MDPI

Article

Generative Adversarial Network-Based Scheme for Diagnosing Faults in Cyber-Physical Power Systems

Hossein Hassani [1,*], Roozbeh Razavi-Far [1,2], Mehrdad Saif [1] and Vasile Palade [3]

1   Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada; roozbeh@uwindsor.ca (R.R.-F.); msaif@uwindsor.ca (M.S.)
2   School of Computer Science, University of Windsor, Windsor, ON N9B 3P4, Canada
3   Center for Data Science, Coventry University, Coventry CV1 5FB, UK; vasile.palade@coventry.ac.uk
*   Correspondence: hassa12t@uwindsor.ca

Abstract: This paper presents a novel diagnostic framework for distributed power systems that is based on using generative adversarial networks for generating artificial knockoffs in the power grid. The proposed framework makes use of the raw data measurements including voltage, frequency, and phase-angle that are collected from each bus in the cyber-physical power systems. The collected measurements are firstly fed into a feature selection module, where multiple state-of-the-art techniques have been used to extract the most informative features from the initial set of available features. The selected features are inputs to a knockoff generation module, where the generative adversarial networks are employed to generate the corresponding knockoffs of the selected features. The generated knockoffs are then fed into a classification module, in which two different classification models are used for the sake of fault diagnosis. Multiple experiments have been designed to investigate the effect of noise, fault resistance value, and sampling rate on the performance of the proposed framework. The effectiveness of the proposed framework is validated through a comprehensive study on the IEEE 118-bus system.

1. Introduction

Complex cyber-physical power systems contain a numerous number of elements such as generation units, bus bars, transmission lines, and loads, which are protected by circuit breakers and protective relays. When a fault happens in an element of the system, a large amount of alarms can potentially be generated by means of the protecting devices to be sent to the supervisory control and data acquisition (SCADA) system [1]. However, it is often difficult to manage the received alarms by means of the SCADA for the sake of fault diagnosis in cyber-physical power systems. Furthermore, the large amount of data measurements collected by means of the sparse measuring devices such as phasor measuring units (PMUs) in SCADA systems makes it even more challenging for the successful implementation of diagnostic frameworks for cyber-physical power systems [2]. Therefore, it is of paramount importance to develop an efficient diagnostic method that can cope with the large amount of data [3].

Data-driven methods have been widely used for fault diagnosis [4]. One of the major challenges in the design of data-driven diagnostic frameworks goes back to the extraction of the most informative features from the numerous number of collected features by means of the SCADA system [5]. This issue, however, can be addressed by resorting to feature selection techniques [6]. Feature selection could be refer to as the process of selecting an informative and relevant subset of the original features [7]. These techniques can generally be divided into three major categories including filters, wrappers, and embedded techniques [8]. Filters makes use of the developed tools for measuring the

mutual information, distances, dependencies, and consistencies in order to extract a set of reliable features [9]. Wrappers, however, are generally constructed based on a classification model and take the classification accuracy as a measure in extraction of a subset of features that leads to the best classification accuracy [10]. In embedded techniques, the learning and selection processes are combined and in contrast to the filter or wrapper techniques, these two processes cannot be performed separately [11]. Other than feature selection, developing a diagnostic system that is free of the type of the data distribution is also important for the generalization purposes. To this end, generative adversarial networks (GANs) have been extensively used in the design of data-driven diagnostic frameworks due to their capability of generating the true data distribution from random distributions [12].

GANs are known for their capability in dealing with unbalanced classification by generating realistic looking data samples. They have also found application in data augmentation based on the artificial data which are very similar to the original data samples. Thanks to their advantages, GANs have been widely used in many applications including fault diagnosis [13]. For instance, an auxiliary classifier GAN (ACGAN) has been proposed in [14] to construct an augmentation mechanism for the sake of fault diagnosis. A deep GAN model has been proposed in [15] to deal with the imbalance data used for fault diagnosis. In [16], the authors have used the infrared thermography and infrared image processing in order to generate a useful set of features to evaluate the working condition of angle grinders through a method called BCAoMID-F. The implementation of GANs for self-supervised [17], semi-supervised [18], and unsupervised [12] fault diagnosis schemes have also been extensively studied. In [19], the authors adapt a framework to generate knockoffs, which are random variables that mimic the correlation structure within the existing set of variables in a way that provides a mechanism for the accurate control of the false discovery rate. Even though this technique shows encouraging results, however, it only adapts to the Gaussian distributions. Thanks to the promising results of GANs, the results of this work has been recently extended to a method called knockoffGAN (kGAN), in which the knockoffs are generated without any assumption on the distribution of variables [20]. It is also proposed that by resorting to statistical measures based on the attained coefficients through the Lasso regression, one can employ the feature selection task based on the generated knockoffs. However, a poorly chosen statistic can yield to unreliable results. Therefore, this work aims to propose a diagnostic model based on the knockoffs to benefit from their advantages, while ensures a reliable feature selection for the sake of fault diagnosis.

This work puts forward a novel framework for fault diagnosis in cyber-physical power systems. In contrast to the model-based techniques that rely on the explicit model of the system [21], the proposed framework in this work is data-driven and benefits from the generated knockoffs by means of the kGAN. We consider three different modules in the proposed framework to deal with the large amount of data collected from different spots in the network of the system. Specifically, it is proposed to collect voltage, frequency, and phase-angle features from each bus in the system. After normalization, we make use of a feature selection module in order to construct a subset of the most informative features from the original set of features. In this regard, we make use of the well-known techniques including infinite feature selection (InfFS) [22], mutual information feature selection (MutInfFS) [23], minimum redundancy maximum relevance feature selection (mRMR) [24], and relief feature selection (Relief) [25]. We then run the kGAN technique in the next module of the proposed diagnostic framework, where the selected features from the previous module are used as basis for the generation of knockoffs. The generated knockoffs of the selected features are then fed into the next module, that is the classification module, where two different classification models including k-nearest neighbour (kNN) and support vector machine (SVM) are used to diagnose different types of faults. The main contribution of this work relies on the design of a novel framework that involves multiple modules including feature selection, kGAN, and classification. This is proposed to benefit the most from the advantages of the generative adversarial networks in the extraction of

knockoffs. The generated knockoffs are free of the distribution of data and can be generated in a way that controls the false discovery rate of the selected variables. Compared with the scenario, in which no knockoffs are generated from the selected features, the attained results denote the superiority of the proposed framework for classification tasks. We implement three different types of faults on the IEEE 118-bus system and investigate the effect of noise, fault resistance value, and sampling rate on the performance of the proposed framework through a very comprehensive analysis of the attained results.

The rest of this paper is organized as follows. We review the literature of GANs in Section 2. The generation of knockoffs is presented in Section 3. Simulation results and analysis of the attained results are represented in Section 4 and concluding remarks are given in Section 5.

