Engineering Applications of Artificial Intelligence 168 (2026) 114093

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

Hybrid Inception-Transformer model for signals classification: The case of electrical faults in power transformers

Elías Herrero Jaraba [a,*], Eduardo Martínez Carrasco [b], Anibal Antonio Prada Hurtado [b], María Teresa Villen Martínez [b], Guillermo Rios Gómez [b], David Hernando Polo [b], Julio David Buldain Pérez [a]

[a] Department of Electronic Engineering and Communications, University of Zaragoza, María de Luna, 1, 50018, Spain
[b] CIRCE Technology Center. Parque Empresarial Dinamiza, Avenida Ranillas Edificio 3D, 1ª Planta 50018, Spain

ARTICLE INFO

Keywords:
Deep neural networks
Electrical fault detection
Time series analysis
Machine learning
Pattern classification

ABSTRACT

This paper presents a hybrid deep learning model for fault detection in power transformers, addressing the limitations of conventional protection schemes under transient operating conditions. The proposed model, TransInception, integrates InceptionTime for efficient feature extraction in multivariate time series and Gated Transformer for capturing dependencies between variables. The architecture is modified by replacing the original gating mechanism with a linear double-layer output and removing a bottleneck layer responsible for handling temporal dependencies. The dataset used for training and testing was generated in a real-time digital simulation (RTDS) environment, consisting of an external grid, a delta-wye transformer, and a dynamic load. After training, the hybrid deep learning model was validated in a test grid specifically designed for this stage, where a parallel transformer configuration was implemented. This validation allowed for the evaluation of its performance in classifying internal, external, and no-fault conditions, as well as assessing cases of current transformer saturation. Additionally, sympathetic inrush conditions were studied to analyse the model's response to interactions between power transformers. As future work, efforts will focus on improving the model's adaptability to transient conditions and optimising its computational efficiency for deployment in substation protection systems.

1. Introduction

This research explores the use of deep learning-based algorithms to complement the protection of power transformers in electrical systems. Although traditional protection schemes, such as differential relays, have demonstrated reliable performance under standard operating conditions and certain types of faults, they present limitations in scenarios where transient phenomena and variations in system dynamics affect their precision and stability.

Among these limitations are the management of inrush and sympathetic inrush currents, which can lead to unintended trips (Bera, 2023); and the effects of saturation in current transformers (CT), which impact in the differentiation between internal and external faults during fault events (Abbasi, 2022). These factors may compromise the selectivity and stability of protection schemes, justifying the need for alternative strategies to enhance their performance under dynamic conditions.

In this context, a hybrid deep learning model called *TransInception* was implemented to help the conventional protection schemes to detect faults in power transformers under dynamic conditions. These hybrid model, integrates two advanced approaches: *InceptionTime* and *Gated Transformer*.

- *InceptionTime* was selected for its ability to efficiently process multivariate time-series signals through parallel convolutional

Abbreviations: AI, Artificial intelligence; CNN, Convolutional Neural Network; CSV, Comma-separated values; CT, Current transformers; dB, Decibels; DTW, Dynamyc time warping; EF, External Fault; eRPE, Efficient relative coding; GPU, Graphics processing unit; GTN, Gated Transformer network; HHT, Hilbert-Huang transform; IF, Internal fault; LSTM, Long short-term memory; ML, Machine Learning; ML-Transformer, Neural model transformer; NF, Non-fault; NIR, No information rate; PCA, Principal component analysis; POW, Phase of the signal; RNN, Recurrent neural networks; RTDS, Real Time Digital Simulation; SNR, Signal-to-noise ratio; SVM, Support Vector Machines; tAPE, Absolute position coding

* Correspondence to: María de Luna, 1 50018 Zaragoza, Spain.
E-mail addresses: jelias@unizar.es (E.H. Jaraba), emartinez@fcirce.es (E.M. Carrasco), aaprada@fcirce.es (A.A.P. Hurtado), mtvillen@fcirce.es (M.T.V. Martínez), grios@fcirce.es (G.R. Gómez), dhernando@fcirce.es (D.H. Polo), buldain@unizar.es (J.D.B. Pérez).

https://doi.org/10.1016/j.engappai.2026.114093
Received 7 February 2025; Received in revised form 17 December 2025; Accepted 3 February 2026
Available online 5 February 2026

operations, enabling the extraction of local feature patterns in electrical signals (Ismail Fawaz et al., 2020).
- *Gated Transformer* was employed due to its capability to model temporal relationships and dependencies among multiple variables, improving the processing of long sequences through attention mechanisms (Liu et al., 2021).

The model was designed to perform real-time classification tasks, focusing on the detection and differentiation of operational events while balancing accuracy and time response.

The data used for training and validating the model were generated through simulations in a *real-time digital simulation (RTDS) environment*. RTDS is a hardware platform widely used worldwide to simulate the behaviour of electrical power systems in real time, with simulation step sizes ranging from 5 to 50 μs. RTDS can be configured to reproduce normal operating conditions, transient phenomena, and fault scenarios in the electrical system, providing a system response that closely matches the behaviour of the real power system (Sidwall and Forsyth, 2022). Several study cases have been defined to generate representative data, and evaluate the solutions with greater adaptability to diverse operating conditions. These simulations included power transformers operating under different load levels, variations in the impedance of the source, and saturation in the current transformers. Additionally, different fault scenarios were analysed, including internal zone short circuits and external zone faults, with the objective of evaluating the model's capability to identify anomalous conditions under various system configurations.

Finally, to assess the deep learning model's performance under challenging conditions not considered during the training process, tests were conducted using a parallel transformers scheme, allowing for an analysis of its response to transformer interactions, coupling effects, and variations in power flow. The evaluation of these scenarios, which represent challenging conditions requiring more advanced detection strategies, aims to determine the applicability of the trained deep learning model in electrical systems with a dynamic configuration.

For the sake of clarity and a better understanding of the content of this article, it is important to emphasise that in order to distinguish the term transformer used in two different technical fields (power electricity and neural networks), it has been decided to use the term decoder instead of transformer when referring to the neural network of the same name. In fact, in this article we only use the decoder part of the Transformer neural network.

2. Background and related work

Power transformer protection devices are based on traditional schemes such as differential protection, which have been widely used for fault detection. While these methods behave well under normal operating conditions, they face several limitations in more complex scenarios.

Studies have shown that differential protection schemes struggle to distinguish between internal and external faults, particularly when current transformer saturation occurs (Abbasi, 2022). Additionally, sympathetic inrush currents have been identified as a key factor causing unintended trips, affecting the reliability of protection systems (Bera, 2023). Several studies have focused on improving the fault identification under these scenarios using advanced signal processing and artificial intelligence techniques. For example, the Hilbert-Huang Transform (HHT) has been applied to differentiate inrush currents from internal faults, reducing false trips and improving fault classification accuracy (Wang et al., 2019a). Similarly, hybrid models combining time-frequency analysis and deep learning have demonstrated improved performance in transformer fault detection by enhancing feature extraction from transient signals (Zhou et al., 2020). Other studies have explored the use of wavelet transforms and support vector machines (SVM) to mitigate misoperations caused by CT saturation (Xiao et al., 2020). Despite these improvements, existing approaches still face challenges in generalisation and adaptability across varying operational conditions, leading to the need for more robust architectures.

At its core, what this paper is focusing on is a time series classification problem. Focused on a very specific application, but the primitives it handles are nothing more than time signals. This is why we should start with studies based on more standard and conventional machine learning techniques.

Firstly, the use of CNN was adapted to the processing of temporal signals, always with the same approach of such networks used in image processing. Even, some studies have explored the combination of CNN with LSTM for time series classification, with promising results. Vaibhava Lakshmi and Radha (2023) and Du et al. (2018) both proposed attention-based LSTM-CNN frameworks, achieving high accuracy in classifying time series data. Peng et al. (2019) and Xu et al. (2023) further improved on this by introducing multi-level networks and multi-scale convolutional neural networks with LSTM, respectively. Liang et al. (2021) and Li et al. (2021) focused on specific applications, such as automatic modulation classification and slide relative position matrix-based models, both achieving high accuracy. Wan et al. (2021) and Gharghory (2021) extended these models to multivariate time series classification, with the latter specifically focusing on remote sensing data.

Algorithms based on CNN opened the door or were the precursors to more deep learning approaches, but the real breakthrough came with the emergence of a model called InceptionTime. This model brought about a very large increase in classification accuracy compared to previous models. In fact, to this day it is a model whose behaviour is very similar to the most current models. InceptionTime, has been shown to outperform traditional methods such as HIVE-COTE in terms of accuracy and scalability (Ismail Fawaz et al., 2020). It has also been compared to other deep learning models, such as InceptionFCN, which was found to be more efficient in terms of training time and overall accuracy (Usmankhujaev et al., 2021) In a study by Wang et al. (2019b), a hybrid model combining Inception and LSTM modules was proposed, achieving a lower error rate than the baseline model. The potential of InceptionTime in early time series classification has been explored, with the model showing promise in terms of accuracy and earliness (Rußwurm et al., 2019). However, the incremental approach has been found to outperform InceptionTime in terms of both accuracy and earliness, albeit with a higher false positive rate (Miao et al., 2023). Lastly, the TEASER algorithm, which models early time series classification as a two-tier problem, has been shown to outperform competitors in terms of both earliness and accuracy (Schäfer and Leser, 2020).

InceptionTime is one of the cornerstones of this work, and shares the limelight with another ML-Transformers-based solution. A model developed in recent years, and which has gained great significance thanks to its fantastic behaviour in text analysis. A range of studies have explored the use of ML-transformers for time series classification. Wen et al. (2023) provides a comprehensive survey of ML-transformer schemes for this purpose, highlighting their strengths and limitations. Uchiyama (2023) and Jiang et al. (2022) both present ML-transformer-based models for time series classification, achieving high accuracy. Kambale et al. (2023) investigates the use of transfer learning with ML-transformer models, while (Woo et al., 2022) proposes ETSformer, a ML-transformer architecture specifically designed for time series forecasting.

Recent advancements in time series classification have seen the emergence of the Gated Transformer Network (GTN), which combines the strengths of ML-Transformer Networks with gating mechanisms (Liu et al., 2021). This approach has been further enhanced by the Multi-Modal Fusion ML-Transformer, which leverages the power of multi-modality and achieves high accuracy in classification tasks (Jiang et al., 2022). The ML-Transformer-based Time Series Classification model has also demonstrated exceptional performance in various data modalities (Uchiyama, 2023). For time series anomaly detection, the Dilated

Transformer Network has been proposed, which uses dilated convolution to extract long-term dependence features (Wu et al., 2022b). In the realm of time series forecasting, the Dateformer and Multi-resolution Time-Series ML-Transformer have been introduced, both of which significantly improve the accuracy and range of forecasting (Young et al., 2022)(Zhang et al., 2023). The Soft-DTW ML-Transformer, which uses soft dynamic time wrapping for early stopping criteria, has also shown promise in reducing prediction error rates (Ho et al., 2020). Lastly, the Probabilistic Decomposition Transformer model has been developed to provide hierarchical and interpretable probabilistic forecasts for complex time series (Tong et al., 2022).

In the literature review conducted, no studies were identified that explicitly combine InceptionTime with a Gated Transformer Decoder for time series classification, nor in the specific case of transformer fault analysis. While independent works address Inception-based models and Transformer-based models separately, the contribution of this work lies in the practical utility of integrating both approaches. The InceptionTime enables a robust multiscale extraction of transient features, while the Gated Transformer Decoder strengthens the ability to model long-term dependencies and filter irrelevant information. This integration aims to enhance the generalisation and adaptability of the classification process under operational conditions.

