IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 October 2023, accepted 31 October 2023, date of publication 2 November 2023, date of current version 8 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3330056

RESEARCH ARTICLE

Hyperparameter Optimization of Long Short Term Memory Models for Interpretable Electrical Fault Classification

BIJU G. M., (Member, IEEE), AND G. N. PILLAI, (Member, IEEE)
Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee 247667, India
Corresponding author: Biju G. M. (bijugm87@gmail.com)

ABSTRACT The reliability of the model significantly affects early detection and accurate classification of electrical faults. In this study, a Long Short Term Memory based fault classification model was developed for the Power System Machine Learning benchmark dataset, focusing on improving reliability by increasing interpretability. First, novel metrics are introduced to measure model interpretability. These interpretability metrics are uniquely defined based on the disentanglement of the fault classification factors. Subsequently, hyperparameter optimization was performed using multi-objective Bayesian Optimization to determine the optimal model architecture. The objective of optimization is to maximize interpretability and classification accuracy. The Pareto-optimal solution presents different model architectures with varying accuracy and interpretability trade-offs. Finally, the manifestation of interpretability in terms of subsequences is studied using Shapley Additive Explanations. The impact of class representation and architectural parameters on interpretability was also analyzed. Furthermore, the most accurate model in the Pareto front achieved highly competitive accuracy for the benchmark data.

INDEX TERMS Artificial intelligence, deep learning, electrical fault detection, hyperparameter optimization, interpretability, long short term memory, pareto optimization, power grids, power system reliability, recurrent neural networks, time series analysis.

I. INTRODUCTION

Fault classification is an important aspect of electric grid operation. It is essential to prevent power outages, ensure grid resilience, enhance efficiency, enable predictive maintenance, and facilitate grid modernization. These efforts support the transition towards carbon neutrality by maintaining a stable and reliable electric grid that can effectively accommodate renewable energy sources, reduce greenhouse gas emissions, and support sustainable development in the face of climate change [1]. The sensitive nature of grid operations limits the accessibility of grid data. The need for well-documented and real datasets for grid fault classification is an important limitation of data-driven models for benchmarking their performance [2]. Recent developments in open-source power system datasets have provided opportunities to address several research gaps in this area [2], [3]. The load, renewable, and grid data in such datasets are presented as time-series readings of voltage, current, power, etc. The faults leave unique signatures on the current and voltage time-series values. Fault classification involves classifying the nature of a fault based on time-series electrical readings.

Traditional model-based methods, such as observers or estimators, typically rely on the development of precise mathematical models based on the system parameters. These models were then used to compare the measured values with the output generated by the model to diagnose faults [4], [5]. However, conventional model-based diagnostic techniques often fail to achieve competitive results owing to the increasing complexity of electrical and electronic systems. The rapid evolution of smart sensors and the Internet of Things (IoT) has led to data-driven methods surpassing model-based approaches in terms of their performance. In classical machine learning, these methods often rely on external

The associate editor coordinating the review of this manuscript and approving it for publication was Gerard-Andre Capolino.

feature-extraction processes to facilitate model learning. For instance, in [6], the authors employed techniques such as the Fast Fourier Transform and the ReliefF algorithm to select the most correlated features. These selected features were then used as inputs for models such as the Extreme Learning Machine (ELM) and Random Vector Functional Link (RVFL) for fault classification. Similarly, in [7], Principal Component Analysis was employed to reduce dimensionality before applying a Bayesian model for fault diagnosis.

By contrast, deep learning models can extract features and perform classification, making them highly effective for nonlinear mapping. Deng et al. [8] developed a Convolutional Neural Network (CNN) model based on LeNet-5 for the localization of traveling wave faults. Liang et al. [9] introduced an approach using an Adaptive Convolutional Neural Network (ACNN) to select fault lines within distribution networks. In addition, feature extraction techniques such as the Short-time Fourier Transform (STFT) have been combined with CNN classifiers for fault detection and classification [10], [11]. Hou et al. [12] employed a Conditional Generative Adversarial Network (CGAN) in a fault identification method tailored to distribution networks. Furthermore, Long Short Term Memory (LSTM) was used for detection of High Impedance Fault (HIF) in solar Photovoltaic (PV) integrated power system [13] and for fault detection in a grid-connected Micro-grid (MG) system [14]. Gated Recurrent Unit (GRU) models have been used in fault detection methods for Ultra High Voltage Direct Current (UHVDC) systems [15] and photovoltaic arrays [16]. The fault location in power systems was also identified using a combination of the attention mechanism and bidirectional GRU [17]. Alrifaey et al. [18] utilized Wavelet Packet Transform as a data preprocessing technique in conjunction with a hybrid LSTM and Stacked Autoencoder approach for fault detection and classification in photovoltaic systems. Deep learning methods have also been applied at the device level for data-driven fault diagnosis using streaming Phasor Measurement Unit (PMU) data [19], and multihierarchy embedding matching [20] has been explored for electrical fault diagnosis. Collectively, these studies underscore the growing popularity of deep learning approaches in fault classification.

The lack of interpretability accompanies the success of deep learning models in improving classification performance, as these are black-box models. Several factors necessitating interpretability include adversarial robustness, regulatory compliance, and ethical concerns [21], [22]. When the inference is incomprehensible to users, the credibility of the model becomes questionable for sensitive operations such as the electric grid. For a successful grid operation, the fault classification model must be reliable, and model interpretability is an important aspect of this domain [23].

There are two different approaches for enhancing the interpretability of deep learning models. One is to use post-hoc tools or architectural enhancements to improve model interpretability. These tools are model-agnostic methods such as Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP) [24], [25]. In [26], Integrated Gradients (IG) were used to interpret the feature importance of interruptions in a power distribution network. In [27], LIME was used for transient stability evaluation in power grids. Some models, such as attention models, have built-in mechanisms that aid post-hoc interpretability, and such structural features have been incorporated for fan fault diagnosis [28] and electricity load forecasting [29]. In [30], the authors augmented a base linear regression learner with recurrent neural networks (RNNs) to improve interpretability for time-series forecasting. Using class activation maps for power converter fault diagnosis [23], [31], decoupling position embedding units [32], time-domain attention [33], and self-organizing maps for encoding power curves [34] have improved the interpretability of fault diagnosis. However, the interpretability approach explored in these studies has significant drawbacks. These methods act as a tool that can be used to interpret the model inference. However, the resulting explanation might not be meaningful. Identifying the most impactful input regions or features may not reveal the domain-related patterns identified by the domain experts. For example, consider two explanations - '*all 40 time-steps are equally important,*' and '*first 10 time-steps contribute 50%, the remaining 30 time-steps contribute 50%*'. Both explanations are equivalent to interpretability tools (attention mechanism, CNN heatmaps, SHAP, LIME, etc.). However, the latter explanation is more interpretable because it highlights the most significant input subsequence.

The second approach focuses on aligning model inferences with concepts and patterns identifiable by human experts. This approach explicitly addresses the drawbacks of previous approaches. The focus of this approach, other than interpreting the importance of features for inference, is to find patterns in the time domain, such as time-series shapelets, patches, and instances [24]. For practical applications, the identification of such subsequences enhances interpretability. The key properties of this aspect of interpretability are differentiated contributions, enhanced explanations in terms of shapelets or subsequences, and expert alignment. In this study, interpretability refers to this approach. Neural basis expansion analysis for interpretable time-series (N-BEATS) enhances this interpretability approach, but only for forecasting problems [35]. Reconstruction loss was also used for interpretability enhancement, which is data-intensive and difficult to converge [36], [37]. An interpretability metric was proposed in [38] to measure the second interpretability approach for image classification.

In this study, we improved the model interpretability for electrical fault diagnosis and avoided the problems of the abovementioned methods (such as the data intensiveness of representation learning, unknown accuracy-interpretability trade-off, and shapelet identification). Specifically, we made

the LSTM model more interpretable for electrical fault classification. LSTM is an intuitively optimal model for time-series data because of factors such as maintaining causality and variable input length, which are unique properties of the time-series data. To this end, novel interpretability metrics were defined for the LSTM model. These metrics measure the ability of the model to identify relevant time-series patterns, are model-agnostic, and can be extended to any model such as GRU, CNN, and artificial neural networks. Our objective was to identify models that scored highly in terms of classification accuracy and interpretability. We performed hyperparameter optimization using Bayesian Optimization (BO) to find the Pareto front of models for the accuracy-interpretability trade-off. We analyzed these models and identified the factors that make them more interpretable. To the best of our knowledge, this is the first work that improves the interpretability of LSTM (or any RNN) models without making architectural enhancements, such as attention layers. The main contributions of this study are as follows.

1) Novel metrics have been proposed to measure the interpretability of the LSTM models. These metrics are agnostic to the disentangling factors of the fault signals.
2) The architectural properties of the LSTM models that contributed to maximizing the fault classification interpretability and accuracy were identified.
3) The manifestation of interpretability metrics in factors such as class similarity, dissimilarity, and subsequence length has been studied.
4) The signature subsequences of the different types of faults that affected the model inference were identified.

The remainder of this paper is organized as follows. The formal definition of Bayesian Optimization is presented in Section II. Section III outlines the fundamental operation of the Long Short-Term Memory network, and Section IV offers a definition of the introspectability metric and related background concepts. Section V provides an overview of related works. Section VI presents novel interpretability metrics, associated algorithms, and model searching methods. Section VII provides details of the experiments conducted. Section VIII presents the results and analysis along with the post-result analysis methods, and Section IX concludes the paper.

II. BAYESIAN OPTIMIZATION

Balancing the pursuit of high accuracy with the imperative of robust interpretability in model selection constitutes a multifaceted optimization endeavor involving two primary goals: accuracy and interpretability. To effectively address this challenge, we employ Bayesian Optimization (BO), a resource-efficient technique specifically designed to optimize these computationally expensive black-box objectives [39]. BO capitalizes on the integration of prior assumptions regarding the black-box function and progressively refines these priors by assimilating data samples collected from the function itself. This iterative procedure aims to construct a posterior distribution that delivers a more precise approximation of the function. In our study, the input to the objective function is the model hyperparameter choices sampled from the search space, and the output is an accuracy or interpretability score. Central to this framework is the surrogate model responsible for approximating the objective function. BO leverages a statistical surrogate model to accurately represent the objectives. Moreover, BO relies on an acquisition function that steers the selection of sampling points towards regions with a higher potential for improvement compared to the current best observation. The choice of the next query point is influenced by the optimization of this acquisition function, which effectively balances the trade-off between exploration and exploitation.