2. Literature Review

It is well-studied that GANs consist of two models called generator and discriminator, which are typically implemented by neural networks. The generator model aims to learn the distribution of true examples in order to generate new data samples. The discriminator model, however, aims to discriminate the generated data examples by means of the generator from the true data examples [26]. GANs are constructed based on the generative algorithms, which are a category of machine learning algorithms alongside the discriminative algorithms. The generative algorithms make use of a fully probabilistic model of the observed data and can be categorized into two classes including explicit density model and implicit density model. The former model is based on the distribution of data and tries to train the model either based on the true examples of the distribution or by fitting the distribution parameters. Techniques based on the maximum likelihood estimation, approximate inference [27], and Markov chain [28] are used in training of the explicit models. The implicit models, however, do not rely on the direct estimation or fitting of the distribution parameters. Without any explicit hypothesis, these models generate data samples from a distribution to modify the existing model. The training is typically based on the ancestral sampling [29].

In this regard, different representative variants of GANs have been recently developed for different applications. For instance, InfoGAN [30] in contrast to the typical GAN that makes use of a single unconstructed noise signal, decomposes the noise signal into two parts and tries to derive a lower bound of the mutual information objective for an efficient optimization. Some variants of the InfoGAN including causal InfoGAN [31] and semi-supervised InfoGAN (ss-InfoGAN) [32] have been recently developed. GANs are also extended to the case, in which some extra conditions are assigned to the generator and discriminator models. This model is called conditional GAN (cGAN) [33] and can generate data samples that are conditioned on the class labels [34,35]. For image-to-image translation tasks, where the aim is to learn a mapping from an input image to an output image, cycle-consistent GANs (CycleGAN) have been developed to deal with the issue of unpaired data samples [36]. DualGAN [37] has the same structure as that of the CycleGAN; however, its loss function is supported by the Wasserstein GAN (WGAN) [38]. In contrast to the original GAN, in which the discriminator is used for a binary classification task, the discriminator in WGAN is applied to a regression task in order to estimate the Wasserstein distance. This idea, however, requires the discriminator to be K-Lipschitz constrained. In [39], a method called Wasserstein-divergence (W-div) is proposed to relax the WGAN Lipschitz constraints, where it was then used in WGAN-div to approximate the W-div based on an optimization scheme. Same as the WGAN, loss sensitive GAN (LS-GAN) has also Lipschitz constraints, where the given distribution is assumed to belong to a set of Lipschitz densities with a compact support [40]. These variant models of GANs are trained based on different training structures.

The original GAN is developed based on the multilayer perceptron (MLP). Specifically, the generator and discriminator are MLP models, which can only be used for small-sized datasets and have no good generalization capability to deal with complex images [41].

Laplacian GAN (LAPGAN) [42] has been proposed for higher resolution images and makes use of a cascade of convolutional neural networks (CNN) in a Laplacian framework. In the framework of general GAN model, SinGAN [43] and InGAN [44] have also been proposed to learn a generative model based on a single natural image. The next structure is deep convolutional GAN (DCGAN), where in contrast to the original GAN that makes use of the MLP models, are based on the deep convolutional neural networks (DCNNs) [45]. Progressive GAN (PGGAN) [46] is another category of GAN models, in which the progressive neural networks are used in order to grow the generator and discriminator models progressively. Self-attention GAN (SAGAN) [47] is also another developed structure that utilizes the spectral normalization for generator and discriminator models so as to improve the training dynamics. BigGAN [48] is a recently-developed structure which is similar to the SAGAN, however, it is more scalable. Furthermore, StyleGAN [49] is known for its high-quality generator model in generation of face images. Other structures based on the autoencoders [50], encoders [51], multi-discriminator learning [52], multi-generator learning [53], and multi-GAN learning [54] have also been recently developed for GAN models.

3. Knockoff Generation

The general framework of the proposed method has been illustrated in Figure 1. As it can be observed from this figure, the proposed framework contains multiple modules including data acquisition, feature selection, kGAN, and decision making. We make use of sparse data measuring devices in order to collect voltage, frequency, and phase-angle measurements form each bus in the distributed power system. The collected data measurements are then fed into a feature selection module, in which multiple state-of-the-art techniques including InfFS, Relief, MutInfFS, and mRMR have been implemented in order to extract the most informative features from the original set of features. The extracted features are then fed into the kGAN module, where the selected features are used as input and a corresponding set of random variables called knockoffs are outputs of the module. The generated knockoffs are then fed into the decision making module, where the kNN and SVM classification models have been used in order to diagnose different types of faults.
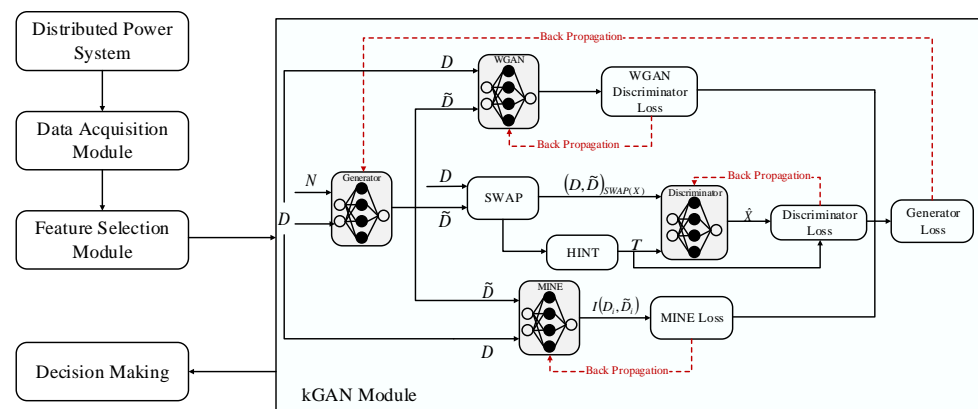
Figure 1. The general framework of the proposed diagnostic method.

Assume that the set of features is denoted by $\mathcal{D}$ and its dimension is d. Suppose that the set of labels is denoted by $\mathcal{C}$ and $\mathcal{D} = \{D_1, \ldots, D_d\}$ and C are random variables. Then, the concept of a null set can be defined as follows [55].

Definition 1. *A variable D_j is null if and only if C is independent of D_j conditional on $\{D_j : i \neq j\}$.*

The set of all null variables is shown by $\mathcal{K}$. In order to select the set of most informative features while controlling the false discovery rate, suppose that the set of selected features is denoted by $\hat{\mathcal{X}} \subset \{1, \ldots, d\}$. The false discovery rate can then be defined as follows:

$$FDR = \mathbb{E}\left[\frac{|\hat{\mathcal{X}} \cap \mathcal{K}|}{|\hat{\mathcal{X}}|}\right]. \tag{1}$$

Based on the given notations, the definition of the knockoffs can be given as follows [55].