The ML-Transformers models together with the idea of using the two dimensions that multivariate time signals implicitly have -one in the time dimension and the other in the variable dimension-, combine the central idea of our work. In the following section we will analyse in detail its final composition, and how the different technologies and models used are combined.

With this basis in terms of the diverse theoretical and methodological background throughout the years prior to this work, it is worth noting the innovative proposals that we believe have been completed without exception in this article:

- From the point of view of faults in electrical transformers, the work presented provides a fast method that is essential in the field of electrical protection. As will be seen later, the entire procedure takes less than 20% of the time of a 50 Hz network cycle.
- On the other hand, the aim is to extract as much information as possible through a hybrid solution that extracts both temporal information and the dependency that can be extracted from the relationship between the signals chosen as inputs to the classifier.
- Likewise, a longer-term study of a solution to the problem of detecting and classifying electrical faults in power transformers is being initiated. In this first phase, the classification is based on three types of faults: internal, external, and no fault. Subsequently, work will be done on a more complex classifier by increasing the number of target classes, taking into account the same requirements presented here in terms of classifier performance and response time.
- Finally, a classifier is presented with performance at the high end of the classifiers studied, used both with temporal signals and specifically in the detection of faults in power transformers.

3. Data acquisition and preprocessing

In order to achieve the objective of detecting and classifying the possible electrical faults that may appear in an electrical power transformer, a set of data on a real or simulated installation is required. In this work we have opted for the simulation of a real transformer in which we will be able to create different types of electrical faults. Obviously, this option allows us great versatility and freedom to create faults without causing physical damage or unnecessary economic costs.

A Real Time Digital Simulator (RTDS) is used to simulate the different operating scenarios (fault scenarios, energising scenarios,..) in the electric system. Fig. 1 shows an image of the RTDS hardware used during the study.

Fig. 1. RTDS hardware.

The electric grid used to carry out the study is composed by a voltage source, a power transformer and a dynamic load, as can be seen in Fig. 2. In this figure we can see the different connections established for the process of data fault types (F1-F4) for the training and subsequent testing of the classifier. On the one hand, we have a 45 kV external voltage source connected to the primary winding of a power transformer. On the other hand, the secondary winding of the power transformer, which provides an output voltage of 13.8 kV, is connected to a dynamic load.

According to Fig. 2, the aim is to configure a protection zone around the transformer, in order to be able to react in a timely manner according to the nature and position of the detected electrical incident. For example, if there is an electrical fault outside this established protection zone, it would be advisable for the transformer to be able to continue operating without being switched off to avoid the costly process of rearming that would be necessary if it were to be switched off. It is these types of decisions that are being sought with practical developments such as the one presented in this article.

3.1. Simulation of fault scenarios

Four fault locations have been configured to simulate both internal and external faults to the protection zone on both sides of the power transformer. They appear in Fig. 2 with the callsigns F1, F2, F3 and F4. In such a way that:

- F1: Internal fault on the high voltage side.
- F2: Internal fault on the medium voltage side
- F3: External fault on the high voltage side.
- F4: External fault on the medium voltage side.

Likewise, and as can be seen in Fig. 3, the proposed scheme enables us to create different types of faults: single line to ground faults (AG,BG,CG), line to line faults (AB,BC,CA), double line to ground faults (ABG, BCG, CAG) and three phase faults (ABC).

But in addition to the faults detailed above, their impedance is also configured in a range between 0 Ω and 30 Ω, in steps of 2 Ω. Likewise, two final variations to be taken into account have been considered: the use of the point in the phase of the wave where the fault occurs, which is what we call the Phase of the signal (POW), of which 3 different values are taken: 0°, 90° and 180°; and the tests carried out with or without saturation of the current transformers.

And finally, different load and short-circuit conditions have been considered as case studies. Modifying the characteristic impedances of the high voltage source (45 kV) by around 20%, and likewise varying both the active and reactive power of the load by the same percentage.

All of the above contributes a high degree of variability to the database, allowing for a wide range of test types to be taken into account, enabling the network to learn and generalise with new examples. This not only enables the study presented in this work, but also serves for future developments, expanding the possibilities and configuration of future classifiers.

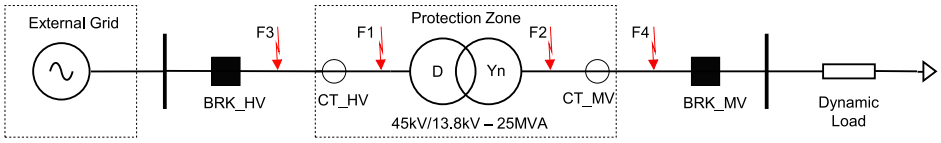


Fig. 2. One-line representation of the power transformer in the study.

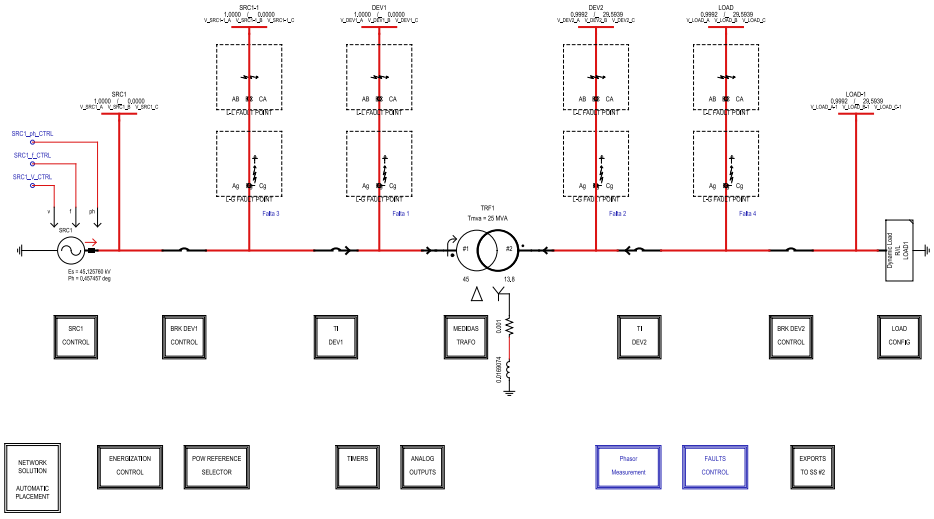


Fig. 3. Detailed single-line diagram showing the configuration of the electrical faults created in the tests using RTDS.

3.2. Measurement collection and dataset configuration

According to the IEC 61850-9-2 (Commission, 2011) standard, which defines the requirements for specific communications mapping systems in electrical substations, the appropriate sampling rate for protection applications, where high temporal resolution is required for fast detection, is 80 samples per cycle, or in other words, each sample must be collected every 250 μs.

This sampling rate is consistent with the Sampled Values streams provided by merging units commonly deployed in digital substations, ensuring that the simulated data are representative of practical operating conditions. At 50 Hz this corresponds to 4 kHz, and at 60 Hz to 4.8 kHz, which are the typical rates adopted in digital substations. It should be noted that no physical transformer was used during the acquisition; all signals were generated in the RTDS environment.

Likewise, and as can be seen in Fig. 3, the complete system offers great versatility for shaping and recording data of different natures. Within the wide range of these possibilities, data is finally recorded at the aforementioned rate according to the following list:

- Instantaneous current values for both the high and low voltage sides - we emphasise their importance in this work as they will be the data presented to the classifier-.
- PMU (Phasor Measurement Unit) values for medium and high voltage currents.
- Circuit breaker status.
- Digital value of each type of fault: No fault (NF), internal fault (FI), and external fault (FE) - this will be our ground truth in order to verify the performance of the classifier-.
- Digital values of the type of fault: AG, BG, CG, AB, BC, CA, ABG, BCG, CAG and ABC - as previously presented at the beginning of Section 3.1

All this data is collected in CSV format files. This makes it easy to read and process them with the programming languages used. However, the distribution and format in which these files are collected is not suitable for the subsequent training and testing phases required by the neural classifier proposed in this work. We will talk about this later.

Once the structure of the data to be recorded had been defined, different tests were carried out in each of the three different operating conditions, each lasting 600 msec, resulting in a total database of 6340 experiments. However, and specifically for this work, of all the data collected and stored in CSV files, we will only actively use both the instantaneous current values and the digital values of each type of fault. However, the number of patterns to be presented to the classifier will be much greater. This is because the classifier will be presented with time windows of 40 samples, corresponding to half a cycle of these (periodic) signals. This means that a larger set of input patterns will be obtained from each experiment.

3.3. Selection, preparation and preprocessing of the dataset

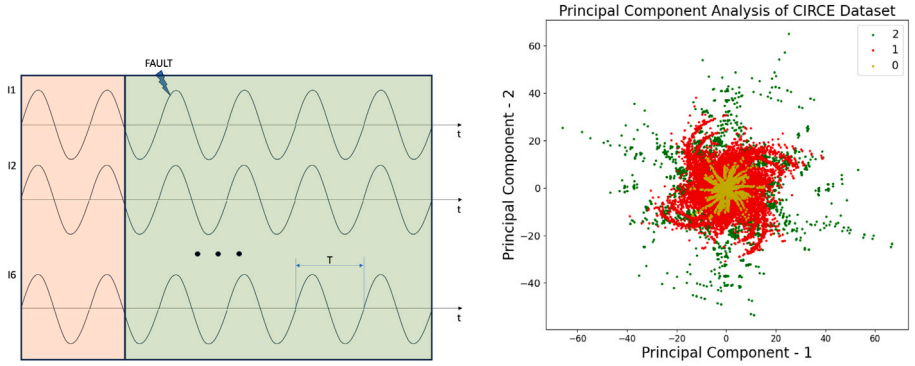
Specifically for this work, for each experiment data, both the instantaneous current values and the digital values of each type of fault stored in the CSV files will be actively used. Considering the number of data experiments available, the number of patterns to be presented to the classifier for training and testing processes will be much greater (> 6340) due to the use of time windows of 40 samples commented just before.

However, a large number of these time windows will be repetitive, and we will only focus on the time period closest to the moment when the power failure occurs. In this sense, it has been decided to consider only the data belonging to the time signals corresponding to 2 cycles before the failure and up to 5 cycles after it (Fig. 4(a)).

However, by way of a numerical summary, Table 1 summarises the entire database that has been created, and the distribution of this data for the training phase and the test phase. Normally this distribution is around 80%–20%, but in our case it is a distribution of 87%/13%. This is a little different to the standard usually observed in the specialised literature, but in our case we wanted to slightly unbalance this relationship due to the need to choose the critical experiments that we were interested in checking. In any case, the randomness in the selection of these two data sets remains intact.

Table 1
Database used for training and testing the classifier. The numeric columns correspond to the number of elements in each one. The number of time windows in this table corresponds to a size of 40 samples.

Phase	# Experiments	# Conditions	# Critical cycles	Total number of cycles	# Time Windows	%
Training	5700	3	(2+5)	119 700	239 400	87,38%
Test	640	3	(4+5)	17 280	34 560	12,61%
Total				136 980	273 960	



(a) Selected cycles around the time of the electrical fault.

(b) PCA projection.

Fig. 4. Data selection and PCA projection of the training data set.