Let $f_1, \ldots, f_m$ be the m objective functions to maximize. The performance space is then m-dimensional. In such a multi-objective optimization problem, a singular optimal solution is typically elusive. Instead, the objective is to discern the collection of optimal solutions, wherein enhancing one objective invariably comes at the expense of another [40]. This collection of solutions is called Pareto set, P_s. A solution, $x^* \in P_s$, is Pareto-optimal if there is no other point x in the search space such that $f_i(x^*) \leq f_i(x)$ for all i and $f_i(x^*) < f_i(x)$ for at least one i. The quality of the Pareto front, P_f, associated with the Pareto set P_s, was measured using the hypervolume $\mathcal{H}(P_f)$.

$$\mathcal{H}(P_f) = \int_{\mathbb{R}^m} \mathbb{1}_{H(P_f)}(z)dz, \tag{1}$$

where $H(P_f) = \{z \in Z \mid \exists 1 \leq i \leq |P_f| : r \preceq z \preceq P_f(i)\}$, and $\mathbb{1}_{H(P_f)}$ is a Dirac delta function which has value 1 when $z \in H(P_f)$ and 0 otherwise. r is the reference point, $\preceq$ is the objective dominance operator, and $P_f(i)$ is the i^{th} solution in P_f. The hypervolume improvement (HVI) in each BO iteration is an indication of how close the current estimate is to the true Pareto front.

$$\text{HVI}(P, P_f) = \mathcal{H}(P_f \cup P) - \mathcal{H}(P_f). \tag{2}$$

where P is a set of new points added in the current BO iteration to the previous estimate of Pareto front P_f.

With access to the Pareto set, decision-makers gain the flexibility to make choices based on their specific preferences, striking a balance between competing objectives. In this study, we opt for a Gaussian Process (GP) as the surrogate model for each of the objectives, given its capacity to achieve competitive modeling performance even with limited query data. The specific acquisition function employed was the Noisy Expected HyperVolume Improvement (NEHVI).

III. LONG SHORT TERM MEMORY

Long Short Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to model sequential data by capturing long-range dependencies and mitigating the vanishing gradient problem affecting traditional RNNs [41]. An LSTM cell is the fundamental building block of an LSTM network and is equipped with various gates to control the flow of information.

The LSTM maintains a cell state (**C**) that runs along the entire sequence, allowing it to capture long-term dependencies. The update to the cell state is governed by three gates: the forget gate (**f**), the input gate (**i**), and the output gate (**o**). The forget gate determines what information from the cell state (C_t) should be discarded or kept. The input gate determines new information that should be stored in the cell state. The candidate cell state ($\tilde{C}_t$) is new information to be added to the cell state. The new cell state is a combination of what was maintained by the forget gate, what was added by the input gate, and the candidate cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{3}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{4}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{5}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{6}$$

where W and b are the corresponding weight matrix and bias term, h_{t-1} is the hidden state at the previous time step, x_t is the input at the current time step, σ is the logistic function, and $\odot$ is element-wise multiplication operator. The output gate determines what the next hidden state should be, based on the cell state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{7}$$
$$h_t = o_t \odot \tanh(C_t). \tag{8}$$

The mechanism expressed in these equations allows LSTM to capture and learn dependencies in sequential data, making it suitable for tasks such as natural language processing, speech recognition, and time-series analysis.

IV. INTERPRETABILITY MODELS AND METRICS

In this section, the introspectability metric and other crucial theoretical concepts are defined, providing essential groundwork for subsequent sections.

A. INTROSPECTABILITY

Introspectability is a metric used to quantify the interpretability of a model in terms of the degree of disentanglement between class representations within a neural network $\mathcal{M}$ [38]. Let $\mathcal{X}^{(c)}$ denote the validation data belonging to class c, and $\Phi^{(c,l)}$ denote the activations of layer l reshaped to a single dimension. The activations of all layers are concatenated to obtain $\Phi^{(c)}$. The mean class activations for class c are

$$\bar{\Phi}^{(c)} = \frac{1}{N^{(c)}} \sum_{i=1}^{N^{(c)}} \Phi^{(c)}, \tag{9}$$

where $N^{(c)}$ denotes the number of validation data instances for class c. The introspectability of model $\mathcal{M}$ is then defined as

$$\text{Introspectability}(\mathcal{M}, \mathcal{X}) = \frac{1}{\binom{N_C}{2}} \sum_{c=1}^{N_C} \sum_{k=c+1}^{N_C} D(\bar{\Phi}^{(c)}, \bar{\Phi}^{(k)}), \tag{10}$$

where $D(\cdot, \cdot)$ is the function for obtaining the cosine distance and N_C is the total number of classes in the data. In our study, we calculated introspectability score only for the encodings.

B. SUBSEQUENCE

Interpreting deep learning models with time-series subsequence identification is essential for applications such as anomaly detection, medical diagnostics, and fraud detection. Explanations rooted in subsequences pinpoint specific segments of a time-series that contribute to the model's classification decisions [24]. In the context of a time-series denoted as $x = \{t_1, ..., t_m\}$, a subsequence $s = \{t_i, ..., t_{i+l-1}\}$ with length of l represents an ordered sequence of values. It is characterized by the condition $1 \leq i \leq m - l + 1$, ensuring that the subsequence lies within the bounds of the original time-series. This interpretability approach provides valuable insights into the functioning of deep learning models on time-series data, thereby enhancing transparency and trust in their outputs.

C. SHAPLEY ADDITIVE EXPLANATIONS

LSTM is not an interpretable model; therefore, we used Shapley Additive Explanations (SHAP) to identify the subsequences. SHAP is based on cooperative game theory and provides a method to fairly distribute the "explanatory power" of each feature across different combinations of features [42].

For a given feature i and a prediction $f(x)$, the SHAP value (ϕ_i) for feature i is computed as the average of the marginal contributions that feature i makes to all possible feature subsets.

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[f(S \cup \{i\}) - f(S) \right], \tag{11}$$

where N represents the set of all features, S is a subset of N excluding feature i, $|S|$ is the number of features in subset S, and $|N|$ is the total number of features.

This formula captures the concept of evaluating the model's prediction for all possible subsets of features, including and excluding the feature of interest. It computes the average of the differences in the model predictions for these subsets, accounting for all possible combinations.

SHAP values are particularly useful for explaining the importance of individual features in the context of a specific prediction, thus providing insight into the model's decision-making process.

V. RELATED WORK

To the best of our knowledge, only a few studies have enhanced the interpretability of electrical time-series data. The neural basis expansion analysis for interpretable time-series (N-BEATS) employs a set of basis functions that can be learned to capture different patterns for time-series forecasting [35]. These basis functions are akin to

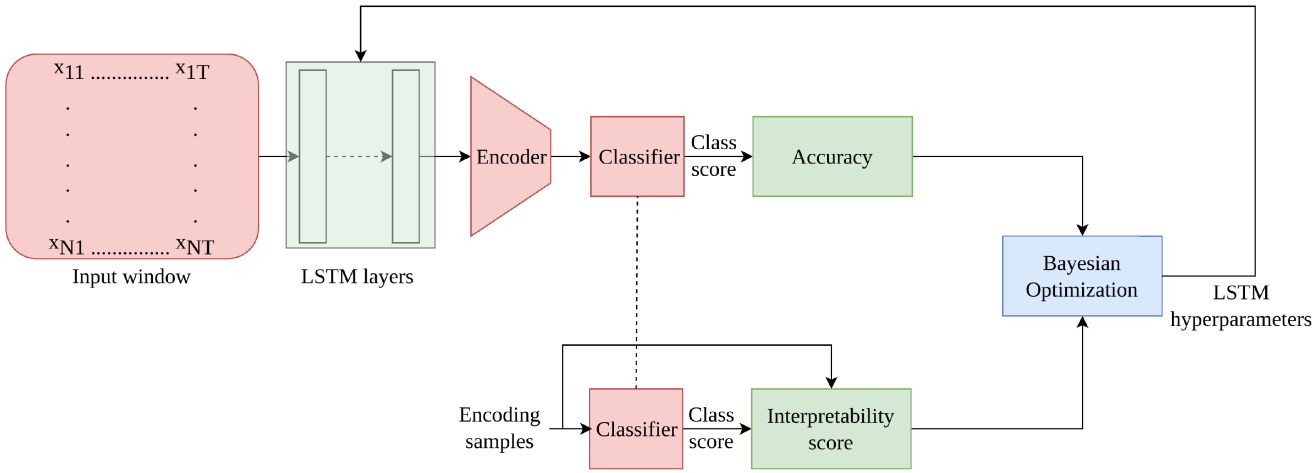

FIGURE 1. General model architecture and objectives for hyperparameter optimization. The accuracy is obtained after training the LSTM-encoder-classifier model, and the interpretability score is obtained using this trained classifier. Bayesian Optimization uses validation accuracy and interpretability score to sample LSTM architecture hyperparameters from the search space.

interpretable components such as trends and seasonality. This methodology considerably enhances interpretability by enhancing explanations of domain-related patterns. However, this method was specifically defined for forecasting. Interpretable fault diagnosis requires the identification of key shapelets or subsequences of the electrical signals. The N-BEATS was not designed for this purpose. In [37], the authors identified the representative shapelets of a time-series signal using disentangled representation learning. Such representative shapelets capture key patterns for time-series classification and aid in interpreting inferences. However, representation learning methods are data-intensive and highly sensitive to hyperparameters. Furthermore, these representations do not focus on the target task. Some studies on image-classification tasks have addressed the second approach of interpretability. In [36], the reconstruction loss function was augmented with an interpretability aware loss term to train an autoencoder-based model for anomaly detection. This forces the latent representation to learn the attention maps. However, the trade-off between these two loss terms is unknown, and a regularization coefficient is empirically defined to represent the trade-off. A neural architecture search was also utilized to optimize the network architecture for accuracy and interpretability by disentangling the class representations [38]. A new interpretability metric called introspectability was defined for this purpose. The introspectability results were successful in obtaining classwise disentanglement. However, the transferability of introspectability to time-series problems has not yet been tested. In summary, there is a lack of electrical fault diagnosis interpretability in terms of shapelets and other fault-related patterns aligned with the domain and no mechanism to achieve this. The methods available in the aforementioned studies are not transferable to electrical fault diagnosis. In addition, there is a lack of understanding regarding the trade-off between the fault classification accuracy and interpretability.

