Evaluation of Critical and Non-Critical Machines under Transient Stability using Classification Algorithms

CRISTIAN TOAQUIZA TOAQUIZA, CARLOS QUINATOA CAIZA
Technical University of Cotopaxi,
Latacunga,
ECUADOR

Abstract: - The lack of studies systematically comparing different supervised classification algorithms for distinguishing critical and non-critical machines under transient stability conditions was addressed using supervised machine learning algorithms. The New England test system, consisting of 39 buses and 10 generators, was used, employing Kron reduction to simplify the network to the generators' internal nodes. A dataset of 9900 instances was created from 990 three-phase fault scenarios (on lines and buses, with varying clearing times). These were classified as critical or non-critical based on the rotor's angular displacement from the center of inertia. Three classification algorithms neural networks (MLP), support vector machines (SVM), and random forests were compared, using only variables existing before the dynamic simulation to avoid information leakage. In a set of test instances (20% of the total, not seen during training), the random forest achieved the best overall performance (93.7% accuracy, F1 score of 0.75, AUC of 0.970). With the validation sets grouped by failure location, the F1 score of Random Forest is 0.70 (AUC 0.942), still above SVM and MLP. On the other hand, the SVM and MLP techniques showed greater sensitivity to the critical class (87.0% recall in both cases), although at the cost of lower accuracy and, therefore, a higher false alarm rate. These results from Random Forest may be a good approximation for replacing exhaustive dynamic simulation as a tool for the rapid detection of critical machines, reserving detailed simulation for the subset of cases identified as critical.

Key-Words: - Transient stability, critical machines, machine learning, Random Forest, SVM, neural networks, classification.

Received: May 17, 2026. Revised: July 19, 2026. Accepted: August 21, 2026. Published: November 23, 2026.

1 Introduction

Electric power systems operate in a constant dynamic state, exposed to disturbances such as short circuits, loss of transmission lines, or sudden generator shutdowns. Transient stability is understood as the system's ability to maintain synchronization between its generators after a significant disturbance [1], [2], [3]. When this ability is lost, the consequences can range from power fluctuations to partial or total power outages, which have a high social and economic cost.

Traditionally, transient stability assessment has been performed using numerical simulation in the time domain, solving the swing equation for each synchronous generator under each contingency being studied [1], [4]. Although this method is accurate, its computational cost increases rapidly with the number of generators and contingency scenarios being evaluated, leading to the development of accelerated simulation techniques [5]. As an alternative, direct or energy function methods (extended equal area criteria, EEAC, PEBS) were developed, which allow for stability calculation without integrating the differential equations over the entire post-fault interval [6], [7]. These methods originated from the practical direct analysis method initially proposed for multi-machine systems [8]. The distinction between critical and non-critical machines is a fundamental principle in these direct methods: in the event of a given disturbance, a small subset of generators, usually those closest to the fault or the most heavily loaded, accumulates most of the kinetic energy and determines whether the system maintains or loses synchronism; the rest of the machines remain virtually unchanged [8], [9]. By determining in advance which machines are essential for a particular contingency, the computational effort of stability analyses can be significantly reduced, focusing the exhaustive analysis solely on the relevant subset.

The development of machine learning techniques in recent years has enabled the creation of an additional approach to traditional methods,

replacing simulation or energy functions with models previously trained on simulated scenarios [10]. Architectures using hierarchical deep neural networks [11], one-dimensional convolutional neural networks [12], and XGBoost in conjunction with factorization machines [13] have been suggested. Core vector machines trained on large volumes of data [14] and support vector machines optimized for real-time evaluation [15] have also been proposed. Furthermore, the use of Random Forest has been suggested to identify critical distance relays in transient stability studies [16]. Random Forest has also been applied to the interpretation of incipient faults in power transformers from gases dissolved in dielectric oil [17] which supports the choice of this algorithm for classification problems in electrical systems with heterogeneous physical variables.

The availability of synchronized phasor measurements (PMU) has also driven methods for optimal sensor placement, with the aim of investigating stability [18]. This publication includes contributions directly related to: the evaluation of transient stability through principal component analysis with kernel and Gaussian processes [19], the analysis of small-signal stability in constant-speed induction generators [20], optimal LQG control in multi-machine systems [21], and the coordinated design of static VAR compensators and power stabilizers, explicitly considering the speed fluctuation between generators as a parameter for optimization [22].

In the context of Ecuador, the transient stability of the southern segment of the National Interconnected System has been analyzed in the face of three-phase faults and generation outages [23]. Likewise, recent studies by the national electricity grid operator have directly addressed the identification of critical generators through supervised artificial intelligence using PMU measurements [24], the separation of critical and non-critical machines through data mining based on the concept of inertia [25], and the prediction of the short-term stability of the electrical system using deep learning [26]. This background also demonstrates the continued relevance of the topic, as well as the lack of analyses that systematically compare different categorization methods using a single database and the same input values.