Having said that, finally the data presented to the classifier will be the time windows of 40 samples around their respective electrical faults, where we will use the instantaneous current values mentioned above. This results in a complete data set of 273,960 time windows.

Finally, it should be noted that all these data are normalised between 1 and −1 so as to eliminate dependence on the amplitude of the instantaneous currents, thereby preventing the model from being influenced by the power transformer rating. In fact, the normalisation is carried out in each time window, calculating the maximum and minimum values of each current. Consequently, this calculation is expressed according to the following equation:

$$X_{norm} = 2 * \frac{X - min(X)}{max(X) - min(X)} - 1 \tag{1}$$

Where X corresponds to both the data used in training and in the inference phase. In this same figure, Fig. 4(b), the PCA projection of the training data set is shown for information purposes.

It is necessary to highlight that during this work, the training and testing of the deep learning models were carried out considering only the events associated with internal and external zone short-circuits faults. Events such as power transformer energisation were excluded from this process because, after normalisation, the waveform performance is similar to the obtained during internal fault conditions. Consequently, they require a different treatment approach to avoid potential misoperation of the model under these scenarios.

All of this processing, together with the training and inference tasks (except for the tests carried out in Section 6), was performed on a basic platform based on an Intel(R) Core(TM) i9-9900K CPU @ 3.60 GHz processor with 32 GB of RAM, equipped with an NVIDIA GeForce RTX 2080 SUPER GPU.

Likewise, care has been taken to balance the training database appropriately so that no class predominates over the others (in this case, it would be the 'No Fault' class). Thus, the percentages for each of the three target classes ($K = 3$) are: $\eta = (36.37\%, 31.48\%, 32.15\%)$ for the classes 'No fault', 'Internal fault' and 'External fault' respectively. This shows a multi-majority imbalance value of less than 1.5, demonstrating the balance of the three classes in this first test (Eq. (2), see Ortigosa-Hernández et al. (2017) for further details).

$$\gamma_K \text{is multi-majority} \iff \sum_{i=1}^{k} \mathbb{1}(\mu_i \geq \frac{1}{K}) = 1 \geq \frac{k}{2} = 1,5 \tag{2}$$

On the other hand, performing the test for the case of multi-minority imbalance (Eq. (3)):

$$\gamma_K \text{is multi-minority} \iff \sum_{i=1}^{k} \mathbb{1}(\mu_i < \frac{1}{K}) = 0 > \frac{K}{2} = 1,5 \tag{3}$$

In both cases, both tests show that neither condition is met, demonstrating that the training data set is balanced.

However, in the inference phase, all windows corresponding to the fault injection time shown in Fig. 4(a) are used.

Finally, Section 6 evaluates the performance of the deep learning model implemented on the AI platform, using a hardware-in-the-loop environment under challenging conditions that were not considered during the training process

4. Methodology

The main objective of this work, as specified in Section 1 within the list of contributions, can be divided into two goals to be achieved in the design of the final application:

- Maximise the accuracy of the classification performed in order to be competitive,
- and at the same time minimise the response time.[1]

Regarding the first premise, we have proceeded to study some of the most recent existing classifiers, which have solutions with fairly high accuracy rates. However, we present here a solution that combines two of those existing approaches, TimeInception (Ismail Fawaz et al., 2020) and Gated Tranformer (Liu et al., 2021), with a notable performance in classification accuracy. On the other hand, the response time must be minimal, due to the requirements of the technical application where the potential classification algorithm is to be used. The combination

[1] It is worth clarifying that the term *response time* will be used in this article to refer to the execution time that the deep learning model takes to process a window of time. On the other hand, we will use the term *operating time* to express the time measured in the RTDS, which encompasses more actions than the one corresponding to classification.

of both classifiers and the decisions made in the effective reduction of each model is essential to achieve this second goal.

This will be a multivariate classifier where a temporal dataset of 6 different electrical variables will be handled as inputs, corresponding to the three instantaneous electrical currents on the high voltage winding in the transformer and the three corresponding ones on the low side.

These inputs are intended to be classified, and complementing the information in Section 3.2, into the following three types of fault:

- Internal fault (IF): the transformer must be tripped to avoid further damage.
- External Fault (EF): the fault current pass through the transformer and the protection must remain stable and avoid the disconnection.
- Non-fault status (NF), corresponding to evaluate in steady state conditions different strength of the sources and load levels.

These three types of fault are intended to provide an optimised manoeuvring margin in order to be able to isolate the minimum part of the affected power system. For example, if it is detected that an external fault has occurred, which is therefore external to the transformer itself, it can be decided and manoeuvred so that the transformer remains connected normally so that the resetting manoeuvre can be carried out more easily at a later time.

The following two subsections will explain in more detail both of the models discussed above, which this article draws on to provide a more optimal practical solution. Subsequently, in a third section, we will explain in detail the final configuration of the hybrid model proposed in this paper.

4.1. InceptionTime-based feature extraction

The first model is the one developed and presented in Ismail Fawaz et al. (2020), known by the authors as Inception Time.

This is a logical evolution of the work (Szegedy et al., 2015) where the model called GoogleLeNet is presented, within which there are different modules that are called Inception. These modules are the centrepiece of this model, and consist of a combination of convolutions of different sizes (1×1, 3×3, 5×5, ...) and a max pooling layer, all of them connected in parallel. The idea behind Inception is to allow the network to learn to automatically select the most relevant features at each scale.

GoogleLeNet, and therefore Inception, are designed to work with images, a type of data that does not fit the purpose of our proposal. So, Inception Time is the module that will help us to work with time series, which by nature are one-dimensional signals. Of all the existing versions of Inception, the model we focus on in our work incorporates residual connections in the style of the ResNet (He et al., 2015) networks, as used in the original Inception Time paper. This type of connection provides several advantages such as solving the problem of the vanishing and explosive gradient, reduces the difficulty of training deep networks, and above all, in what we are interested in, improves accuracy and performs an implicit regularisation.

Let us focus on the original Inception Time module. This module incorporates a one-dimensional convolutional layer called B1, as can be seen in Fig. 5. The authors have labelled it a bottleneck, but its main function is to conjugate and extract the different relationships between the different input time series. There is also a second bottleneck, called B2, which incorporates the complete time series, a fact that improves the accuracy of the network in general terms. It should be noted that B2 receives the result of applying the max pool operation to the input time series (in the style proposed in Szegedy et al. (2015)).

In addition, the result of the processing of B1 is filtered with different filter sizes by convolutional layers. Its specific function is to extract the different particularities of the input signals at different scales, taking into account their relationships between them. These three filters, together with the output of B2, are finally concatenated.

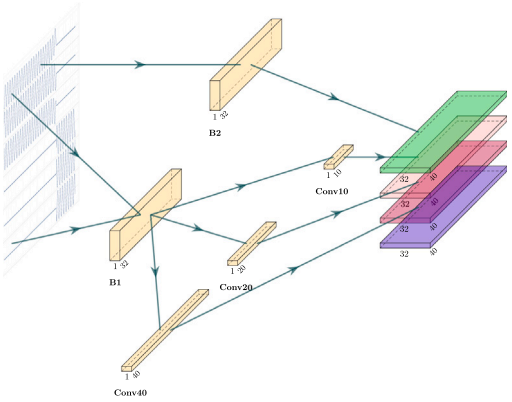


Fig. 5. Architecture of the original Inception module using 32 filters for layers B1 and B2, and filters of 10, 20 and 40 for the convolutional layers.

Table 2
Size of the weights of a single Inception Time module.

Layer	Parameters
B1	192
Conv5	5120
Conv10	10 240
Conv20	20 480
B2	192
	36 224

By way of example, two representations of both the module and the model used in Szegedy et al. (2015) are attached. In particular, in Fig. 5, the module is represented in which a number of filters of 32 (hyperparameter f) and a temporal length of the input data of 40 have been chosen as parameters of the representation.

On the other hand, in Fig. 6, the complete original model, configured with a depth of layers of 6 (we will refer to this hyperparameter as M throughout the rest of the article) and with the same number of filters than before, can be observed. Similarly, the expression 4×32 in this figure indicates that it admits 4 input signals obtained from 32 filters, and I or R indicates the number of the Inception or Residual module.

On another note, and with the aim of improving stability in the training process, the residual connection is configured both in the original work and in Fig. 6 at levels that are multiples of 3.

Finally, the number of weights for each of the Inception modules can be seen in Table 2. From this it can be immediately concluded that the size of the complete model in Fig. 6 is approximately 180,000 parameters. It is worth mentioning that the number of weights corresponding to the two residual connections is negligible (192 for each of them) with respect to the number corresponding to each Inception Time module.

4.2. Gated Decoder for temporal dependency modelling

The second model on which we are going to base the presentation of the final solution is the one corresponding to the work presented in Liu et al. (2021). In this case, its fundamental characteristic lies in the fact that the processing is divided into two 'towers' where in each one of them the input time signals are analysed in two different and independent ways.

The first 'tower' will focus on analysing the time series on the time axis, trying to obtain particular characteristics that can occur in each series over time (Time Channel). In contrast, the second 'tower' focuses on detecting the intrinsic particularities of the relationships between each of the input series (Time Step). This means that this

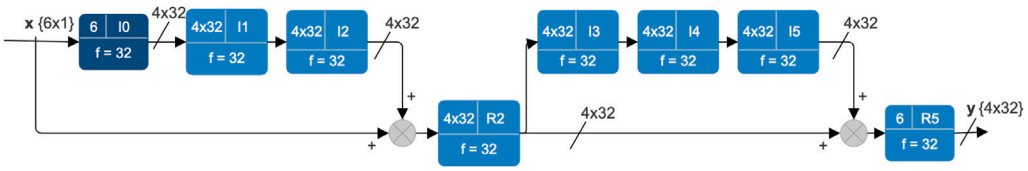


Fig. 6. Complete architecture of the original Inception Time using, as an illustrative example, 32 filters and 6 layers.

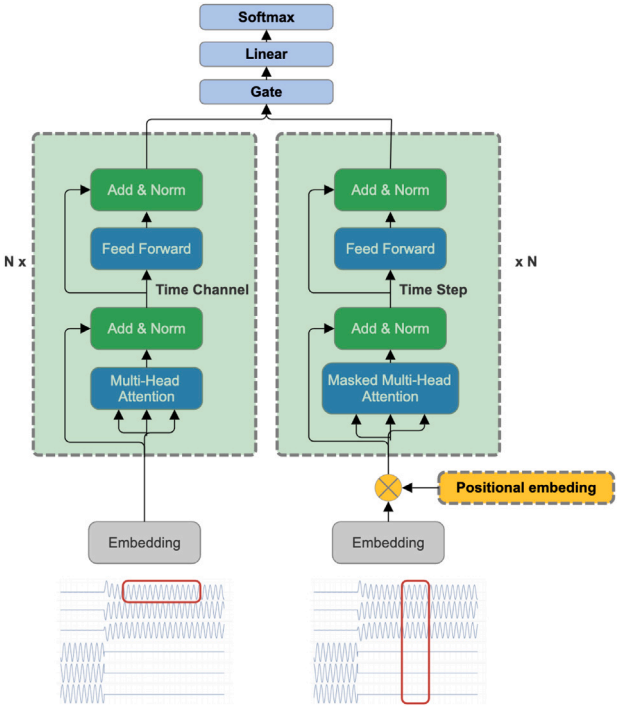


Fig. 7. Architecture of the original Gated Transformer as described in Liu et al. (2021). The two processing towers can be seen.

second network tries to find dependencies between each of the input time series.

