a better energy balance between producers and consumers. SG systems must be able to react quickly and predictably, adapting to changes in energy demand and supply through the control of energy consumption and storage devices. The applications managing the SG must rely on a secure, highly scalable, and always available communication network, as they need to handle large volumes of real-time data to consistently respond to changes in the network state [4,5].

One of the most important aspects of SGs is the reconfiguration of the power distribution within the network. This is crucial because it allows the system to reroute power in case of a failure in one of the network links, or simply to optimize the distribution based on a given criterion [6]. Many existing contributions [7–9] place more emphasis on the latter case, in which reconfiguration through optimization improves power losses due to propagation or line overload, or by addressing other electrical parameters. Generally, most reconfiguration algorithms applied to SGs focus on network resilience in case of faults aim to provide backup routes or paths to perform the distribution. However, the proposed strategies are always reactive to the fault, whether in a centralized or distributed manner.

Therefore, in this work, we aim to generate a preventive, and thus proactive, reconfiguration strategy for network faults by leveraging real-world SG datasets and Machine

Learning (ML) and Deep Learning (DL) techniques. The proposed approach is based on a previously existing routing and reconfiguration solution for SGs, known as DEN2DE [10]. We build upon this foundation to develop a novel fault prediction mechanism that classifies alternative routes to preemptively identify those likely to cause network errors. Our work contributes to the field by providing a fault prediction solution tailored specifically for SG reconfiguration, which is trained and validated using real datasets as detailed in Section 3. Furthermore, we ensure its generalizability across diverse network topologies by testing it with random configurations generated through the Boston University Representative Internet Topology Generator (BRITE) tool, using parameters from the IEEE 123 Node Test Feeder model [11]. Through these validations, we examine the optimization of multiple ML and DL models for DEN2DE by employing feature selection, assessing the most suitable predictive model for each scenario.

This article is organized as follows. Section 2 provides an overview of related reconfiguration algorithms applied to SGs. Section 3 presents the main real-world datasets analyzed for this study. Section 4 briefly explains the foundation routing and reconfiguration solution and how our proposal extends it. Section 5 presents the validation environment and the evaluation of the different Artificial Intelligence (AI) models along with their respective optimizations. Section 6 outlines the results obtained, as well as the best configuration of the evaluated models, together with a brief discussion regarding the generalization of our study, and finally, Section 7 provides the conclusions and future work.

2. Related Work

In order to analyze the contributions of our work, we first briefly examine the works in relation to SG reconfiguration and fault tolerance. To start with, the reconfiguration of SGs represents a complex discrete optimization problem that can be addressed through various methods, including optimization solvers or metaheuristic algorithms, as well as ML or DL models [12–16]. The choice of approach typically depends on the complexity of the network and the specific system requirements. In the literature on reconfiguration for SGs, two primary directions have been identified: the first focuses on defining an objective function aimed at optimizing the internal performance of the grid (e.g., minimizing power propagation losses, capacity overloading, and costs), while the second concentrates on reconfiguring the network to enhance resilience and fault tolerance.

Delving into specific examples and algorithms, Rodriguez et al. [9] propose a distributed reconfiguration approach for smart grids based on the Open Shortest Path First (OSPF) routing protocol, aiming to enhance fault detection and minimize power losses. This method is implemented in a multi-agent system deployed in secondary substations and validated on the IEEE 123 Node Test Feeder, where agents execute the OSPF algorithm to reconfigure the grid in response to network changes. For Islanded Microgrids (IMGS), Hemmatpour et al. [17] introduce a reconfiguration methodology focused on optimizing voltage stability and power flow. Using an Adaptive Multi-Objective Harmony Search Algorithm and validated on IEEE 33-bus and 69-bus systems, this approach improves IMGS performance by increasing loadability and reducing power losses, which is crucial for systems lacking a main grid. Similarly, Sun et al. [18] present a self-healing strategy for microgrid islanding, optimizing generation re-dispatch, network reconfiguration, and load shedding. The problem is modeled as a Mixed-Integer Quadratic Programming (MIQP) to minimize costs and ensure smooth islanding and is validated only on the IEEE 9-bus system. Moreover, for offshore High-Voltage Direct Current (HVDC) transmission systems connected to wind farms, Sanz et al. [19] address the challenge of minimizing power losses by proposing a reconfiguration algorithm based on Particle Swarm Optimization (PSO), which optimizes the topology of meshed HVDC grids. However, these works mainly focus on optimizing the internal operating parameters of SGs, typically power losses and network configuration costs, but not potential network failures.

Regarding the works focused on providing resilience and fault tolerance in SGs, a growing trend has been observed to place more emphasis on ML and DL. Hosseinzadeh et al. [20] present an enhanced K-Nearest Neighbors (KNN) algorithm combined with Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for efficient fault detection and classification, demonstrating robustness and precision suitable for real-time applications in SGs. Li et al. [21] propose a Convolutional Neural Network (CNN)-based method for fault localization using voltage data, achieving superior accuracy in low-observability scenarios. The approach is validated on IEEE 39-bus and 68-bus systems, showing resilience and effectiveness in complex conditions. Another similar work using CNNs is by Alhanaf et al. [22], who employ Artificial Neural Networks (ANNs) and one-dimensional CNNs (1D-CNNs) for fault management, extracting fault characteristics from raw signals and demonstrating high accuracy in real-time applications on the IEEE 6-bus system. Furthermore, Kaplan et al. [23] presented a Long Short-Term Memory (LSTM) network for fault diagnosis, targeting the complexities of renewable energy sources. Their method was validated using Simulink SG models in Matlab, and it was found to improve fault prediction accuracy, outperforming traditional techniques and enhancing the reliability of SGs. Efatinasab et al. [24] address vulnerabilities to adversarial attacks in fault zone prediction using a Bayesian Neural Network (BNN) framework. The model improves prediction robustness and introduces an uncertainty-based detection scheme to counter adversarial threats, achieving high accuracy. Other works, such as Marashi et al. [25], investigate fault propagation in cyber-physical systems, proposing a neural network model that achieves high predictive accuracy in the IEEE 14-bus and 57-bus topologies, enabling early fault detection and preventive actions in SGs. Finally, Ding et al. [26] propose a resilient microgrid formation strategy, combining distributed generator control and topology reconfiguration to optimize load restoration after outages using CPLEX, validated on the IEEE 33-bus and 615-bus test systems. However, these works focus on optimizing a model for a single topology, and the use of real datasets is missing.

Therefore, after analyzing the state of the literature, the main contributions of our work consist of (1) a fault prediction solution for SG reconfiguration, (2) based on real datasets, and (3) generalized for any type, shape, or size of topology (i.e., evaluated with the random topology generator BRITE, according to real parameters). For this purpose, ML and DL models are analyzed, followed by their optimization.

3. Data Source Search, Analysis, Purge, and Selection

To comprehensively evaluate the different AI techniques for fault prediction in reconfiguration scenarios for SGs, a high-quality dataset is essential. Both the quality and quantity of data are crucial for defining a consistent model. In the energy context, our experience proves there is a lower degree of availability of datasets. The protection of user privacy in the measurement and analysis of their energy behavior reduces the number of publicly available datasets, as they primarily depend on energy companies [27]. Nevertheless, the pursuit of optimizing energy distribution and reducing user consumption, along with other environmental motivations, has led to the creation of public datasets in recent years. Table 1 represents several analyzed residential datasets, along with details about their implementation, such as location, number of residences, deployment period, sampling frequency, and measured electrical parameters. As part of our research analysis, to evaluate and select the most appropriate dataset from the ones available in Table 1, the following requirements were considered:

- *Quantity*: it is crucial to review the quantity of data available in each dataset, ensuring that they include measurements from a large number of residential buildings and cover extended periods to analyze energy consumption patterns effectively.
- *Quality*: The quality of the data depends on the resolution of the measurements. Datasets such as *REDD* and *BLUED* offer high sampling frequencies, allowing for better energy disaggregation and a more representative analysis of energy behavior.
- *Location*: The location of the data is important because it affects voltage differences between countries. For example, datasets from the U.S., such as *BLUED*, operate at less than 120 V, whereas European datasets like *ECO* handle up to 230 V.

- *Parameterization*: The samples include voltage (V), current (I), and power (P, Q, S), each associated with a timestamp and an identifier corresponding to the respective residence. Some datasets, such as *HUE* and *SustDataED*, also include environmental data, which is important for understanding the impact of climatic conditions.
- *Photovoltaic generation*: in the context of SGs, it is essential to choose datasets that include information on renewable energy generation, such as *Smart** and *SustDataED*.

Table 1. Comparison of public power consumption datasets.

Name	Location	Residences	Period (Days)	Resolution	Parameters
AMPds2 [28]	Vancouver (Canada)	1	730	60 s	I, V, P, S, F, pf
BLUED [29]	Pittsburgh (USA)	1	8	83.33 µs	I, V, switch events
ECO [30]	Switzerland	6	244	1 s	P
GREEND [31]	Italy and Austria	9	310	1 s	P
HUE [32]	British Columbia, Canada	28	60	1 s	P
iAWE [33]	New Delhi (India)	1	73	1 s	V, I, P, S
REDD [34]	Boston (USA)	6	119	66.66 µs	I, V, P
*Smart** [35]	Massachusetts (USA)	3	90	60 s	P, S, V, I
SustDataED [36]	Madeira (Portugal)	50	1144	60 s	I, V, P, Q, S
UK-DALE [37]	UK	5	499	62.5 µs	P, switch state

Among the analyzed datasets, *SustDataED* was particularly selected due to the large number of multiple users, both in terms of consumption and production, over a long period. Likewise, it is important that the samples have a good resolution and provide real-time information on the climatic conditions of the location where the residences are situated. In the following sections, we examine this dataset in detail and outline the main design criteria applied to generate the final dataset, curated from *SustDataED* and complemented with the PVWatts tool, which was eventually leveraged for our evaluation.

3.1. SustDataED—Dataset Analysis

The SustDataED dataset [36] was created as part of the SINAIS research project, aimed at providing ecological feedback to users to promote sustainable energy behavior and greater use of renewable energy sources. This dataset encompasses five years of energy consumption and production data from 50 households in the city of Funchal (Madeira, Portugal), divided into three different deployments. This implies there are no measurements from 50 households simultaneously throughout the five years of the project but separately in three groups.

Figure 1 depicts the three aforementioned deployments. Since we were working with consumption and generation measurements that were partially correlated with weather and temporal conditions, we selected the time period with the highest number of households deployed simultaneously. In our case, we chose the first deployment, in which measurements from up to 23 households were available simultaneously.

The data available in SustDataED are quite varied. Table 2 lists the different types of measurements included in the dataset. However, not all of them were relevant for this study. For instance, a high level of granularity, such as power event measurements or user event measurements, was not required for our analysis. As shown in Table 2, the measurements related to energy production were uncertain. At first glance, SustDataED appeared suitable because it provided real photovoltaic production measurements. However, these data were reported as aggregate values and pertained to a photovoltaic system that supplied all the households, with no information about the system's dimensions, as they were not specified. It was also unclear whether the size of the photovoltaic system increased over time, making household-level analysis more complex.