This article presents and evaluates a procedure for differentiating critical equipment from non-critical equipment under transient instability conditions. This procedure combines: (i) conventional dynamic simulation of the 39-bus New England test system, employing the oscillation equation and Kron reduction from the grid to the generator internal nodes; (ii) a marking standard based on the rotor-angle deviation with respect to the center of inertia; and (iii) the development and comparison of three supervised classification schemes: Random Forest, Support Vector Machines (SVM), and Multi-Layer Perceptron-class Neural Networks (MLP), using only data available before running the dynamic simulation. This allows the final classifier to be used as a fast-screening tool without information leakage between the predictors and the label. The remainder of the article is organized as follows: Section 2 presents the mathematical formulation of the problem of classifying critical and non-critical machines; Section 3 sets out the suggested solution (methodology, data set and models); the conclusions of the study are found in Section 4.

2 Problem Formulation

Figure 1, summarizes the seven stages of the process with their inputs and outputs.

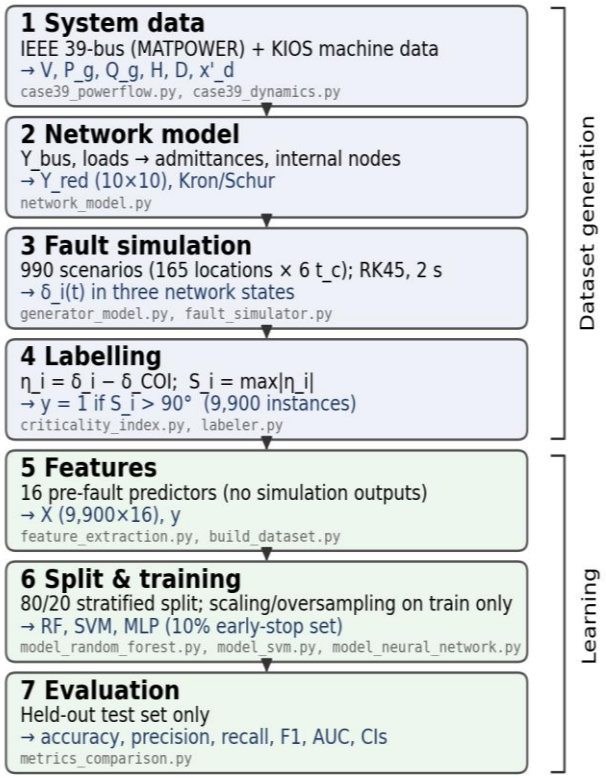

Fig. 1: Flowchart of the methodology
Source: Created by the authors.

Stages 1 through 4 generate the labeled dataset of 9900 generator scenario instances; stages 5 through 7 construct the 16 pre-failure predictors, split the data in an 80/20 ratio, and train the

classifiers. It's worth noting that they evaluate only on the reserved test set. Standardization and oversampling are applied only to the training set. Random Forest and SVM do not use a validation set (the hyperparameters are set a priori), and MLP uses 10% of its training data for early detection. The Python modules for each stage are named in Figure 1.

2.1 Dynamic Model of the Generator

The classical model allows the evaluation of the transient stability of a power system with n synchronous generators, where each machine i is represented by an internal emf with constant magnitude E_i, while the rotor angle δ_i evolves according to the swing equations, with δ_i in rad, $\Delta\omega_i$ in rad/s, $\omega_s = 2\pi f_0$, H_i in s, $P_{m,i}$ and $P_{e,i}$ in p.u. on the system base, in equations (1) and (2) [27].

$$\frac{d\delta_i}{dt} = \Delta w_i \quad (1)$$

$$\frac{d\Delta w_i}{dt} = \frac{\omega_s}{2H_i}\left(P_{mi} - P_{ei} - D_i\frac{\Delta\omega_i}{\omega_s}\right) \quad (2)$$

where H_i is the inertia constant, D_i the damping coefficient, ω_s the synchronous speed, $P_{m,i}$ the mechanical power (constant during the analysis period) and $P_{e,i}$ the electrical power delivered by machine i, calculated from the network reduced to the internal nodes of the generators by means of the Kron reduction [2], as given by equation (3), with $E_i = E_{i0}e^{j\delta_i}$, and Y_{ij} the elements of the reduced admittance matrix Y_{red}, obtained by eliminating all nodes of the network except the internal nodes of the n generators, as detailed in Section 2.2.

$$I_i = \sum_{j=1}^{n} Y_{ij}E_j, P_{e,i} = \mathrm{Re}\{E_i I_i^*\} \quad (3)$$

2.2 Construction of the Reduced Network

The Y_{red} matrix is obtained in four steps equation (4)-(7). First, the full bus admittance matrix of the m-bus system Y_{bus} is constructed from the line parameters (r, x, b) and the transformers with transformation ratios outside their nominal value represented by a complex tap $t = ae^{j\emptyset}$ [1], [2]. in equations (4) and (5). Where $y_s = \frac{1}{r+jx}$ is the series admittance of branch i-j. Second, the constant power loads are converted into equivalent admittances evaluated at the voltage of the previous power flow solution, a standard simplification of the network reduction method in classical transient stability [6], in equations (6). The third step adds an internal node for each generator, connected to its terminal bus by means of the transient reactance adjusted to the base of the system $x_{d,i}^{\prime sys} = x'_{d,i}\left(\frac{S_{\text{base}}}{S_{\text{gen},i}}\right)$ the Kron reduction (Schur complement) is applied to eliminate all nodes of the network, keeping only the n internal nodes of the generators in equations (7). Where the subscripts k and e correspond to the nodes to be preserved (internal to the generator) and the nodes to be eliminated (the rest of the network) respectively. At the pre-fault point the power computed from Y_{red} reproduces the power-flow generation with a maximum error of $5.7x10^{-7}$ p.u.