In Fig. 7 you can see the model presented in Liu et al. (2021) where the two processing towers mentioned appear. Basically, the two towers differ in two specific stages of processing. On the one hand, there is the Position Embedding required for the Time Step 'tower', whose function is to unequivocally encode each of the input time series. And on the other hand, the use of Masked Multi-Head Attention appears in the same tower, which adds degrees of freedom when presenting the data from each channel in a more varied and information-rich way.

On the other hand, the Time Channel 'tower' becomes a standard decoder, as used in other models based on the Transformer neural network. Except that the embedding performed in both 'towers' is a simple linear layer (this is a method widely used when the input data corresponds to time series).

4.3. Hybrid inception-decoder arquitecture (TransInception)

Finally, in this third subsection, and once both models that have inspired us have been explained in detail, we explain the changes made to both original models and the final proposal carried out in this work.

On the one hand, the proposed hybrid model (see Fig. 8) will focus on the idea of Liu et al. (2021) in which the total processing is compartmentalised into two independent 'towers', one at the channel level and the other on the time axis. However, in our proposal we are going to use a different model for each of the two 'towers'. The Time Channel 'tower' is made up of the modified version of Inception Time, and the Time Step 'tower' is built with the decoder used in the Gated Transformer model. This configuration is only endorsed by the results presented later in Section 5, where the performance is much better in the chosen option.

We will therefore go into detail on the various changes made to each of the 'towers'.

4.3.1. Channel-wise processing: InceptionTime"tower"

In this case, the model chosen is Inception Time. In which we are going to make a strategic change to achieve, we believe, a more optimal functioning.

This change consists of eliminating the Bottleneck layer $B1$, which can be seen in Fig. 5. The reason for this decision is that $B1$ has the purpose of finding dependencies between channels, something that will be done by the other 'tower' in this hybrid model.

This decision greatly simplifies the Inception modules that are to be used, as can be seen in Table 3. Each of these modules will have a little more than 19% of the parameters needed in the original Inception Time module. This means that this particular "tower" has around 30,000 parameters, something that will be necessary when adding the necessary decoder.

The most significant parameters that most directly affect the performance of the modified Inception Time model are:

- Filters (f): Number of filters used in the convolutional layers of Inception Time.
- M: Depth used in the complete structure of the Inception Time model.

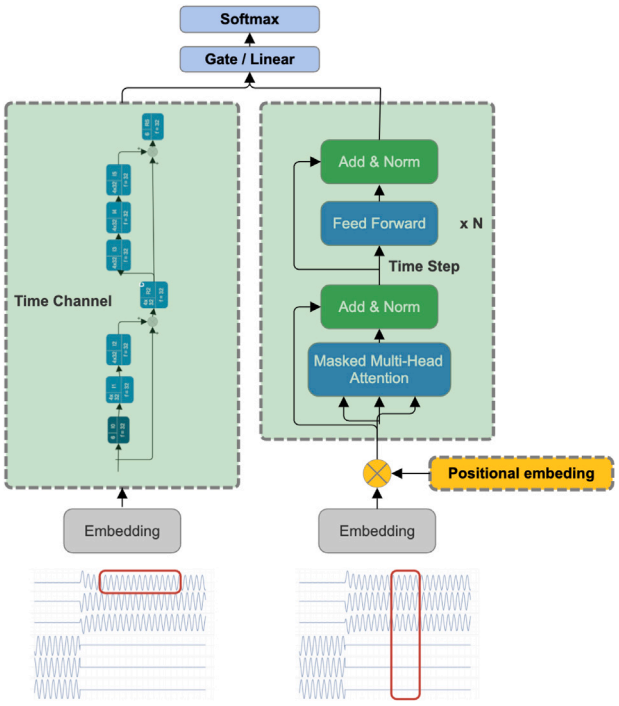


Fig. 8. Final architecture of the hybrid model proposed in this work.

Table 3
Size of the weights of a single Inception Time module without the bottleneck layer B1.

Layer	Parameters
Conv5	960
Conv10	1920
Conv20	3840
B2	192
	6048

4.3.2. Temporal processing: Decoder "tower"

On the other hand, in the 'tower' designed to process the different relationships between the different input channels (instantaneous electrical currents), the decoder model presented in Liu et al. (2021) will be used.

No significant changes will be made to this proposal, except for the part of joining and mixing the results of each processing 'tower'. In this way, two very different options will be used. On the one hand, a gating layer similar to the one used by the authors in the original version of the model will be used. And on the other hand, a simple affine linear network will be incorporated that mixes the results of each 'tower' in a more profound way.

It should be noted that the gating process in the original model performs a weighted sum of each of the towers as a block. That is to say, the result of each "tower" is weighted with a single parameter. On the contrary, with the proposed linear network, each element of the result of both "towers" will be processed by a particular weight for each of them. This means that the latter obtains individual relationships at the level of each element of the result.

However, the main problem with these linear layers will be the increase in the memory required for the total classifier model, where this will depend directly on the output dimensionality of these layers.

Therefore, and by way of summary, the parameters to be considered for the optimisation process of the hybrid model will be:

- d_{model}: Dimensionality used in the Decoder for embedding, multi-head attention and feed forward.
- $Heads(h)$: Number of heads used in multi-head attention.
- N: Number of block sequences in the Decoder.

And on the other hand, the parameters that define the process of joining the results of the two 'towers', and therefore of the hybrid model, are:

- *Linear*: Output dimensionality of the linear layer.
- *Gating*: Weighting method for the result of each "tower".

5. Experimental evaluation

In this section we are going to summarise the results obtained on the basis of the data discussed in Section 3 with the model proposed in 4.3.

The different tests are divided into several sections. But initially we want to justify that the choice of each model in each of the processing 'towers' is the right one. To do this, we will study the results of the two possible configurations once the parameters of each of them have been optimised. These results can be seen in Table 4.

From these results it can be concluded that configuration 1 has clearly better statistics even with a shorter response time. For this reason, the rest of the tests and experiments will be carried out using configuration 1. In fact, in Section 4, both models have already been placed in their corresponding tower following the configuration 1 scheme.

On the other hand, the parameters that are to be optimised due to their notable influence on the results of the classifier are going to be those defined in Table 5, and which have already been highlighted and duly defined in each subsection of 4.3. This table also shows their starting values in this optimisation process. It should be noted that the parameters *Linear* and *Gating* both represent a binary decision. In this sense, the model will have its processing unit based on a gating of both towers or a basic affine linear neural network. The influence and characteristics of this linear network will be studied later in Section 5.2.

Table 4
Numerical results and models responsible for each 'tower' for both possible configurations.

Configuration	Tower		Statistics				Time (msec)
	Time channel	Time step	f1	Recall	Precision	Accuracy	
1	Inception	Decoder	0.9959	1	0.9918	99.02%	3.539
	Time		0.9885	0.9843	0.9926		
			0.9802	0.9762	0.9842		
2	Decoder	Inception	0.9755	1	0.9610	97.72%	5.083
		Time	0.9843	0.9745	0.9842		
			0.9754	0.9746	0.9746		

Table 5
Initial parameter configuration. The results marked in the different tables with the label 100% correspond to this configuration.

Inception		Decoder			Hybrid		General	
Filters (f)	M	d_{model}	Heads (h)	N	Linear	Gating	Overlap	W_{size}
32	6	256	8	4	1024	OFF	20	40

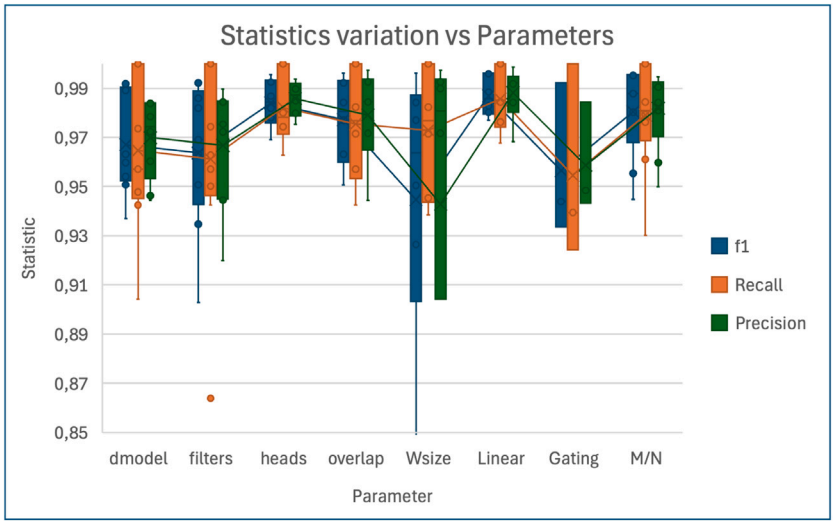


Fig. 9. Influence of the statistics (f1, recall and precision) measured for each of the parameters of interest.

5.1. Analysis of key hyperparameters

Once configuration 1 has been justified, it is important to explain the process of optimising the parameters considered when configuring the two models that make up both towers. To do this, let us start by looking at Fig. 9.

In Fig. 9 we can see the influence that each parameter/decision has on the performance of the proposed hybrid model, this performance being identified by means of the statistical values f1, recall and precision that have been obtained in the different tests carried out.

It is worth highlighting the parameter W_{size}, which corresponds to the size of the window chosen as input to the classifier. We can obtain from more than 99% in both recall and precision to less than 85% in f1. Likewise, the influence of activating gating or using a linear layer on the performance of the classifier can be clearly identified. Later, in Section 5.2, we will go into more detail on this aspect.

As an added summary to what has been said about Fig. 9, it is worth mentioning that the parameters d_{model}, filters or $overlap$ can worsen the performance of the classifier to values of 95% or less, while the parameters $heads$, $linear$, M or N have less influence on the result (with performances above 97%).

Having said that, it is worth looking a little more closely at the influence of these parameters, both in terms of the accuracy of the classifier and the response time required.

If we look at Figs. 10 and 11, they show how each of the parameters mentioned affects both accuracy and response time respectively. Both figures show different experiments with different values of each of the parameters, with the values corresponding to 100% in the figure with the values that appear in Table 5.

It is worth highlighting, for example, the case of the parameter M with a value of 200% ($M = 12$). An accuracy value of 99.26% is obtained, but at the expense of a computational cost of 6.679 ms. And another representative case, in this case for a value of N of 200% ($N = 8$), where a worse performance is obtained, specifically 98.88% at the expense of a response time of 6922 ms.

As for the D_{model} parameter, it should be noted that it has considerable influence, but with values different from the starting one its accuracy drops considerably, increasing the computational cost to 7538 ms. The same happens, in terms of accuracy, with the $Filters$, $Heads$ and $Overlap$ parameters. Accuracy drops considerably, but in the case of these three parameters, the response time is significantly reduced.

From all this, it can be deduced that the final value of the parameters with which the classifier performs best is shown in Table 6.

5.2. Evaluation of output fusion strategies

In the previous sections we have seen that the option of using gating in the classifier can significantly alter the performance of the classifier.