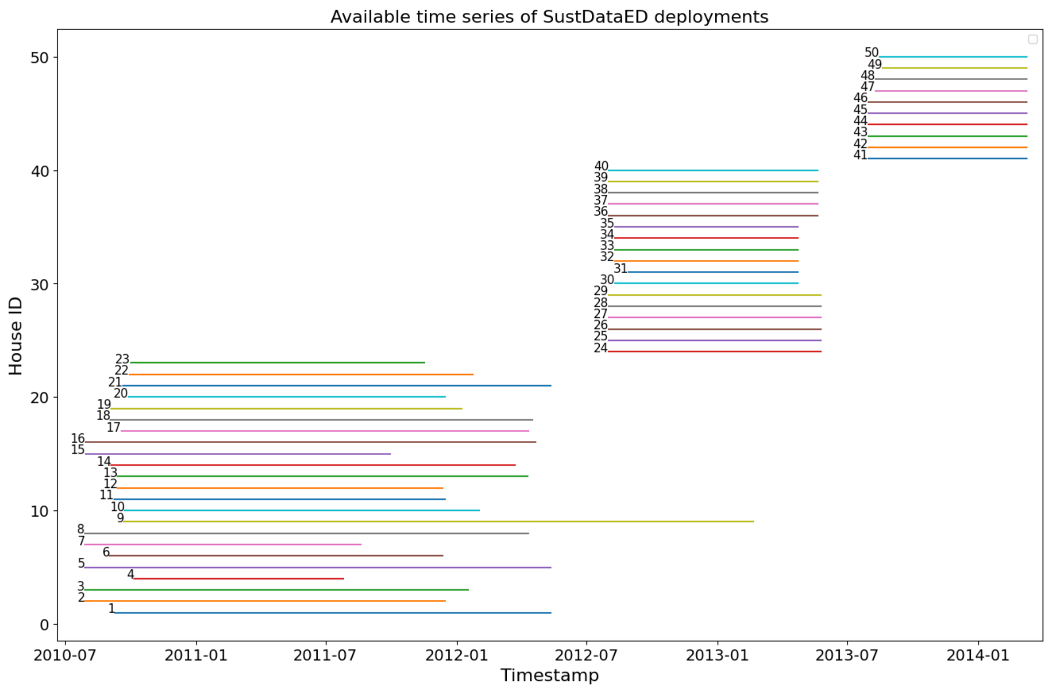


Figure 1. Time series of measurements of the SustDataED dataset.

Table 2. Types of measurements in SustDataED dataset. Utility: useful (✓), useless (X), questionable (**?**).

Type of Measurement	Utility
Energy consumption measurements	✓
Energy production measurements	**?**
Demographic measurements	✓
Environmental and climatic conditions measurements	✓
User event measurements	X
Power event measurements	X

3.2. SustDataED—Dataset Shortcomings

Since the literature on SustDataED did not provide clarity on household-level photovoltaic energy production measurements, an alternative was sought to form a new production dataset. To perform this production data simulation, two tools were chosen in order to compare and ensure that the results obtained from them were consistent:

- Global Solar Atlas [38]: this platform is funded by the Energy Sector Management Assistance Program (ESMAP) with the aim of mapping renewable energy resources globally, providing access to both long-term averaged data and real-time data for any location on Earth.
- PVWatts [39]: this platform is provided by the National Renewable Energy Laboratory (NREL), which is the national laboratory of the U.S. Department of Energy.

For each tool, the location was set to Funchal (Portugal), since the analysis should correspond to the data collected in the SustDataED dataset. Once the location was established, the photovoltaic system for each household was configured, and the relevant data layers extracted like the Direct Normal Irradiation (DNI) ($\frac{\mathrm{kWh}}{\mathrm{m}^2}$) and Photovoltaic Power Potential (PVOUT) ($\frac{\mathrm{kWh}}{\mathrm{kWp}}$), which are key parameters for estimating the power output of photovoltaic systems. Additionally, climatic factors play a crucial role in verifying the consistency of the generated power. According to the Köppen and Trewartha's climate classifications, Funchal features a Mediterranean climate with oceanic influences or a temperate climate with dry summers. Located in a subtropical zone, the city experiences minimal daily temperature variations, leading to relatively stable temperatures throughout the year.

After configuring both systems, the dates corresponding to the SustDataED time series were aligned to collect photovoltaic production data. In both tools, as shown in Figure 2, a comparative analysis of the same parameters, DNI and PVOUT, was conducted for the same time periods. The results, as illustrated in Figure 2, were found to be nearly identical. Following this verification of consistency between the two tools for simulating photovoltaic production values, PVWatts was selected due to its more streamlined export capabilities, facilitating the integration with the SustDataED data for the construction of the final dataset.

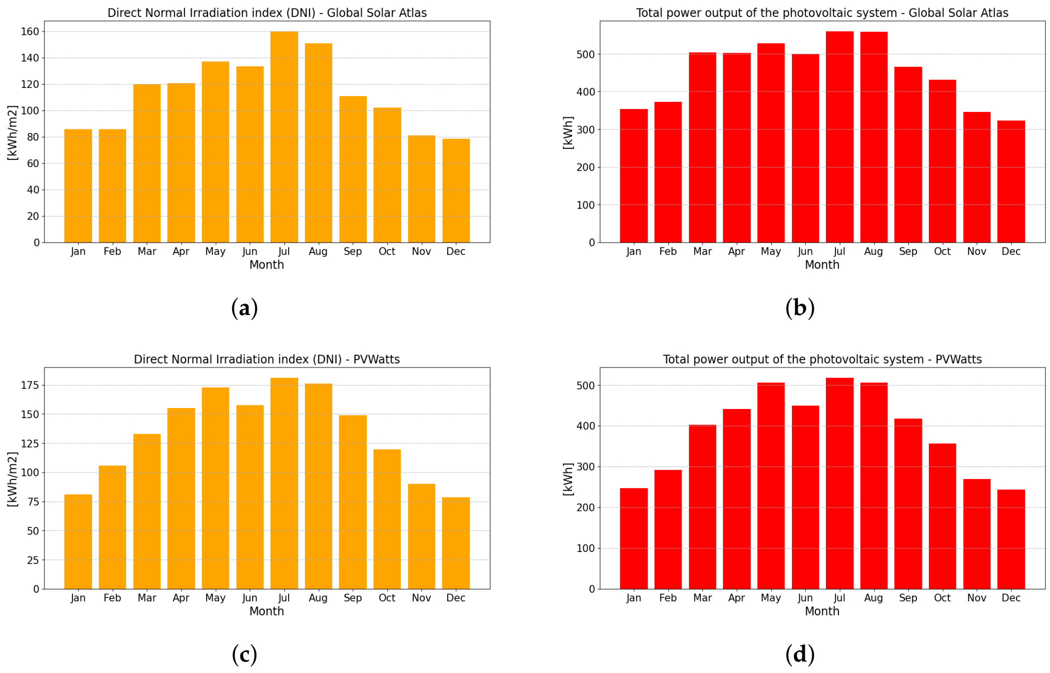
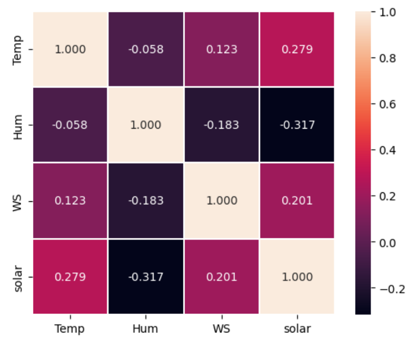
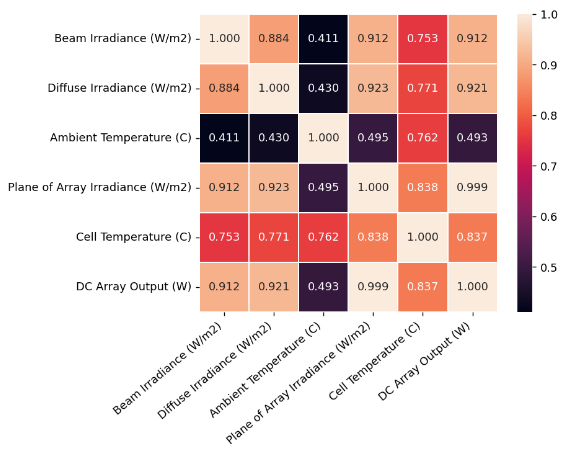


Figure 2. Comparison of Global Solar Atlas and PVWatts, total monthly values of photovoltaic energy production (PVOUT) versus normal direct irradiation (DNI). (**a**) Global Solar Atlas—DNI. (**b**) Global Solar Atlas—PVOUT. (**c**) PVWatts—DNI. (**d**) PVWatts—PVOUT.

In addition, a comprehensive correlation analysis was performed on the photovoltaic power generation parameters for both SustDataED and PVWatts datasets to conclusively evaluate the reliability of the energy production data. When examining the correlation between the climatic variables and the power generation for both datasets, as shown in Figure 3, distinct patterns emerged. For SustDataED (Figure 3a), it was evident that the temperature exhibited a weak correlation with the generated power, yielding a coefficient of only 0.279. This lack of strong correlation was consistent across other variables, making it difficult to analyze energy production in relation to climatic conditions. In contrast, for the PVWatts dataset (Figure 3b), most parameters showed a high degree of correlation, approaching unity. For instance, the correlation between array plane radiation and power generation was nearly ideal, with a coefficient of 0.999, as clearly shown in the graph.



(**a**)



(**b**)

Figure 3. Comparison using correlation matrices of photovoltaic production parameters. (**a**) SustDataED—Photovoltaic parameters. (**b**) PVWatts—Photovoltaic parameters.

As illustrated in Figure 4, the scatter plot for the PVWatts correlation revealed that the most significant parameters for photovoltaic power generation were the plane of irradiance hitting the solar panel and the temperature of the solar cells. Given the limitations of the production data from SustDataED, particularly in terms of accuracy and granularity, the decision was to use the simulated production data from the PVWatts tool. This choice was supported by the consistency of the PVWatts data when compared to the results from the Global Solar Atlas.

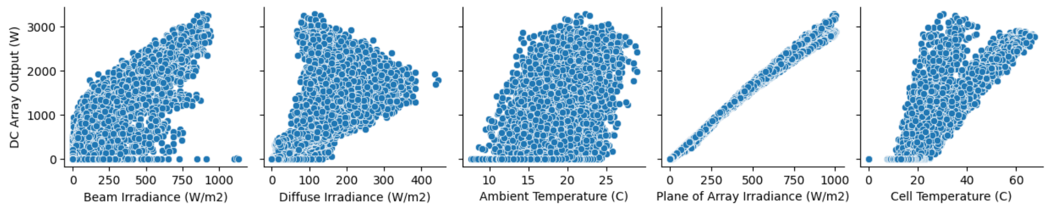


Figure 4. Correlation patterns between production parameters in PVWatts.

3.3. SustDataED and PVWatts Tool—Final Dataset (KeyMonData)

After identifying and addressing the limitations of SustDataED by incorporating the PVWatts tool for photovoltaic power generation data, the entire dataset was processed to create the final dataset. The data processing workflow is outlined in Figure 5. As shown, there were two main branches: the first branch originates from the SustDataED dataset and included all household consumption data, while the second branch corresponded to the photovoltaic generation data. Out of the 50 households, only 23 were selected from the first deployment, as previously mentioned. Subsequently, 23 photovoltaic generation profiles were generated, ultimately forming 23 files of real net loads, representing the sum of consumption and generation data.

This new dataset required further processing to limit both its size and the temporal series to align with the specified deployment. Additionally, to make the dataset more manageable, the data were synthesized. Specifically, SustDataED provides measurements at a minute-level resolution; however, working with minute-by-minute samples is unnecessary, as the variations in consumption are negligible. Therefore, the resolution was reduced to hourly samples. Thus, if the first deployment was limited to a duration of one year, there were 8760 (24 h $\times$ 365 days) time intervals per household, for a total of 23 households.

