IEEE Open Journal of Signal Processing

Received 21 August 2020; revised 1 December 2020; accepted 1 December 2020. Date of publication 18 December 2020; date of current version 25 January 2021. The review of this article was arranged by Associate Editor A. Majumdar.

Digital Object Identifier 10.1109/OJSP.2020.3045829

EnerGAN++: A Generative Adversarial Gated Recurrent Network for Robust Energy Disaggregation

MARIA KASELIMI [1], NIKOLAOS DOULAMIS [1] (Member, IEEE), ATHANASIOS VOULODIMOS [2] (Member, IEEE), ANASTASIOS DOULAMIS [1] (Member, IEEE), AND EFTYCHIOS PROTOPAPADAKIS [1]

[1] Department of Electrical and Computer Engineering, National Technical University of Athens, 15780 Zografou, Greece
[2] Department of Informatics and Computer Engineering, University of West Attica, 12243 Athens, Greece

CORRESPONDING AUTHOR: MARIA KASELIMI (e-mail: mkaselimi@mail.ntua.gr)

This work was supported by the EU H2020 project BENEFFICE under Grant Agreement 768774.

ABSTRACT Energy disaggregation, namely the separation of the aggregated household energy consumption signal into its additive sub-components, bears resemblance to the signal (source) separation problem and poses several challenges, not only as an ill-posed problem, but also, due to unsteady appliance signatures, abnormal behaviour that is usually detected in appliances operation and the existence of noise in the aggregated signal. In this paper, we propose EnerGAN++, a model based on Generative Adversarial Networks (GAN) for robust energy disaggregation. We attempt to unify the autoencoder (AE) and GAN architectures into a single framework, in which the autoencoder achieves a non-linear power signal source separation. EnerGAN++ is trained adversarially using a novel discriminator, to enhance robustness to noise. The discriminator performs sequence classification, using a recurrent convolutional neural network to handle the temporal dynamics of an appliance energy consumption time series. In particular, the proposed architecture of the discriminator leverages the ability of Convolutional Neural Networks (CNN) in rapid processing and optimal feature extraction, among with the need to infer the data temporal character and time dependence. Experimental results indicate the proposed method's superiority compared to the current state of the art.

INDEX TERMS Convolutional neural networks, denoising autoencoders, energy disaggregation, generative adversarial networks, non-intrusive load monitoring, recurrent neural networks, robustness to noise, sequence-to-sequence learning.

I. INTRODUCTION

Non-Intrusive Load Monitoring (NILM) or energy disaggregation can be considered as an efficient and cost effective framework to reduce energy consumption [1]. Energy disaggregation uses the aggregate power signal of a household as input to estimate which appliances contribute and to which extent to the aggregate energy consumption signal. To this day, various approaches have been proposed to solve the NILM problem, as presented in Section I-A. Some of the most successful ones exploit deep learning neural network structures for modelling an energy disaggregation problem (e.g. recurrent neural network structures [2]). Nevertheless, there are barriers and limitations that until recently, have not been properly addressed. In particular, the proposed techniques have not been applied successfully across different households and datasets [3]. Thus, it is difficult to create algorithms with a good *generalization ability*. In addition, noisy aggregate energy consumption measurements significantly deteriorate the performance of NILM methods. Usually, the detected appliances have unsteady signatures or present abnormal behavior and additionally, the existence of noisy as well as inadequate datasets deteriorates overall models' performance [4]. The latter issue is more evident in case that deep learning structures are used for energy disaggregation, since a large number of

labeled samples is required during training. However, the collection of labeled training data is an arduous task and therefore the need for training-less solution (unsupervised learning) has recently arisen [5].

In this study, we investigate the ability of generative adversarial networks to create robust appliance power patterns for energy disaggregation in the presence of noise. In this context, the term noise implies that the aggregate energy consumption signals, used during the testing phase of a deep learning NILM model, deviates, sometimes significantly, from the aggregate inputs used during the training phase. Usually, this deviation worsens models performance, as the distinctive characteristics of each appliance are no longer distinguishable in the aggregate signal. Generative adversarial networks (GAN) [6], which is a special category of generative models, can learn data distributions through adversarial learning, even in the presence of noise. Therefore, they are capable of representing noisy time series with high consistency [7]. Given that GAN input is a noise vector, the model is inherently more robust to noisy input signals and less sensitive to inconsistencies or gaps in the training input samples.