Definition 2. *A knockoff for the variable $\mathcal{D}$ is a random variable denoted by $\tilde{\mathcal{D}} \in \mathcal{D}$ that satisfies the following constraints:*

$$\left(\mathcal{D}, \tilde{\mathcal{D}}\right) \stackrel{d}{=} \left(\mathcal{D}, \tilde{\mathcal{D}}\right)_{swapX} \tag{2}$$

$$\tilde{D} \perp C|D \tag{3}$$

where $X \in \{1, \ldots, d\}$ and $(.,.)_{swap(X)}$ is used to show the vector that can be obtained by swapping the ith component with the $(i + d)$th component and $\stackrel{d}{=}$ denotes the equality on distribution.

In order to make use of the generated knockoffs for the sake of feature selection, it is required to define a feature statistic $\mathcal{F}_j$ that only relies on $\mathcal{D}$, $\tilde{\mathcal{D}}$, and $\mathcal{C}$. This statistic is defined as $\mathcal{F}_j = f_j((\mathcal{D}, \tilde{\mathcal{D}}), \mathcal{C})$ for $f_j \in \mathbb{R}$. The f_j function is required to satisfy the following constraint:

$$\mathcal{F}_j([\mathcal{D}, \tilde{\mathcal{D}}]_{swap(X)}, \mathcal{C}) = \begin{cases} -f_j([\mathcal{D}, \tilde{\mathcal{D}}], \mathcal{C}), & j \notin X \\ f_j([\mathcal{D}, \tilde{\mathcal{D}}], \mathcal{C}), & \text{otherwise.} \end{cases} \tag{4}$$

In order to utilize the above statistic, one way is to resort to the LASSO coefficients in order to regress on the augmented set of knockoffs-feature. Denoting the LASSO coefficients by $w_1, \ldots, w_{2d}$, one can define the LASSO coefficient difference as follows:

$$\mathcal{F}_j = |w_j| - |w_{j+d}|. \tag{5}$$

Then, based on the given statistic and the definition of knockoffs, the following theorem can be given for the sake of feature selection [55].

Theorem 1. *Suppose that $q \in [0, 1]$. Given the statistics $\mathcal{F}_1, \ldots, \mathcal{F}_d$, define:*

$$\tau = \min\left\{t > 0 : \frac{1 + |\{j : \mathcal{F}_j \leq -t\}|}{|j : \mathcal{F}_j \geq t|} \leq q\right\}. \tag{6}$$

Then, the selection of variables $\hat{\mathcal{X}} = \{j : \mathcal{F}_j \geq \tau\}$ will lead to the control of false discovery rate at level q.

In order to satisfy the given constraints in (2), a modified GAN model, called kGAN, has been used to generate knockoffs without any assumption on the distribution of data. The kGAN module has been illustrated in Figure 1.

As it can be observed from this figure, the kGAN module contains a generator network, denoted by $\mathcal{G}$, that is a function that satisfies $\mathcal{G}(.,., \xi) : \mathcal{D} \times [0, 1]^c \rightarrow \mathcal{D}$, where its parameters are shown by ξ and takes a random realization of $\mathcal{D}$ and random noise $n \sim \mathcal{U}([0, 1]^c)$ as input and outputs the set of knockoffs $\tilde{\mathcal{D}}$.

The discriminator network is designed so as to deal with the given constraint in (2). In this regard, a discriminator network is defined to have a loss which is minimized only for distributions that satisfy the condition given in Equation (2). To this end, the discriminator is denoted by $\mathcal{S}$, which is a function satisfying $\mathcal{S}(., \psi) : \mathcal{D} \times \mathcal{D} \rightarrow [0, 1]^d$, and takes the

swapped sample-knockoff pair $(\mathcal{D}, \tilde{\mathcal{D}})_{swap(X)}$ and its output is a vector in $[0, 1]^d$, where the ith component of the output is denoted by $\mathcal{S}\left((\mathcal{D}, \tilde{\mathcal{D}})\right)_{swap(X)}$ and denotes the probability of $i \in X$. To this end, the loss of the discriminator can be given as follows:

$$\mathcal{L}_{\mathcal{S}} = \sum_{X \in \{0,1\}^d} \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\tilde{\mathcal{D}}}\left[X.\log(\mathcal{S}(\mathcal{D}, \tilde{\mathcal{D}}))_{swap(X)} + (1 - X).\log(1 - \mathcal{S}((\mathcal{D}, \tilde{\mathcal{D}})_{swap(X)}))\right]\right], \quad (7)$$

where '.' denotes the dot product. In order to deal with the computational complexity of this loss function, it is suggested to utilize the stochastic gradient descent algorithm for minibatches of X that are uniformly sampled. Furthermore, a hint vector $\mathcal{T}$ is introduced, which is a random variable to be passed into the discriminator. The introduction of the hint vector involves the sampling of a multivariate Bernoulli random variable $\mathcal{B}$ that takes the value of 1 with the probability of 0.9. Then, given $\mathcal{T}_i = \mathcal{B}_i$ in case that $\mathcal{B}_i = 1$ and $\mathcal{T}_i = 0.5$ if $\mathcal{B}_i = 0$, the discriminator will then aim to predict only values of X for which $\mathcal{B}_i = 0$. To this end, the final loss of the discriminator will be of the following form:

$$\mathcal{L}_{\mathcal{S}} = \sum_{X \in \{0,1\}^d} \mathbb{E}_{\mathcal{D}}\Big[\mathbb{E}_{\tilde{\mathcal{D}}}\big[\mathbb{E}_{\mathcal{T}}[X \odot (1 - \mathcal{B}).\log(\mathcal{S}((\mathcal{D}, \tilde{\mathcal{D}})_{swap(X)}, \mathcal{T})) +$$
$$(1 - \mathcal{T}) \odot (1 - \mathcal{B}).\log(1 - \mathcal{S}((\mathcal{D}, \tilde{\mathcal{D}})_{swap(X)}, \mathcal{T}), \mathcal{T})]\big]\Big], \quad (8)$$

where $\odot$ is the element-wise product.

In order to make the discriminator algorithm more stable, a regularization term of the form of WGAN, denoted by f is added to loss function (8). Therefore, the general loss of the discriminator will be of the following form:

$$\mathcal{L}_{\mathcal{F}} = \mathbb{E}\Big[f(\mathcal{D}) - f(\tilde{\mathcal{D}}) - \eta(\| \triangledown_{\hat{\mathcal{D}}f(\hat{\mathcal{D}})} \|_2 - 1)^2\Big], \quad (9)$$

where $\hat{\mathcal{D}} = \epsilon\mathcal{D} + (1 - \epsilon)\tilde{\mathcal{D}}$ with $\epsilon \sim \mathcal{U}[0, 1]$ and η is parameter to be tuned.