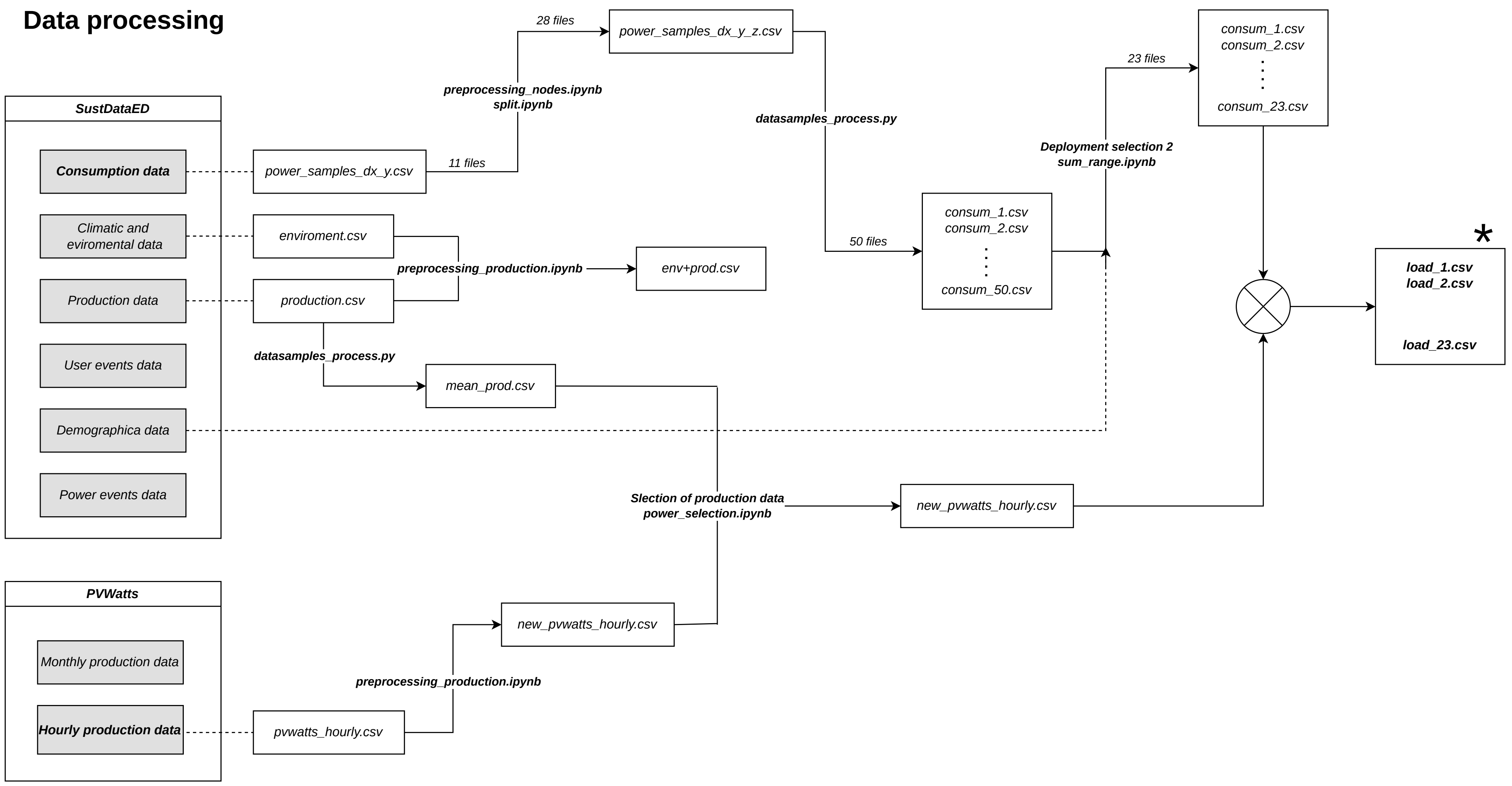


Figure 5. Data processing scheme for the generation of the final dataset (*).

Regarding the file naming convention, each file was labeled as *load_x.csv*, where *x* refers to the identifier of the corresponding household for the calculated loads. Consequently, this resulted into a total of 23 load files (*load_x.csv*), each containing 8760 time intervals. The final fields included in the new dataset are outlined in Table 3. After being processed, cleaned, and synthesized, the new dataset was named *KeyMonData* and has been made publicly available for use by anyone [40].

Table 3. Resulting dataset (*KeyMonData*) from the processing stage.

Field	Description	Units
timestamp	Time of measurement	datetime
datetime	Date of the average value	datetime
H	Hour of the average value	-
iid	Housing identifier	-
Diffuse Irradiance	Diffuse radiation index (DIF)	W/m^2
Plane of Array Irradiance	Plane of array radiation index (POA)	W/m^2
Ambient Temperature	Ambient temperature	C
Cell Temperature	Solar cell temperature	C
DC Array Output	DC array output power	W
AC System Output	AC system output power	W
Pavg	Consumed power	W
Dif	Net calculated load	W

4. Reconfiguration Algorithm and Proposed Extension

The reconfiguration algorithm for the SG, upon which the ML and DL models were built, was DEN2DE [10]. DEN2DE is an algorithm designed for discovering paths in dense and heterogeneous networks, enabling the automation of the routing process from leaf nodes to the root node of the topology, while efficiently managing resource sharing. In the specific context of an SG, the root node is defined as the one having a direct connection to the electrical distribution network, whereas the leaf nodes represent the remaining nodes in the SG.

To further explain the reconfiguration process followed by DEN2DE, Figure 6 depicts a scenario of load redistribution within an SG, similar to the IEEE 5-bus topology example. In green, we can observe the nodes that have a surplus of load, which can be offered to other nodes in the network. In orange, the nodes demanding power from neighboring nodes, or ultimately, from the root node, are highlighted. However, calculating all possible load redistribution solutions can be computationally complex within a reasonable time frame. For instance, Figure 6a illustrates a suboptimal redistribution of resources, which occurs as neighbors decide based on local decisions, which leads to most requests directed to the node with a surplus of +100 (insufficient to meet all the demands), as represented by the arrows. On the other hand, Figure 6b depicts an optimal distribution scenario, which requires a more sophisticated approach.

To address this issue, DEN2DE proposes a three-phase solution. The first phase, as shown in Figure 7, involves exploring all possible routes from the root node to each leaf node in the topology. This exploration, as depicted in the figure, is carried out using labels. The exploration begins at the root node, which generates the first label (*1*) and transmits it to all its immediately connected neighbors, in this case to node 2. Node 2 receives the label, appends its node identifier, stores it in its learning table, and then proceeds to transmit it similarly to all its contiguous neighbors. In this way, node 3 receives the label *1.2*, indicating that it is two hops away from the root node, and it repeats the process by assigning the label *1.2.3* to its neighbors, nodes 4 and 5. These nodes, in turn, perform the same process, exchanging labels *1.2.3.4* and *1.2.3.5*, which allow them to learn an alternative route to the root node. This is because they now have a direct route through node 3, as well as a longer route through the node that just assigned them the new label. Ultimately, nodes 4 and 5 will attempt to transmit the last learned labels (*1.2.3.4* and *1.2.3.5*) back to node 3. However, DEN2DE logic incorporates a loop prevention mechanism: if the label being offered is part of any previously learned label in the learning table, it will be discarded. This is why labels such as *1.2.3.5.4* and *1.2.3.4.5* are discarded by node 3, as illustrated in Figure 7.

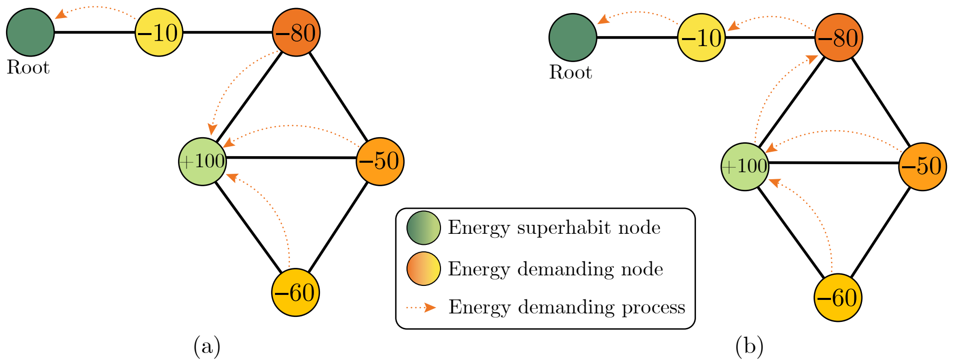


Figure 6. Example of load redistribution in an existing SG. (**a**) Suboptimal energy demanding scenario. (**b**) Optimal energy demanding scenario.

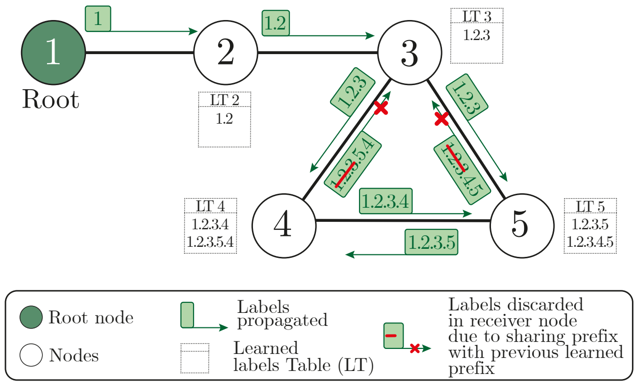


Figure 7. Operation of the DEN2DE algorithm.

Once the entire labeling process across the network is finished, the first phase of the algorithm is considered complete. The second phase of DEN2DE involves selecting the best label per node for load redistribution, based on the criteria defined in the algorithm itself (such as the number of hops, distance, losses, available load, etc.). In this example, nodes with only one label do not need to apply this phase. However, nodes 4 and 5 will need to select which label they will use, as it will later determine how they will redistribute the load within the SG. The third phase of the algorithm focuses on the redistribution of loads between the nodes. This phase assumes that all nodes have selected a label, i.e., a route to reach the root node. As a result, a logical topology will be formed over the physical topology of the SG, potentially leaving certain links unused. This enables optimal power distribution within the SG, as illustrated in Figure 6b.

In summary, DEN2DE first explores the whole network topology (assigning labels during the process), then it defines the order to follow for resource distribution (this order is represented by the selected label) and, finally, it reconfigures the network based on that order. In this work, several ML and DL models were evaluated to support the DEN2DE algorithm during its second phase. More specifically, these models provided additional information to enhance the selection of labels (and, hence, the order of resource distribution), as they were chosen considering the prediction of potential faults that may arise in the SG, which eventually improved the reconfiguration process as a whole.

It is important to note that although we evaluated these models exclusively with DEN2DE, both our dataset and our analysis of AI techniques could also provide a foundation for analyzing other reconfiguration algorithms in SG networks. Ultimately, the techniques examined in our study offer supplementary insights into fault prediction, and it is at the discretion of the specific reconfiguration mechanism to incorporate this information to enhance its procedure.

5. Evaluation

Considering the objective of this article, which aimed to detect and predict errors, or faults more specifically, an error criterion had to be defined first to label the results from the algorithm. Considering that the simulations were configured with a real scenario involving losses and capacity limitations of the links, the decision was made to establish the failure condition based on the presence of excess capacity in an energy exchange between two nodes in the network. These errors were not introduced arbitrarily; rather, we simulated the power balance based on the established configuration of the smart grid. In scenarios where losses occurred due to excess capacity, we denoted that for a given entry in the dataset, under those specific conditions, a fault would be marked accordingly. Subsequently, our objective was to classify whether a fault would occur based on predetermined conditions and configurations. In this step, it was decided to apply a link configuration based on the *IEEE 123 Node Test Feeder* [11] to ensure that the process of assigning capacities to nodes in the algorithm was random. For simplicity, three types of links are defined in Table 4.

Table 4. Link configuration based on the *IEEE 123 Node Test Feeder*.