$$Y_{\text{bus},ji} = Y_{\text{bus},ji} - \frac{y_s}{t} \quad \frac{y_s + jb/2}{|t|^2} \quad (4)$$

$$Y_{bus,ij} = Y_{bus,ij} - \frac{y_s}{t^*}\ ;\ Y_{bus,ij} = Y_{bus,ij} - \frac{y_s}{t} \quad (5)$$

$$S_{\text{load},k} = P_L + jQ_L$$

$$Y_{load,k} = \frac{S^*_{loads,k}}{|V_k|^2} \quad (6)$$

$$Y_{\text{red}} = Y_{kk} - Y_{ke}Y_{ee}^{-1}Y_{ek} \quad (7)$$

2.3 Critical Machine Criterion

To remove the common-mode motion of the system, the rotor angle of each generator is referenced to the center of inertia (COI), and is calculated using equations (8) and (9). The severity index of machine i in failure scenario k is defined as the maximum relative angular oscillation observed during the simulation in (10), and machine i is also classified as critical in contingency k in (11) converts simulated angles into a severity index and labels. The threshold $\theta_c = 90°$ is adopted as a conservative engineering criterion motivated by the classical power–angle curve. For a machine connected to an infinite bus through a lossless reactance, $P_e = P_{max}\sin\delta$ who maximum transfer occurs at 90° and the synchronizing power coefficient $\frac{dP_e}{d\delta} = P_{max}\cos\delta$ vanishes at 90°; beyond this angle, a further increase in angle no longer produces additional restoring power. In a multimachine system, the relevant angle is the separation between the machine and the rest of the system. Treating machine i as the critical group and the remaining machines as the non-critical group N, the one-machine-infinite-bus angle is $\delta_{i,N} = \delta_i - \delta_N = \eta_i \cdot \frac{M_T}{M_T - M_i}$, with $M_i = 24H_i/\omega_s$ and $M_T = \sum_k M_k$. Hence $\eta_i = 90°$ corresponds to $\delta_{i,N} = 90° \cdot \frac{M_T}{M_T - M_i}$, $(98.9° - 100.3°$ for the machines, sice MT/(MT-Mi)=1.10-1.11), With network conductances, $P_{e,i} = E_i^2 G_{ii} + \sum_{j\neq i} E_i E_j\left(G_{ij}\cos\delta_{ij} + B_{ij}\sin\delta_{ij}\right)$, whose maximum is shifted from 90° by an angle that depends on G_{ij}/B_{ij}. Therefore, 90° is an early-

warning reference rather than an exact stability limit, and actual loss of synchronism is identified separately with $\theta_u = 180°$.

$$\delta_{COI}(t) = \frac{\sum_{i=1}^{n} H_i \delta_i(t)}{\sum_{i=1}^{n} H_i} \quad (8)$$

$$\eta_i(t) = \delta_i(t) - \delta_{COI}(t) \quad (9)$$

$$S_i^k = max_t|\eta_i(t)| \quad (10)$$

$$y_i^k = \begin{cases} 1, si\ S_i^k > \theta_c \\ 0, \quad if\ otherwise \end{cases} \quad (11)$$

2.4 Classification Problem Formulation

The objective is to formulate a binary classification model $f: X \rightarrow \{0,1\}$. Starting from a vector of variables x_i^k known before solving (1)-(3) for scenario k, the machine's own parameters, pre-failure condition and contingency description, predict the label y_i^k defined in (11), without resorting to the complete dynamic simulation, for which equation (12). Where $d_{e(i,k)}$ represents the electrical distance (equivalent transfer impedance) between the internal node of generator i and the fault point of scenario k, and t_c^k corresponds to the fault clearing time. The performance of $f(\cdot)$ is evaluated using the standard binary classification metrics: accuracy, precision, recall (sensitivity), and F1-score, defined in equation (13). where TP, TN, FP, and FN correspond to the true positives, true negatives, false positives, and false negatives, respectively, represented in equations (14) through (16).

$$\hat{y}_i^k = f(x_i^k), x_i^k \in \mathbb{R}^{16} \quad (12)$$

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN} \quad (13)$$

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

$$F1 = 2 * \frac{Precision \cdot Recall}{Precision + Recall} \quad (16)$$

In Section 3, the criteria are compared primarily in terms of sensitivity (recall) and F1-score for the critical category, as well as overall accuracy. This is done in accordance with background research using large-scale simulation [15], given the class imbalance characteristic of contingencies, where a small part of the system consists of critical generators. Since only 13.2% of cases are critical, accuracy alone is not informative; emphasis is placed on completeness, F1 score, area under the ROC curve (AUC), and Matthews correlation coefficient (MCC), along with false positive and false negative rates (FPR, FNR).