Finally, in order to generate knockoffs that are as independent as possible of the original features, the mutual information neural estimation (MINE) [56] is used to minimize the mutual information between the set of features and their corresponding knockoffs. In this regard, the mutual information between each pair of the feature and knockoff is estimated by means of d neural networks, denoted by $\mathcal{N}^1, \dots, \mathcal{N}^d$ with the set of parameters $\theta_1, \dots, \theta_d$. By considering a trade-off parameter λ, the following loss of estimation is added to the loss of the generator:

$$\mathcal{L}_{\mathcal{P}} = \sum_{j=1}^{d}\left(\sum_{i=1}^{n}(\mathcal{N}_{\theta^j}^j(\mathcal{D}_j^{(i)}, \tilde{\mathcal{D}}_j^{(i)})) - \log(\sum_{i=1}^{n}\exp(\mathcal{N}_{\theta_j}^j(\mathcal{D}_j^{k(i)}, \tilde{\mathcal{D}}_j^{(i)})))\right), \quad (10)$$

in which k is supposed to be a permutation of $[n]^2$ and superscript (i) is used to demonstrate the ith sample. Based on the discussion in this section, the general loss of the proposed method is defined as follows:

$$\min_{\mathcal{G}}\left(\max_{\mathcal{S}}(\mathcal{L}_{\mathcal{S}}) + \lambda\max_{\mathcal{P}}(\mathcal{L}_{\mathcal{P}}) + \mu\max_{f}(\mathcal{L}_f)\right), \quad (11)$$

where μ is a parameter to be tuned.

4. Simulation Results

In this section, we firstly introduce the IEEE 118-bus power system, and, then, we discuss the types of faults and the generated datasets, and finally, we present the results of the proposed diagnostic framework.

As mentioned earlier, we aim to diagnose different types of faults on the IEEE 118-bus system.This system contains 118 buses, 91 loads, and 19 generation buses. In this work,

we simulate three different types of faults on this system. These faults are called load loss (LL), generator outage (G), and generator ground (GG). Together with the normal operational state of the system, there will be four types of states to be diagnosed. As for the simulation of the 'LL'/'G' faults, we have disconnected the corresponding load/generation unit from its corresponding bus for a short period of time. As for the 'GG' faults, we have simulated a three-phase short-circuit fault between the generation units and ground. We have simulated 31 'LL' faults by disconnecting each single of them from the corresponding bus. In the same way, 19 'G' faults and 19 'GG' faults are simulated. By adding the normal operational state of the system to the above-mentioned simulated faults, there exist 70 classes of operational states to be diagnosed. For each class or operational state, we have collected 500 samples from the sample that fault has been injected into either loads or generators, to the sample that fault has been cleared. Furthermore, voltage, frequency, and phase-angle features are collected from each bus of the system. In Figure 2, the voltage, frequency, and phase angle measurements collected from the first bus of the system in presence of an LL fault on bus #1 are illustrated. The fault has been injected at $t = 1$ second and the simulation period is set to five seconds. As there are 118 buses in the system and three types of features are collected from each of them, there exist a total number of 354 features to be used in construction of datasets.
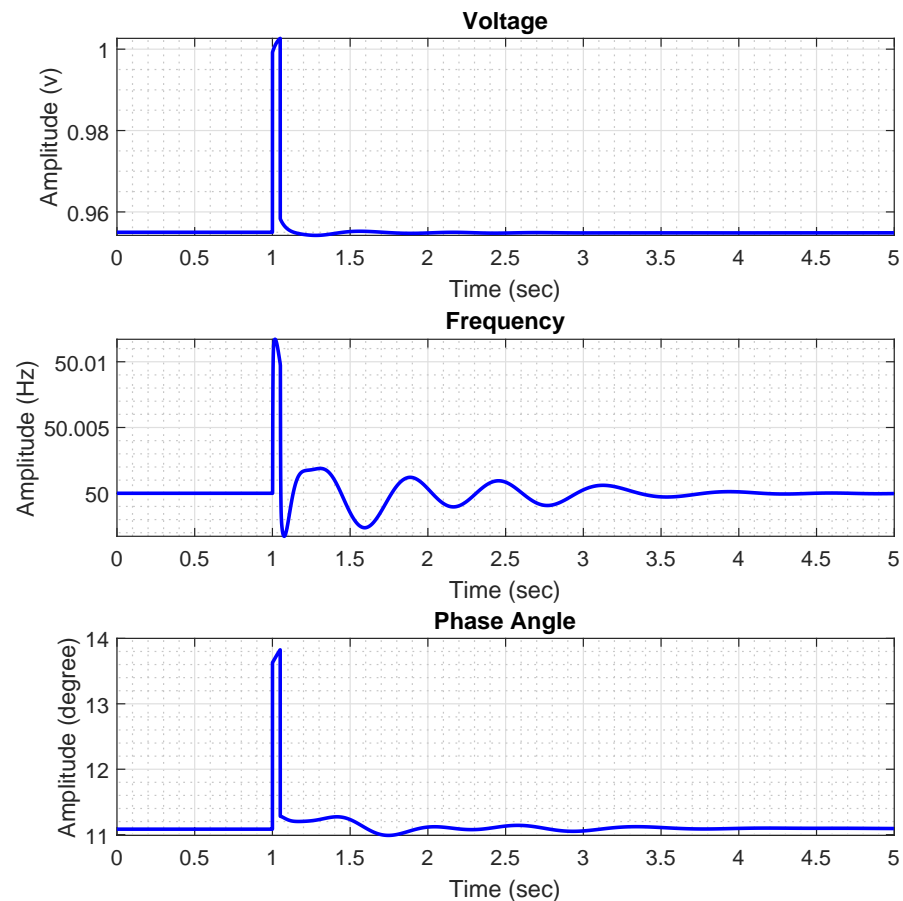
Figure 2. The collected voltage, frequency, and phase angle measurements following an LL fault on bus #1 at $t = 1$ second of the simulation.

In order to study the effect of fault resistance (FR), signal to noise ratio (SNR), and sampling rate (SR) on the performance of the proposed diagnostic framework, 12 different datasets have been created. In this regard, two different SR values have been considered which are 20 KHz to 10 KHz. The FR values are supposed to be 1 Ω and 10 Ω, and the SNR values have been selected to be 50 dB, 40 dB, and 30 dB. By making a combination of the FR, SNR, and SR, 12 datasets $\{\mathcal{A}_1, \ldots, \mathcal{A}_{12}\}$ are generated.