R **(ohm/km)**	***I max*** **(A)**	***Section*** **(mm^2)**
0.272	185	70
0.78	100	25
1.91	53	10

Now that the definition of error is clear, in this section, we evaluate various ML and DL techniques in terms of error prediction. Specifically, we define Random Forest (RF) and Support Vector Machine (SVM) as ML methods, and ANN as a DL method, based on their structural characteristics. Our evaluation follows the pipeline illustrated in Figure 8, which first generates a comprehensive set of random topologies, then sets up the simulation environment using DEN2DE, and finally performs an extensive analysis of the different ML and DL models, each of which is examined in separate sections.

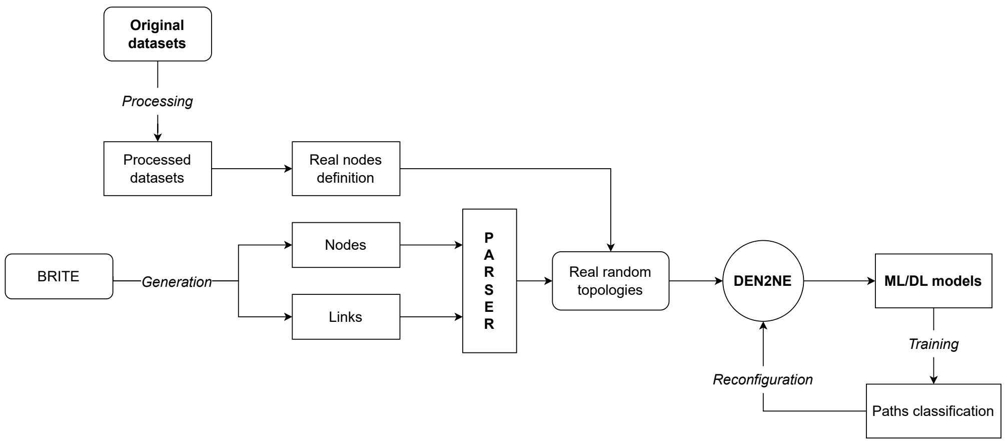


Figure 8. Defined pipeline to detect and predict errors during the energy distribution process.

5.1. Random Topologies' Generation

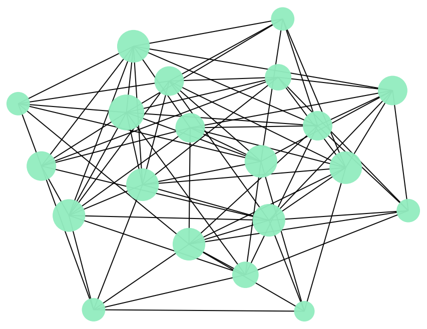
The random topology generation process plays an important role in testing and validating the ML and DL models. Creating diverse network scenarios establishes a robust foundation for the precise detection and prediction of faults.

5.1.1. Brite Tool

BRITE [41] is a widely used tool designed to generate random network topologies that simulate real-world network structures. In the context of this project, BRITE was used to create random topologies representing different configurations of SGs with the aim of importing them into the DEN2DE algorithm and simulating various network conditions. BRITE allows the definition of multiple parameters and topological models, such as node distribution, degree connectivity, and link probabilities, bringing flexibility and making BRITE particularly useful for generating diverse scenarios. In addition, the tool supports several models to generate network topologies, and two of them are established [42]:

- Waxman model [43]: This model is a probabilistic approach where nodes are placed randomly, and the likelihood of a connection between two nodes decreases with their distance. It is suitable for simulating the physical distance constraints in power grids.
- Barabasi–Albert model [44]: This model creates topologies based on preferential attachment, where new nodes are more likely to connect to nodes with higher degrees. This reflects the real-world nature of power grids, where certain nodes are hubs with many connections.

As shown in Figure 9, both topological models are depicted with an identical number of nodes. Node size reflects the number of connections within the topology, with larger nodes representing those with more connections. In the Waxman model (see Figure 9a), the topology exhibits a relatively random structure, resulting in an approximately uniform average number of connections per node. Conversely, in the Barabási–Albert topological model (see Figure 9b), initial nodes have a higher likelihood of connecting with newly added nodes, thus creating what is known as a scale-free topology.


(**a**)

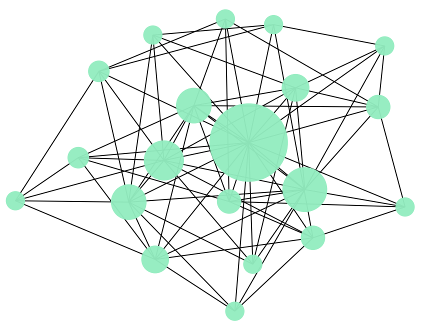

(**b**)

Figure 9. Comparison of Router Waxman and Barabasi–Albert topology examples. (**a**) Router Waxman model topology example. (**b**) Router Barabasi–Albert topology example.

5.1.2. Generation Process Automation

A key aspect of this project was the automation of topology generation, which is essential for conducting large-scale simulations and enabling a comprehensive evaluation of ML and DL models for error detection. By using customized scripts, BRITE topology generation was automated, producing a total of 180 topologies that could then be used for different simulation executions in DEN2DE. The total number of topologies generated was determined by the product of several configured parameters:

- Topology models: two models, Router Waxman and Router Barabasi–Albert.
- Node counts: topologies were generated with 100, 150, and 200 nodes.

- Connectivity levels: three different degrees of connectivity were specified, with each topology having a degree of 2 m (due to bidirectional links).
- Random seeds: 10 different seed files were applied to ensure variety in the generated topologies.

5.2. Topology-Based Simulations in DEN2DE

Once the topologies were generated, they were imported into the DEN2DE framework, and the algorithm executed simulations to model the energy distribution across the nodes. These simulations incorporated the real load profiles that were extracted and processed in Section 3.3 allowing for the simulation of power distribution processes in realistic SG environments.

Considering that the objective was to discover energy transactions between nodes that exceeded their link's capacity, the scenario of lossy-constrained link capacity was defined. Also, as specified for BRITE, five different seed files were applied to DEN2DE to obtain different simulations from the same input topology. Based on the configured parameters and the 8760 real-data rows available (representing one year of hourly data), the algorithm could potentially run 47,304,000 unique simulations. As it was inefficient and impractical to execute all of them, it was essential to decide which tests to conduct in DEN2DE, and 12 specific time points were chosen, each representing a specific hour of a day for each month. To simplify the process, since the sample time frame started on 28 November 2010, the 28th day of each month was selected. The time chosen for the extraction of load profiles was 11:00 a.m., as this was when average energy production tended to be highest. This selection strategy increased the likelihood of encountering energy exchanges that exceeded the link capacity, leading to a higher number of errors in the load distribution process. This was important for effectively training the ML and DL models.

5.3. Machine Learning Techniques

To detect and predict errors in the energy distribution process, a binary classification of the dataset instances was performed using two supervised ML techniques: RF and SVM. On one hand, an RF [45] combines multiple decision trees and obtains a final prediction through majority voting of these trees, using criteria such as entropy or the Gini index to determine the splits at each node. On the other hand, an SVM [46] aims to find the optimal hyperplane to maximize the separation between two classes, adjusting the regularization parameter C to balance the margin and classification errors, and configuring the parameter γ along with the kernel type to handle non-linearity in the data. The development of these techniques followed a sequence of necessary steps to design optimal models and configure an effective training, based on the use of various classes and methods provided by the open-source software library *scikit-learn* https://scikit-learn.org/stable/.

5.3.1. Feature Scoring

In this first step, a default model was designed for each ML technique to find the features of the dataset that were the most important in the classification process.

For the RF technique, a model was trained using 10 estimators and the entropy criterion. The importance of the dataset features was estimated using the *feature_importances_* attribute provided by the classifier. This calculation was based on the mean and standard deviation of the impurity reduction that occurred within each tree.

In Figure 10, the results show the relative importance values assigned to each feature, with a total sum equal to one. There was a significant impact on the distance between nodes and in the parameters resulting from DEN2DE (*total_balance* and *abs_flux*), which corresponded to the gateway node load after energy balancing and the total flow of resources exchanged in the distribution process, respectively. However, this method introduced a bias towards features with high cardinality, or in other words, those with a large number of unique values. This bias occurred because such features created deeper split nodes in the trees, as there were more options for dividing the dataset. Therefore, these types of features were more likely to receive higher importance scores [47].

Taking this into account, the permutation importance method was used for a comparative analysis. In this second method, the feature values were permuted randomly one at a time, and the classifier's performance was evaluated at each iteration, so a feature that significantly decreased the model's accuracy when its values were permuted scored higher in importance. In this case, as shown in Figure 11, the parameters resulting from DEN2DE (*total_balance* and *abs_flux*) still showed high importance scores. However, now, the lengths of the node labels and the link capacity also exhibited high importance scores.

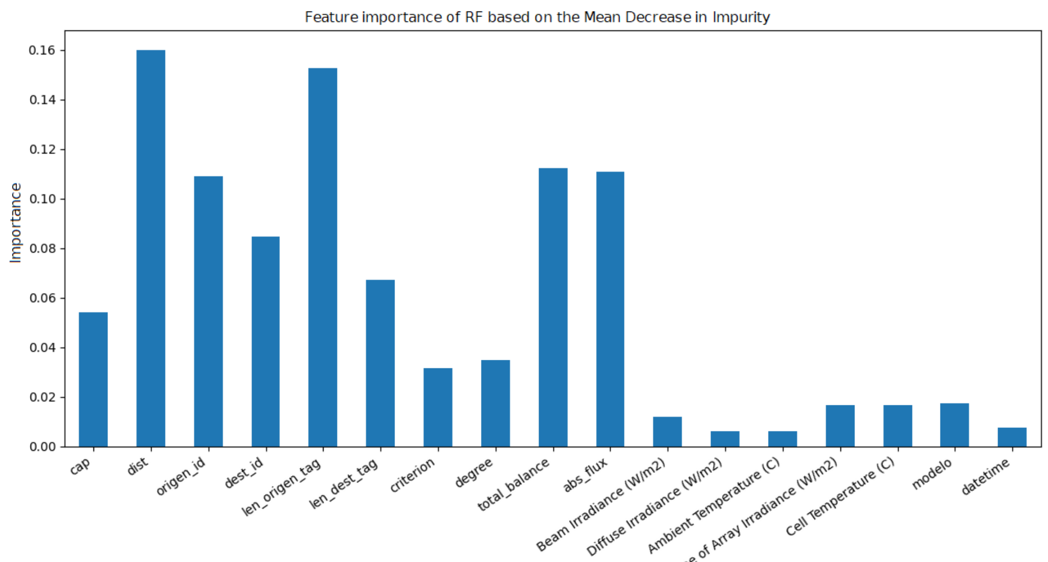


Figure 10. Feature importance scoring for the RF using the *feature_importances_* attribute.

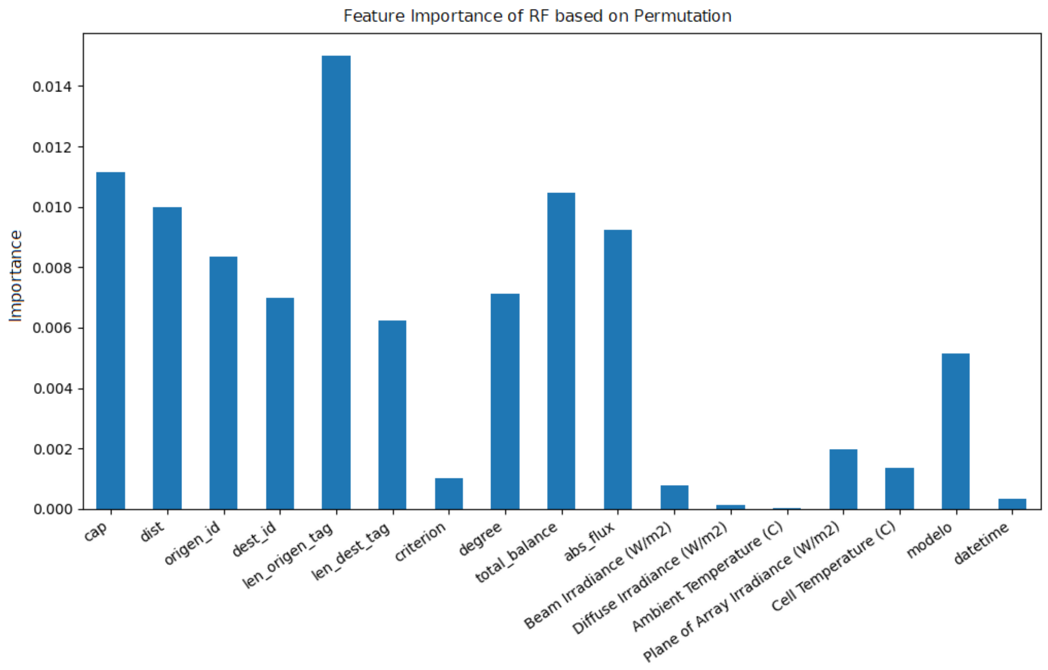


Figure 11. Feature importance scoring for the RF using the *permutation_importance* method.