3 Problem Solution

3.1 Simulation Methodology and Data Set Generation

The methodology suggested in Section 2 was implemented entirely in Python. The power flow and dynamic data module uses only the inertia H, damping D, and transient reactance x'_d from the KIOS data in the classical model; the AVR and regulator models are not included in the simulation. The result is a reduced admittance matrix Y_{red} of size $10x10$whose nodes are the generator internal nodes. The swing equations (1)–(2), with the electrical power given by Eq. (3), are integrated with the explicit Runge–Kutta 4(5) method (RK45, scipy.integrate.solve_ivp) using a maximum step of 2 ms.

Integration is prematurely stopped by an event when any generator exceeds 360° of angular excursion about the center of inertia. This condition indicates that the system has irreversibly lost synchronism. Because the IOC includes the diverging machine, the severity indices of the remaining machines are evaluated only up to the point of stopping; therefore, the labels describe the system when the first machine crosses 360°. The 990 scenarios comprise: (i) 234 self-clearing bus faults (39 buses × 6 clearing times $t_c \in \{0.05,0.08,0.10,0.15,0.20,0.30\}$ s); and (ii) 756-line faults in which the faulted branch is permanently disconnected after clearing. The 34 branches without transformers were faulted at 10%, 50% and 90% of their length, and the 12 transformer branches at both terminals, each with the 6 compensation times: (34×3 + 12×2) × 6 = 756. Each scenario was simulated for 2 s (or until loss of synchronism) and evaluated for each of the 10 generators, yielding $990 \times 10 = 9{,}900$ generator scenario instances. For each instance, the severity index S_i^k defined in Eq. (10), the maximum angular excursion with respect to the COI, the time at which it occurs, and the kinetic energy at fault clearing were stored. Labels were assigned with the absolute-threshold criterion of Eq. (11), using $\theta_c = 90°$ and $\theta_u = 180°$. The composition of the dataset is shown in Table 1. The 990 scenarios correspond to 165 distinct fault locations (39 nodes and 126-line positions).

The resulting dataset is naturally unbalanced (Table 1): critical instances represent only 13.2% of the total, which aligns with the physical principle that only a small group of machines is severely affected in each contingency. Of the 1310 instances labeled as critical ($\theta_c = 90°$), 791 (60.4%) also

correspond to actual loss of synchronism ($\theta_c = 180°$), while the remaining 519 are critical cases that remain synchronized.

Table 1. Composition of the generated dataset

Concept	Value
Bus failure scenarios	234
Line failure scenarios	756
Total number of failure scenarios	990
Generator-scenario instances	9900
Critical instances (label = 1)	1310 (13.2 %)
Non-critical instances (label = 0)	8590 (86.8 %)
Instances from line faults	7560 (76.4 %)
Instances in bus faults	2340 (23.6 %)
Instances with real loss of synchronization (>180°, $is_{unstable}$)	791(8.0%)
Different fault locations	165 (39 bus + 126 line)
Scenarios stopped at 360°	521 (52.6%)

Source: Created by the authors.

This subdivision is consistent with the 90° criterion identifying machines at risk, including those that subsequently lose synchronism (791 of the 1,310 critical instances). The Figure 2 shows the trajectory of $\eta_{i(t)}$ for the 10 generators in response to a three-phase fault at bus 16 cleared in 0.15 seconds (one of the 990 scenarios in the dataset). Generator 7 (bus 36) loses synchronism (η > 180°), while generator 6 (bus 35) is labeled critical (η > 90°) but remains synchronized. Only 42 of the 1,310 critical cases (3.2%) return to below 45° before the end of the simulation, meaning the label mainly identifies machines that do not return.

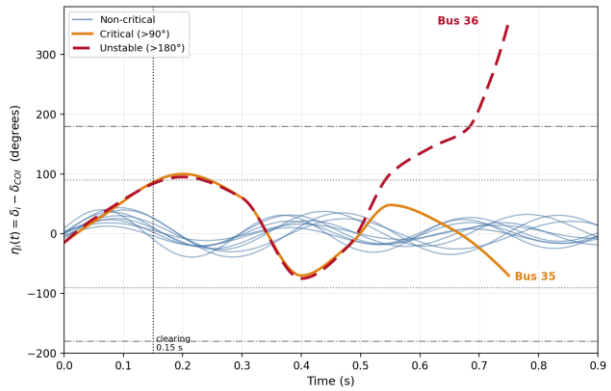

Fig. 2: Three-phase fault at Bus 16, clearance tc =0.15 s
Source: Created by the authors.

The network is represented in three states: pre-fault $\left(Y_{\text{red}}^{\text{pre}}\right)$; fault-on (for a bus fault, the faulted node is grounded through a shunt admittance of 10^4 p.u. before the Kron reduction; for a line fault, an auxiliary node at 10, 50 or 90% of the line is grounded); and post-fault $\left(Y_{\text{red}}^{\text{post}}\right)$, with the faulted branch removed for line faults and the original network for bus faults). Eqs. (1)–(2) are integrated with $Y_{\text{red}}^{\text{fault}}$ for $0 \leq t < t_c$ and with $Y_{\text{red}}^{\text{post}}$ for $t \geq t_c$.