Following the given description in Section 1, we consider two different scenarios and compare them with a baseline. Our baseline is the case, in which the raw data measurements are directly and without any processing fed into the classification models. Furthermore, in order to investigate the effectiveness of the proposed framework, we compare it with a scenario, in which the raw data measurements are firstly fed into the feature selection module, and, then, the selected features are directly fed into the classification module [6]. This is the first scenario ('S#1'). In the second scenario, which is the proposed diagnostic framework in this study, we propose to generate the knockoffs of the selected features by means of the kGAN module, and, then, set these knockoffs as inputs to the classification models. Therefore, in the second scenario ('S#2'), the raw data measurements are firstly fed into the feature selection module, where the selected features are further processed by the kGAN module and the generated knockoffs are fed into the classification models. As for the feature selection module, we resort to four well-known feature selection techniques including InfFS, MutInfFS, mRMR, and Relief. In the feature selection module, in order to find the best number of features to be selected, we start with two features and increase the number of features up to the value, for which no significant performance improvement can be observed for each classification model. The performance of each classification model has been reported based on the F-Measure.

We start with the kNN classification model, where the attained results by means of this classifier are illustrated in Figure 3. It worth noting that each classification model is validated through a 10-fold cross-validation manner. As there are 12 datasets and a 10-fold cross-validation is performed, there are 120 F-Measure values for each experiment. In Figure 3, we have reported the results for all datasets for the baseline and the aforementioned feature selection techniques w.r.t. scenarios 'S#1' and 'S#2'. As it can be observed from this figure, both scenarios have successfully improved the results compared with the baseline case. However, the second scenario (our proposed method) outperforms the first scenario [6] in all experiments, despite of the type of the feature selection technique. As for the first scenario, the attained results denote that InfFS leads to the highest average F-Measure value, which is then followed by mRMR, Relief, and MutInfFS. As for the second scenario, InfFS, MutInfFS, and mRMR show almost the same performance in terms of the average F-measure value, while Relief is the worst technique. Another worthwhile point to be mentioned is that the attained results for the second scenario show lower variation in the attained F-Measure values compared with the first scenario in the case of the InfFS, Relief, and MutInfFS, showing its robustness in dealing with different datasets. We have summarized the attained results of the kNN classification model in Table 1 w.r.t. each dataset in order to check for the effect of noise, fault resistance, and sampling rate on the performance of this classifier.
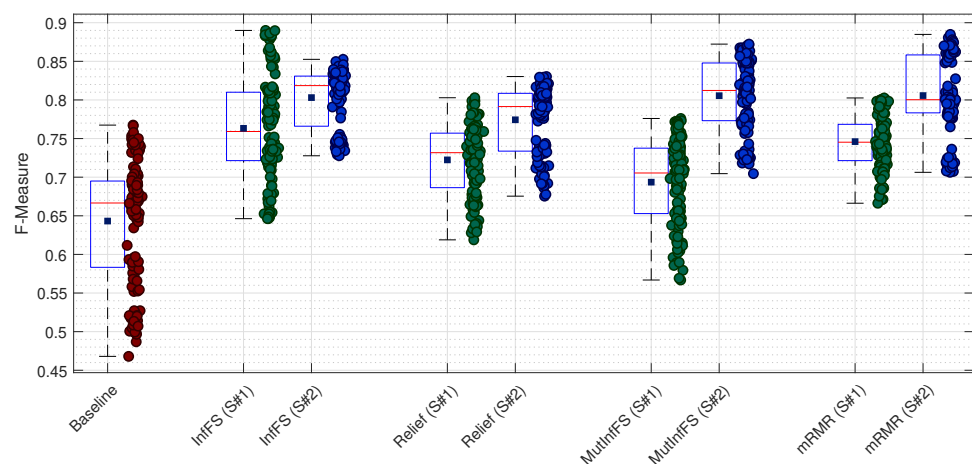
Figure 3. The attained F-Measure values for the kNN classification model w.r.t. each feature selection technique.
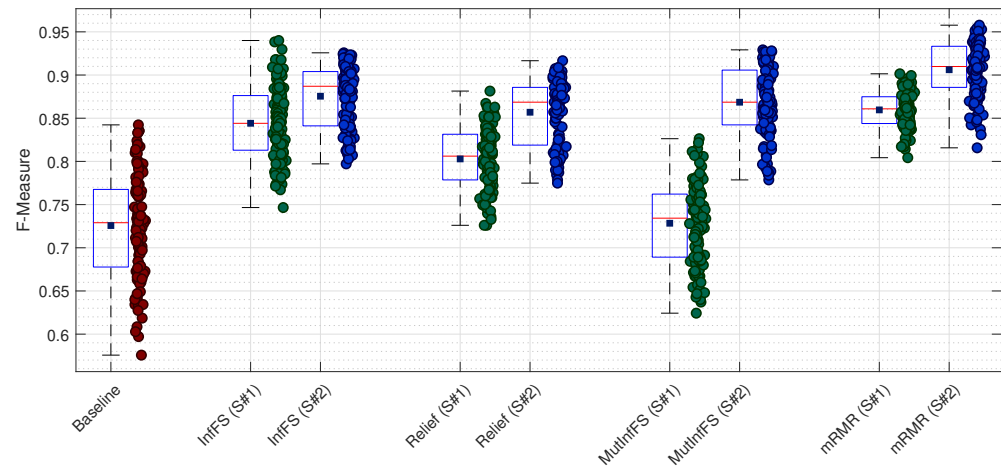
Table 1. The attained F-Measure (FM) values by means of the kNN classification model w.r.t. each dataset.

Dataset	Baseline	InfFS		Relief		MutInfFS		mRMR	
		S#1	S#2	S#1	S#2	S#1	S#2	S#1	S#2
$\mathcal{A}_1$	0.7379	0.8527	0.8312	0.7718	0.7963	0.7479	0.8560	0.7758	0.8647
$\mathcal{A}_2$	0.6809	0.7847	0.8209	0.7386	0.7858	0.7111	0.8478	0.7468	0.8563
$\mathcal{A}_3$	0.6519	0.7149	0.7966	0.7169	0.6973	0.6962	0.7968	0.7279	0.7818
$\mathcal{A}_4$	0.7522	0.8828	0.8408	0.7928	0.8052	0.7695	0.8632	0.7956	0.8733
$\mathcal{A}_5$	0.7006	0.8020	0.8305	0.7587	0.7887	0.7329	0.8511	0.7599	0.8639
$\mathcal{A}_6$	0.6620	0.7418	0.7415	0.7388	0.7019	0.7113	0.7757	0.7432	0.7885
$\mathcal{A}_7$	0.6691	0.7754	0.8244	0.7217	0.8149	0.6571	0.8155	0.7658	0.8046
$\mathcal{A}_8$	0.5668	0.7274	0.8183	0.6705	0.8070	0.5905	0.8051	0.7157	0.7957
$\mathcal{A}_9$	0.4973	0.6535	0.7406	0.6371	0.7331	0.6180	0.7236	0.6828	0.7141
$\mathcal{A}_{10}$	0.6884	0.8125	0.8298	0.7544	0.8194	0.7515	0.8211	0.7910	0.8077
$\mathcal{A}_{11}$	0.5925	0.7317	0.8181	0.6981	0.8106	0.6880	0.8084	0.7391	0.7956
$\mathcal{A}_{12}$	0.5187	0.6799	0.7438	0.6691	0.7343	0.6485	0.7287	0.7090	0.7217
Avg.	0.6432	0.7633	0.8030	0.7224	0.7743	0.6935	0.8055	0.7460	0.8057