For the SVM technique, a default model was trained with an Radial Basis Function (RBF) kernel, suitable for non-linearly separable datasets. Consequently, the importance of the dataset's features was evaluated through the *support_vectors_* and *dual_coef_* attributes provided by the class. As depicted in Figure 12, the results indicated that the lengths of the source and destination labels had a significant influence on the dataset. As for the RF, the permutation method was applied, obtaining similar results. In Figure 13, the lengths of the labels, the link capacity, and the total energy flow (*abs_flux*) were highlighted in the classification.

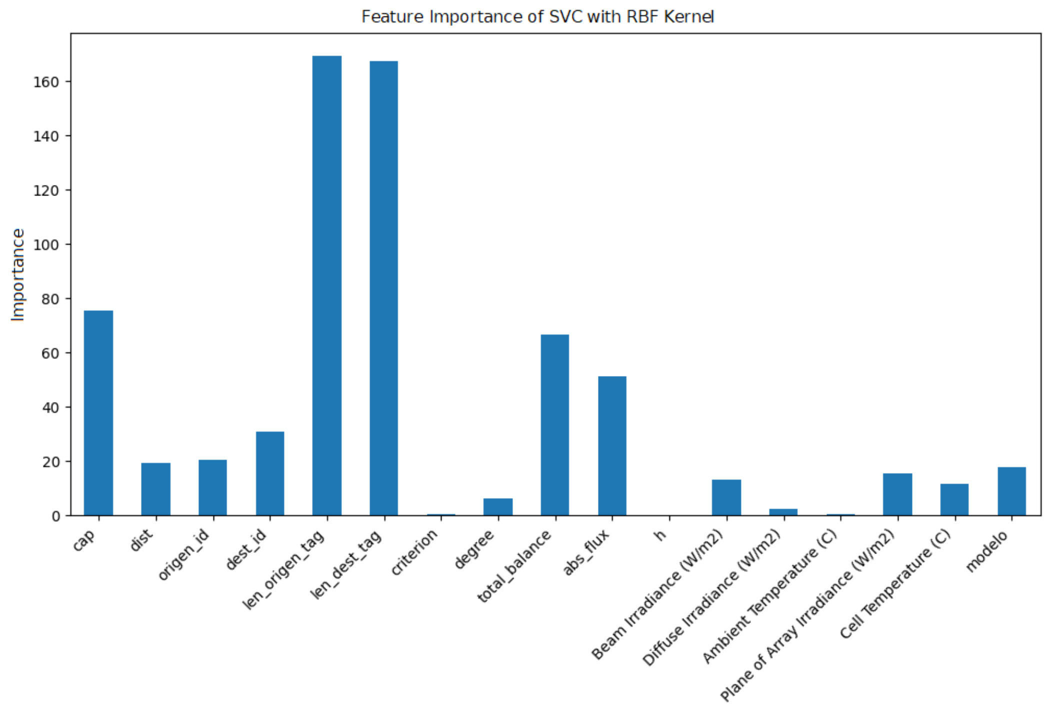


Figure 12. Feature importance scoring for the SVM using the *support_vectors_* and *dual_coef_* attributes.

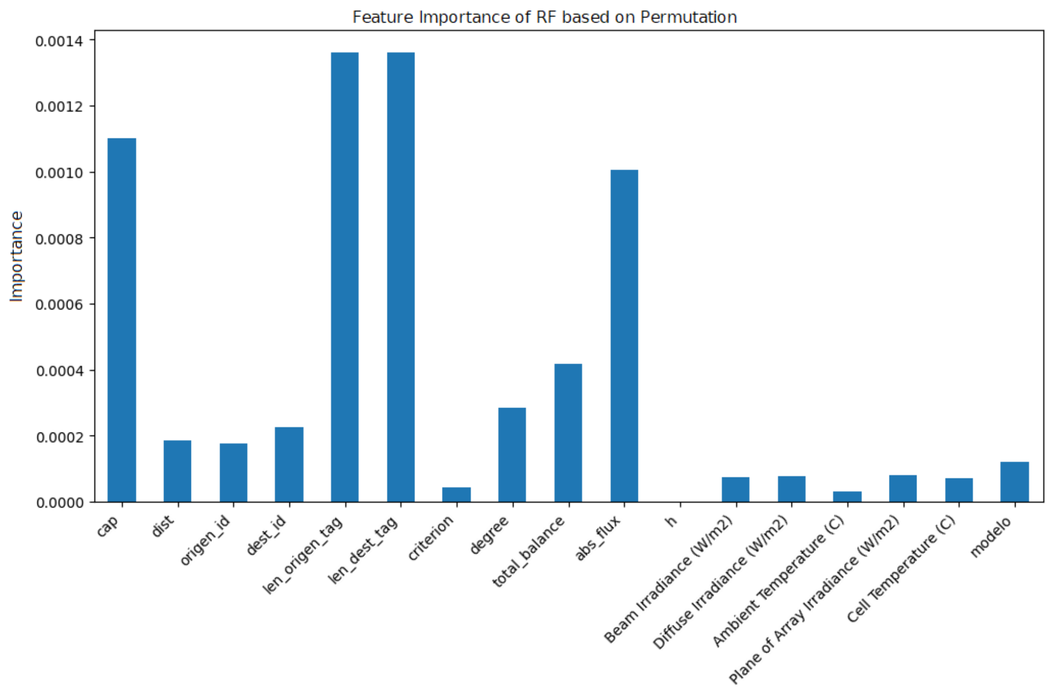


Figure 13. Feature importance scoring for the SVM using the *permutation_importance* method.

5.3.2. Hyperparameters' Optimization

To find the best combination of hyperparameters for each model, the Grid Search method [48] was used. This method performs an exhaustive search through a grid of hyperparameters to identify the combination that offers the highest accuracy. Additionally, since it is based on cross-validation, the dataset was divided into five folds.

In the case of the RF technique, the focus was on two main hyperparameters: the number of estimators or trees and the impurity measurement criterion. The execution of the method determined, based on the *best_score_* and *best_params_* attributes, the best combination of hyperparameters, achieving an accuracy of 99.18% with a classifier of 100 estimators using the entropy criterion (Table 5). For a more detailed analysis, the *cv_results_* attribute was used, which provided comprehensive information about the search. It can be observed that, although the variations in accuracy between different parameter combinations were minimal, the fitting times showed significant differences. As shown in Table 6, the duration of the training increased proportionally with the number of estimators configured in the model. Therefore, since accuracy did not improve significantly when using more than 25 estimators, the classifier with 25 estimators based on the entropy criterion was determined to be the optimal model.

Table 5. Accuracy (%) (*mean_test_score*) of *cv_results_* attribute from the *Grid Search* in the RF.

Number of Estimators	*Criterion*	
	Entropy	*Gini*
10	99.07	99.02
25	99.16	99.14
50	99.16	99.15
75	99.17	99.17
100	99.18	99.16

Table 6. Time (s) (*mean_fit_time*) of *cv_results_* attribute from the *Grid Search* in the RF.

Number of Estimators	*Criterion*	
	Entropy	*Gini*
10	147	146
25	373	382
50	747	761
75	1072	1002
100	1439	987

To model an optimal SVM, two hyperparameters were mainly studied: the type of kernel and the regularization parameter C (See Tables 7 and 8). The parameter γ was omitted due to the prior scaling of the data, which eliminated its impact. In this case, different from the RF, the search for the best combination of hyperparameters required more execution time, taking approximately 350 h. The results of this process showed that the optimal model was an SVM with an RBF kernel and a regularization parameter C equal to one, achieving an accuracy of 97.78% and requiring a relatively low training time compared to other hyperparameter combinations. The analysis of the *cv_results_* attribute revealed that the *sigmoid* kernel offered the worst performance and that the polynomial (*poly*) kernel showed an increase in training time as C increased, suggesting that a high C value was not suitable for this kernel.

Table 7. Accuracy (%) (*mean_test_score*) of *cv_results_* attribute from the *Grid Search* in the SVM.

C	***Kernel***		
	poly	***rbf***	***sigmoid***
0.25	97.65	97.66	95.96
0.5	97.65	97.71	95.94
0.75	97.65	97.75	95.82
1	97.65	97.78	95.81

Table 8. Time (s) (*mean_fit_time*) of *cv_results_* attribute from the *Grid Search* in the SVM.

C	***Kernel***		
	poly	***rbf***	***sigmoid***
0.25	21,030	13,876	14,659
0.5	28,807	10,277	18,695
0.75	42,183	9328	12,235
1	47,460	9446	12,869

5.3.3. Dimensionality Reduction

After analyzing the best combinations of hyperparameters for each ML technique, the application of some dimensionality reduction methodologies on the dataset was studied. This aimed to eliminate irrelevant or redundant information from the dataset to achieve more efficient model training. Three different techniques were tested to later compare the performance of the RF and SVM models in Section 6:

- Recursive Feature Elimination with Cross-Validation (RFECV) [49]: This method iteratively discards the least influential features until the model's performance no longer significantly improves. With 5-fold cross-validation, it can be seen in Figure 14 that the maximum accuracy for the optimal RF model was achieved with eight features. Additionally, it was confirmed that the identified features (*cap*, *dist*, *origen_id*, *dest_id*, *len_origen_tag*, *len_dest_tag*, *total_balance*, *abs_flux*) matched those with the highest scores in Section 5.3.1. However, in the case of SVM, the classifier class did not have the necessary attributes for implementing RFECV (*coef_*, *feature_importances_*), so this technique was only applied to RF.
- Univariate Feature Selection [50] uses the *SelectKBest()* method, which requires prior scaling and the specification of a fixed number of features to work with. In this case, values of k equal to five and eight features were selected.
- PCA [51] reduces the dimensionality of the dataset using the Singular Value Decomposition (SVD) technique. It works by finding the principal directions of variation (k vectors) in the dataset to construct a projection matrix, which establishes a new k-dimensional feature space. To define the optimal number of components, the elbow method was used. Figure 15 shows the variance leveling off at three components, so for comparative purposes, the performance of RF and SVM was studied for two and four components.

5.4. Deep Learning Techniques

The development of DL techniques was divided into the optimal design and training of two ANN models, using two different libraries: *Scikit-Learn* and the *Keras* module of *TensorFlow*. ANNs, and particularly Multilayer Perceptrons (MLPs), are built from various layers of nodes and connections that replicate the neural structure of the human brain [52].