VI. PROPOSED METHODOLOGY

This section details the concepts and formulations of novel interpretability metrics and methods used to obtain interpretable models. In the proposed method, the hyperparameters that affect both accuracy and interpretability are the model architecture. These are the number of layers and hidden dimensions. Bayesian Optimization (BO) was used to determine the optimal hyperparameters. Because there are two objectives to maximize, that is, accuracy and interpretability (see Fig. 1), we obtain a series of models that form the Pareto-optimal solution.

A. INTERPRETABILITY

Identifying subsequences or shapelets in the time window is important for interpreting time-series model inferences [24]. This is particularly significant for fault classification because of the unique signatures of the different fault types. Interpreting the inference through this lens helps improve the reliability of the model. It has been shown that disentangled encoding leads to factorization of a time-series sequence into subsequences [37], but this is a time-consuming task. The approach presented in our study differs from representation learning for reconstructions. Without training the models for representation learning, our study used surrogate interpretability metrics to identify models that learn the disentangled encoding of the input time-series.

The general architecture of the model is shown in Fig. 1. The input window has all features of the input signal for a time sequence of length T. The LSTM layers process this input sequentially and recursively along the time dimension, thereby producing a single-instance output of the hidden dimension size. The encoder is a dense layer with a fixed

output dimension; therefore, all the encoded representations of the models belong to the same encoding space. A linear softmax classifier was used to obtain the class scores.

For a more interpretable model, the encoding representation should be disentangled with respect to the representative factors [37]. Many metrics are available to measure disentanglement, but they are defined by considering known factors in representation learning [43]. However, these factors are unknown for fault classification. Hence, we used the classifier of the model as the arbitrator of the relevant factors rather than explicit factors. The classifier of any model implicitly identifies the deciding factors for the different classes. These factors are not directly accessible from either the encoding or class scores (encoding traversals for representation learning are used to visualize the factors; this is outside the scope of this study). Regardless, we have the advantage that the factors identified by the classifier are not representative, but class-identifying. We call these factors Class Identity Factors (CIFs). When we seek the disentanglement of the encoding representation through the classifier, we implicitly seek the disentanglement of class-specific CIF. This makes the model more interpretable, specifically for fault-classification tasks. The new interpretability metrics are explained in the following subsections.

1) CIF-ENCODING (CIF-e)

CIF-e measures CIF disentanglement by measuring how each encoding dimension impacts CIF. For a disentangled CIF, the encoded representation dimensions should be modular and compact [43]. CIF-e was measured using a trained model, and only the trained classifier layer was used. The fault classification factors are those learned by the model classifier. Each factor that the classifier learns corresponds to the determining factor for a particular class. CIF-e measures the degree to which a factor occupies the encoding space. For interpretability, this subspace should be compact for all the factors. The CIF-e score measures the average size of the subspace for all the factors. This was measured as the average number of encoding dimensions that influence a single CIF. CIF variance was measured using class scores. For this purpose, all encoding dimensions were kept as noise/background, and only one dimension was maintained near the average class value. If the class score varies significantly (from the noise score), then that dimension contains part of the CIF representation. In short, an encoding dimension contains a class-relevant factor if it can pull a noisy data point into a class representation space by varying the corresponding dimension value alone. The fewer the number of dimensions that contain a class-relevant factor, the more disentangled is the CIF representation.

Fault datasets are typically small and cannot be used to conduct statistically consistent CIF studies. Therefore, we generated random encodings based on the distribution of dataset encodings. Let $\mathbf{e} = \{e_1, e_2, \ldots, e_D\}$ denote the encoding obtained at the output of the encoder for an input time-window, where D is the dimension of the encoding space and e_i is the encoding element at the i$^{\text{th}}$ dimension. When an encoding is sampled from the encoding space, it is denoted as e_{cj} such that the j$^{\text{th}}$ dimension is sampled from the cluster distribution of class $c \in C$. All other dimensions are sampled from the entire data distribution, forming the noise or background for a larger sample size N. A collection of N samples is denoted as E_{cjN}, and the classifier block of the model is denoted as $M_{classifier}$.

Algorithm 1 CIF-Encoding

Input: $M_{classifier}$, Data
Output: CIF-e score

for $c \leftarrow 1$ to $|C|$ **do**
  initialize $dimension\_count = 0$
  for $d \leftarrow 1$ to D **do**
    generate samples E_{cdN}
    if $c = \underset{k \in C}{\text{argmax}} \sum_{n \in N} [M_{classifier}(E_{cdn}) = k]$ **then**
      $dimension\_count = dimension\_count + 1$
    end if
  end for
  if $dimension\_count = 0$ **then**
    $class\_dimension\_count_c = D$
  else
    $class\_dimension\_count_c = dimension\_count$
  end if
end for
$mean\_class\_dimension = \frac{1}{C} \sum_{c \in C} class\_dimension\_count_c$
CIF-e score = $1/mean\_class\_dimension$

Algorithm 1 explains how the CIF-e score was obtained from these samples and the trained classifier. When the classifier assigns a higher score to a particular class, it signifies the recognition of at least one corresponding CIF. The dimensions associated with the CIF were identified using the algorithm. A non-zero *dimension_count* indicates the presence of a CIF, which plays a dominant role in one or more dimensions. When the *dimension_count* is one, it suggests that the encodings have a disentangled representation of the CIF. When *dimension_count* is greater than one, it signifies that these non-zero dimensions contain a CIF with or without duplicated dimensions. This situation can also arise when multiple dimensions are necessary to represent the CIF. A lower number of dimensions per CIF is preferable, making *dimension_count* a measure of the extent of the required penalization. Maximum *dimension_count* represents the scenario with the highest level of dimension duplication and entanglement. However, when *dimension_count* is zero, it does not imply the absence of CIF because the classifier identifies the CIF for the corresponding class. Instead, it indicates that the CIF is distributed across all or some encoding dimensions, with none of the dimensions distinctly standing out compared to the others. This represents the

Algorithm 2 CIF-Class

Input: $M_{classifier}$, Data
Output: CIF-c score

generate samples E_N
for $c \leftarrow 1$ to $|C|$ **do**
  get $E_c = \{e_k : M_{classifier}(e_k) = c\}$
  initialize $dimension\_count = 0$
  for $d \leftarrow 1$ to D **do**
    if $var(E_c^d) < \alpha \times var(E_N^d)$ **then**
      $dimension\_count = dimension\_count + 1$
    end if
  end for
  if $dimension\_count = 0$ **then**
    $class\_dimension\_count_c = D$
  else
    $class\_dimension\_count_c = dimension\_count$
  end if
end for
$mean\_class\_dimension = \frac{1}{C} \sum_{c \in C} class\_dimension\_count_c$
CIF-c score = $1/mean\_class\_dimension$

worst-case scenario, representing maximum entanglement and dilution. To address duplication and dilution effectively, we assigned a maximum *dimension_count* of D in such cases. CIF-e is designed to remain in the range of [0,1], with a higher score indicating greater interpretability.

2) CIF-CLASS (CIF-c)

The CIF-c is an alternative method for measuring the disentanglement of CIF. CIF-e looked at each encoding dimension individually to determine if it influenced CIF. However, in CIF-c, we consider the encoding dimensions of only one class-subspace at a time. First, we collected randomly generated encodings based on the data distribution. We then isolated the group of encodings identified by the classifier as a particular class. Within this group, the CIF relevant to that class was dominant. We identified the dimensions with the least variability for the group. These define the dominant CIF. The dimensions with higher variability were irrelevant. Fewer dimensions with lower variability indicate modularity of the class factors and, hence, a disentangled representation. Algorithm 2 explains the steps involved in determining the CIF-c score, where E_N is the collection of N samples generated from the entire data distribution, E_c is the subset of E_N whose class is $c \in C$, $var(E_N^d)$is the variance of the d^{th} dimension of E_N. α defines the threshold, which determines whether the distribution has low or high variability. Just like CIF-e, in this context, we also mitigate maximum entanglement and dilution concerns by designating a maximum *dimension_count* of D in these scenarios. The CIF-c is designed to remain in the range [0,1], and a higher score indicates greater interpretability.

These two metrics are defined using a classifier-based CIF, and constitute the main proposition. In addition, we define three more complementary metrics, independent of the model classifier. These are defined using only certain properties of encoding itself. Hence, the factors, on which these metrics are focused, are abstract and self-contained. Of the three complementary metrics (elaborated in subsequent sections), the first two (KL-introspectability and KL-compact) are grounded in the notion of introspectability, which is defined in Section IV. We adjusted the similarity measure and incorporated additional components that represent compactness. The final metric offers a rough approximation of the disentanglement based on the covariance among the encodings, serving as a fundamental benchmark for comparison.

3) KL-INTROSPECTABILITY

Kullback-Leibler-introspectability (KL-introspectability) is a variant of introspectability, defined in [38]. Introspectability measures the dissimilarity of disparate classes as the cosine distance between all the latent representations within the model. In our study, the cosine distance between classes was counter-intuitive and did not consider the distribution of classes. KL-introspectability measures the degree of dissimilarity between the encodings of different classes as the Kullback-Leibler (KL) divergence between class clusters. Let $\mathbf{e_i}$ denote the encoding of i^{th} data point in the validation data, and y_i be the corresponding output of the model. The encoding cluster of class c is represented by $E_c = \{\mathbf{e}_i \in \mathbb{R}^D : y_i = c\}$ over the validation dataset, where D is the encoding dimension. The dissimilarity between the two class distributions is measured using a symmetrical variant of KL divergence, as shown in (13).

$$\mathrm{KL}(\hat{E}_c||\hat{E}_{c'}) = \sum_x \hat{E}_c(x) \log \frac{\hat{E}_c(x)}{\hat{E}_{c'}(x)}, \tag{12}$$

$$\mathrm{KL}_{\mathrm{sym}}(\hat{E}_c||\hat{E}_{c'}) = \min(\mathrm{KL}(\hat{E}_c||\hat{E}_{c'}), \mathrm{KL}(\hat{E}_{c'}||\hat{E}_c)), \tag{13}$$

where $\hat{E}_c$ denotes the distribution that corresponds to E_c. KL-introspectability was calculated as the mean inter-class dissimilarity between the encoding clusters of the validation set.

$$\text{KL-introspectability} = \frac{1}{\binom{|C|}{2}} \sum_{c=1}^{|C|} \sum_{c'=c+1}^{|C|} \mathrm{KL}_{\mathrm{sym}}(\hat{E}_c||\hat{E}_{c'}). \tag{14}$$

Because dissimilarity is the desired result, a higher KL-introspectability indicates a more interpretable model. For better visualization, log(1 + KL-introspectability) was used for the experiments and is referred to as KL-introspectability.