3.2 Severity versus Clearance Time

The severity of a fault depends not only on its location but, crucially, on how long it takes to clear. Figure 3 shows, for four representative locations, the number of generators labeled as critical based on the clearing time: at bus 16, no generator is critical if the fault clears in 0.10 seconds or less, but 10 out of 10 are critical if it clears in 0.30 seconds; bus 30, on the other hand, remains benign across almost the entire range. This behavior explains why clearing_time is the second most important variable in the model.

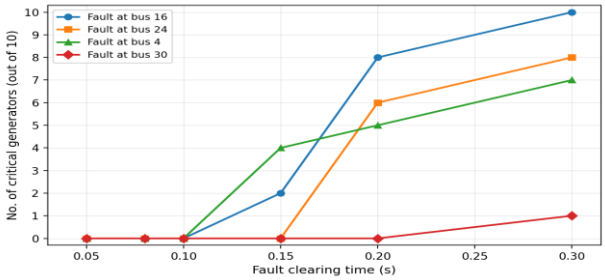

Fig. 3: Severity vs. clearance time
Source creador por los autores

3.3 Criticality Frequency per Generator

Table 2 summarizes, for the 234 bus faults (39 buses × 6 clearing times), the frequency with which each generator was labeled critical. The denominator of each percentage is 234 (one instance per generator, as well as the scenarios). The average angular excursion is the mean of

Table 2. Criticality per generator in 234 Bus Failures

Generator bus	Number of times critical	% of times critical	Average angular excursion
31	37 / 234	15.8 %	76.0°
32	35 / 234	15.0 %	70.3°
39	33 / 234	14.1 %	64.3°
38	30 / 234	12.8 %	74.4°
30	27 / 234	11.5 %	54.8°
36	22 / 234	9.4 %	58.7°
35	20 / 234	8.5 %	59.0°
34	19 / 234	8.1 %	51.6°
33	16 / 234	6.8 %	48.1°
37	15 / 234	6.4 %	49.2°

Source: Created by the authors.

S_i^k (in degrees) over these 234 scenarios; it is capped at 360° because integration stops at that angle, so it is a lower bound for unstable cases. The generator at bus 31, the reference bus (reference angle) of the power flow solution, was critical in 15.8% of cases, followed by buses 32 (15.0%) and 39 (14.1%); the generators at buses 37, 33, and 34 were the least affected (6.4–8.1%).

3.4 Predictor Variables and Model Configuration

Strictly following the formulation of (12), the predictor variable vector x_k^i was created using only data available before solving the dynamics: machine parameters ($H, D, x_d^{'}$ in system function, E_0 and δ_0), pre-fault condition (P_g and Q_g inertia share H_{share}, generation share P_share, and load factor $\frac{P_g}{P_{max}}$), as well as the contingency description, fault type, clearing time, whether it occurs in a transformer, relative location along the line, whether the fault occurs at the generator's own bus, and the electrical distance (modulus of the transfer impedance) within the pre-fault network between the generator's internal node and the fault point. Information leakage between the predictor variables and the label was avoided by not using columns derived from the dynamic simulation (max_eta_deg kinetic energy). D is identical for all machines and contains no information; machine-level predictors take only ten distinct values and act as a generator identifier.

To ensure a fair comparison, the dataset was split in a stratified manner into training (80%, 7920 instances: 6872 non-critical and 1048 critical), test (20%, 1980 instances: 1718 non-critical and 262 critical), with a fixed seed (random_state = 42), and using the same split for all three algorithms. The random Forest was trained with 300 trees and balanced weighting (class_weight = "balanced"), without the need to scale the variables. The variables were standardized to zero mean and unit standard deviation, with only the training sets, before the SVM was trained with the radial kernel (RBF), C = 10, γ = "scale", and balanced class weighting. The neural network (MLP) was configured with ReLU activation, L2 regulation ($\alpha = 10^{-3}$), early stopping, and two hidden layers (32 and 16 neurons). Since scikit-learn does not allow weighting classes in MLPClassifier, the imbalance was resolved by overrepresenting the minority class only in the training set; the test set was left unchanged so that the metrics reflected the authentic performance on unbalanced data. MLP uses an internal 10% of the oversampled training set for early detection; Random Forest and SVM do not use a validation set, and their hyperparameters are fixed a priori, without adjustment. This is because the ten instances of a scenario, as well as the six clearance times for a location, are correlated.

3.5 Learning Model Results

Table 3 summarizes the performance of the three algorithms on the test set consisting of 1980 instances, which were not used during the training phase. The area under the ROC curve (AUC) is included as a complementary measure of overall discriminatory power. Random Forest exhibits the best overall performance (Table 3): 93.7% accuracy, 78.4% precision, and an F1-score of 0.75. This is accompanied by a confusion matrix showing 73 false negatives and 52 false positives out of a total of 1980 instances (Table 4). SVM and MLP achieve the same recall of the critical class 87.0%, meaning that only 34 of 262 critical machines were not detected by either of them, bu at the cost of substantially lower precision (49.0% and 53.4%, respectively), corresponding to 237 and 199 false alarms compared to 52 for the Random Forest algorithm. Its overall accuracy (86.3% and 88.2%) is also lower than that of the Random Forest algorithm (93.7%).