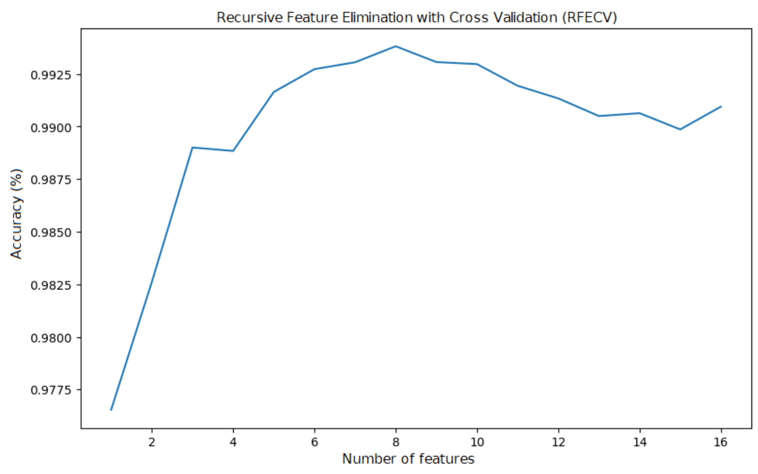


Figure 14. Analysis of RF model accuracy based on the number of features used with RFECV method.

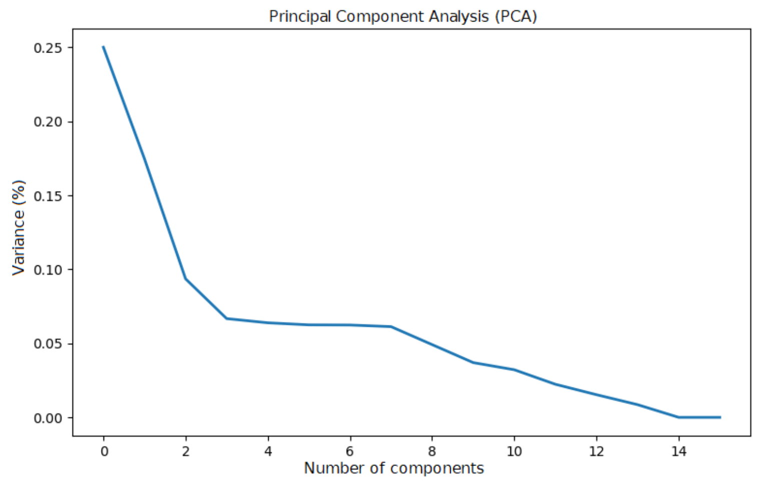


Figure 15. Analysis of variance based on the number of components used in PCA.

Regarding their development, it is important to note that the feature scoring and dimensionality reduction steps applied to the ML models in Section 5.3 were omitted. This is because ANNs internally analyze and score the importance of features within the network, so these steps are not necessary to include [53].

5.4.1. Hyperparameters' Optimization

In this step, hyperparameter tuning was performed to achieve optimal performance of the ANN. To do this, a Grid Search method was applied to the MLP model provided by the *Scikit-Learn* library. The study was focused on configuring hidden layers and the number of neurons per layer (*hidden_layer_sizes*), as well as other hyperparameters such as the activation function (*activation*) and the optimization algorithm (*solver*). To prevent overfitting during Grid Search execution, a maximum number of 100 iterations was set along with early stopping, which was activated if no significant improvements occurred in 10 consecutive iterations.

The entire search process took around 2.34 h on a server with 32 processors. In Table 9, it can be seen that the Stochastic Gradient Descent (SGD) optimization algorithm generally outperformed the *Adam* algorithm. Regarding training times, Table 10 represents the impact of both the layer structure and the number of neurons configured in the network. Similarly, the dimensions also affected the achieved accuracies, with some cases of overfitting leading to lower classification accuracies.

Table 9. Accuracy results (%) (*mean_test_score*) extracted from the *cv_results_* attribute of the MLP.

Solver	***Adam***		***Sgd***	
Hidden Layers	***Activation Function***			
	Relu	***Tanh***	***Relu***	***Tanh***
(5,)	90.17	97.61	97.66	97.54
(8,)	92.03	89.73	97.65	97.65
(10,)	87.00	89.90	97.48	89.76
(50,)	86.89	89.82	89.53	97.65
(100,)	90.95	89.71	84.56	89.72
(5, 5)	89.80	89.72	97.66	97.65
(8, 8)	88.49	89.72	89.89	97.65
(10, 10)	81.68	89.70	90.30	97.65
(50, 50)	81.91	82.04	97.66	89.73

Table 10. Time results (s) (*mean_fit_time*) extracted from the *cv_results_* attribute of the MLP.

Solver	***Adam***		***Sgd***	
Hidden Layers	***Activation Function***			
	Relu	***Tanh***	***Relu***	***Tanh***
(5,)	88	37	63	35
(8,)	96	43	35	36
(10,)	94	40	56	43
(50,)	157	168	143	70
(100,)	236	223	170	131
(5, 5)	136	50	52	44
(8, 8)	159	54	69	46
(10, 10)	149	55	54	48
(50, 50)	333	319	166	155

Therefore, the analysis of the results determined that the optimal MLP configuration was given by a structure of two hidden layers with five neurons each (5, 5), the Rectifier Lineal Unit (RELU) activation function, and the SGD optimization algorithm. Figures 16 and 17 represent the evolution of the accuracy values and the loss function, respectively. The optimal model converged at iteration 41, achieving an accuracy of 97.66% and a loss function value of 0.0712.

5.4.2. Model Execution and Results' Evaluation

After identifying the optimal hyperparameters for the MLP model from the *Scikit-Learn* library, the configuration was replicated in the ANN model defined by the *Keras* module of *TensorFlow*. The goal was to verify that the results obtained with Grid Search were consistent and that two different ANN models with the same hyperparameter values achieved similar results.

In Figures 18 and 19, it can be observed that unlike the previous model, the *Keras* ANN completed all 100 iterations in this case. However, it was confirmed that the loss function (0.068) and the accuracy values (97.87%) obtained in the last iteration were very similar to those of the previous model.

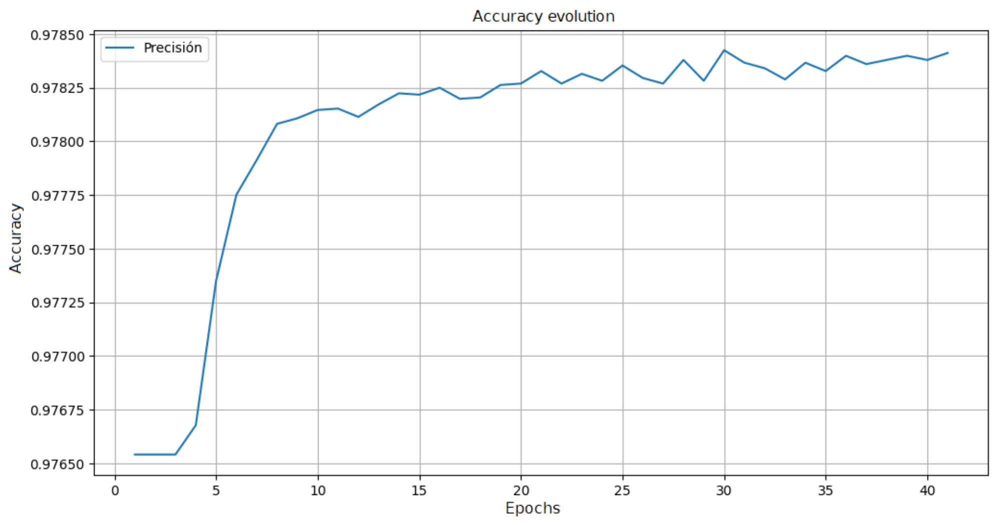


Figure 16. Evolution of the accuracy value for the ANN from *Scikit-Learn*.

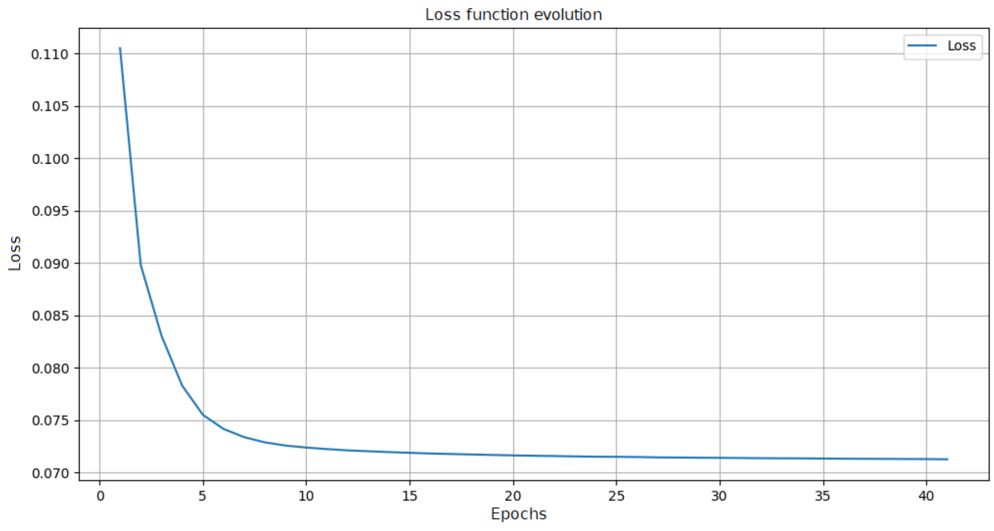


Figure 17. Evolution of the loss function value for the ANN from *Scikit-Learn*.

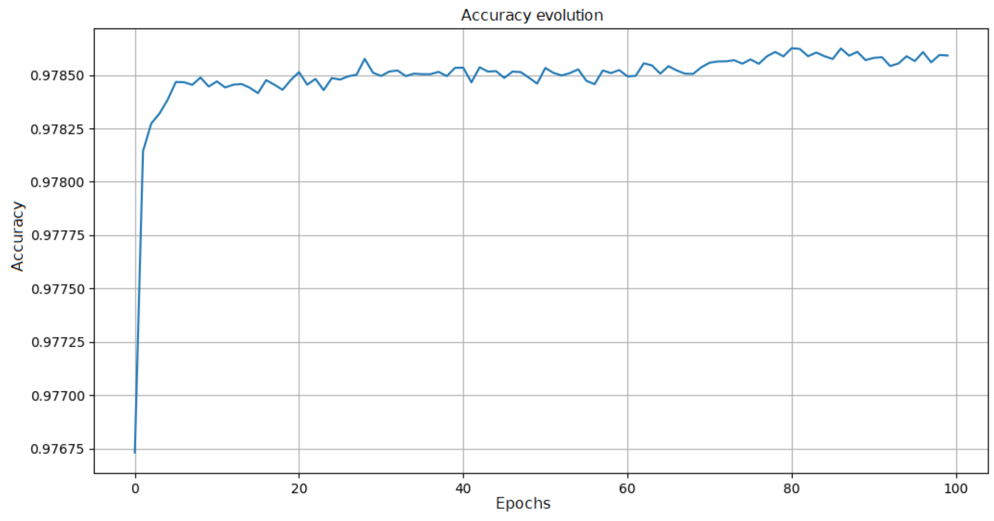


Figure 18. Evolution of accuracy value for the ANN from *Keras*.