4) KL-COMPACT

Compactness is defined as the property that the subset in the encoding space that represents a factor is small [43]. KL-introspectability ensures class disentanglement, but this

does not affect compactness. To add compactness, we added another term to KL-introspectability.

$$\text{KL-compact} = \text{KL-introspectability} - \frac{1}{|C|}\sum_{c=1}^{|C|} \text{KL}_{\text{sym}}(\hat{E}_c||\hat{E}_N). \quad (15)$$

The second term in the above equation penalizes the class cluster with larger variance. Thus, KL-compact ensures class disentanglement by ensuring compactness of the factor clusters. Similar to KL-introspectability logarithm was used here.

5) COVARIANCE OF ENCODINGS (COVARIANCE-e)

The fundamental concept behind disentanglement is that the encoding factors are independent. Each independent factor influencing fault classification should preferably be encoded by only one of the encoding dimensions for modularity. For this effect, the covariance between the encoding dimensions, $\text{cov}(E_c^d, E_c^{d'})$, should be minimal for all classes. For class c,

$$\text{covariance}_c = \frac{1}{\binom{|D|}{2}} \sum_{d=1}^{D} \sum_{d'=d+1}^{D} \text{cov}(E_c^d, E_c^{d'}). \quad (16)$$

The covariance-e is defined so that a higher score means more interpretability,

$$\text{covariance-e} = \frac{1}{\frac{1}{|C|}\sum_{c\in C} \text{covariance}_c}. \quad (17)$$

B. HYPERPARAMETER OPTIMIZATION

Variational autoencoders are typically used to obtain the disentangled encoding of data. However, the drawback of this method is that the trade-off between accuracy and interpretability is unknown and tuning for an acceptable balance is not feasible. To avoid this, we use hyperparameter optimization to find models that form the Pareto front of the accuracy-interpretability trade-off. We used LSTM architecture hyperparameters for the optimization. These are the hidden dimensions and number of LSTM layers. The Evolutionary Algorithm (EA) has a high sample complexity and is not suitable when the evaluation process is expensive. Therefore, multi-objective hyperparameter optimization with Bayesian Optimization (BO) was used to implement this. BO is a sample-efficient optimization method that significantly reduces the computational resource consumption.

Objective Functions: In our multi-objective optimization scenario, we grapple with two conflicting objectives: accuracy and interpretability. To accommodate this duality, we establish a Gaussian Process (GP) surrogate model for each objective. Each GP model is dedicated to representing an objective function which takes model hyperparameters as input and produces an accuracy or interpretability score as output. These surrogate models were trained using available data points pertinent to their respective objectives.

Acquisition Function: We deployed an acquisition function for multi-objective optimization: Noisy Expected Hypervolume Improvement (NEHVI). This specialized function guides the selection of the next query point, with the aim of enhancing the hypervolume.

Optimization Process: Bayesian optimization follows a sequential procedure for selecting query points by optimizing the acquisition function. In each iteration, it identifies the point poised to maximize the hypervolume within the objective space. Subsequently, this point was evaluated using both surrogate models to derive objective function values.

VII. EXPERIMENTAL SETUP

A. PSML DATASET

The Power System dataset for Machine Learning benchmarking (PSML) is a unique dataset for ML-based grid operations, with open access and multiscale time-series data [2]. The dataset covers three years of minute-level real-world load, weather, and renewable time-series data across 66 areas in the United States, as well as one year of minute-level synchrophasor measurements in three scenarios and over 1000 disturbance cases with millisecond-level synchrophasor measurements. This dataset is comprehensive and maintains consistency across various timescales, encompassing both the transmission and distribution-level dynamics. It encompasses diverse energy resources and dynamic events, rendering it exceptionally well suited for implementing machine-learning-based algorithms.

A schematic of the mechanism used by the authors in [2] is provided in the Supplementary Material. The authors first gathered real-world weather and load time-series data. These datasets were used in conjunction with physical renewable generation models to generate the solar and wind power generation profiles. Three distinct time-series datasets were acquired: load, solar-power, and wind-power datasets. A co-simulation model was constructed to simulate the combined transmission and distribution readings. The transmission grid model was implemented using PSS/E, adapted from the original PSS/E 23-bus test system. The distribution grid models are based on an IEEE 13-bus feeder. These models are linked to the corresponding load buses in the transmission system model. Solar photovoltaic (PV) and power inverter models were attached to load buses within each distribution grid, effectively representing aggregated residential rooftop solar installations. A series of simulations was conducted to obtain multiscale measurement data. These included steady-state power flow simulations under various load and renewable generation scenarios, and transient dynamic simulations involving random disturbances. The simulations used an innovative joint transmission and distribution (T + D) grid platform.

In our research, we focused solely on the occurrence of faults as events of interest, as illustrated in the Supplementary Material. The fault classification dataset in PSML consists of five types of faults: branch fault, branch trip, bus fault, bus

trip, and generator trip. Branch and bus faults are short-circuit faults between the conductors and ground faults. Trip-type faults occur during equipment tripping. These faults were distributed in 549 instances of time-series sequences, with 439 instances in the training set and 110 in the test set. Each instance spans 4 s with 91 different voltage, current, and power readings from various locations in the transmission system. These were recorded at 240Hz, resulting in 960 time steps in one instance of the time-series sequence. Because this is a long sequence, sub-sampling was performed to obtain a shorter sequence length of 38 time steps for a 4 s duration.

B. SEARCH SPACE

1) LSTM MODEL

The architecture of the LSTM model is shown in Fig. 1. The search space includes only the architectural hyperparameters of the LSTM model, as mentioned in the previous section. The model has an LSTM module (with many layers and hidden dimensions) followed by an encoding layer. The encoding layer was a dense layer with linear activation. The dimensions of the encodings were fixed at 10 for all models in the search space. This method assumes a maximum of two encoding dimensions in order to represent each CIF. The classification layer is also a dense layer with a linear activation function followed by a softmax operation. The bounds of the search space were [1, 5] for the number of LSTM layers and [8, 512] for the hidden dimensions. The search space had a total of 2525 models.

The following models were also considered in the experiments to validate the model-agnostic nature of interpretability metrics.

2) BIDIRECTIONAL LSTM MODEL

The bidirectional LSTM (Bi-LSTM) model processes the input in both the forward and backward directions of time sequences. In this model, only the LSTM module shown in Fig. 1 is replaced with the Bi-LSTM module. All hyperparameter choices were identical to those of the LSTM model hyperparameters (as explained in the previous section). The architecture of the Bi-LSTM model is presented in the Supplementary Material.

3) CNN-LSTM MODEL

In the CNN-LSTM model, the CNN layers were added before the LSTM module. The CNN captures short-term patterns in the time-series data. The output of the CNN layers was condensed time-series data with explicit short-term features. The LSTM module can capture the long-term patterns from these short-term features. Two CNN layers with 32 and 64 filters (kernel_size=3, stride=1, and no padding) are used in the CNN module. Further details regarding the CNN-LSTM model architecture are provided in the the Supplementary Material. The search space parameters are the LSTM module hyperparameters. For a fair comparison, all hyperparameters were kept the same as those in the previous sections.

4) LSTM-CNN MODEL

In the LSTM-CNN model, CNN layers were added after the LSTM module. The LSTM module generates a context vector as a latent-variable vector at each time step. All these were stacked to form a time-series of context vectors. The CNN module captures the temporal patterns of the context vectors. For the CNN module, we used two CNN layers with 32 and 64 filters (kernel_size=3, stride=1, and no padding), followed by a max-pooling layer (kernel_size=2, stride=1, and no padding) and a flatten layer. The Supplementary Material provides further details on the architecture of the model. All hyperparameters were kept the same as those in the previous sections.

C. BO HYPERPARAMETERS

We used BOTORCH [44] to implement multi-objective BO. The model hyperparameters serve as inputs to the objective functions, generating accuracy or interpretability scores, thereby striving to optimize both aspects concurrently. We consider the class imbalance for the accuracy objective by calculating the balanced accuracy, as done by the authors in [2]. The macro-averaged mean absolute error was used to penalize false positives and false negatives. The training data of PSML were randomly (balanced) split in a 70:30 ratio to obtain the validation set. The model was trained on this training split for 500 epochs with cross-entropy loss. In addition to early stopping with the validation set, a dropout layer was added before the classification layer, with a probability of 0.2. These provide sufficient regularization to prevent overfitting during the training phase. Balanced accuracy of the validation set was used as the accuracy objective. Each of the metrics mentioned in Section VI and introspectability [38] scores were used for the interpretability objective. Experiments were repeated for each of the six interpretability metrics. For the CIF-c, the value of α was empirically set to 0.6. The sample size, N, was 2000. The interpretability scores were obtained for the validation split. The multi-objective acquisition function used for BO is Noisy Expected HyperVolume Improvement (NEHVI). The reference point used for the hypervolume is (0, 0) because all metrics are defined as positive values and maximization objectives.

VIII. RESULTS

A hyperparameter search using BO is conducted for each interpretability metric. Fig. 2 shows the performance of the proposed methodology during the experiments using LSTM model. Fig. 2a shows the convergence of the training and validation set losses during the training phase of one of the sampled LSTM models. The plot shows that there was no over-fitting during training. Fig. 2b shows the improvement in the dominated hypervolume (hypervolume of the Pareto front) as the search progressed for the CIF-e. BO can escape

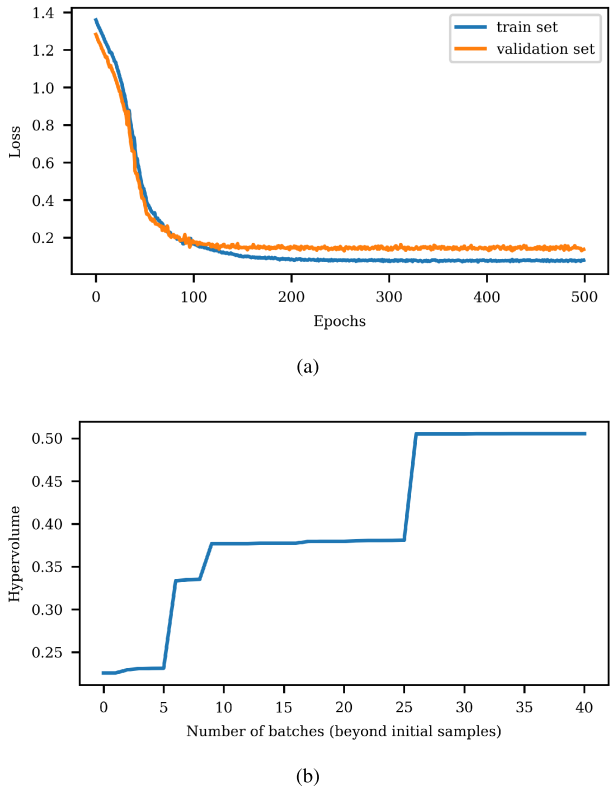

FIGURE 2. Convergence of (a) the train set loss and the validation set loss during the training phase of the model, indicating stable learning, and (b) the dominated hypervolume during the BO search progression, showing the growth of Pareto front.

from local optimum solutions and eventually converge. Fig. 3 shows the LSTM model Pareto front obtained by BO for each case. The Pareto fronts obtained for the other models exhibited similar trends, as shown in the Supplementary Material. The accuracy and interpretability scores reported in this section were calculated on the PSML test set. There was a drop in the accuracy for all cases when a more interpretable model was obtained. The trade-off was minimal for introspectability score, suggesting a correlation between introspectability and accuracy. The most accurate models in all cases had similar accuracy scores; however, the most interpretable models in the Pareto front had different accuracy scores. This indicates that different interpretability metrics have different trade-off dynamics with respect to the accuracy. Table 1 summarizes the accuracy scores of Pareto fronts. The BO search found a model with highly competitive accuracy (**77.63**%) compared to all previous benchmark models [2]. The previous sota accuracy was 74.2±2.9% for the LSTM-FCM model.