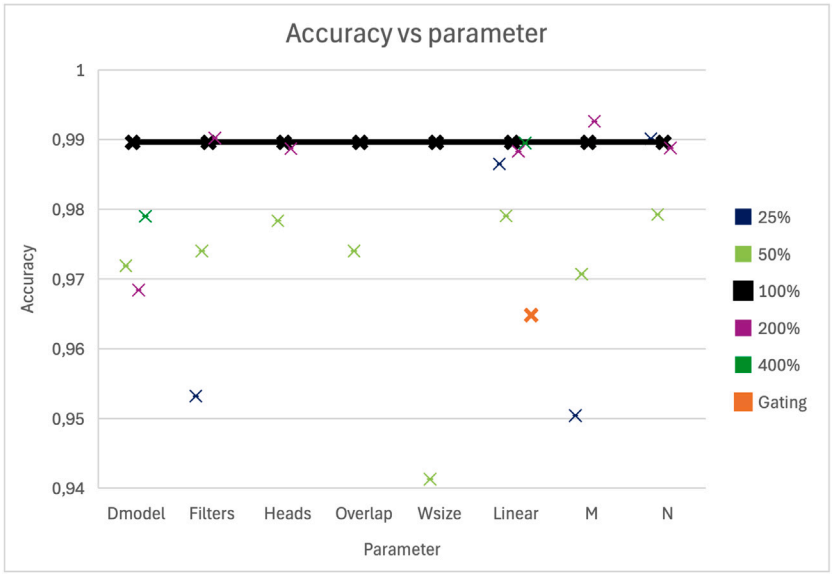


Fig. 10. Measurement of accuracy for different values of each of the parameters of interest. Effects of the gating process are also included.

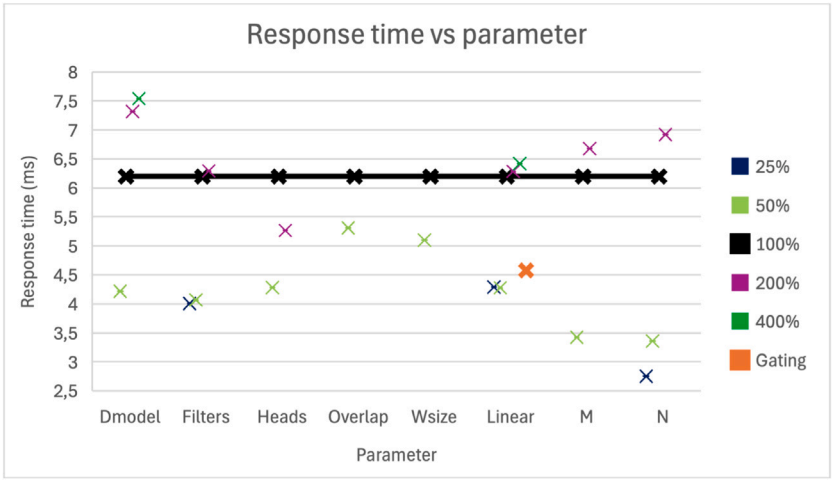


Fig. 11. Response time for different values of each of the parameters of interest. Effects of the gating process are also included.

Table 6
Final configuration of the parameters. The value of 256 corresponding to the *Linear* column has some extra connotation that will be discussed in Section 5.2.

Inception		Transformer			Hybrid		General	
Filters (f)	M	d_{model}	Heads (h)	N	Linear	Gating	Overlap	W_{size}
32	6	256	8	1	256(*)	OFF	20	40

Thus, although the response time with the use of gating is reduced to values around 4.5 ms (Fig. 11), the accuracy values (Fig. 10) fall to 96.48%, and their f1, recall and precision values suffer similar drops (Fig. 9).

This has led us to reconsider the use of gating in the process of joining the two processing towers. And as an alternative, we use a simple double layer of linear transformations where the output information from each tower is received, and an intermediate result is obtained for classification. As can be seen in Fig. 8, the final softmax-type processing, or normalised exponential function, is added to the result of this layer. This will produce an output for each of the target classes bounded between 0 and 1, that is, it will come to represent the probability distribution over the 3 possible output classes.

Figs. 10 and 11 show the numerical results for the use of this double linear layer in terms of classifier performance (specifically accuracy) and response time obtained (see *Linear* column in these tables). The parameter taken as a reference for analysing the performance of this double layer has been the dimensionality of the output of the first linear layer. In the tests, dimensions of 1024, 512 and 256 have been chosen, corresponding respectively to the data series 100

The results offer a curious outcome, where the calculated accuracy specifically for a dimension of 512 drops sharply to 97.9%. In the rest of the cases experimented with in relation to the dimensionality of the linear layer, results close to 99% are maintained, even that corresponding to 256 (the latter with a notable decrease in response time).

Table 7
Comparison on the size of the models obtained with the use of gating or with linear layers of different sizes. Percentage column calculated with respect to the gating option.

Output option	Time channel	Time step	Total	Percentage
Gating	43 008	379 651	422 659	100%
Linear 1024	43 008	7 208 195	7 251 203	1716%
Linear 512	43 008	3 798 787	3 841 795	909%
Linear 256	43 008	2 094 083	2 137 091	506%

One of the problems involved in using linear layers such as those proposed here is the excessive use of memory (see Table 7). A size that can be optimised, obviously, by reducing the dimensionality of the linear layer. Because of this, by adjusting this dimensionality it is possible to obtain a classifier with a higher performance and with less computational cost, but with an adverse effect, a significantly larger size than in the case of using gating.

5.3. Performance assessment of the optimised hybrid model

After the exhaustive study of the different parameters of the model presented, it is time to conclude with an optimal solution for the classifier.

The conclusions that can be drawn from the results shown in Table 8 tell us that dimensionality is a fairly significant adjustment parameter in terms of the size of the model in memory. But adjusting this parameter does not result in a loss in the output statistics that define the performance of the classifier.

The three versions of the proposed model obtained according to the dimensionality of the linear layer output will be called TransInception (1024), TransInception (512), and TransInception (256) respectively.

Finally, it would be interesting to talk about the confusion matrix obtained from one of the three previous models, specifically the TransInception (1024) model, which is represented in Fig. 12. This matrix indicates that there is a minimum percentage of prediction failures. But what is more important is the appearance of these failures between the external fault and internal fault classes. Due to the nature of the application where the classifier is to be used, an electrical power transformer, the confusion between these two classes, although minimal, has malicious effects on the operation of the protection system. This is an unresolved issue in this work, and is a very important aspect for the improvement of the classifier.

5.4. Comparative analysis with state-of-the-art approaches

Once the study of the main parameters that influence the performance of the network in both training and inference has been carried out, it is time to compare the absolute performance of the model proposed in this article with that measured in its most direct competitors.

In this case we have chosen to include in the comparison the two works we have been inspired by, trying to show the improvement in the evaluation statistics calculated. But in addition, we also wanted to include a work from 2023 in which they work with Transformers and time series (Foumani et al., 2024), which has indeed an interesting contribution related with the coding applied to the temporal position of the data characteristics. Both absolute position coding (tAPE) and efficient relative coding (eRPE) are incorporated. In the comparison we test the complete model and algorithm proposed in Foumani et al. (2024) in order to be able to perform the comparison fairly.

The results, therefore, of the comparative study are shown in Table 9, and visually in Fig. 13. It shows the final result of the four models compared. Especially in its graphical representation (Fig. 13), it is clear what we have achieved in our case. Both versions of the TransInception model presented in this article have the best accuracy values by a considerable margin, while their execution times remain among the fastest of the methods tested.

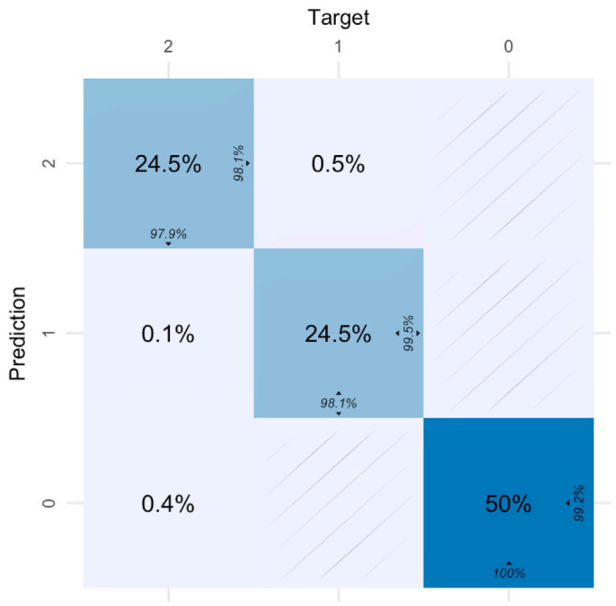


Fig. 12. Confusion matrix corresponding to the TransInception classifier version (1024).

5.5. Cross-dataset comparative study of related works

In the previous section, we sought to compare the model proposed in this article with other well-known models that are considered state-of-the-art. All of these models have been adapted to our database, thanks to the access that the various authors have made available to the scientific community. Therefore, the results are somewhat fair between models, but always taking into account the fact that each model can normally be conditioned by the application it is intended to solve, and therefore by the data set with which each author has worked.

There are other works where it is more complicated or tedious to implement their models, but we believe it is interesting to provide a reference of their results in order to assess and specify the performance of several recent models with data sets that are not identical but are very similar, given that they all work with the aim of identifying internal faults in power transformers.

Most of the works presented in Table 10 share many of the methods, and it should be noted that the numerical range of accuracy is quite wide. In other words, the same method offers very different performance depending on the database used or the pre-processing employed to identify signal characteristics.

This fact leads us to a first conclusion, which is that the performance of a classification method depends on the two factors mentioned in the previous paragraph. And it is this distinguishing feature that stands out most in our proposal: no prior signal analysis process is required to obtain specific characteristics, thus saving processing time that is not spent on the classification process.

On the other hand, the second conclusion drawn from this simple study of methodologies carried out in recent years is that accuracy ranges from a modest 62% to 99.9%. Methods such as Support Vector Machine (SVM) fluctuate from 76.78% (Zou et al., 2023) to 99.7% (Bera et al., 2021), or Gradient Boost from 91.42% (Çuhadaroğlu and Uyaroğlu, 2025) to 99.95% (Bera et al., 2021).

We highlight the work (Bera et al., 2021), which classifies a wide variety of cases related to power transformer faults: internal faults, turn-to-turn faults, winding-to-winding faults, magnetising inrush, sympathetic inrush, external faults with saturation, non-linear load switching, capacitor switching and ferroresonance. From all these experiments, we have focused on the study of internal faults over time windows of 12 samples, obtaining a database of 23,760 + 9504 total samples.

Table 8
Optimal configurations as a function of the dimensionality of the output linear layer. The accuracy and other statistics (f1, Recall and precision) remain very stable in the 3 configurations tested, and only differences in the response time of each of them can be seen.

Linear dimensions	Gate	Accuracy	f1	Recall	Precision	Tproc (msec)
1024	No	**99,02%**	0.99590228	1.	0.991838	**3539**
			0.98849372	0.984375	0.99264706	
			0.98023219	0.97625	0.98424701	
256	No	**98,89%**	0.99678123	1.	0.99358311	**2754**
			0.98296477	0.96770833	0.99870995	
			0.97697946	0.98583333	0.9682832	
512	No	**98,8%**	0.99621232	1.	0.99245322	**3066**
			0.98341033	0.975625	0.99132091	
			0.97594502	0.97625	0.97564022	

Table 9
Table comparing the results between the two best options of the proposal in this article with other models selected as potential competitors.