Table 3. Comparison of metrics between models on the test set

Model	Accuracy	Precision	Recall	F1-score	AUC
Random Forest	0.9369	0.7842	0.7214	0.7515	0.970
SVM	0.8631	0.4903	0.8702	0.6272	0.940
Neural Network (MLP)	0.8823	0.5340	0.8702	0.6618	0.953

Source: Created by the authors.

Table 4 shows the confusion matrices of the three models on the same test set, which makes it possible to directly observe the trade-off between false positives and false negatives mentioned previously.

Table 4. Results of the confusion matrices of Random Forest, SVM and Neural Network (MLP) on the test set (1980 instances).

Model	Real/Predicted	No Crit.	Crit.
Random Forest	No Crit	1666	52
	Crit	73	189
SVM	No Crit	1481	237
	Crit	34	228
MLP	No Crit	1519	199
	Crit	34	228

Source: Created by the authors.

Figure 4 shows the four metrics from Table 3 for the three models. The horizontal axis lists the classifiers, the vertical axis shows the metric value (0-1), and each group of bars represents accuracy, completeness, and F1 score. The dashed line represents the accuracy of a trivial classifier that always predicts non-critical values (0.868). Random Forest is the best in accuracy, completeness, and F1 score, thus producing the fewest false alarms, while SVM and MLP are the best in completeness (0.870 vs. 0.721), meaning they miss fewer critical machines.

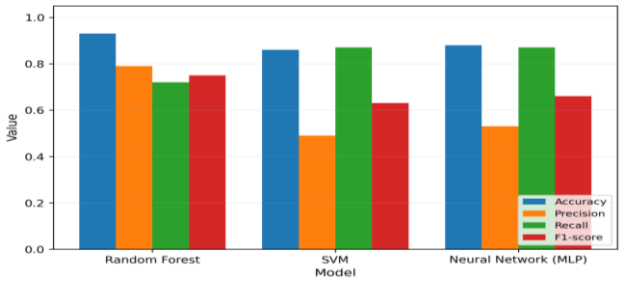

Fig. 4: Comparison of accuracy, precision, recall and F1-score between the three models
Source: Created by the authors.

Table 5. Validation scheme: F1-score / AUC (mean of 20 repetitions)

Validation	Random Forest	SVM	MLP
Random split	0.752 / 0.970	0.611 / 0.931	0.639 / 0.944
By contingency	0.755 / 0.971	0.605 / 0.926	0.636 / 0.939
By location	0.696 / 0.942	0.582 / 0.912	0.596 / 0.920

Source: Created by the authors.

The evaluation was repeated 20 times with three types of partitioning (by instance, by contingency and by location of the failure) and it was found that Random Forest maintains its performance when grouping by contingency (F1 ≈ 0.75), but reduces it when grouping by location (F1 of 0.696), and still outperforms SVM and MLP in 19 of the 20 repetitions, described in Table 5.

Table 6 and Figure 5 summarize the most influential variables, according to Random Forest, as the mean decrease in Gini impurity over the 300 trees, normalized so that the 16 importances sum to 100%, together with the permutation importance on the test set (drop in F1-score after shuffling a variable, 30 repetitions).

Table 6. Importance of variables — top 10 (Random Forest)

Variable	Permutation ΔF1	Importance
electrical distance	0.52	57.0 %
Clearance time	0.186	21.0 %
Relative position along the line	0.011	4.2 %
Fault at the generator's own bus	0.041	3.6 %
Load factor	0.006	2.6 %
Initial angle (δ0)	-0.010	1.8 %
Internal FEM	-0.005	1.7 %
Pg_share participation	-0.008	1.6 %
Active power (P_{share})	-0.007	1.6 %
Fault on a transformer branch	-0.033	1.5 %

Source: Created by the authors.

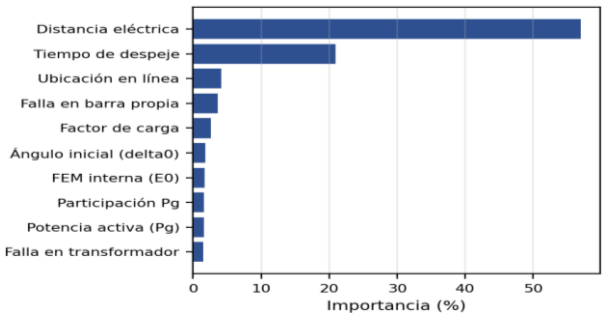

Fig. 5: Importance of the ten most influential predictor variables (Random Forest).
Source: Created by the authors.

Table 6 show that 57% of the total importance is concentrated in the electrical distance to the fault point, followed by the clearing time (21%). In sum, these two variables, which are directly related to electrical proximity and the temporal severity of the contingency, represent approximately 78% of the total. On the other hand, the intrinsic parameters of the machines ($H, D, x_d^{'}$) do not appear among the ten most significant variables. This result is consistent with the physical basis of the direct methods (PEBS/EEAC) described in Section 1, which determine that electrical proximity to the fault is the main element of criticality. Furthermore, the results Demuestran que Random Forest supera a SVM y MLP en puntuación F1 y AUC para la clasificación de contingencias críticas. Because impurity-based importance is biased by correlated predictors, it was supplemented with permutation importance and a feature group ablation (Random Forest, grouped by contingency). Removing electrical distance reduces F1 from 0.747 to 0.403, while on its own it reaches 0.576. Generator parameters have individual importance close to zero, but removing the group reduces F1 to 0.555: they are strongly correlated.