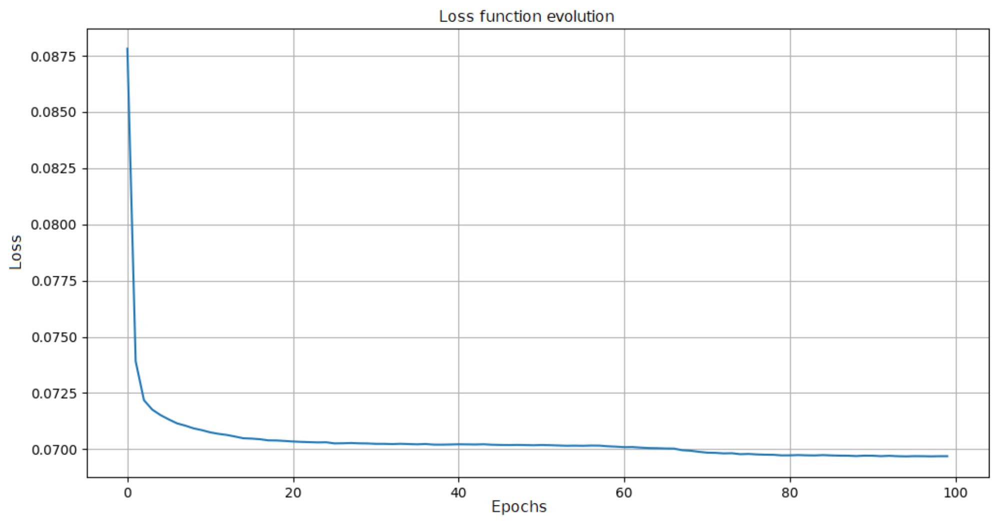


Figure 19. Evolution of the loss function value for the ANN from *Keras*.

6. Results and Discussion

The results presented in Table 11 provide a comprehensive view of the performance of the developed models. To validate the performance in ML models, two methods were employed: the confusion matrix [54] and the K-fold cross-validation [55]. In the case of DL techniques, in addition to the confusion matrix, the evaluation methods provided by the *Scikit-Learn* library and the *Keras* module were used, which provided the values of losses and accuracy obtained.

In the following sections, we analyze these three models, highlighting first the obtained results and afterwards, discussing them. Additionally, we finally discuss the potential generalization to other topologies and reconfiguration algorithms.

All trained models, along with their optimization processes and configurations, as well as the final dataset (*KeyMonData*), are available in a public GitHub repository [40].

Table 11. Summary of results obtained from the development of ML and DL models.

		Confusion Matrix				*K-Fold Cross-Validation*	
		Accuracy	*Precision*	*Recall*	*F1 Score*	*Accuracy*	*Standard Deviation/(*,**) Loss*
Random Forest [25 estimators, entropy criterion]	*None Applied*	99.29	94.40	74.27	83.16	99.30	0.02
	RFECV	99.44	94.28	81.05	87.17	99.47	0.02
	kbest (n = 5)	97.79	65.97	12.55	21.08	97.80	0.02
	kbest (n = 8)	97.74	53.95	27.37	36.32	97.79	0.01
	PCA (n = 2)	97.55	12.52	00.67	01.27	97.35	0.01
	PCA (n = 4)	98.06	81.66	22.60	35.41	98.04	0.00
SVM [C = 1, kernel = rbf]	*None Applied*	97.23	89.32	06.95	12.83	97.79	0.01
	kbest (n = 5)	97.11	65.25	12.29	20.73	97.80	0.01
	kbest (n = 8)	97.05	71.19	09.30	16.46	97.79	0.01
	PCA (n = 2)	97.61	-	0	-	97.76	0.00
	PCA (n = 4)	97.61	-	0	-	97.76	0.00
ANN [(5, 5), relu, sgd]	*sklearn*	97.84	69.03	14.98	24.66	97.83	* -
	keras	97.84	72.17	13.49	22.74	97.85	** 0.0690

Note: * *Score()*/ ** *Evaluate()*.

6.1. Random Forest

The confusion matrix provided some metrics that were essential to evaluate the performance of the RF model. Firstly, if we only focus on *Accuracy*, the results seemed quite good for the five tests conducted with different dimensionality reduction techniques applied during the RF model training. However, when analyzing the *Precision* and *Recall* metrics, notable differences were observed. For example, PCA application with two components obtained a result with a good *Accuracy* value (97.55%), but the *Precision* was poor (12.52%), which indicated that the percentage of detected errors out of the total errors in the dataset was extremely low (0.67%). Therefore, the *F1 Score* was also significantly low (1.27%), as it is a metric that combines the values of *Precision* and *Recall*.

Accordingly, by analyzing all the metrics provided by the confusion matrix, it was determined that the dimensionality reduction technique that worked best in training the RF was the RFECV. It presented the lowest error prediction cost because it reduced the probability of false alarms and undetected real errors with high percentages in *Precision* (94.28%) and *Recall* (81.05%), respectively. The second best option to detect and predict errors was to use all the features of the dataset without applying any reduction technique. The results achieved by evaluating with the K-fold cross-validation method showed the same accuracy values that the ones obtained with the confusion matrix, confirming the good performance of the RFECV technique application.

6.2. Support Vector Machine

To evaluate the performance of the SVM model, it is important to emphasize the significant computational cost and time required to train the classifier. In this case, 325 h were needed to complete the process of running and validating the results of the five tests. Regarding the confusion matrix, it can be noticed at first glance that in both cases of PCA application, undefined values were obtained for the *Precision* and *F1 Score* metrics. This was because the SVM model failed to make any error predictions.

Therefore, although a high percentage of *Accuracy* was achieved (97.61%), the model was not performing the classification correctly, and the error cost was very high. In the case of the *Precision* metric, the best results were obtained by not applying any dimensionality reduction technique to the dataset (89.32%), as it could be observed that in the Univariate Feature Selection technique or *SelectKBest()*, efficiency decreased as the number of configured features was reduced (65.25%). On the other hand, the *Recall* metric showed low values in all tests, indicating that the proportion of errors that were correctly identified was very small.

With the analysis presented, it can be conclusively determined that the SVM was not sufficiently efficient to meet the objective of detecting and predicting errors. Regarding the K-fold cross-validation method, similar accuracy values were obtained, just as with the RF. We deem it essential to delineate the limitations of the SVM model, as this will substantiate our rationale for eschewing its application in error detection within SGs. The findings elucidate the inherent unsuitability of the SVM model for this particular task, thereby offering critical insights that can inform a more judicious model selection in analogous applications or use cases.

6.3. ANNs

Given that in the development of DL techniques, no dedicated steps are performed for feature scoring and dimensionality reduction of the dataset, the evaluation focused on comparing the results obtained by the two ANNs. As mentioned in Section 5.4.2, the performance of both developed ANNs was very similar. The achieved accuracies were identical (97.84%), and the other metrics obtained from the confusion matrix showed minimal differences between the two models. For example, a slightly higher *Recall* was observed for the *Scikit-Learn* ANN (14.98%), but both models indicated poor effectiveness in error detection. However, the *Keras* ANN produced higher *Precision* and therefore, a lower probability of false alarms (72.17%).

Furthermore, the evaluation methods provided by *Scikit-Learn* (*score()*) and *Keras* (*evaluate()*) were applied to both ANNs. In the first case, only the accuracy value of the model was obtained, while in the second case, the loss function was also provided. As expected, the results remained similar and, although the differences between the two models were minimal; if the effectiveness in detecting real errors was prioritized, the *Keras* ANN would be chosen. However, a higher *Recall* was observed in some simpler ML methods (such as RF when applying RFECV) compared to these more sophisticated DL methods, which may be attributed to the different levels of effectiveness of dimensionality reduction techniques used in ML models and the internal feature scoring performed by ANNs.

6.4. Generalization to Other Topologies and Reconfiguration Algorithms

These findings highlight the effectiveness of ML and DL techniques in improving the reliability of SGs and offer valuable insights for the future development of intelligent grid management systems.

It is important to clarify that our developed models are not intended as tailor-made solutions for a specific, fixed topology. Rather, they are optimized by employing randomized topologies that vary systematically in node numbers, connectivity degrees, and probabilistic topological models, applying the IEEE 123 Node Test Feeder parameters to achieve this diversity. While it is true that our model may not deliver peak performance for any particular topology, we emphasize that it achieves optimal average performance across a broad range of topologies, representing a trade-off approach that generalizes effectively across diverse network conditions.

Accordingly, additional investigation with different reconfiguration algorithms, or with AI techniques specifically tailored for certain topologies, would be desirable in the long term to compare to what extent the customization of models would yield better results. In any case, this initial study serves as a foundation for future analysis and proves its ability to enhance the DEN2DE algorithm, which was our main objective.

7. Conclusions

In this manuscript, we proposed a novel approach for predicting faults in SGs through the application of ML and DL models to support the reconfiguration process. Building upon the DEN2DE algorithm, we addressed the complexities of load redistribution in dense and heterogeneous SGs by applying advanced ML and DL techniques to enhance decision-making during the grid's reconfiguration phase. By utilizing real-world datasets such as SustDataED and integrating simulated photovoltaic energy production data from tools like PVWatts, this work demonstrated the feasibility of employing AI-driven models to predict potential grid faults and improve system resilience.

The proposed methodology effectively identified optimal routes for load balancing while minimizing errors in energy distribution. Among the evaluated models, the Random Forest classifier, optimized with RFECV as a dimensionality reduction technique, consistently demonstrated superior performance. This configuration achieved high levels of Precision (94.28%) and Recall (81.05%), ensuring a balanced error detection capability with a reduced false alarm rate, as compared to other models. Extensive testing and validation in simulated environments confirmed that this RF-RFECV configuration enhanced the robustness and adaptability of the SG, managing the uncertainties associated with renewable energy sources and dynamic changes within the grid topology effectively.

Future research could build on the current findings by investigating additional ML and DL models to further enhance fault prediction capabilities. Additionally, extending the application of these models by simulating additional time samples to gain further insights, or to a broader range of datasets beyond SustDataED, potentially encompassing different regions, scales, and topological configurations, even different reconfiguration algorithms, would contribute to improving the generalization and robustness of the proposed approach. An important direction for future work also includes evaluating the developed models in light of the intrinsic characteristics of smart grids, such as distributed generation, fluctuating load demands, and the integration of renewable energy sources, to better understand their adaptability and performance in real-world smart-grid environments.

Author Contributions: Conceptualization, D.C. and P.B.; methodology, D.C.; software, P.B.; validation, D.C. and P.B.; formal analysis, D.C., P.B., and D.L.-P.; investigation, D.C., P.B., and E.R.; resources, E.R.; data curation, D.C. and P.B.; writing—original draft preparation, D.C. and P.B.; writing—review and editing, E.R., D.L.-P., N.M., and J.D.-F.; visualization, D.L.-P., N.M., and J.D.-F.; supervision, D.C. and E.R.; project administration, E.R. and D.L.-P.; funding acquisition, E.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by project TED2021-131301B-I00 funded by MICIU/AEI/10.13039/501100011033, and by a grant from Universidad de Alcalá through "Contratos Predoctorales de Formación de Personal Investigador—FPI-UAH 2022".