Nevertheless, adopting a GAN-inspired NILM model implies that certain adjustments to the conventional GAN structures are required. A GAN is a framework for estimating generative models through an adversarial process, in which simultaneously two models are trained; the *Generator* and the *Discriminator*. The generator maps a latent space to the data space follows a learned distribution and *is optimized, during training, to cheat the discriminator*. The discriminator distinguishes the generated data produced by the generator. Therefore, *it is optimized not to be cheated by the generator*. GANs usually reproduce realistic outputs, after an adversarially learning process, having the same statistics as the labeled data. However, in our NILM context, the purpose is to reproduce the real appliance consumption time series instead of generating a realistic time sequence of the same statistics. This means that we know at advance the desired generated-output value. Thus, stepping backwards, the input vector shouldn't be a completely random generated vector, but an alternative input vector that help GAN to reproduce real consumption outputs of an appliance. Under this scope, the idea of inverse GAN is adopted [8]. In an inverse GAN, real data signals are generated in contrast to traditional GAN structures where realistic, but not real, outputs are produced. Hereby, the aggregated energy consumption is considered as the alternative random noise trigger of the GAN. The aggregated signal composes of the sum of independent energy consumption appliance patterns and therefore, it is considered as the NILM-based GAN noise signal trigger. In addition, labeled appliance energy consumption data sequences are used as input signals to the GAN generator in order to learn to generate almost identical data sequence for an appliance. Thus, our GAN NILM model is trained to produce faithful replica of appliance's energy consumption signals to cheat the discriminator.

In our recent work in [9], a NILM GAN-based model, called EnerGAN, has been introduced for solving the energy disaggregation problem; estimation of appliance energy consumption time-series, triggered by aggregate energy measurements. In the adversarial competitive learning, the EnerGAN NILM model uses as the discriminator component a binary classifier, which fails to model temporal dependencies and recurrent properties that existing inherently in energy consumption signals. Therefore, these models fail to discriminate patterns of abrupt changes in the energy consumption signals. Additionally, the use of a simple discriminator makes the model to be more sensitive to noisy aggregate signal inputs since it can be easily cheated by the generator. To address the aforementioned difficulties, this paper proposes an extension of the previous EnerGAN model, named EnerGAN++ which enriches the discriminator with recurrent properties to increase robustness and precision accuracy in adversarial energy disaggregation modelling.

A. RELATED WORK

1) DEEP LEARNING IN NILM MODELLING

With the rise of deep learning, a new family of methods has been introduced that exploit deep neural network structures to solve the ill-pose problem of NILM. Deep learning techniques have been applied to low frequency NILM since 2015 [10]. Recurrent Neural Netowrks (RNN), and their variants, such as Long Short-Term Memory Networks (LSTM) and Gated Recurrent Units (GRU) have been primarily used, as they are very popular and effective with 1D time series data. Relevant studies have been carried out in the past ([3], [10], [11]). In a previous work of ours, we have also proposed a Bayesian optimized bidirectional LSTM model for NILM [2], whereas in [1] a context-aware LSTM model adaptable to external environmental conditions is presented. Although Convolutional Neural Networks (CNN) are traditionally developed for two-dimensional imagery data ([12]), one-dimensional CNN can be used to model the temporal character of sequential time-series data. Few researches [13] have enriched the CNN-based structures with a recurrent character, such as CNN-LSTM and Recurrent Convolutional Networks. In [14] a causal 1-D convolutional neural network for NILM is proposed. Some researches introduce the concept of data sequences [10] to feed the classic structure with historical past values of power load. Among them, [15] propose a sequence to point CNN architecture, underscoring the importance of sliding windows to handle long-term timeseries. Alternatively, sequence to sequence architectures have also been proposed [16]. Among other deep learning schemes applied for NILM, is worth-remembering to mention denoising autoencoders, that primarily proposed by [17].