From the perspective of its application as a screening tool, the choice between models depends on which of the two error classes is more costly in

relative terms (Table 3). If the aim is to reduce the number of detailed verification simulations performed after screening, "Random Forest is preferable when the goal is to limit unnecessary verification simulations: it yields 52 false positives versus 237 (SVM) and 199 (MLP). However, it misses 73 of the 262 critical machines (false-negative rate 27.9%), whereas SVM and MLP miss only 34 (13.0%). Therefore, the choice depends on the relative cost of each error type. If, on the other hand, the predominant risk is failing to identify a truly critical machine, which could lead to an unexpected crash, SVM or MLP are more appropriate due to their higher recall, accepting a higher number of unnecessary verifications as an acceptable cost given the risk of a false negative. The classification is not an artifact of the decision threshold: with a coincident recovery of 87%, the false positive rate is 7.6% for Random Forest, 11.1% for MLP, and 13.5% for SVM.

3.6 Implementation in Python and Reproducibility

The research was implemented as a modular Python pipeline (numpy, scipy, pandas, scikit-learn, joblib, matplotlib) organized into the seven stages shown in Figure 1, each an independent module of pure functions. A single seed (random_state = 42) controls splitting, model initialization, and oversampling, and splitting is calculated only once, so all classifiers are evaluated on the same 1980 test instances. main.py runs the entire pipeline in approximately 2 minutes, and config.py stores all adjustable parameters. dataset.csv (9900 instances) can be regenerated, and Tables 3 and 4 can be reproduced, even with scikit-learn version differences.

3.7 Limitations and Sensitivity Analyses

Dos análisis evalúan la robustez de las etiquetas. Primero, para umbrales θc de 60°, 75°, 90°, 105°, 120° y 180°, la prevalencia crítica es 29,5%, 22,0%, 13,2%, 10,2%, 8,8% y 8,0%, respectivamente, mientras que Random Forest mantiene F1 = 0,74–0,87 y AUC = 0,95–0,99 (divisiones agrupadas por contingencia). Segundo, debido a que el IOC incluye la máquina divergente, los ángulos de las otras máquinas derivan una vez que una máquina pierde el sincronismo; La integración en el dominio del tiempo se detuvo en 521 de los 990 escenarios, y los niveles de parada de 180°, 270°, 360°, 540° y 720° dan prevalencias críticas del 8,2%, 9,4%, 13,2%, 27,9% y 32,6%, respectivamente (53,0% sin parada). Por lo tanto, las etiquetas describen el sistema cuando la primera máquina cruza los 360° y no son comparables con las etiquetas de otras reglas de parada; el análisis de COI del grupo no divergente (formulación de clúster crítico) se pospone para trabajos futuros.

4 Conclusion

This article presents a comprehensive method for classifying critical and non-critical machines in terms of transient stability. This method combines classical dynamic simulation of the 39-bus New England system (a Kron-reduced oscillation equation connecting the network to the internal nodes of the ten generators) with three supervised machine learning algorithms: an MLP-type Neural Network, a Random Forest, and an SVM. These algorithms were trained using only variables accessible before the simulation, thus preventing any information leakage between the target labels and the predictive variables. A dataset of 9900 instances (Table 1) was created from 990 systematically generated three-phase fault scenarios (line and bus faults, each with six clearing times). These were labeled using a physics-based criterion: the maximum angular excursion of the rotor about the center of inertia, set at a lower limit of 90°. This allowed us to simulate the expected imbalance between classes in this type of problem (13.2% of critical instances). The observations (Table 3) indicate that Random Forest presents the most optimal overall balance between precision (78.4%), accuracy (93.7%), and discriminative capacity (AUC = 0.970), while SVM and MLP also achieve a high level of sensitivity to the critical class (recall = 87.0% in both cases) at the expense of a higher number of false alarms. No single model strictly outperforms the others in all five metrics simultaneously, so the choice of algorithm should be based on the relative cost of a false negative (critical machine not detected) compared to the cost of a false positive (unnecessary detailed verification) in the specific application. With folds grouped by fault location, Random Forest still obtains F1 = 0.70 and AUC = 0.94, above SVM and MLP (F1 = 0.58–0.60). The labels depend on the 90° threshold and on the 360° stopping rule, to be revisited with a critical-cluster reference.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively

reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

[1] P. Kundur, *Power System Stability and Control*. New York, NY, USA: McGraw-Hill, 1994.

[2] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*, 2nd ed. Wiley-IEEE Press, 2003.

[3] P. Kundur, J. Paserba, and V. Ajjarapu, "Definition and classification of power system stability," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1387–1401, 2004, doi: 10.1109/TPWRS.2004.825981.

[4] C. Tang, C. Graham, M. El-Kady, and R. Alden, "Transient stability index from conventional time domain simulation," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1524–1530, 1994.

[5] P. Zadkhast, J. Jatskevich, and E. Vaahedi, "A multi-decomposition approach for accelerated time-domain simulation of transient stability problems," *IEEE Trans. Power Syst.*, vol. 30, no. 5, pp. 2301–2311, 2015, doi: 10.1109/TPWRS.2014.2361529.