As it was mentioned earlier, in the first scenario, InfFS leads to the highest average F-Measure that is 0.7633, which is then followed by mRMR (0.7460), Relief (0.7224), and MutInfFS (0.6935). As for the second scenario, the best performance has been achieved by means of the mRMR (0.8057), which is then followed by MutInfFS (0.8055), InfFS (0.8030), and Relief (0.7743). In order to check the effect of FR on the performance of the proposed technique, we resort to the results of datasets $\{\mathcal{A}_1, \ldots, \mathcal{A}_6\}$ for which the FR is 1 Ω, and compare them with those of datasets $\{\mathcal{A}_7, \ldots, \mathcal{A}_{12}\}$, for which the FR is 10 Ω. For datasets with FR = 1 Ω, the average F-Measure for the second scenario is 0.8362, whereas it is 0.8331 for datasets with FR = 10 Ω. In the same vein, the average F-Measure for the first scenario are 0.7888 and 0.7718, respectively. Therefore, the attained results denote that the performance of the proposed method is not significantly affected by the value of FR. In order to check for the effect of noise, we regroup datasets into three groups of $\{\mathcal{A}_1, \mathcal{A}_4, \mathcal{A}_7, \mathcal{A}_{10}\}$ with SNR = 50 dB, $\{\mathcal{A}_2, \mathcal{A}_5, \mathcal{A}_8, \mathcal{A}_{11}\}$ with SNR = 40 dB, and $\{\mathcal{A}_3, \mathcal{A}_6, \mathcal{A}_9, \mathcal{A}_{12}\}$ with SNR = 30 dB. For the second scenario, the average F-Measure values for the aforementioned three groups are 0.8291, 0.8190, and 0.7433, respectively. In the case of the first scenario, the attained results are 0.7762, 0.7247, and 0.6931, respectively. The attained results for both scenarios denote the superiority of the proposed method (the second scenario) in dealing with noisy data measurements compared with the first scenario. Finally, we aim to check for the effect of SR on the proposed method by regrouping the given datasets into two groups $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_7, \mathcal{A}_8, \mathcal{A}_9\}$, for which the sampling rate is 10 KHz, and $\{\mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_6, \mathcal{A}_{10}, \mathcal{A}_{11}, \mathcal{A}_{12}\}$, for which the sampling rate is 20 KHz. The attained F-Measure values for the first scenario are 0.7167 and 0.7459 w.r.t. the aforementioned group of datasets, respectively. The average F-Measure values for the second scenario are 0.7958 and 0.7985, respectively. The attained results, on one hand, denote the superiority of the second scenario in comparison with the first scenario in dealing with datasets with different SR values. On the other hand, there is no significant changes for the second scenario when the SR decreases from 20 KHz to 10 KHz, denoting its robustness against the sampling rate issues.

We repeat the same experiments for the SVM classification model. The attained results are represented in Figure 4. As it can be observed, the baseline shows much variation in terms of F-Measure in dealing with different datasets. However, it almost shows the same average F-Measure value in comparison with the results of the MutInfFS used in the first scenario, but lower average F-Measure values compared with the second scenario. By comparing the first and second scenarios, the attained results, on one hand, show the superiority of the second scenario for each feature selection technique. On the other hand,

the second scenario leads to lower variation in F-Measure values when InfFS, Relief, and MutInfFS techniques are applied. The attained results for the SVM classification model are summarized in Table 2.

Figure 4. The attained F-measure values by means of the SVM classification model w.r.t. each dataset.

The collected results in Table 2 denote that mRMR leads to the best performance in both scenarios, which is then followed by InfFS, Relief, and MutInfFS in the first scenario, and InfFS, MutInfFS, and Relief in the second scenario. In order to investigate the effect of FR value, we regroup datasets as before, where the FR value is 1 Ω for the first group and 10 Ω for the second group. The average F-Measure value for the first group has been obtained as 0.8365 for the first scenario, whereas it is 0.7840 for the second group of datasets. In the same vein, the attained results for the second scenario are 0.8909 and 0.8627, respectively. The attained results of this experiment, on one hand, denote that the second scenario outperforms the first scenario. On the other hand, they verify the more robust performance of the second scenario compared with the first scenario against the changes in the FR value. In order to check for the effect of noise on the performance of the proposed scheme, same as what was done for the kNN classification model, we regroup datasets into three groups w.r.t. SNR = 50 dB, SNR = 40 dB, and SNR = 30 dB. For the first scenario, the average F-Measure values for each group of datasets can be computed as 0.8335, 0.8044, and 0.7928, respectively. As for the second scenario, the attained results are 0.9022, 0.8949, and 0.8331, respectively. As it can be observed from the attained average values of F-Measure for both scenarios, the second scenario outperforms the first one in dealing with noisy measurements. Finally, we regroup datasets into two groups based on the SR values, where SR = 10 KHz for the first group and SR = 20 KHz for the second group. The average F-Measure values for these two groups are 0.8048 and 0.8157, respectively, for the first scenario, whereas the average values for the second scenario are 0.8756 and 0.8779, respectively. The attained results verify the superiority of the second scenario over the first scenario.

Following the presented results for the kNN and SVM classification models, some general remarks can be made. Generally speaking, SVM classification model has outperformed the kNN by considering both scenarios plus the baseline, where the average F-Measure for the SVM is 0.8304, whereas it is 0.7508 for the kNN. The results of of the second scenario suggest that mRMR shows the best performance in dealing with the aforementioned datasets, however, its combination with the SVM classification model leads to a better combination for the sake of fault diagnosis in the IEEE 118-bus system. Furthermore, the results of both classification models verified the superiority of the proposed technique in comparison with the baseline and the first scenario.

Table 2. The attained F-Measure (FM) values by means of the SVM classification model w.r.t. each dataset.