Many popular NILM algorithms, such as Non-Negative Matrix Factorization (NMF) [18], hidden Markov models [19], or tensor factorization models [20], [21] have their origins in audio modeling/source separation algorithms, and are all types of generative models that specify how to generate data that fit to a distribution. With the rise of deep learning, a new family of methods, called deep generative models

(DGMs), is formed through the combination of generative models and deep neural networks [22]. Popular DGMs include the Variational Autoencoder (VAE) and GANs. Bellow, we present the relevant related work of the GAN model, which is the basis framework of our EnerGAN++ configuration.

2) GENERATIVE ADVERSARIAL NETWORKS

GANs are popular in a variety of application domains, including photorealistic image super-resolution [23], image inpainting [24], text to image synthesis [25]. Several studies have shown promising results for reproducing data, in spite of labels corrupted by random noise [7], [26], [27]. Generative adversarial networks learn a deep generative model that is able to synthesize high dimensional data samples. New data samples are synthesized by passing latent samples, drawn from a chosen prior distribution, through the generative model. However, GANs do not offer an "inverse model," a mapping from data space back to latent space, making it difficult to infer a latent representation for a given data sample. There are several relevant studies trying to propose an inverse GAN such as [8], [28], [29].

3) GENERATIVE ADVERSARIAL NETWORKS IN NILM MODELLING

Bao *et al.* [30] adopt a GAN-based framework for solving NILM in an early attempt. Then, Kaselimi *et al.* [9] propose a generative adversarial network for sequence to sequence learning, whereas Pan *et al.* [31] achieve sequence to subsequence learning with conditional GANs. Chen *et al.* [32] propose a context aware convolutional network for NILM that has been trained adversarially. Recently, GANs have been used as a method for generating realistic energy consumption data. Under this framework [33] proposes an algorithm of using GANs to sufficiently learn from a limited number of real data, whereas the work of [34] proposes a synthetic appliance power signature generator, called PowerGAN, to mitigate the data limitations arising from the insufficient labeled appliances power data.

B. MOTIVATION AND CONTRIBUTION

The main limitation of the aforementioned GAN-based NILM approaches is that the discriminator is represented by a generic classifier which has not been optimised for the specific particularities of energy disaggregation. More specifically, energy consumption follows long-range dependencies which cannot be approximated by the traditional shallow-based (short-term) discriminant components. Moreover, data consumption time-series of an appliance present non-linear auto-regressive behavior; the output values at a time instance is non-linearly related with the output values of previous time instances. Therefore, the discriminator classifier should simulate recurrent capabilities. Finally, a GAN model used for NILM modelling is implemented using the inverse GAN framework, since we need to approximate real energy consumption signals of an appliance instead of generating realistic outputs.

In this paper, we extend the EnerGAN approach of [9] by including a deep learning classifier in the discriminator component of GAN. In particular, we enrich the concept of adversarial learning in NILM by introducing a more efficient discriminator in our proposed EnerGAN++ model which is now constitute of a combined convolutional layer with a recurrent GRU unit instead of a simple binary classifier. Advanced structures such as the recurrent GRU approximate long range recurrent dependencies in a better way, compared to traditional recurrent neural networks that suffer from the vanishing gradient problem [35], [36]. In this work, we leverage the strengths of rapid progress in CNNs and the desire to apply these models to time-varying power consumption data sequences, under an adversarial training framework [37]. Here, we emphasize that alternative ways of applying CNNs for sequential data with temporal character for NILM have been proposed, such as [14]. However, none of the above mentioned methods that combine CNNs properties with temporal character have been used in adversarial learning for energy disaggregation. In particular, EnerGAN++ model has good performance, especially in case that (a) noisy aggregate signals are used as input triggers and (b) abrupt changes in the appliance energy signals are encountered. We name this extended model, which is also based on an inverse GAN structure, EnerGAN++. Therefore, the main contributions of this paper can be summarized as follows:

- The use of a deep learning recurrent classifier to model the discriminator component of the EnerGAN++, with the capability to approximate long-term and regressive data signal behaviour, instead of the traditional GAN discriminators relied on shallow and generic classifiers.
- EnerGAN++ generates the entire signal waveform and it is capable of making more long-term estimations, rather than predicting solely the single current value of the individual power signal
- Our approach uses the aggregate energy measurements as the noise input vectors triggering the GAN model. In addition, since our target is to approximate real energy consumption of a specific appliance, EnerGAN++ uses labeled time-series of single appliance as input vectors of the generator during the training phase in contrast to conventional GAN modelling where only random noise signals are considered as input triggers. The non-linear deep discriminator unit competitively acts with the generator to reject generated sequences which are not well approximates of energy time-series of an appliance during training. Instead during testing, only the aggregate energy signal is used as input and the EnerGAN++ isolates the energy consumption sequence of a specific appliance from aggregate measurements.
- EnerGAN++ are inherently more robust to noisy input signals and less sensitive to inconsistencies or gaps in the input aggregate signal, since more sophisticated deep learning GAN components are considered.

The remainder of this paper is structured as follows: In Section II a detailed description of the NILM problem

formulation is provided. Sections III and IV describe the two EnerGAN++ network's components: the generator and the discriminator, explaining at the same time the challenges and barriers that we should overcome to adopt a GAN network for solving NILM problem as well as, the respective adaptations that EnerGAN++ introduces. Section V presents the proposed EnerGAN++ method configuration. In Section VI, the EnerGAN++ approach is experimentally evaluated against state of the art energy disaggregation methods in publicly available datasets, whereas Section VII concludes the paper with a summary of findings.

II. NILM PROBLEM FORMULATION

The residential total power consumption is measured using smart meter devices, thus the consumers be aware of their total (aggregate) power consumption. However, as far as energy efficiency is concerned, energy consumption awareness at appliance level is essential. One way to measure consumption per appliance is through the usage of smart plugs, a solution which is economically unaffordable. For this reason, usually NILM or energy disaggregation methods are applied. NILM is the problem of decomposing the total power consumption of a household, into individual appliance power signal components, using signal processing and machine learning tools, without prior existence of smart-plug equipment.

At a discrete time index t, we assume $\tilde{p}(t)$ the noisy aggregate measured energy signal for the whole household under study. Signal $\tilde{p}(t)$ is the sum of the individual appliances' power consumption $p_j(t)$ plus an additional noise $\epsilon(t)$. Thus, in a NILM framework [38], we express the total power consumption $\tilde{p}(t)$ as:

$$\tilde{p}(t) = \sum_{m=1}^{M} p_m(t) + \epsilon(t) \tag{1}$$

In Eq. (1) variable m refers to the m-th out of M available appliances. Here, we need a robust to noise model able to separate the total noisy power measurements $\tilde{p}(t)$ into the individual -free of noise- appliance source signals $p_m(t)$. Under a NILM framework, the individual appliance power consumption $p_m(t)$ is not a priori available, assuming the absence of installed smart plugs. Instead, only $\tilde{p}(t)$ is given. Therefore, the problem is to calculate the best estimates $\hat{p}_m(t)$ of the appliance power consumption, given the noisy $\tilde{p}(t)$ values.

NILM methods often look at the problem as decomposing a mixture signal into individual appliances signals (based on single-channel source separation) and formulate the task as an optimization problem [see Eq. (1)]. Traditional generative models such as independent component analysis [39] and non-negative matrix factorization [18] have been proposed to solve the NILM problem. Linear independent component analysis (ICA) has been especially popular as a method for blind source separation (BSS), with applications in various domains including audio source separation and image processing. However, it would be interesting to replace the linear ICA model, with an alternative model to model the non-linearities existing in energy data consumption signals. In the light of the recent success of deep learning methods, auto-encoders (AEs) have been proposed as an approach to supervised non-linear source separation, by for example Pandey *et al.* [40] and Grais and Plumbley [41]. In the context of NILM, the work of Kelly [10] is one of the first works that have used denoising autoencoders.