[6] H. D. Chiang, *Direct Methods for Stability Analysis of Electric Power Systems: Theoretical Foundation, BCU Methodologies, and Applications*. John Wiley & Sons, 2011.

[7] D.-Z. Fang, T. Chung, Y. Zhang, and W. Song, "Transient stability limit conditions analysis using a corrected transient energy function approach," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 804–810, 2000, doi: 10.1109/59.867177.

[8] T. Athay, R. Podmore, and S. Virmani, "A practical method for the direct analysis of transient stability," *IEEE Trans. Power Apparatus Syst.*, vol. PAS-98, no. 2, pp. 573–584, 1979, doi: 10.1109/TPAS.1979.319407.

[9] S. Wang, J. Yu, and W. Zhang, "Power system transient stability assessment using couple machines method," 2017, doi: 10.48550/arXiv.1711.04256.

[10] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, "An artificial intelligence framework for on-line transient stability assessment of power systems," *IEEE Trans. Power Syst.*, vol. 4, no. 2, pp. 789–800, 1989, doi: 10.1109/59.193853.

[11] L. Zhu, D. J. Hill, and C. Lu, "Hierarchical deep learning machine for power system online transient stability prediction," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2399–2411, 2020, doi: 10.1109/TPWRS.2019.2957377.

[12] K. Gao, S. Yang, S. Liu, and X. Li, "Transient stability assessment for power system based on one-dimensional convolutional neural network," *Autom. Electr. Power Syst.*, vol. 43, no. 12, pp. 18–26, 2019.

[13] N. Li, B. Li, and L. Gao, "Transient stability assessment of power system based on XGBoost and factorization machine," *IEEE Access*, vol. 8, pp. 28403–28414, 2020, doi: 10.1109/ACCESS.2020.2969446.

[14] W. Hu et al., "Real-time transient stability assessment in power system based on improved SVM," *J. Mod. Power Syst. Clean Energy*, vol. 7, no. 1, pp. 26–37, 2019, doi: 10.1007/s40565-018-0453-x.

[15] B. Wang, B. Fang, Y. Wang, H. Liu, and Y. Liu, "Power system transient stability assessment based on big data and the core vector machine," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2561–2570, 2016, doi: 10.1109/TSG.2016.2549063.

[16] R. Vakili and M. Khorsand, "A machine learning-based method for identifying critical distance relays for transient stability studies," *Energies*, vol. 15, no. 23, p. 8841, 2022, doi: 10.3390/en15238841.

[17] A. Freire, J. Astudillo, C. Quinatoa, and F. Arias, "Interpretación de gases disueltos en aceite dieléctrico mediante bosques aleatorios para la detección de anomalías en transformadores de potencia," *Revista Técnica "energía"*, vol. 19, no. 2, pp. 90–98, 2023.

[18] M. Rashidi and E. Farjah, "Lyapunov exponent-based optimal PMU placement approach with application to transient stability assessment," *IET Sci. Meas. Technol.*, vol. 10, no. 5, pp. 492–497, 2016, doi: 10.1049/iet-smt.2015.0232.

[19] Y. Li and X. Gu, "Transient stability assessment of power systems based on KPCA and Gaussian process," *WSEAS Trans. Power Syst.*, vol. 9, pp. 178–184, 2014.

[20] B. Mehta, P. Bhatt, and V. Pandya, "Modelling of fixed speed squirrel cage induction generators for small signal stability assessment," *WSEAS Trans. Power Syst.*, vol. 9, pp. 360–375, 2014.

[21] A. S. Oshaba, "Stability of multi-machine power system by used LQG controller," *WSEAS Trans. Power Syst.*, vol. 9, pp. 143–154, 2014.

[22] E. S. Ali and S. M. Abd-Elazim, "Power system stability enhancement via new coordinated design of PSSs and SVC," *WSEAS Trans. Power Syst.*, vol. 9, pp. 428–438, 2014.

[23] D. Pila, C. Quinatoa, L. Camacho, and J. Vaca, "Transient stability analysis of the Ecuadorian electrical system: Case of the southern segment," *WSEAS Trans. Power Syst.*, vol. 19, pp. 360–373, 2024, doi: 10.37394/232016.2024.19.31.

[24] J. León, G. Colomé, and E. Tapia, "Identificación de generadores críticos ante problemas de estabilidad transitoria," *Revista Técnica energía*, vol. 21, 2025, doi: 10.37116/revistaenergia.v21.n2.2025.684.

[25] W. Noroña, D. Cajas, J. Lozada, and F. Chamba, "Análisis de estabilidad transitoria utilizando el concepto de inercia y minería de datos," *Revista Técnica energía*, vol. 22, 2025.

[26] E. Guañuna, G. Colomé, and E. Tapia, "Predicción del estado de estabilidad de corto plazo del sistema eléctrico de potencia utilizando aprendizaje profundo," *Revista Técnica energía*, vol. 22, 2026, doi: 10.37116/revistaenergia.v22.n2.2026.723.

[27] I. Boldea and L. N. Tutelea, Electric Machines. Boca Raton, FL, USA: CRC Press, 2021.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US