To understand the relationship between the interpretability metrics and the different manifestations of interpretability in the fault classification data, we examined the correlation of these metrics with various factors such as the likelihood of finding subsequences and the similarity of different classes in the encoding space. The following subsections present detailed analyses of these factors.

A. ANALYSIS

1) SUBSEQUENCES AND ACCURACY

Identifying relevant subsequences is an essential aspect of the post-hoc interpretability of time-series models [24]. The subsequence of a time-series refers to the window of a time-series with an identifiable pattern. In fault classification, the subsequences correspond to the unique signatures of different types of faults. In this section, we test whether a higher score on the interpretability metric leads to the identification of more relevant subsequences. We used post-hoc interpretation with Shapley Additive Explanations (SHAP) [42] to determine the relevant subsequences. A trained model with one sample input was required to obtain the SHAP values. The SHAP values represent the contribution of each input element to the output of the model. The fault classification problem has 91 input features and a 4-second time window. This study finds SHAP values corresponding to only the time dimension to obtain the time-step contribution of the input time-series to the model prediction. To implement this, we modified the codebase of TimeSHAP [45]. The time-step SHAP values were used to identify regions with high-contribution plateaus. These plateaus are the windows that contribute the most to the model inference. The input time steps corresponding to these plateaus form the subsequences.

For the trained model, subsequences were identified for the test set of the data. Table 1 shows a summary of the Pareto front solutions concerning accuracy and subsequences. The CIF-based metrics exhibited constant trends across all the models for the subsequences. CIF-e consistently found the longest subsequence for all models. However, CIF-c had the highest average length of subsequences. This suggests that the CIF-e has a larger variability with respect to the Pareto solutions. The randomness involved in CIF-e leads to such variability. In contrast, CIF-c does not have much randomness in the algorithm; hence, there is less variability, even though the average performance metrics are better than those of CIF-e. The LSTM model finds the longest subsequence among all the models. However, the variation in the longest subsequence among the different models was minimal, indicating consistency of the CIF-e algorithm. However, the best average subsequence length is obtained using the Bi-LSTM model. Here, the variation is minimal. For all models, introspectability was the least effective metric for finding the subsequences. KL-introspectability was slightly better at identifying subsequences than introspectability. KL-compact and covariance-e are less effective than CIF-based metrics in identifying subsequences but are better than introspectability-based metrics. In addition, the performance of a metric changes slightly when the model changes, but the relative performance compared with the other metrics remains the same for all models. The LSTM model has the longest subsequence, whereas the Bi-LSTM model has the longest average length.

For the majority of the models, the Pareto front for accuracy-introspectability had the best accuracy score (both maximum and mean). The results are presented in Table 1.

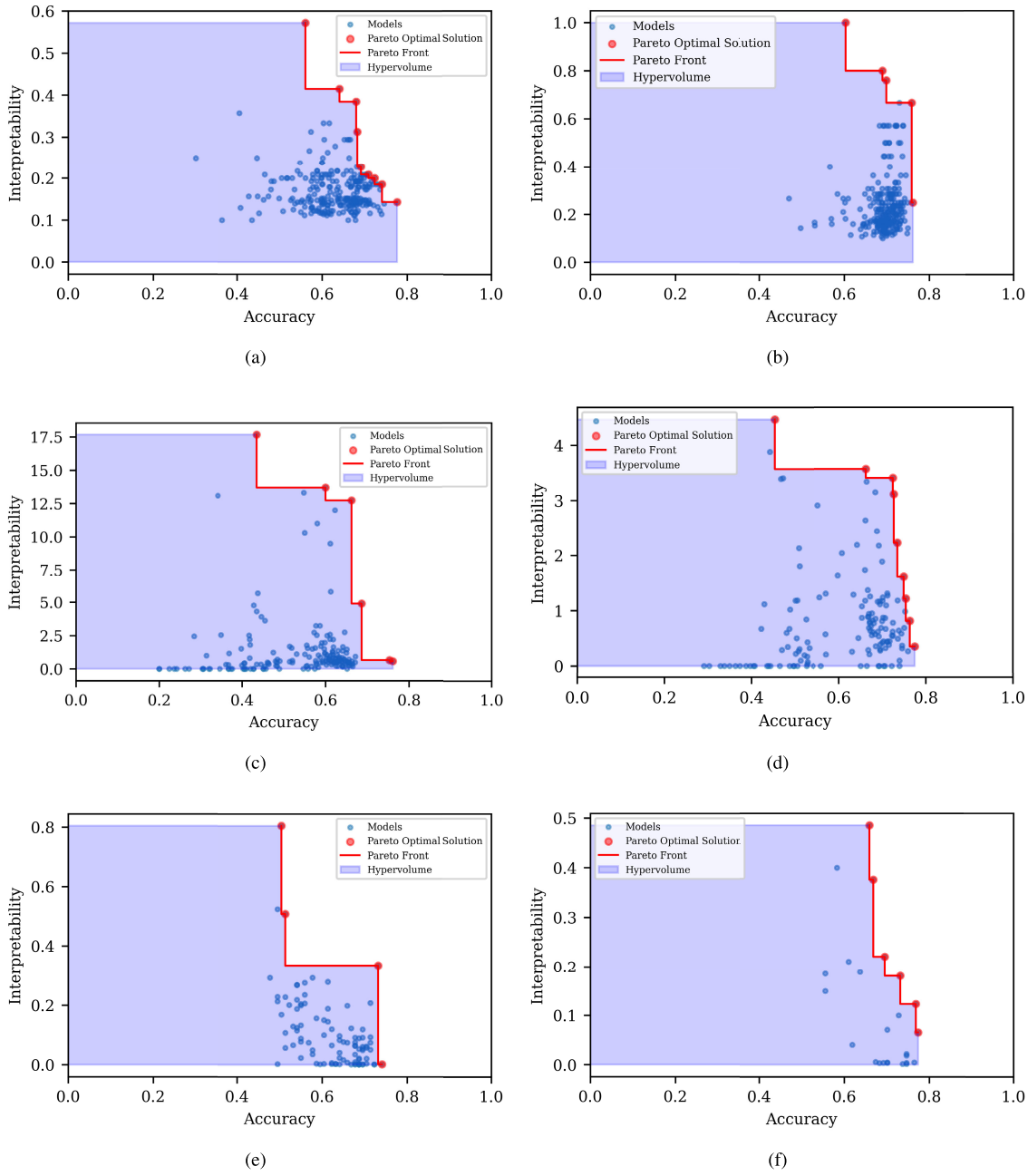

FIGURE 3. LSTM pareto solutions obtained for (a) CIF-e, (b) CIF-c, (c) KL-Compact, (d) KL-Introspectability, (e) Covariance-e, and (f) Introspectability, showing different accuracy-interpretability trade-offs for different Pareto fronts. The Pareto front is shown by the red line, while the blue dots show the sampled models, and the red dots show the individual models which form the Pareto solution.

However, introspectability is poor for identifying subsequences, as previously explained. From the Pareto front in Fig. 3, the accuracy-introspectability trade-off is minimal for the LSTM model. The Supplementary Material shows the Pareto front for the other models, where this phenomenon can be observed. This indicates a correlation between inter-class distance and accuracy. KL-introspectability has a larger trade-off for accuracy than introspectability, as is evident from the Pareto front and lower average accuracy. However, the low average subsequence length suggests that this score does not consistently indicate interpretability. Because KL-introspectability reduces the overlap between class clusters, this result implies that dissimilar class clusters do not necessarily cause disentanglement of the factors. These results suggest that class disentanglement, as measured by these metrics, does not translate into signature subsequences. The KL-compact metric has a worse mean accuracy score than the introspectability metric but is slightly better at

TABLE 1. Summary of Pareto solutions for different interpretability metrics, showing the dominance of CIF-based metrics in enhancing interpretability. Mean and maximum accuracies of the Pareto solutions for each model, along with the corresponding mean and maximum subsequence lengths, are reported for each of the interpretability metrics. The highlighted values indicate the best scores for each model.

Model	Metric	Accuracy (%)		Subsequence length (time-steps)	
		Maximum	Mean	Maximum	Mean
LSTM	CIF-e	**77.63**	68.97	**20**	12.3
	CIF-c	76.11	70.23	18	**13.8**
	KL-compact	76.24	64.96	15	8.8
	KL-introspectability	77.57	70.56	13	8.5
	Covariance-e	74.09	62.27	15	10.9
	Introspectability	77.36	**71.60**	10	7.5
Bi-LSTM	CIF-e	77.34	65.82	**19**	13.9
	CIF-c	76.93	67.75	18	**14.1**
	KL-compact	76.21	64.84	18	7.7
	KL-introspectability	76.83	**70.38**	11	7.5
	Covariance-e	75.84	61.08	13	9.4
	Introspectability	**77.42**	67.90	9	6.2
CNN-LSTM	CIF-e	76.99	64.71	**19**	13.0
	CIF-c	76.93	68.79	18	**13.7**
	KL-compact	74.69	62.08	16	10.8
	KL-introspectability	76.43	58.15	12	9.7
	Covariance-e	75.70	52.49	14	10.7
	Introspectability	**77.17**	**72.68**	11	7.4
LSTM-CNN	CIF-e	77.16	67.40	**18**	12.6
	CIF-c	75.29	70.19	17	**13.2**
	KL-compact	72.54	66.57	13	10.8
	KL-introspectability	74.59	66.25	10	8.3
	Covariance-e	72.05	62.52	12	8.8
	Introspectability	**77.17**	**72.43**	9	6.3

finding subsequences. Covarance-e, on the other hand, has a sharper trade-off in the Pareto front, as is evident from its poor mean accuracy. Although the accuracy trade-off is very high, it performs better than other non-CIF-based methods in terms of interpretability. It had the lowest accuracy score among all the metrics. In summary, introspectability-based metrics exhibit a poor relationship with interpretability. KL-compact and covariance-e have a high accuracy trade-off, whereas CIF-based metrics maintain a better accuracy trade-off and simultaneously achieve good interpretability.