Model	Year	Accuracy	f1	Recall	Precision	Execution time (msec)
InceptionTime (Ismail Fawaz et al., 2020)	2019	94,31%	0989	1	0978	
			0909	0915	0904	3,273/118,85%
			0889	0864	0915	
TimesNet (Wu et al., 2022a)	2022	96,35%	0994	1	0989	
			0941	0983	0902	10,001/363,14%
			0925	0874	0983	
GTN (Liu et al., 2021)	2023	96,35%	0994	1	0989	
			0941	0983	0902	10,001/363,14%
			0925	0874	0983	
ConvTran (Foumani et al., 2024)	2024	93,39%	0981	1	0963	
			0902	0869	0938	2,829/102,72%
			0873	0875	0871	
TransInception (256)	2025	98,89%	**0,997**	**1**	**0,994**	
			0983	0,968	**0,999**	**2,754/100,00%**
			0977	**0,986**	0968	
TransInception (1024)	2025	**99,02%**	0996	1	0992	
			0,988	**0,984**	0993	**3,539/128,50%**
			0,980	0,976	**0,984**	

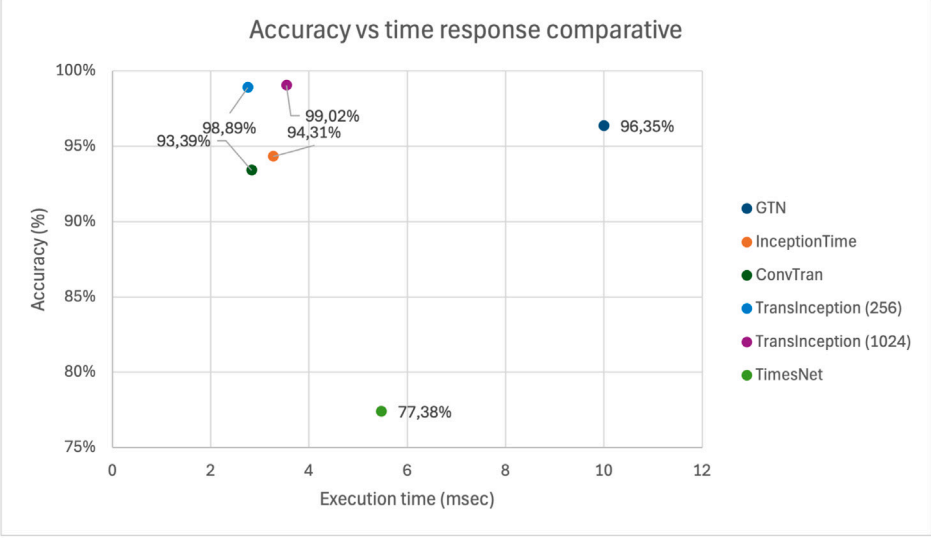


Fig. 13. Graphical representation of the tested methods in terms of accuracy and response time.

This article studies classification in all the cases mentioned above with an average performance well above 99% and a very competitive response time of around 9 ms + 16,67 ms (much of which is spent on the feature extraction phase).

Finally, it should be noted that this is how the progress in the classification task that this article contributes can be appreciated in terms of being among the highest performance rates of similar works within the field of study of power transformers. Likewise, it achieves very competitive computation times with respect to these listed works.

5.6. Analysis of the impact of data error on classifier performance

Up to this point, the classification process has been discussed using only data obtained directly from the RTDS system, which provides a clear and reliable picture of the classifier's performance in a real power transformer. However, an important aspect to consider in this study is the impact of different noise levels and their effect on the classifier's performance.

Table 10
List of works related to the topic of the article with their respective performance in the classification task.

Paper	Methodology	Year	Number of fault types	Training Acc	Testing Acc	System modelling	Simulation time/ Fault inception/ Fault duration	Features selection	Time consuming	Hardware
Pani et al. (2020)	Random Forest (RF) Gradient Boost (GB) Decision Tree (DT) Gradient Boost (GBC-1)	2020	7	93,61% 93,95% 93,65% –	95,10% 95,40% 93,60% 99,95%/98,5%	315 MVA 400 kV/230 kV 400 MVA/500	0,2 s/0.1 s/0,05 s (3 cycles)	Yes	–	i7-6560U CPU @ 2.20 GHz having 8 GB RAM
Bera et al. (2021)	Decision Tree (DT) Support Vector (SVM) Random Forest (RFC)	2020	11	– – –	99,5%/95,3% 99,7%/89,2% 99,9%/94,6%	MVA 230 kV & 500 kV	Fault inception time: 15,3 ms	Yes	25,37 ms	i7-8700 CPU 3.2 GHz 64 GB RAM
Sudha et al. (2022)	Linear discriminant analysis (LDA) Quadratic discriminant analysis (QDA) K-Nearest Neighbour (K-NNA)	2021	25	98% 90% 98,00%	62%	2 KVA and 10 KVA	–	Yes	–	–
Tahir and Tenbohlen (2023)	Transformer condition assessment (TCA)	2023	6	–	95,60%	240 MVA, 400 kV/132 kV	–	Yes	–	–
Zou et al. (2023)	Probabilistic Neural Network (PNN) Super Vector Machine (SVM)	2023	5	67,50% 76,78%	71,43% 82,43%	Dissolved Gas Analysis (Not specified)	– –	Yes	– –	–
Çuhadaroğlu and Uyaroğlu (2025)	Deep Learning (DBN) Logistic Regression (LR) Random Forest (RF) K-Nearest Neighbour (KNN) Support Vector (SVM) Decision Tree (DT) Gradient Boost (GB)	2025	Detection only	81,48% – – – – – –	90% 77,68% 88,68% 69,41% 85,69% 79,61% 91,42%	10 kVA 2 kV/230 V	– – – – – – –	Yes	– – – – – – –	Omicron FRAnalyzer. Klaus, Austria

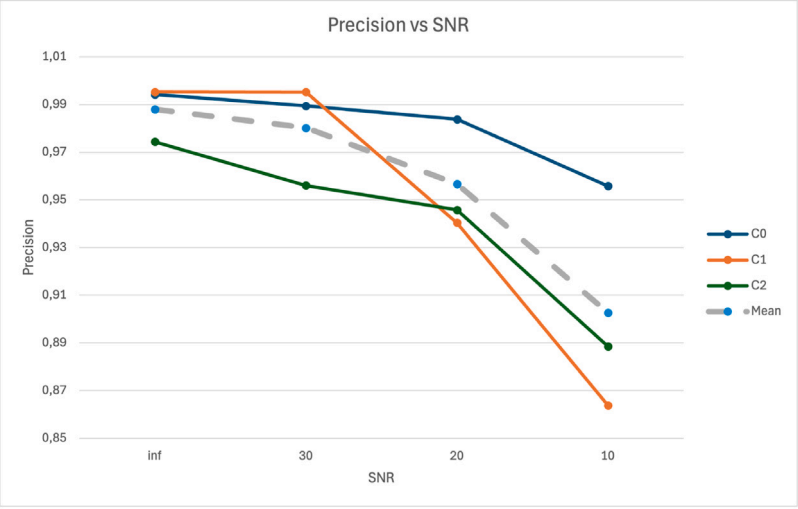


Fig. 14. Measurement of model accuracy based on the SNR introduced on the original data, assuming that noise is present in both the training and inference phases.

That is why in this section we delve into this study, and we do so from two points of view. The first is that noise is present both in the training data and during the inference process; and on the other hand, we will analyse this impact by training with noise-free data, where noise only appears during the inference process.

5.6.1. Noise present in both the training and inference phases

In this subsection, we will retrain our model with different noise levels. Starting with a clean signal, we will then apply signal-to-noise ratio (SNR) values of 30 dB, 20 dB, and 10 dB.

In Fig. 14, as expected, a drop in model accuracy can be seen when the SNR value decreases (increase in the amplitude of the noise added to the original signal). The SNR value of 20 dB indicates, for us, a characteristic point to be considered, and is where the accuracy value falls below 95%.

Fig. 14 also shows that in the 'No Fault' class (C0 in the figure), there is less impact from the increase in noise present. Meanwhile, the other two classes fall to accuracy values below 90%.

5.6.2. Noise present only in the inference phase

On the other hand, we wanted to present the impact of noise on the original signals by training the model with noise-free data and then adding noise in the inference phase.

In this case, Fig. 15, the point of interest remains only at the noise level of 30 dB, which is where the accuracy of the classifier falls below 90% again.

Likewise, stability can be seen in the behaviour of the 'No fault' class (C0) with respect to the noise level present up to a value of 10 dB, falling dramatically to an accuracy value of 0 with a noise level of 5 dB.

It is also worth noting the behaviour of class C1 (Internal Fault), where accuracy improves above 20 dB (unlike at lower noise levels, where logic would suggest an explanation).

6. Robustness assessment under challenging operational conditions

In Section 3, the hardware necessary for data extraction using the RTDS product on a simulated power transformer has been illustrated. This same hardware is going to be used for the implementation of a complete system on the necessary electronic components together with a future real transformer. This system can be seen in the laboratory testbench illustrated in Fig. 16, where RTDS appears as the source of simulation data that represents the electrical element to be protected: in our case, an electrical power transformer. Subsequently, the data is taken to a switch capable of distributing it to different processing,

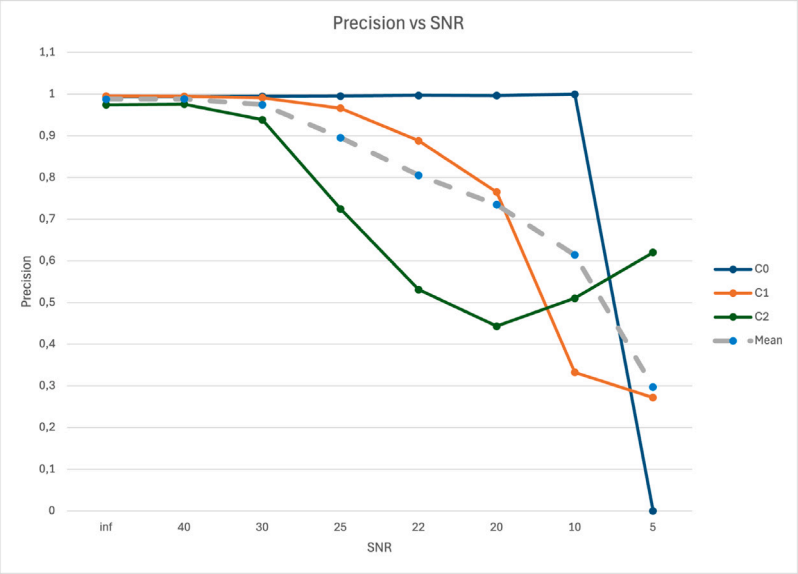


Fig. 15. Measurement of model accuracy based on the SNR introduced on the original data, assuming that noise is present only in the inference phase.

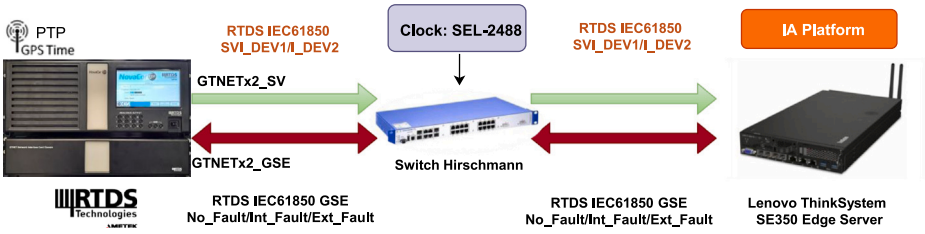


Fig. 16. Laboratory testbench.