A. ADVERSARIAL LEARNING IN ENERGY DISAGREGGATION - THE ENERGAN++ APPROACH

In general estimates of $\hat{p}_m(t)$ given as input the aggregate signal $\tilde{p}(t)$ is approximated through a non-linear relationship and therefore neural network structures are considered as universal approximators [42], [43]. However, unlike other deep learning neural networks that are trained with a loss function until convergence, a GAN model is trained in an adversarial manner between two main components: *the generator* $\mathcal{G}$ and *the discriminator* $\mathcal{D}$. In our NILM modelling, the *generator* $\mathcal{G}$ is trained to produce time sequences resembling energy consumption of an appliance, while the *discriminator* $\mathcal{D}$ is trained to check whether the produced time series by the *generator* coincide with a specific (real) appliance energy consumption sequence or not. Therefore $\mathcal{G}$ is trained in a way to cheat $\mathcal{D}$ in the sense that it produces data sequences that they are not be able to be distinguished by $\mathcal{D}$. In other words $\mathcal{G}$ and $\mathcal{D}$ play a two-play minmax game with value function of $V(\mathcal{D}, \mathcal{G})$

$$\min_{\mathcal{G}}\max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) = E_{p_m \sim p_{data(p_m)}}[\log \mathcal{D}(p_m)] + \\ + E_{z \sim p_z(z)}[\log(1 - \mathcal{D}(\mathcal{G}(z)] \tag{2}$$

Eq. (2) means that the discriminator $\mathcal{D}$ is trained to maximize the probability of assigning the correct label between the ground truth samples and the "fake" samples produced by $\mathcal{G}$, while simultaneously $\mathcal{G}$ to minimize $log(1 - \mathcal{D}(\mathcal{G}(z))$ that is the generator to produce samples of being indistinguishable by $\mathcal{D}$. The discriminator is trained to reject the artificially generated sequences by the $\mathcal{G}$ and the generator to produce sequences confusing $\mathcal{D}$.

A typical generative model takes a noise signal vector z in space R^k and generates another signal, $\mathcal{G}(z)$, of a higher dimensional space R^n with $n > k$. The generative models produce data that resemble the real data [6]. Probabilistic generative models demonstrate excellent performance for various tasks such as denoising, inpainting, texture synthesis, video and natural language processing [44].

In our case, a typical generative model will try to successfully reproduce the power consumption signal of an appliance in households. In particular, in the following, we denote as $\{p_{m,t}^{label}\}_{t=t}^{t=t+T}$ the target signals over a time window T and as $\{\hat{p}_{m,t}^{\mathcal{G}}\}_{t=t}^{t=t+T}$ the data generated by the generator $\mathcal{G}$ over the same time interval. A traditional GAN generator is optimized to create data sequence resembling the labeled data. This is achieved by propagating the noise signal into a higher

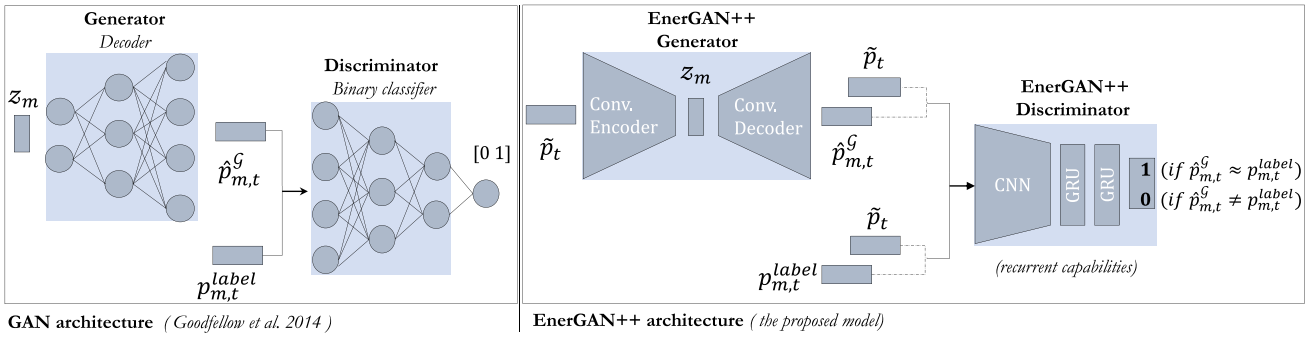

FIGURE 1. **Comparison between a typical GAN approach and the EnerGAN+ structure approach. Three main specifications are introduced there: (i) the use of the aggregated signal as input noise, (ii) the encoding layer used to map the aggregated signal into a desired latent feature, and (iii) the advanced CNN-GRU discriminator dedicated to achieve the optimal sequence classification, that ensure the EnerGAN++ ability to achieve energy disaggregation, even if the data input samples are noisy.**

dimension space through a neural network structure (i.e., a decoder) with the purpose of transforming the noise input z into a data sequence $\{\hat{p}^{\mathcal{G}}_{m,t}\}^{t=t+T}_{t=t}$. This is illustrated in Fig. 1, where the noise input signal z is forwarded into a convolutional decoder to generate a data sequence of $\hat{p}^{\mathcal{G}}_{m}$ of similar statistical properties with the labelled data p^{label}_{m}.