Data Availability Statement: The original data presented in the study are openly available in KeyMonData at https://github.com/PaulaBartolomeMora/TFM (accessed on 14 October 2024).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANN	Artificial Neural Network
BNN	Bayesian Neural Network
BRITE	Boston University Representative Internet Topology Generator
CNN	Convolutional Neural Network
DL	Deep Learning
DNI	Direct Normal Irradiation
HVDC	High-Voltage Direct Current
IMGS	Islanded Microgrids
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory
MIQP	Mixed-Integer Quadratic Programming
ML	Machine Learning
MLP	Multilayer Perceptron
OSPF	Open Shortest Path First
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
PVOUT	Photovoltaic Power Potential
RBF	Radial Basis Function
RELU	Rectifier Lineal Unit
RF	Random Forest
RFECV	Recursive Feature Elimination with Cross-Validation
SG	Smart grid
SGD	Stochastic Gradient Descent
SVD	Singular Value Decomposition
SVM	Support Vector Machine

References

1. Vu, K.; Begovic, M.M.; Novosel, D. Grids get smart protection and control. *IEEE Comput. Appl. Power* **1997**, *10*, 40–44. [CrossRef]
2. Amin, S.M.; Wollenberg, B.F. Toward a smart grid. *IEEE Power Energy Mag.* **2005**, *3*, 34–41. [CrossRef]
3. Hauser, C.H.; Bakken, D.E.; Bose, A. A failure to communicate. *IEEE Power Energy Mag.* **2005**, *3*, 47–55. [CrossRef]
4. Zhang, Y.; Huang, T.; Bompard, E.F. Big data analytics in smart grids: A review. *Energy Inform.* **2018**, *1*, 1–24. [CrossRef]
5. Vale, Z.; Faria, P.; Abrishambaf, O.; Gomes, L.; Pinto, T. MARTINE—A platform for real-time energy management in smart grids. *Energies* **2021**, *14*, 1820. [CrossRef]
6. Moreno Escobar, J.J.; Morales Matamoros, O.; Tejeida Padilla, R.; Lina Reyes, I.; Quintana Espinosa, H. A comprehensive review on smart grids: Challenges and opportunities. *Sensors* **2021**, *21*, 6978. [CrossRef]

7. Pál, D.; Beňa, L.; Kolcun, M.; Čonka, Z. Optimization of Active Power Losses in Smart Grids Using Photovoltaic Power Plants. *Energies* **2022**, *15*, 739. [CrossRef]
8. Storti, G.L.; Possemato, F.; Paschero, M.; Alessandroni, S.; Rizzi, A.; Mascioli, F.M.F. Active power losses constrained optimization in smart grids by genetic algorithms. In Proceedings of the Neural Nets and Surroundings: 22nd Italian Workshop on Neural Nets, WIRN 2012, Salerno, Italy, 17–19 May 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 279–288.
9. Rodriguez, F.J.; Fernandez, S.; Sanz, I.; Moranchel, M.; Bueno, E.J. Distributed approach for smartgrids reconfiguration based on the OSPF routing protocol. *IEEE Trans. Ind. Inform.* **2015**, *12*, 864–871. [CrossRef]
10. Carrascal, D.; Rojas, E.; Carral, J.A.; Martinez-Yelmo, I.; Alvarez-Horcajo, J. Topology-aware scalable resource management in multi-hop dense networks. *Heliyon* **2024**, *10*, e37490. [CrossRef]
11. Schneider, K.P.; Mather, B.A.; Pal, B.C.; Ten, C.W.; Shirek, G.J.; Zhu, H.; Fuller, J.C.; Pereira, J.L.R.; Ochoa, L.F.; de Araujo, L.R.; et al. Analytic Considerations and Design Basis for the IEEE Distribution Test Feeders. *IEEE Trans. Power Syst.* **2018**, *33*, 3181–3188. [CrossRef]
12. Bhattarai, B.P.; Paudyal, S.; Luo, Y.; Mohanpurkar, M.; Cheung, K.; Tonkoski, R.; Hovsapian, R.; Myers, K.S.; Zhang, R.; Zhao, P.; et al. Big data analytics in smart grids: State-of-the-art, challenges, opportunities, and future directions. *IET Smart Grid* **2019**, *2*, 141–154. [CrossRef]
13. Koshy, S.; Rahul, S.; Sunitha, R.; Cheriyan, E.P. Smart grid–based big data analytics using machine learning and artificial intelligence: A survey. *Artif. Intell. Internet Things Renew. Energy Syst.* **2021**, *12*, 241.
14. Massaoudi, M.; Abu-Rub, H.; Refaat, S.S.; Chihi, I.; Oueslati, F.S. Deep learning in smart grid technology: A review of recent advancements and future prospects. *IEEE Access* **2021**, *9*, 54558–54578. [CrossRef]
15. Kotsiopoulos, T.; Sarigiannidis, P.; Ioannidis, D.; Tzovaras, D. Machine learning and deep learning in smart manufacturing: The smart grid paradigm. *Comput. Sci. Rev.* **2021**, *40*, 100341. [CrossRef]
16. Rojas, E.; Carrascal, D.; Lopez-Pajares, D.; Alvarez-Horcajo, J.; Carral, J.A.; Arco, J.M.; Martinez-Yelmo, I. A Survey on AI-Empowered Softwarized Industrial IoT Networks. *Electronics* **2024**, *13*, 1979. [CrossRef]
17. Hemmatpour, M.H.; Mohammadian, M.; Gharaveisi, A.A. Optimum islanded microgrid reconfiguration based on maximization of system loadability and minimization of power losses. *Int. J. Electr. Power Energy Syst.* **2016**, *78*, 343–355. [CrossRef]
18. Sun, W.; Ma, S.; Alvarez-Fernandez, I.; Roofegari nejad, R.; Golshani, A. Optimal self-healing strategy for microgrid islanding. *IET Smart Grid* **2018**, *1*, 143–150. [CrossRef]
19. Sanz, I.; Moranchel, M.; Moriano, J.; Rodríguez, F.J.; Fernandez, S. Reconfiguration algorithm to reduce power losses in offshore HVDC transmission lines. *IEEE Trans. Power Electron.* **2017**, *33*, 3034–3043. [CrossRef]
20. Hosseinzadeh, J.; Masoodzadeh, F.; Roshandel, E. Fault detection and classification in smart grids using augmented K-NN algorithm. *SN Appl. Sci.* **2019**, *1*, 1627. [CrossRef]
21. Li, W.; Deka, D.; Chertkov, M.; Wang, M. Real-time faulted line localization and PMU placement in power systems through convolutional neural networks. *IEEE Trans. Power Syst.* **2019**, *34*, 4640–4651. [CrossRef]
22. Alhanaf, A.S.; Balik, H.H.; Farsadi, M. Intelligent fault detection and classification schemes for smart grids based on deep neural networks. *Energies* **2023**, *16*, 7680. [CrossRef]
23. Kaplan, H.; Tehrani, K.; Jamshidi, M. Fault diagnosis of smart grids based on deep learning approach. In Proceedings of the 2021 World Automation Congress (WAC), Taipei, Taiwan, 1–5 August 2021; pp. 164–169.
24. Efatinasab, E.; Sinigaglia, A.; Azadi, N.; Susto, G.A.; Rampazzo, M. Adversarially Robust Fault Zone Prediction in Smart Grids with Bayesian Neural Networks. *IEEE Access* **2024**, *12*, 121169–121184. [CrossRef]
25. Marashi, K.; Sarvestani, S.S.; Hurson, A.R. Identification of interdependencies and prediction of fault propagation for cyber–physical systems. *Reliab. Eng. Syst. Saf.* **2021**, *215*, 107787. [CrossRef]
26. Ding, T.; Lin, Y.; Bie, Z.; Chen, C. A resilient microgrid formation strategy for load restoration considering master-slave distributed generators and topology reconfiguration. *Appl. Energy* **2017**, *199*, 205–216. [CrossRef]
27. Himeur, Y.; Alsalemi, A.; Bensaali, F.; Amira, A. Building power consumption datasets: Survey, taxonomy and future directions. *Energy Build.* **2020**, *227*, 110404. [CrossRef]
28. Makonin, S.; Ellert, B.; Bajić, I.V.; Popowich, F. Electricity, water, and natural gas consumption of a residential house in Canada from 2012 to 2014. *Sci. Data* **2016**, *3*, 160037. [CrossRef]
29. Anderson, K.; Ocneanu, A.; Benitez, D.; Carlson, D.; Rowe, A.; Berges, M. BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SustKDD), Beijing, China, 12–16 August 2012.
30. Beckel, C.; Kleiminger, W.; Cicchetti, R.; Staake, T.; Santini, S. The ECO data set and the performance of non-intrusive load monitoring algorithms. In Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, BuildSys'14, Memphis, TN, USA, 3–6 November 2014; pp. 80–89. [CrossRef]
31. Monacchi, A.; Egarter, D.; Elmenreich, W.; D'Alessandro, S.; Tonello, A.M. GREEND: An energy consumption dataset of households in Italy and Austria. In Proceedings of the 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), Venice, Italy, 3–6 November 2014; pp. 511–516. [CrossRef]
32. Makonin, S. HUE: The hourly usage of energy dataset for buildings in British Columbia. *Data Brief* **2019**, *23*, 103744. [CrossRef]
33. Batra, N.; Gulati, M.; Singh, A.; Srivastava, M. It's Different: Insights into home energy consumption in India. In Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings, Roma, Italy, 11–15 November 2013. [CrossRef]

34. Kolter, J.; Johnson, M. REDD: A Public Data Set for Energy Disaggregation Research. *Artif. Intell.* **2011**, *25*, 59–62.
35. Barker, S.; Mishra, A.; Irwin, D.; Cecchet, E.; Shenoy, P.; Albrecht, J. Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes. *Proc. SustKDD.* **2012**, *111*, 108.
36. Pereira, L.; Quintal, F.; Gonçalves, R.; Nunes, N.J. SustData: A public dataset for ICT4S Electric Energy Research. In *ICT for Sustainability 2014 (ICT4S-14)*; Atlantis Press: Dordrecht, The Netherlands, 2014; pp. 359–368. [CrossRef]
37. Kelly, J.; Knottenbelt, W. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Sci. Data* **2015**, *2*, 150007. [CrossRef]
38. Global Solar Atlas Info. Available online: https://globalsolaratlas.info/ (accessed on 30 January 2024).
39. PVWatts Calculator. Available online: https://pvwatts.nrel.gov/pvwatts.php (accessed on 2 February 2024).
40. Development of Fault Detection and Prediction Machine Learning Models in Smart Grids Enviroments. Available online: https://github.com/PaulaBartolomeMora/TFM (accessed on 12 July 2024).
41. Medina, A.; Lakhina, A.; Matta, I.; Byers, J. BRITE: An approach to universal topology generation. In Proceedings of the MASCOTS 2001, Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Cincinnati, OH, USA, 15–18 August 2001; pp. 346–353. [CrossRef]
42. Zegura, E.; Calvert, K.; Bhattacharjee, S. How to Model an Internetwork. In Proceedings of the IEEE INFOCOM'96. Conference on Computer Communications, San Francisco, CA, USA, 24–28 March 1996.
43. Waxman, B.M. Routing of multipoint connections. *IEEE J. Sel. Areas Commun.* **1988**, *6*, 1617–1622. [CrossRef]
44. Barabási, A.L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512. [CrossRef] [PubMed]
45. Yehoshua, R. Random Forests. 2023. Available online: https://medium.com/@roiyeho/random-forests-98892261dc49 (accessed on 28 February 2024).
46. Sachinsoni. Unlocking the Ideas Behind of SVM (Support Vector Machine). 2023. Available online: https://medium.com/@sachinsoni600517/unlocking-the-ideas-behind-of-svm-support-vector-machine-1db47b025376 (accessed on 26 February 2024).
47. Importance of Permutation vs. Importance of Random Forest Features. Available online: https://qu4nt.github.io/sklearn-doc-es/auto_examples/inspection/plot_permutation_importance.html (accessed on 9 March 2024).
48. Tuning the Hyper-Parameters of an Estimator. Available online: https://scikit-learn.org/stable/modules/grid_search.html#grid-search (accessed on 9 March 2024).
49. Sklearn.feature_selection.RFECV. Available online: http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html (accessed on 10 March 2024).
50. Sklearn.feature_selection.SelectKBest. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.feature_selection.SelectKBest (accessed on 10 March 2024).
51. Sklearn.decomposition.PCA. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html#sklearn.decomposition.PCA (accessed on 11 March 2024).
52. Neural Networks. Available online: https://www.ibm.com/es-es/topics/neural-networks (accessed on 2 March 2024).
53. DLOps: MLOps for Deep Learning. Available online: https://valohai.com/blog/dlops/ (accessed on 26 February 2024).
54. Metrics and Scoring: Quantifying the Quality of Predictions. Available online: https://scikit-learn.org/stable/modules/model_evaluation.html#confusion-matrix (accessed on 11 March 2024).
55. Cross-Validation: Evaluating Estimator Performance. Available online: https://scikit-learn.org/stable/modules/cross_validation.html#computing-cross-validated-metrics (accessed on 12 March 2024).