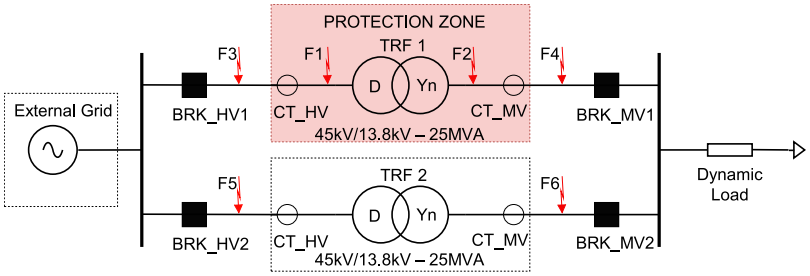


Fig. 17. AI model validation setup in RTDS. Parallel transformer system and measurement points.

representation or management devices. For our tests we will use an Edge Lenovo ThinkSystem SE350 server running Rocky Linux 8.5, equipped with 16 cores (2.2 GHz) and 64 GB of RAM, that will function as an AI platform. It is on this server that we will host our models for the classifier discussed in this article.

This section describes the TransInception model performance, which was previously trained with the data obtained from (Fig. 9), in new operating scenarios. These new scenarios consider internal faults, external faults, and no-fault conditions along with sympathetic inrush and instrument transformer saturation.

Furthermore, the most representative test results obtained during the evaluation process are also included.

6.1. Test system description and experimental scenarios

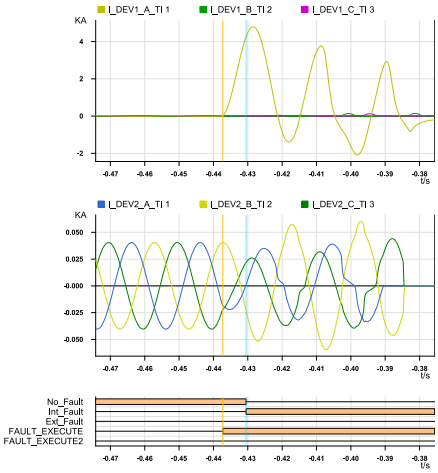
Fig. 17 describes the electrical grid model implemented on the RTDS to evaluate the deep learning model's performance in fault classification process.

This model consists of an external grid, which is represented as a Thevenin equivalent, two power transformers connected in parallel, and a load. The grid model allows for the analysis of the deep learning models behaviour under different operating scenarios, including internal faults (F1 and F2), external faults (F3, F4, F5 and F6), and no-fault conditions, with particular attention to the differentiation of transient events such as sympathetic inrush and instrument transformer saturation.

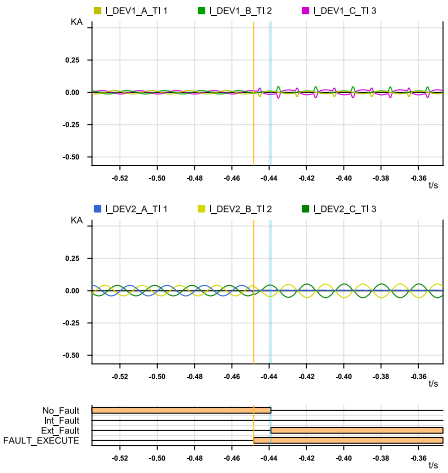
The simulation considers both power transformers operating in parallel and the energisation of one transformer (TRF2) while the other (TRF1) remains in service, enabling an assessment of how the model responds to the interaction between both units. Additionally, coupling effects and their impact on fault detection are also analysed.

To evaluate the deep learning model several fault types were applied:

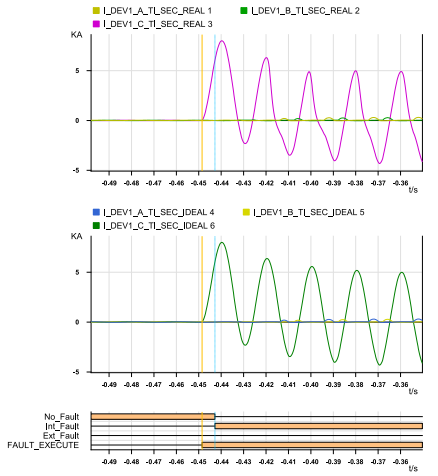
- *Single line to ground faults:* AG, BG and CG.
- *Line to line faults:* AB, BC and CA.



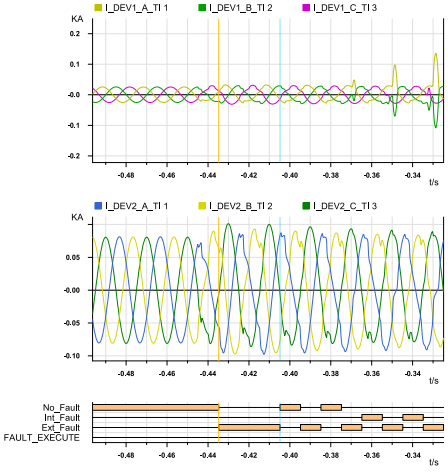
(a) Model response for a single-phase AG fault in F2 with 0 Ω fault impedance.



(b) Model response for an external fault in F3, phase AB, with 0 Ω fault impedance.



(c) Operating time for a CG phase fault in F1 with 0 Ω fault resistance.



(d) Impact of sympathetic inrush on TR1 during the energisation process.

Fig. 18. Model, operating time and sympathetic inrush impact in special examples.

- *Double line to ground faults:* ABG, BCG and CAG
- *Three phase faults:* ABC

Furthermore, different fault resistances were considered during the study: Solid fault 0 Ω, 3 Ω and 10 Ω.

To evaluate the deep learning model performance in the hardware in the loop environment and under these new operating scenarios, it is implemented on a generic hardware platform refer to Fig. 16, which illustrates the test setup. using the deep learning model ".ONNX" (Elena Ranguelova, 2025) obtained during the training process. The model will use the measurements include the phase currents in both transformers and the signals at the common bus.

During tests, the operating time is measured, defined as the duration between the fault occurrence in the simulation and the moment when the deep learning model provides the classification. It should be noticed that this time includes the windowing process of current measurements, scaling and the response time of the deep learning model.

6.2. Results under diverse fault and operational conditions

Internal faults were analysed by simulating short circuits within the protection zone (see Fig. 17), applying different fault impedance conditions. As an example, In Fig. 18(a), one of illustrates the model's responses is observed under a solid single-phase AG fault in F2 initial fault impedance conditions of 0 Ω, executing a single-phase AG fault in F2. As can be shown in the figure, the deep learning modelAI algorithm is able to classify the fault as "Internal Fault", correct classification of the fault is observed with an execution response operating time of 6.8 ms measured from the moment the fault occurs until its classification. Similarly, external faults located outside the protected zone were evaluated. In Fig. 18(b) shows, the model's behaviour when generating a line to n line fault AB is applied exteoutside the protection zone (rnal fault in F3) is observed, considering a with a fault impedance of 0 Ω in phase AB. The correct classification of the external fault is verified with an operating response time of 9.3 ms. The same deep learning model AI performance is observed for all fault types applied in the internal, external protection zone, with an operating time range of 2.5 ms to 10 ms. Furthermore, during no fault conditions, the deep learning AI model performance is stable, indicating that there is not a fault condition in the system.

In the evaluation of IA performance during transient phenomena, scenarios of current transformer saturation were included. From test results it was concluded that the distortion in current measurement

does not significantly affect event classification in internal faults. As an example, Fig. 18(c) shows the classification provided by the deep learning model when a solid CG fault is applied in F1. As is shown in the figure, the model can classify the fault as Internal Fault and with an operating time of 5.8 ms.

Additionally, the behaviour of sympathetic inrush, a phenomenon that occurs when parallel transformers are energised, was analysed. This effect was evaluated with power transformer 1 (TR1) in service while power transformer 2 (TR2) was energised. During tests, it was observed that current transients induced during the energisation process were interpreted as faults by the deep learning model in most of the cases. This occurs because the model has not yet been trained for these classification scenarios. An example of this performance is illustrated in (Fig. 18(d)). As can be observed, the deep learning model initially classifies this event as an external fault, but over time (around 30 ms after the TR2 energisation) and due to the waveform characteristics, the model erroneously classifies it as an internal fault.

7. Conclusions and future work

This work is a first phase with very interesting results, which give rise to expanding the research as initially planned with a greater number of faults to be recognised, providing a greater degree of freedom to the future protection system. The model has proven to meet its objective detecting internal and external faults and distinguish them from no-fault conditions. However, its robustness can be further enhanced by progressively incorporating new operational scenarios and electrical transients, allowing the development of an adaptive and scalable model for real-time transformer protection in interconnected networks. It is worth mentioning that during the data acquisition process we have obtained the necessary data and their more specific classification (Section 3.1) to be able to check the model presented in this article, or to improve or optimise it to meet a more complex classification with a greater number of classes, or as mentioned, more complex scenarios. And on the other hand, our most imminent requirement is the reduction of the response time. A time that is already very small in this version, but due to the application where we want to apply this type of neural model (electrical fault detection), time plays a crucial role.

The tests conducted in the RTDS environment demonstrate that the TransInception model has the capability to accurately differentiate between internal, external and no-faults in power transformers, meeting the criteria initially established. Event classification is achieved within an operating time range of 2.5 ms to 10 ms, enabling its application in advanced protection schemes and as a complement to conventional differential relays, enhancing selectivity and stability in dynamic networks.

The initially stated dual objective of achieving a classifier, with high accuracy together with a reduced response time and feasible for legal requirements, has been achieved. With a model which has obtained a great accuracy of 99,02% with a response time of 3539 ms, even better than the ConvTran model with much better results in accuracy achieved, and with a brilliant result when comparing its response time with its closest competitor in accuracy rate (GTN). Nevertheless, a recognition rate slightly below that of the fault classification work (Bera et al., 2021), but with much lower processing time.

In short, it has been demonstrated with experimental data that a substantial improvement has been achieved with respect to the deep learning models used as inspiration, as well as to a recent model, specifically from the year 2022–2024. And this improvement has been achieved while maintaining a more than reasonable response time, which, speaking in characteristic times of the application where we are working, would reach 19,42% of the period of the electrical network (20 msec) in Europe, i.e. below one quarter of the period.

These results are key for future protection systems as they provide a rapid response. And what we think could be a great possibility is that this speed of execution gives us enough time to carry out subsequent automatic checks to ensure greater security, but also to incorporate other cascade models that can be used in the more detailed classification of the potential active fault.

Furthermore, from tests it is also concluded that the model maintains stability in the presence of current transformer (CT) saturation events, mitigating the risk of incorrect trips in external faults. Its ability to identify distortion induced by CT saturation represents an advantage for improving the reliability of differential protection schemes, allowing better discrimination between normal operating conditions and actual fault events. During the study several limitations were detected. One limitation is related with energisation scenarios where it is suggested the need for specific training with incorporating frequency-domain analysis to improve the discrimination of operational transients.

Another limitation has to do with the appearance in real power transformers of the reclosing and start-up phases after the electrical fault has been corrected or cancelled. In fact, among the most common errors in the present model are erroneous classifications that occur during this start-up process.