Dataset	Baseline	InfFS		Relief		MutInfFS		mRMR	
		S#1	S#2	S#1	S#2	S#1	S#2	S#1	S#2
$\mathcal{A}_1$	0.7970	0.9143	0.9156	0.8258	0.8936	0.7776	0.9159	0.8696	0.9413
$\mathcal{A}_2$	0.7407	0.8898	0.9060	0.8157	0.8877	0.7520	0.9061	0.8600	0.9344
$\mathcal{A}_3$	0.6799	0.8768	0.8414	0.7998	0.8084	0.7351	0.8480	0.8516	0.8840
$\mathcal{A}_4$	0.8265	0.8780	0.9091	0.8571	0.8973	0.8007	0.9195	0.8864	0.9407
$\mathcal{A}_5$	0.7749	0.8527	0.9003	0.8330	0.8901	0.7671	0.9090	0.8701	0.9361
$\mathcal{A}_6$	0.7211	0.8481	0.8418	0.8254	0.8219	0.7560	0.8456	0.8611	0.8867
$\mathcal{A}_7$	0.7341	0.8259	0.8886	0.7893	0.8694	0.7159	0.8697	0.8546	0.9156
$\mathcal{A}_8$	0.6706	0.7977	0.8839	0.7611	0.8640	0.6746	0.8867	0.8377	0.9050
$\mathcal{A}_9$	0.6187	0.7856	0.8170	0.7518	0.8000	0.6549	0.8019	0.8259	0.8513
$\mathcal{A}_{10}$	0.7697	0.8373	0.8934	0.8147	0.8766	0.7353	0.8742	0.8823	0.9157
$\mathcal{A}_{11}$	0.7143	0.8206	0.8865	0.7839	0.8668	0.6954	0.8656	0.8595	0.9085
$\mathcal{A}_{12}$	0.6611	0.8030	0.8221	0.7767	0.8054	0.6762	0.8000	0.8568	0.8543
Avg.	0.7257	0.8441	0.8755	0.8029	0.8569	0.7284	0.8685	0.8656	0.9061

The main advantage of the proposed diagnostic scheme is that it is data-driven, and, therefore, there is no need to have knowledge about the explicit model of the system. Further to this, we have proposed the use of kGAN module in order to generate a set of informative features from the selected measurements. This module can generate this set of features despite of the type of the distribution of data. Furthermore, the proposed diagnostic scheme can be easily extended to involve semi-supervised and unsupervised feature selection techniques in order to benefit from their advantages. The main drawback of the proposed framework goes back to the fact that this technique is offline and cannot be used for a real-time implementation.

5. Conclusions

This work is devoted to the design of a novel diagnostic framework for distributed power systems. The proposed diagnostic framework involves three modules including feature selection, kGAN, and decision making for the sake of fault diagnosis. It makes use of the voltage, frequency, and phase angle measurements collected by means of sparse measuring devices attached to each bus of the power system. The collected data measurements are firstly fed into the feature selection module in order to find the most informative features. The selected feature are then further processed by feeding them into the kGAN module, where a technique based on the GANs has been used in order to generate the corresponding set of knockoffs of the selected features. Generated knockoff are finally fed into the decision making module, where two different classification models are utilized to diagnose different types of faults. A very comprehensive comparative study has been provided in order to investigate the performance of the proposed method in dealing with noisy data measurements, datasets with high fault resistance values, and datasets with different sampling rate values. The attained results verify the applicability, effectiveness, and superiority of the proposed framework in comparison with a literature work. Verifying the results of this work for other large-scale power systems by making use of other state-of-the-art feature selection techniques and classification models could be investigated in a future work.

Author Contributions: Conceptualization, H.H. and R.R.-F.; methodology, H.H. and R.R.-F.; software, H.H.; validation, H.H.; formal analysis, H.H. and R.R.-F. and M.S. and V.P.; investigation, H.H. and R.R.-F. and M.S. and V.P.; resources, R.R.-F.; data curation, H.H.; writing—original draft preparation, H.H.; writing—review and editing, H.H. and R.R.-F. and and V.P.; visualization, H.H.; supervision,

R.R.-F. and M.S.; project administration, R.R.-F. and M.S.; funding acquisition, R.R.-F. and M.S. and V.P. All authors have read and agreed to the published version of the manuscript.

References

1. Hassani, H.; Razavi-Far, R.; Saif, M. Fault location in smart grids through multicriteria analysis of group decision support systems. *IEEE Trans. Ind. Inform.* **2020**, *16*, 7318–7327. [CrossRef]
2. Wang, Q.; Yu, Y.; Ahmed, H.O.A.; Darwish, M.; Nandi, A.K. Fault Detection and Classification in MMC-HVDC Systems Using Learning Methods. *Sensors* **2020**, *20*, 4438. [CrossRef] [PubMed]
3. Hassani, H.; Razavi-Far, R.; Saif, M.; Capolino, G.A. Regression Models With Graph-Regularization Learning Algorithms for Accurate Fault Location in Smart Grids. *IEEE Syst. J.* **2021**, *15*, 2012–2023. [CrossRef]
4. Razavi-Far, R.; Palade, V.; Zio, E. Optimal detection of new classes of faults by an invasive weed optimization method. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 91–98.
5. Kari, T.; Gao, W.; Zhao, D.; Abiderexiti, K.; Mo, W.; Wang, Y.; Luan, L. Hybrid feature selection approach for power transformer fault diagnosis based on support vector machine and genetic algorithm. *IET Gener. Transm. Distrib.* **2018**, *12*, 5672–5680. [CrossRef]
6. Hassani, H.; Hallaji, E.; Razavi-Far, R.; Saif, M. Unsupervised concrete feature selection based on mutual information for diagnosing faults and cyber-attacks in power systems. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104150. [CrossRef]
7. Cui, Q.; El-Arroudi, K.; Weng, Y. A feature selection method for high impedance fault detection. *IEEE Trans. Power Deliv.* **2019**, *34*, 1203–1215. [CrossRef]
8. Gangavarapu, T.; Patil, N. A novel filter–wrapper hybrid greedy ensemble approach optimized using the genetic algorithm to reduce the dimensionality of high-dimensional biomedical datasets. *Appl. Soft Comput.* **2019**, *81*, 105538. [CrossRef]
9. Cekik, R.; Uysal, A.K. A novel filter feature selection method using rough set for short text data. *Expert Syst. Appl.* **2020**, *160*, 113691. [CrossRef]
10. Nouri-Moghaddam, B.; Ghazanfari, M.; Fathian, M. A novel multi-objective forest optimization algorithm for wrapper feature selection. *Expert Syst. Appl.* **2021**, *175*, 114737. [CrossRef]
11. Fu, Y.; Liu, X.; Sarkar, S.; Wu, T. Gaussian mixture model with feature selection: An embedded approach. *Comput. Ind. Eng.* **2021**, *152*, 107000. [CrossRef]
12. Liu, H.; Zhou, J.; Xu, Y.; Zheng, Y.; Peng, X.; Jiang, W. Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks. *Neurocomputing* **2018**, *315*, 412–424. [CrossRef]
13. Farajzadeh-Zanjani, M.; Hallaji, E.; Razavi-Far, R.; Saif, M. Generative adversarial dimensionality reduction for diagnosing faults and attacks in cyber-physical systems. *Neurocomputing* **2021**, *440*, 101–110. [CrossRef]
14. Shao, S.; Wang, P.; Yan, R. Generative adversarial networks for data augmentation in machine fault diagnosis. *Comput. Ind.* **2019**, *106*, 85–93. [CrossRef]
15. Zhang, W.; Li, X.; Jia, X.D.; Ma, H.; Luo, Z.; Li, X. Machinery fault diagnosis with imbalanced data using deep generative adversarial networks. *Measurement* **2020**, *152*, 107377. [CrossRef]
16. Glowacz, A. Ventilation diagnosis of angle grinder using thermal imaging. *Sensors* **2021**, *21*, 2853. [CrossRef] [PubMed]
17. Zhang, W.; Chen, D.; Kong, Y. Self-Supervised Joint Learning Fault Diagnosis Method Based on Three-Channel Vibration Images. *Sensors* **2021**, *21*, 4774. [CrossRef]
18. Farajzadeh-Zanjani, M.; Hallaji, E.; Razavi-Far, R.; Saif, M.; Parvania, M. Adversarial Semi-Supervised Learning for Diagnosing Faults and Attacks in Power Grids. *IEEE Trans. Smart Grid* **2021**, *12*, 3468–3478. [CrossRef]
19. Barber, R.F.; Candès, E.J. Controlling the false discovery rate via knockoffs. *Ann. Stat.* **2015**, *43*, 2055–2085. [CrossRef]
20. Jordon, J.; Yoon, J.; van der Schaar, M. KnockoffGAN: Generating knockoffs for feature selection using generative adversarial networks. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2018
21. Razavi-Far, R.; Davilu, H.; Palade, V.; Lucas, C. Model-based fault detection and isolation of a steam generator using neuro-fuzzy networks. *Neurocomputing* **2009**, *72*, 2939–2951. [CrossRef]
22. Roffo, G.; Melzi, S.; Cristani, M. Infinite feature selection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–15 December 2015; pp. 4202–4210.