For the LSTM model, Fig. 4 shows the mean length of the subsequences identified by the Pareto solution models for the CIF-e and CIF-c. The mean length varied more uniformly for CIF-c than CIF-e. This can be attributed to the variability and randomness of the CIF-e algorithm. The corresponding plots for the other models are shown in the Supplementary Material, where the same trend is observed.

To complete the subsequence-based analysis, we determined the correlation of the interpretability score with the subsequences of 200 models that were randomly sampled from the LSTM model search space. The results are presented in Table 2. The interpretability score of the trained model was obtained from the test data-set, and all the subsequences identified by the model for the test set were obtained using the SHAP plateaus, as before. The table shows that CIF-c has the highest correlation with the mean length of the subsequences identified by the model, whereas CIF-e has the highest correlation with the longest subsequence. Again, this can be attributed to variability in the CIF-e algorithm. Introspectability had the smallest correlation with both the mean and longest subsequences. This implies that without other mechanisms to improve disentanglement, increasing only the class dissimilarity of encoding dimensions could be counterproductive for interpretability.

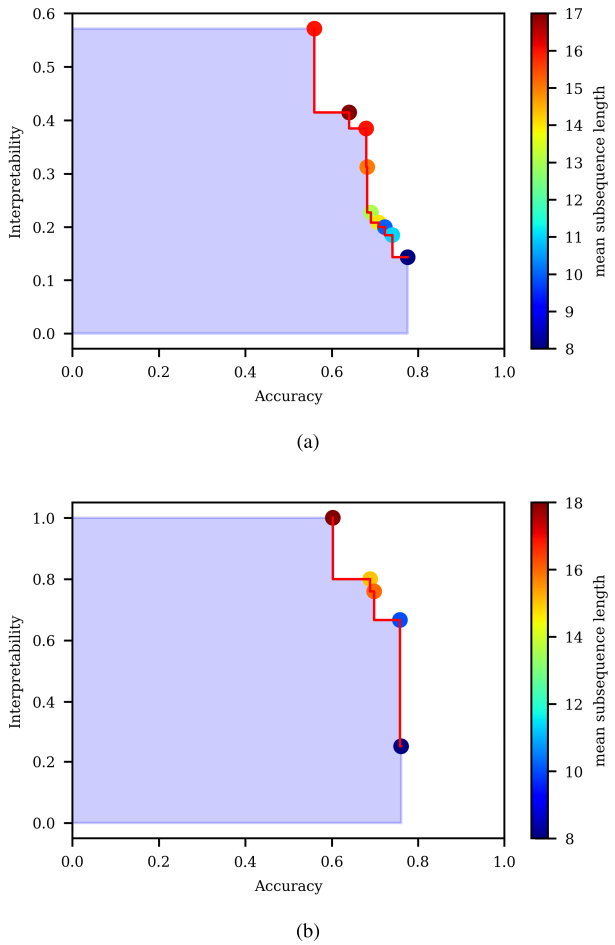

FIGURE 4. Distribution of mean subsequence length in the LSTM Pareto front for (a) CIF-e and (b) CIF-c, indicating consistent variation across the front. The color-coded dots represent the scores of each model in the Pareto front, as displayed in the corresponding color bar.

TABLE 2. Correlation of different interpretability metrics with maximum and mean subsequence lengths of the corresponding Pareto front models, showing a stronger correlation for the CIF-based metrics, whereas all metrics show at least weak correlation.

Metric	Length (max)	Length (mean)
CIF-e	**0.73**	0.67
CIF-c	0.69	**0.71**
KL-compact	0.58	0.61
KL-introspectability	0.45	0.44
Covariance-e	0.41	0.52
Introspectability	0.37	0.33

Fig. 5 shows some sample subsequences identified by the most interpretable LSTM model as per CIF-c. For each fault type, the subsequences were unique and almost always contained the peak value, except for the branch fault. However, the subsequence of the branch fault is distinct from that of the other faults. It can also be observed that the time step when fault triggering occurs is not included in any of the subsequences. Instead, distinctive shapelets become identifiers of subsequences. The less interpretable models did not lead to plateaus in the SHAP values and the contributions were uniform for the entire input time window.

2) CLASS SIMILARITY

To visualize the similarity of the fault classification factors in the encoding dimension, the encoding dimension of ten was reduced to two using T-distributed Stochastic Neighbor Embedding (t-SNE). The t-SNE creates a low-dimensional embedding in two dimensions, where the similarities between the data points are preserved as much as possible. Fig. 6 shows the t-SNE plot for the most interpretable and accurate LSTM models from the Pareto solutions of the CIF-c. The accurate model had more than one cluster for the same class, and the clusters were separated with clear margins and no overlap. This indicates that the model implicitly learns multiple subcategories in the same class. In addition, class representations are more complex. The class clusters were evenly and uniformly separated, suggesting that the characteristics of each class are learned as distinct entities. This is inherently less interpretable, because no inferences can be made regarding the relationship between different classes. No insight into the failure cases can be learned from this plot, and the occurrences of failures not anticipated in the application make this an unreliable model.

By contrast, the t-SNE plot of the interpretable model has overlapping but separable clusters with fuzzy boundaries. There was a stark dissimilarity between the fault and trip-type categories. Bus and branch faults form an adjoining cluster set with a long linear distribution. This suggests that the model learns the bus and branch faults as variants with similar properties. Similarly, branch, bus, and generator trips form adjoining clusters with similar patterns. The model considers all trip-fault categories to have common characteristics. Among these, bus trips are similar to branch and generator trips, whereas the latter are farther clusters. These inferences are in agreement with the theoretical assumptions regarding grid faults. The disadvantage of the interpretable model is the trade-off in accuracy. This is evident from the fuzzy boundaries between some class clusters, which lead to more misclassifications. The advantage is that the model is more reliable and the failure cases can be explained. If the trade-off is within the acceptable range for the application, the interpretable model is the best choice.

3) MODEL ARCHITECTURE

The experiments conducted in the previous section described the hyperparameter search space containing the number of hidden layers and the hidden layer dimensions. This leads to models with different interpretabilities and accuracies. Fig. 7 shows how accuracy and interpretability are distributed in the search space. The figure shows only the CIF-c pattern for the LSTM model. The accuracy distribution has a clear pattern, with the cluster for higher accuracy existing in the region with high hidden dimensions and approximately two or three hidden layers. A lower number of layers and lower

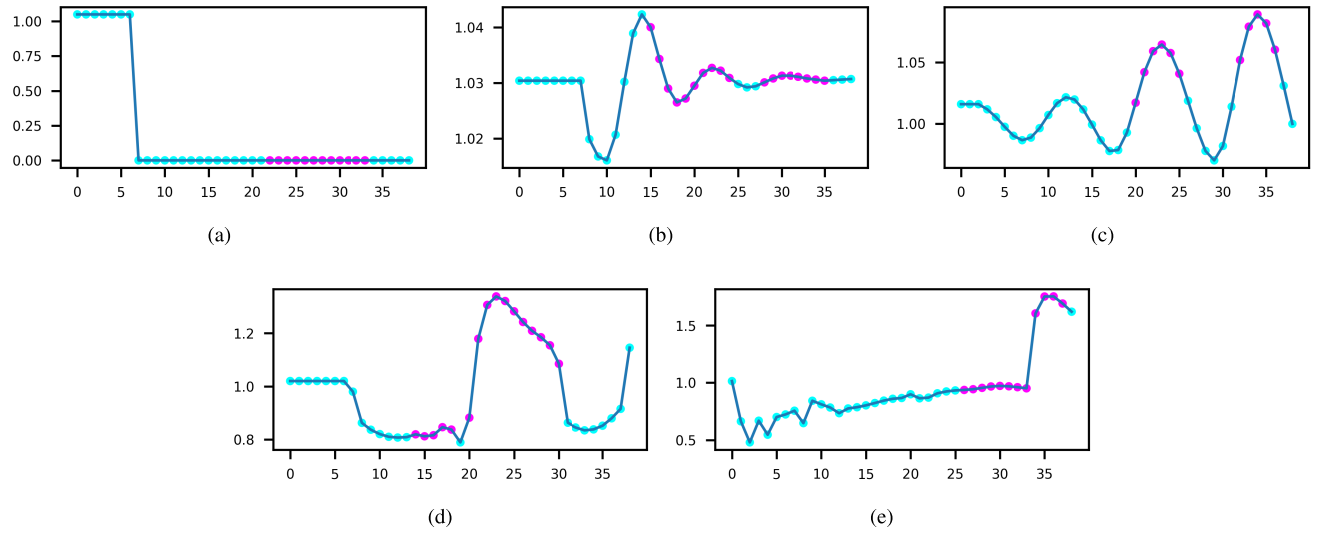

FIGURE 5. Sample subsequences identified by the LSTM models for (a) branch fault, (b) branch trip, (c) bus fault, (d) bus trip, and (e) generator trip, show distinct signature subsequence for each fault type. 38 samples are taken for one time-window of 4 seconds (x-axis), and the voltage values are normalized (y-axis). The highlighted adjacent points in magenta form the subsequence.