Finally, a limitation that is difficult to compensate for is the impossibility of achieving a fault-free protection system, no matter how good the detection and classification system may be. It is therefore necessary to set a minimum safety threshold that allows sufficient confidence in a protection system, even if it is not perfect. But we dare to say that the Deep Learning models can serve as a complement to differential protection schemes, providing an additional validation layer to prevent unwanted trips under complex transient conditions, such as transformer energisation and load exchanges in parallel operation. Its low latency and high accuracy make it viable for applications in digital protection systems.

As future work, this research will be mainly focused on two aspects. The first one concerns different power transformer connections (wye-wye, wye-delta, delta-delta…) and phase shift angles. The second will include different operating scenarios such as capacitor switching, ferroresonance, or non-linear load switching (inspired by Bera et al., 2021). These scenarios are considered to assess the generalisability of the proposed model and to determine whether retraining or fine-tuning processes are required. Nevertheless, the current result is already generalisable to other voltage and power levels due to the scaling method selected during the study. However, this study concludes that the chosen normalisation method also presents limitations, as it does not discriminate the magnitude of faults. Therefore, future work will have to find out and analyse alternative scaling and normalisation methods.

It should also be noted that although the data used in this study were generated with sampling rates commonly used in digital substations, a logical next step is the validation of the proposed model with field measurements in an operational environment. This would allow the assessment of its behaviour under real data availability and quality conditions, complementing the simulation and HIL-based evaluations performed in this work.

CRediT authorship contribution statement

Elías Herrero Jaraba: Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Eduardo Martínez Carrasco:** Validation, Supervision, Conceptualization. **Anibal Antonio Prada Hurtado:** Writing – review & editing, Software, Investigation. **María Teresa Villen Martínez:** Writing – review & editing, Investigation, Data curation. **Guillermo Rios Gómez:** Writing – review & editing, Validation, Software, Investigation, Data curation. **David Hernando Polo:** Investigation, Data curation. **Julio David Buldain Pérez:** Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

Abbasi, A.R., 2022. Fault detection and diagnosis in power transformers: a comprehensive review and classification of publications and methods. Electr. Power Syst. Res. 209, 107990. http://dx.doi.org/10.1016/j.epsr.2022.107990, URL https://linkinghub.elsevier.com/retrieve/pii/S0378779622002176.

Bera, P.K., 2023. Data-driven protection of transformers, phase angle regulators, and transmission lines in interconnected power systems. arXiv preprint arXiv:2302.03826, URL https://arxiv.org/abs/2302.03826.

Bera, P.K., Isik, C., Kumar, V., 2021. Discrimination of internal faults and other transients in an interconnected system with power transformers and phase angle regulators. IEEE Syst. J. 15 (3), 3450–3461. http://dx.doi.org/10.1109/JSYST.2020.3009203.

Commission, I.E., 2011. Part 9-2: Specific Communication Service Mapping (SCSM) - sampled values over ISO/IEC 8802-3. In: Communication Networks and Systems for Power Utility Automation. IEC.

Çuhadaroğlu, H., Uyaroğlu, Y., 2025. Detection of transformer faults: AI-supported machine learning application in sweep frequency response analysis. Energies 18 (10), http://dx.doi.org/10.3390/en18102481, URL https://www.mdpi.com/1996-1073/18/10/2481.

Du, Q., Gu, W., Zhang, L., Huang, S.-L., 2018. Attention-based LSTM-CNNs for time-series classification. In: ACM International Conference on Embedded Networked Sensor Systems.

Elena Ranguelova, 2025. ONNX tutorials. URL https://github.com/onnx/tutorials. Last access: 4 de february de 2025.

Foumani, N.M., Tan, C.W., Webb, G.I., Salehi, M., 2024. Improving position encoding of transformers for multivariate time series classification. Data Min. Knowl. Discov. 38 (1), 22–48.

Gharghory, S.M., 2021. A hybrid model of bidirectional long-short term memory and CNN for multivariate time series classification of remote sensing data. J. Comput. Sci. 17 (9), 789–802.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. arXiv:1512.03385, URL https://arxiv.org/abs/1512.03385.

Ho, K.-H., Huang, P.-S., Wu, I.-C., Wang, F.-J., 2020. Prediction of time series data based on transformer with soft dynamic time wrapping. In: 2020 IEEE International Conference on Consumer Electronics - Taiwan. ICCE-Taiwan, IEEE.

Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.-A., Petitjean, F., 2020. InceptionTime: Finding AlexNet for time series classification. Data Min. Knowl. Discov. 34 (6), 1936–1962. http://dx.doi.org/10.1007/s10618-020-00710-y.

Jiang, H., Liu, L., Lian, C., 2022. Multi-modal fusion transformer for multivariate time series classification. In: 2022 14th International Conference on Advanced Computational Intelligence. ICACI, IEEE.

Kambale, W.V., Kadurha, D.K., El Bahnasawi, M., Al Machot, F., Benarbia, T., Kyamakya, K., 2023. Transformers in time series forecasting: A brief transfer learning performance analysis. In: 2023 27th International Conference on Circuits, Systems, Communications and Computers. CSCC, IEEE.

Li, T., Zhang, Y., Wang, T., 2021. SRPM–CNN: a combined model based on slide relative position matrix and CNN for time series classification.

Liang, R., Yang, L., Wu, S., Li, H., Jiang, C., 2021. A three-stream CNN-LSTM network for automatic modulation classification. In: 2021 13th International Conference on Wireless Communications and Signal Processing. WCSP, IEEE.

Liu, M., Ren, S., Ma, S., Jiao, J., Chen, Y., Wang, Z., Song, W., 2021. Gated transformer networks for multivariate time series classification. CoRR abs/2103.14438.

Miao, L., Luo, G., Liu, X., 2023. Incremental approach for early time series classification. In: 2023 International Symposium on Intelligent Robotics and Systems. ISoIRS, IEEE.

Ortigosa-Hernández, J., Inza, I., Lozano, J.A., 2017. Measuring the class-imbalance extent of multi-class problems. Pattern Recognit. Lett. 98, 32–38. http://dx.doi.org/10.1016/j.patrec.2017.08.002, URL https://www.sciencedirect.com/science/article/pii/S016786551730257X.

Pani, S.R., Bera, P.K., Kumar, V., 2020. Detection and classification of internal faults in power transformers using tree based classifiers. In: 2020 IEEE International Conference on Power Electronics, Drives and Energy Systems. PEDES, IEEE, pp. 1–6. http://dx.doi.org/10.1109/pedes49360.2020.9379641.

Peng, L., Qu, W., Zhao, Y., Wu, Y., 2019. A multi-level network for radio signal modulation classification. In: International Conferences on Artificial Intelligence, Information Processing and Cloud Computing.

Rußwurm, M., Lefèvre, S., Courty, N., Emonet, R., Körner, M., Tavenard, R., 2019. End-to-end learning for early classification of time series. aXiv.org.

Schäfer, P., Leser, U., 2020. TEASER: early and accurate time series classification. Data Min. Knowl. Discov. 34 (5), 1336–1362.

Sidwall, K., Forsyth, P., 2022. A review of recent best practices in the development of real-time power system simulators from a simulator manufacturer's perspective. Energies 15 (3), http://dx.doi.org/10.3390/en15031111, URL https://www.mdpi.com/1996-1073/15/3/1111.

Sudha, B., Praveen, L., Vadde, A., 2022. Classification of faults in distribution transformer using machine learning. Mater. Today: Proc. 58, 616–622. http://dx.doi.org/10.1016/j.matpr.2022.04.514, URL https://www.sciencedirect.com/science/article/pii/S2214785322026359, International Conference on Artificial Intelligence and Energy Systems.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the inception architecture for computer vision. arXiv:1512.00567, URL https://arxiv.org/abs/1512.00567.

Tahir, M., Tenbohlen, S., 2023. Transformer winding fault classification and condition assessment based on random forest using FRA. Energies 16 (9), http://dx.doi.org/10.3390/en16093714, URL https://www.mdpi.com/1996-1073/16/9/3714.

Tong, J., Xie, L., Yang, W., Zhang, K., 2022. Probabilistic decomposition transformer for time series forecasting. arXiv.

Uchiyama, T., 2023. Transformer-based time series classification for the OpenPack challenge 2022. In: 2023 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events. PerCom Workshops, IEEE.

Usmankhujaev, S., Ibrokhimov, B., Baydadaev, S., Kwon, J., 2021. Time series classification with InceptionFCN. Sensors 22 (1), 157.

Vaibhava Lakshmi, R., Radha, S., 2023. Time series classification using attention-based LSTM and CNN. In: 2023 International Conference on Data Science, Agents and Artificial Intelligence.

Wan, S., Chen, T., Ni, X., Xu, C., Wang, R., Wan, Y., 2021. Research on classification algorithm based on multivariate time series. In: 2021 International Conference on Aviation Safety and Information Technology.

Wang, X., Li, Y., Zhang, Q., 2019a. A new identification method of the transformer inrush current based on improved Hilbert-Huang transform algorithm. IEEE Trans. Power Deliv. 34 (4), 1234–1242. http://dx.doi.org/10.1109/TPWRD.2019.2901234.

Wang, J., Wang, W., Wei, S., Zeng, Y., Luo, F., 2019b. Time series sequences classification with inception and LSTM module. In: 2019 IEEE International Conference on Integrated Circuits, Technologies and Applications. ICTA, IEEE.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L., 2023. Transformers in time series: A survey. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization.

Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S., 2022. ETSformer: Exponential smoothing transformers for time-series forecasting. arXiv.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M., 2022a. TimesNet: Temporal 2D-variation modeling for general time series analysis. arXiv abs/2210.02186, arXiv:2210.02186, URL https://api.semanticscholar.org/CorpusID:252715491.

Wu, B., Yao, Z., Tu, Y., Chen, Y., 2022b. A dilated transformer network for time series anomaly detection. In: 2022 IEEE 34th International Conference on Tools with Artificial Intelligence. ICTAI, IEEE.

Xiao, J., Liu, Y., Zhang, B., 2020. Wavelet transform and SVM-based fault detection for power transformers under CT saturation. IEEE Trans. Power Deliv. 35 (3), 1230–1238. http://dx.doi.org/10.1109/TPWRD.2020.2971234.

Xu, G., Sun, W., Xue, H., Feng, X., 2023. Time series classification method based on multi-scale convolution with LSTM. In: IEEE Joint International Information Technology and Artificial Intelligence Conference.

Young, J., Chen, J., Huang, F., Peng, J., 2022. Dateformer: Time-modeling transformer for longer-term series forecasting.

Zhang, Y., Ma, L., Pal, S., Zhang, Y., Coates, M., 2023. Multi-resolution time-series transformer for long-term forecasting. arXiv.

Zhou, H., Yang, Y., Liu, Z., 2020. Deep learning approach for transformer fault diagnosis based on time-frequency representation. IEEE Access 8, 19612–19625. http://dx.doi.org/10.1109/ACCESS.2020.2968654.

Zou, D., Li, Z., Quan, H., Peng, Q., Wang, S., Hong, Z., Dai, W., Zhou, T., Yin, J., 2023. Transformer fault classification for diagnosis based on DGA and deep belief network. Energy Rep. 9, 250–256. http://dx.doi.org/10.1016/j.egyr.2023.09.183, URL https://www.sciencedirect.com/science/article/pii/S2352484723014294, The 8th International Conference on Sustainable and Renewable Energy Engineering.