23. Zaffalon, M.; Hutter, M. Robust feature selection using distributions of mutual information. In Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence (UAI-2002), Edmonton, AB, Canada, 1–4 August 2002; pp. 577–584.
24. Li, X.; Xie, S.; Zeng, D.; Wang, Y. Efficient l_0-norm feature selection based on augmented and penalized minimization. *Stat. Med.* **2018**, *37*, 473–486. [CrossRef]
25. Kira, K.; Rendell, L.A. The Feature Selection Problem: Traditional Methods and a New Algorithm. In Proceedings of the AAAI-92, San Jose, CA, USA, 12–16 July 1992; Volume 2, pp. 129–134.
26. Gui, J.; Sun, Z.; Wen, Y.; Tao, D.; Ye, J. A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv* **2020**, arXiv:2001.06937.
27. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In Proceedings of the International Conference on Machine Learning; JMLR, Beijing, China, 21–26 June 2014; pp. 1278–1286.
28. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
29. Bengio, Y.; Yao, L.; Alain, G.; Vincent, P. Generalized denoising auto-encoders as generative models. *arXiv* **2013**, arXiv:1305.6663.
30. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5 December 2016; pp. 2180–2188.
31. Kurutach, T.; Tamar, A.; Yang, G.; Russell, S.; Abbeel, P. Learning plannable representations with causal infogan. *arXiv* **2018**, arXiv:1807.09341.
32. Spurr, A.; Aksan, E.; Hilliges, O. Guiding infogan with semi-supervision. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer: Skopje, Macedonia, 2017; pp. 119–134.
33. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
34. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier gans. In Proceedings of the International Conference on Machine Learning; JMLR, Sydney, Australia, 6–11 August 2017; pp. 2642–2651
35. Nguyen, A.; Clune, J.; Bengio, Y.; Dosovitskiy, A.; Yosinski, J. Plug & play generative networks: Conditional iterative generation of images in latent space. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4467–4477.
36. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
37. Yi, Z.; Zhang, H.; Tan, P.; Gong, M. Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2849–2857.
38. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning; JMLR, Sydney, Australia, 6–11 August 2017; pp. 214–223.
39. Wu, J.; Huang, Z.; Thoma, J.; Acharya, D.; Van Gool, L. Wasserstein divergence for gans. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 653–668.
40. Qi, G.J. Loss-sensitive generative adversarial networks on lipschitz densities. *Int. J. Comput. Vis.* **2020**, *128*, 1118–1140. [CrossRef]
41. Wiatrak, M.; Albrecht, S.V.; Nystrom, A. Stabilizing generative adversarial networks: A survey. *arXiv* **2019**, arXiv:1910.00927.
42. Denton, E.; Chintala, S.; Szlam, A.; Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv* **2015**, arXiv:1506.05751.
43. Shaham, T.R.; Dekel, T.; Michaeli, T. Singan: Learning a generative model from a single natural image. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019; pp. 4570–4580.
44. Shocher, A.; Bagon, S.; Isola, P.; Irani, M. Ingan: Capturing and retargeting the "dna" of a natural image. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019; pp. 4492–4501.
45. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
46. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
47. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the International Conference on Machine Learning; JMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 7354–7363.
48. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv* **2018**, arXiv:1809.11096.
49. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.
50. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial autoencoders. *arXiv* **2015**, arXiv:1511.05644.
51. Donahue, J.; Krähenbühl, P.; Darrell, T. Adversarial feature learning. *arXiv* **2016**, arXiv:1605.09782.
52. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. It takes (only) two: Adversarial generator-encoder networks. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
53. Hoang, Q.; Nguyen, T.D.; Le, T.; Phung, D. Multi-generator generative adversarial nets. *arXiv* **2017**, arXiv:1708.02556.
54. Liu, M.Y.; Tuzel, O. Coupled generative adversarial networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 469–477.

55. Candès, E.J.; Fan, Y.; Janson, L.; Lv, J. *Panning for Gold: Model-Free Knockoffs for High-Dimensional Controlled Variable Selection*; Department of Statistics, Stanford University: Stanford, CA, USA, 2016.

56. Belghazi, M.I.; Baratin, A.; Rajeshwar, S.; Ozair, S.; Bengio, Y.; Courville, A.; Hjelm, D. Mutual information neural estimation. In Proceedings of the International Conference on Machine Learning; JMLR, Stockholm, Sweden, 10–15 July 2018; pp. 531–540.