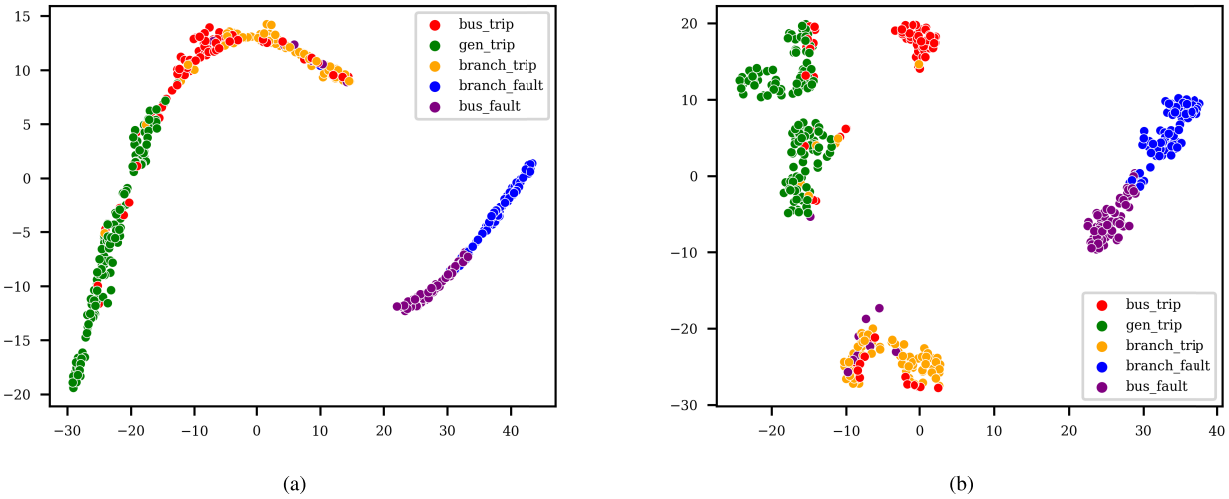

FIGURE 6. t-SNE similarity between the different types of faults for (a) the most interpretable model from the Pareto front of the LSTM models, where clusters form patterns aligned with fault types, and (b) the most accurate model from the Pareto front of the LSTM models, where clusters are scattered and without explicit patterns.

hidden dimensions lead to simpler models, and hence, lower accuracy. The accuracy is relatively low with more hidden layers, suggesting over-fitting in the region. However, the trend for interpretability is different. Interpretable models are rarer than accurate models, and no single clusters exist for the hidden dimensions. The number of layers is lower than that of the high-accuracy models, and most models are clustered around hidden dimensions between 250 and 350, with some scattered on the lower side. This shows that although simpler LSTM models are more interpretable than complex ones, minimum complexity is necessary for a satisfactory interpretation of the inferences.

The clusters of accurate and interpretable models overlap less, although the most accurate and interpretable models are farther apart. Thus, a trade-off exists during the search. This is also evident in Fig. 3. This demonstrates that interpretability is a property of the architecture of a model. Choosing an appropriate architecture (even with the same LSTM modules) can make the model more reliable for applications. The distributions for the Bi-LSTM, CNN-LSTM, and LSTM-CNN models are shown in the Supplementary Material. The trend of generating different accuracy and interpretability clusters also holds true for these models.

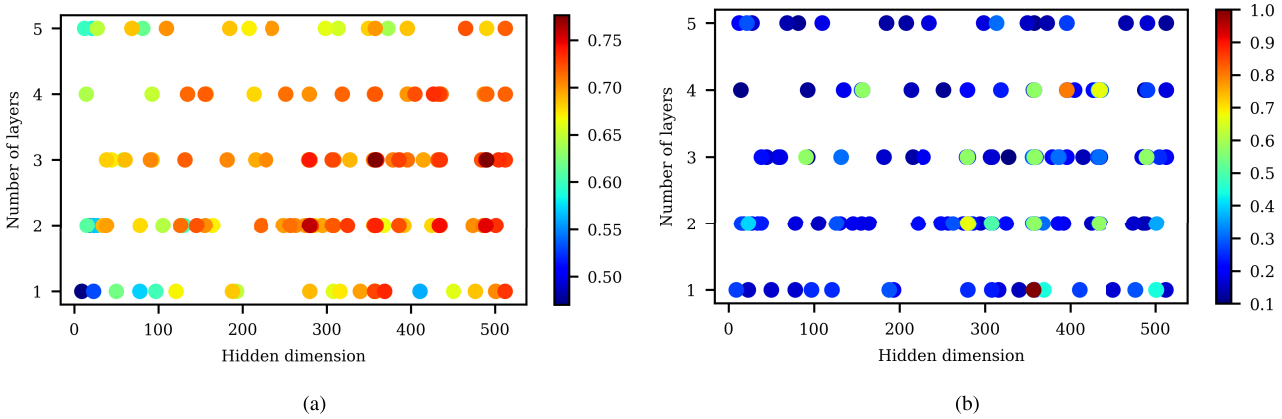

FIGURE 7. Distribution of (a) accuracy and (b) CIF-c score in the search space of the LSTM models, indicating strong clustering for accuracy, and weak clustering for interpretability, with small overlapping. Each dot represents a sampled model, color-coded based on the accuracy or CIF-c score. The search space consists of the number of LSTM layers and the corresponding hidden dimension.

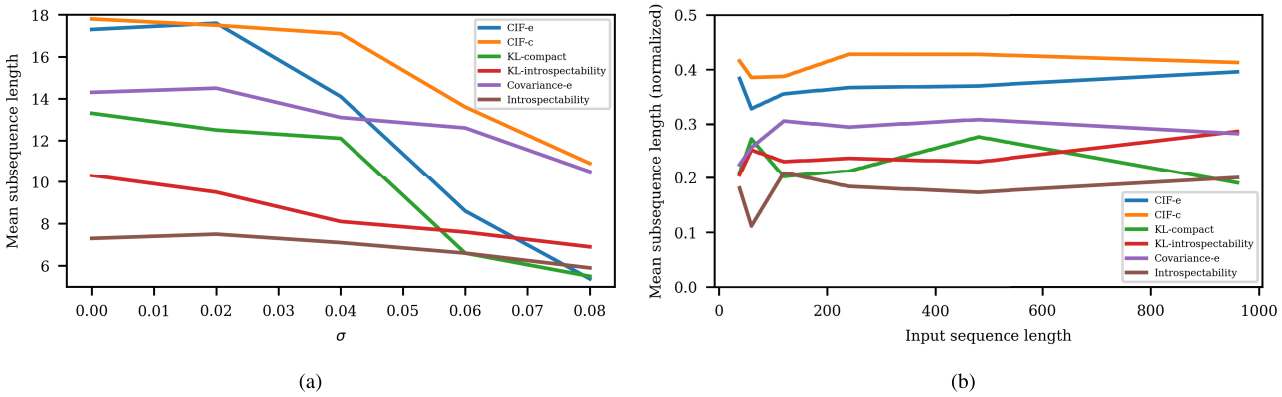

FIGURE 8. Evaluation of (a) robustness by varying the input noise level (σ), illustrating increased performance degradation at higher noise levels, and (b) scalability by varying the complexity of the input signal, demonstrating good scalability. The mean subsequence length is normalized for the assessment of scalability.

4) ROBUSTNESS AND SCALABILITY

To test the robustness of the interpretability metrics for the LSTM model, noise is added to the model input. Gaussian noise $\mathcal{N}(0, \sigma^2)$, was added to each element of the input signal with a fixed σ. A BO search is then conducted to generate the Pareto front for each interpretability metric. For a fair comparison, only the top 10% of the interpretable models were chosen from the Pareto front to calculate the mean subsequence length. This process was repeated for different values of σ. Fig. 8a shows the variation in the mean subsequence length against σ for all metrics. It can be observed that for small noise levels, all the metrics are resilient. At higher noise levels, CIF-e degrades rapidly. This can be explained by the high variance of the algorithm, which makes it highly sensitive to noise. Although CIF-c exhibits mild degradation at higher noise levels, it is the most effective metric in that noise range by a large margin. Covariance-e is the most noise-resilient and has a performance similar to that of CIF-c at the highest noise level. Introspectability-based metrics are the least effective for finding subsequences; hence, degradation is not pronounced. KL-compact also showed poor resilience to noise. In summary, the proposed methodology has high robustness for moderate noise levels; however, for very high noise levels, there is a mild degradation in the performance. In addition, the proposed methodology performed better than the other metrics even at high noise levels.

The scalability is tested using a longer input signal. The input time-series of the PSML data has 960 time steps. This was sampled at different rates to obtain the input signals with different sequence lengths. The input signal length ranged from 38 to 960 for scalability analysis. We repeated the same processes as in the robustness test but with different input signal lengths. The results are shown in Fig. 8b. The mean subsequence length was normalized for comparison. It can be observed that the normalized mean subsequence length is nearly constant for all the input signal lengths, except for minor fluctuations for the smaller lengths. These results demonstrate good scalability. This is possible because of the ability of LSTM to learn long-sequence relationships. This

ability of LSTM is directly transferred to interpretability metrics because of the encoder architecture used in the model, as shown in Fig. 1. Because the encoder dimension is constant regardless of the input sequence length, the interpretability metrics are scalable as far as the model itself is concerned. The interpretability metrics are scalable because the LSTM and CNN models are scalable in terms of the input size.

IX. CONCLUSION AND FUTURE WORK

Fault classification for grid operations is a highly sensitive application, and this study improves the reliability of LSTM models for this task by improving the interpretability of the model. To this end, different novel interpretability metrics were defined. These are unique in defining the disentanglement factor as a property of the classifier of the model, which implicitly identifies factors relevant to fault classification alone. It has been shown that these factors correspond to the subsequences or shapelets present in the fault signals. These subsequences are unique to different fault classes, and interpreting the model inference in terms of the signature subsequences of various faults enhances the reliability of the model, even with a trade-off in accuracy. Based on an acceptable threshold for the application, the Pareto solution offers models with different accuracy and interpretability trade-offs. It was also shown that class relations can be inferred more evidently for interpretable models when such inferences are not evident in less-interpretable models. In summary, this study established a new method to enhance the post-hoc interpretability of LSTM models (as well as other variants and hybrids) for fault classification problems. This was achieved without incorporating interpretable mechanisms into the model architecture. For the future direction of this work, the methodology can be extended to other models, such as CNN and attention. The transferability to grid forecasting problems can be tested. Furthermore, the methodology can be modified to include constraints, such as latency, model size, and minimum accuracy.

REFERENCES

[1] Y. Wang, C.-F. Chen, P.-Y. Kong, H. Li, and Q. Wen, "A cyber-physical-social perspective on future smart distribution systems," *Proc. IEEE*, vol. 111, no. 7, pp. 694–724, Jul. 2023.

[2] X. Zheng, N. Xu, L. Trinh, D. Wu, T. Huang, S. Sivaranjani, Y. Liu, and L. Xie, "A multi-scale time-series dataset with benchmark for machine learning in decarbonized energy grids," *Sci. Data*, vol. 9, no. 1, p. 359, Jun. 2022.

[3] C. Iturrino, M. Bindi, and F. Corti, 2022, "Power quality disturbance," *IEEE Dataset*, doi: 10.21227/567h-yw15.

[4] H. Al Hassan, A. Reiman, G. Reed, Z.-H. Mao, and B. Grainger, "Model-based fault detection of inverter-based microgrids and a mathematical framework to analyze and avoid nuisance tripping and blinding scenarios," *Energies*, vol. 11, no. 8, p. 2152, Aug. 2018. [Online]. Available: https://www.mdpi.com/1996-1073/11/8/2152

[5] Y.-S. Oh, C.-H. Kim, G.-H. Gwon, C.-H. Noh, S. B. A. Bukhari, R. Haider, and T. Gush, "Fault detection scheme based on mathematical morphology in last mile radial low voltage DC distribution networks," *Int. J. Electr. Power Energy Syst.*, vol. 106, pp. 520–527, Mar. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061518322324

[6] Y. Xia, Y. Xu, and B. Gou, "A data-driven method for IGBT open-circuit fault diagnosis based on hybrid ensemble learning and sliding-window classification," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5223–5233, Aug. 2020.

[7] B. Cai, Y. Zhao, H. Liu, and M. Xie, "A data-driven fault diagnosis methodology in three-phase inverters for PMSM drive systems," *IEEE Trans. Power Electron.*, vol. 32, no. 7, pp. 5590–5600, Jul. 2017.

[8] F. Deng, Y. Zhong, Z. Zeng, Y. Zu, Z. Zhang, Y. Huang, S. Feng, and X. Zeng, "A single-ended fault location method for transmission line based on full waveform features extractions of traveling waves," *IEEE Trans. Power Del.*, vol. 38, no. 4, pp. 2585–2595, Aug. 2023.

[9] J. Liang, T. Jing, H. Niu, and J. Wang, "Two-terminal fault location method of distribution network based on adaptive convolution neural network," *IEEE Access*, vol. 8, pp. 54035–54043, 2020.

[10] T. Sirojan, S. Lu, B. T. Phung, D. Zhang, and E. Ambikairajah, "Sustainable deep learning at grid edge for real-time high impedance fault detection," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 2, pp. 346–357, Apr. 2022.

[11] H. Liu, S. Liu, J. Zhao, T. Bi, and X. Yu, "Dual-channel convolutional network-based fault cause identification for active distribution system using realistic waveform measurements," *IEEE Trans. Smart Grid*, vol. 13, no. 6, pp. 4899–4908, Nov. 2022.

[12] S.-Z. Hou, W. Guo, Z.-Q. Wang, and Y.-T. Liu, "Deep-learning-based fault type identification using modified CEEMDAN and image augmentation in distribution power grid," *IEEE Sensors J.*, vol. 22, no. 2, pp. 1583–1596, Jan. 2022.

[13] V. Veerasamy, N. I. A. Wahab, M. L. Othman, S. Padmanaban, K. Sekar, R. Ramachandran, H. Hizam, A. Vinayagam, and M. Z. Islam, "LSTM recurrent neural network classifier for high impedance fault detection in solar PV integrated power system," *IEEE Access*, vol. 9, pp. 32672–32687, 2021.

[14] B. Roy, S. Adhikari, S. Datta, K. J. Devi, A. D. Devi, F. Alsaif, S. Alsulamy, and T. S. Ustun, "Deep learning based relay for online fault detection, classification, and fault location in a grid-connected microgrid," *IEEE Access*, vol. 11, pp. 62674–62696, 2023.

[15] Y. Ren, S. Yuan, G. Cheng, Q. Zhao, L. Wang, D. Liang, and M. Yuan, "Fault diagnosis of UHVDC transmission system based on gated recurrent unit," in *Proc. Panda Forum Power Energy (PandaFPE)*, Apr. 2023, pp. 1792–1796.

[16] W. Gao and R.-J. Wai, "A novel fault identification method for photovoltaic array via convolutional neural network and residual gated recurrent unit," *IEEE Access*, vol. 8, pp. 159493–159510, 2020.

[17] F. Zhang, Q. Liu, Y. Liu, N. Tong, S. Chen, and C. Zhang, "Novel fault location method for power systems based on attention mechanism and double structure GRU neural network," *IEEE Access*, vol. 8, pp. 75237–75248, 2020.

[18] M. Alrifaey, W. H. Lim, C. K. Ang, E. Natarajan, M. I. Solihin, M. R. M. Juhari, and S. S. Tiang, "Hybrid deep learning model for fault detection and classification of grid-connected photovoltaic system," *IEEE Access*, vol. 10, pp. 13852–13869, 2022.

[19] A. Ahmed, K. S. Sajan, A. Srivastava, and Y. Wu, "Anomaly detection, localization and classification using drifting synchrophasor data streams," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3570–3580, Jul. 2021.

[20] X. Liu, X. Miao, H. Jiang, J. Chen, and Z. Chen, "Fault diagnosis in power line inspection using normalized multihierarchy embedding matching," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–10, 2023.

[21] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision making and a 'right to explanation,'" *AI Mag.*, vol. 38, no. 3, pp. 50–57, Oct. 2017. [Online]. Available: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2741

[22] R. El Shawi, Y. Sherif, M. Al-Mallah, and S. Sakr, "Interpretability in healthcare: A comparative study of local machine learning interpretability techniques," in *Proc. IEEE 32nd Int. Symp. Computer-Based Med. Syst. (CBMS)*, Jun. 2019, pp. 275–280.

[23] W. Li, H. Lan, J. Chen, K. Feng, and R. Huang, "WavCapsNet: An interpretable intelligent compound fault diagnosis method by backward tracking," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[24] A. Theissler, F. Spinnato, U. Schlegel, and R. Guidotti, "Explainable AI for time series classification: A review, taxonomy and research directions," *IEEE Access*, vol. 10, pp. 100700–100724, 2022.

[25] A. A. Ismail, M. Gunady, H. C. Bravo, and S. Feizi, "Benchmarking deep learning interpretability in time series predictions," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 1–12.

[26] O. F. Eikeland, I. S. Holmstrand, S. Bakkejord, M. Chiesa, and F. M. Bianchi, "Detecting and interpreting faults in vulnerable power grids with machine learning," *IEEE Access*, vol. 9, pp. 150686–150699, 2021.
[27] Y. Wu, X. Han, Z. Niu, and B. Yan, "Data-driven method and interpretability analysis for transient power angle stability assessment," in *Proc. IEEE 6th Conf. Energy Internet Energy Syst. Integr. (EI)*, Nov. 2022, pp. 1806–1810.
[28] Z. Fan, X. Xu, R. Wang, and H. Wang, "Fan fault diagnosis based on lightweight multiscale multiattention feature fusion network," *IEEE Trans. Ind. Informat.*, vol. 18, no. 7, pp. 4542–4554, Jul. 2022.
[29] M. F. Azam and M. S. Younis, "Multi-horizon electricity load and price forecasting using an interpretable multi-head self-attention and EEMD-based framework," *IEEE Access*, vol. 9, pp. 85918–85932, 2021.
[30] L. Munkhdalai, T. Munkhdalai, V.-H. Pham, M. Li, K. H. Ryu, and N. Theera-Umpon, "Recurrent neural network-augmented locally adaptive interpretable regression for multivariate time-series forecasting," *IEEE Access*, vol. 10, pp. 11871–11885, 2022.
[31] T. Ren, T. Han, Q. Guo, and G. Li, "Analysis of interpretability and generalizability for power converter fault diagnosis based on temporal convolutional networks," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.
[32] C. Yang, J. Zhang, Y. Chang, J. Zou, Z. Liu, and S. Fan, "A novel deep parallel time-series relation network for fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.
[33] H. Zhang, P. Wang, S. Liang, T. Zhou, and W. Wang, "Heterogeneous feature based time series classification with attention mechanism," *IEEE Access*, vol. 11, pp. 19373–19384, 2023.
[34] F. Bilendo, H. Badihi, N. Lu, P. Cambron, and B. Jiang, "Imaging wind turbine fault signatures based on power curve and self-organizing map for image-based fault diagnosis," in *Proc. IEEE Int. Symp. Adv. Control Ind. Processes (AdCONIP)*, Aug. 2022, pp. 204–209.
[35] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–31.
[36] R. Jiang, Y. Xue, and D. Zou, "Interpretability-aware industrial anomaly detection using autoencoders," *IEEE Access*, vol. 11, pp. 60490–60500, 2023.
[37] Y. Li, Z. Chen, D. Zha, M. Du, J. Ni, D. Zhang, H. Chen, and X. Hu, "Towards learning disentangled representations for time series," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, vol. 1, 2022, pp. 3270–3278.
[38] Z. Carmichael, T. Moon, and S. A. Jacobs, "Learning interpretable models through multi-objective neural architecture search," 2021, *arXiv:2112.08645*.
[39] J. Mockus, "On Bayesian methods for seeking the extremum," in *Proc. Optim. Techn. IFIP Tech. Conf.*, G. I. Marchuk, Ed. Berlin, Germany: Springer, 1975, pp. 400–404.
[40] K. Miettinen, *Nonlinear Multiobjective Optimization* (International Series in Operations Research & Management Science). Cham, Switzerland: Springer, 1999. [Online]. Available: https://books.google.co.in/books?id=ha_zLdNtXSMC
[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[42] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 4768–4777.
[43] M.-A. Carbonneau, J. Zaïdi, J. Boilard, and G. Gagnon, "Measuring disentanglement: A review of metrics," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 14, 2022, doi: 10.1109/TNNLS.2022.3218982.
[44] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "BOTORCH: A framework for efficient Monte–Carlo Bayesian optimization," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 1–15.
[45] J. Bento, P. Saleiro, A. F. Cruz, M. A. T. Figueiredo, and P. Bizarro, "TimeSHAP: Explaining recurrent models through sequence perturbations," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 2565–2573.

BIJU G. M. (Member, IEEE) received the M.Tech. degree from the Electrical Engineering Department, Indian Institute of Technology Roorkee (IIT Roorkee), Roorkee, India, in 2017, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include hyperparameter optimization, neural architecture search, image classification, and time-series.

G. N. PILLAI (Member, IEEE) received the M.Tech. degree in control systems from the National Institute of Technology Kurukshetra, Kurukshetra, India, in 1981, and the Ph.D. degree from the Indian Institute of Technology Kanpur (IIT Kanpur), Kanpur, India, in 2001. He is currently a Professor with the Department of Electrical Engineering and the Mehta Family School of Data Science and Artificial Intelligence, IIT Roorkee, Roorkee, India. His research interests include time-series forecasting, reinforcement learning, deep learning, power systems and control, system engineering, and soft computing techniques.