Even though the generated data $\{\hat{p}^{\mathcal{G}}_{m,t}\}^{t=t+T}_{t=t}$ are indistinguishable from the real data $\{p^{label}_{m,t}\}^{t=t+T}_{t=t}$, as they follow the same statistical properties, their resemblance to the actual power values is purely coincidental. Until now, we have underlined the GAN's ability to reproduce appliances power signals that resemble to the actual appliance operation power signal. However, solving NILM problem implies the need for a network trained not only to reproduce the power signal of an appliance but to know the exact operation and consumption of the appliance at a given time epoch. Thus, our method extends beyond the need for randomly generated appliance power consumption value having the same statistics. The trained network should be able to provide information regarding the estimated appliance power consumption value at a specific time, and this, in turn, means that z values should not be purely randomly generated but to move in a space appropriate for extracting the desirable sequential data $\{p^{label}_{m,t}\}^{t=t+T}_{t=t}$ [45]. In case that the desirable output (i.e. the disaggregated appliance values) $\{p^{label}_{m,t}\}^{t=t+T}_{t=t}$ is a priori known, we need to solve the inverse problem, which corresponds to finding the latent vector z that explains the output measurements as much as possible.

To address the aforementioned difficulties, in this paper, we modify the traditional GAN configuration to fit the particularities of energy disaggregation problem. In energy disaggregation the purpose of a NILM-based GAN network is to generate real appliance power sequence which is almost identical with the labelled data over a time interval T, that is $\{\hat{p}^{\mathcal{G}}_{m,t}\}^{t=t+T}_{t=t} \approx \{p^{label}_{m,t}\}^{t=t+T}_{t=t}$ instead of producing realistic data sequences. Thus, it is important to find a way of inverting the "non-invertible" generator. This is achieved through the following modifications of the traditional GAN structure (see Fig. 1).

- First, instead of using a noise input signal z the aggregate energy consumption signal $\tilde{p}(t)$ is considered. Aggregate signal $\tilde{p}(t)$, which is the sum of appliance's energy consumption signals, can be considered as an "noise" input trigger of the EnerGAN++ model. In this way, the generator $\mathcal{G}$ is capable of producing real and not realistic energy consumption sequences for the m-th appliance, confusing the discriminator $\mathcal{D}$.
- Second, in case that the aggregate signal $\tilde{p}(t)$ is used as input trigger of the proposed GAN-based model for NILM, an additional layer is required for the generator $\mathcal{G}$. In particular, we add an encoder prior to the decoder with the main purpose of compressing the input aggregate signal $\tilde{p}(t)$ in a way to produce a noise vector resembling z of the traditional GAN structure.
- Third, the traditional GAN models considers simple classification structure for the discriminator $\mathcal{D}$. Energy consumption signal presents high temporal relationships and auto-regressive properties. Therefore, the traditional GAN discriminator can be easily cheated by the generator, reducing the overall NILM performance. For this reason, the proposed EnerGAN++ model modifies the conventional GAN discriminator by using GRU unit which has recurrent capabilities in order to address the temporal appliance's energy consumption properties. EnerGAN++ has a more clever discriminator, forcing the generator to produce almost identical appliance energy consumption to cheat $\mathcal{D}$.

III. ENERGAN++ GENERATOR

Fig. 2 presents the structure of the generator of the proposed EnerGAN++ model. Our method extends the traditional operation of a GAN beyond the need for randomly generated appliance's power data sequence of the same statistics with the labeled data. The NILM framework implies the need for a network trained to generate the exact operation and consumption of an appliance at a given time instance t.

This is achieved by using as input trigger of the EnerGAN++ the aggregate signal $\{\tilde{p}_t\}^{t=t+T}_{t=t}$ over a time window

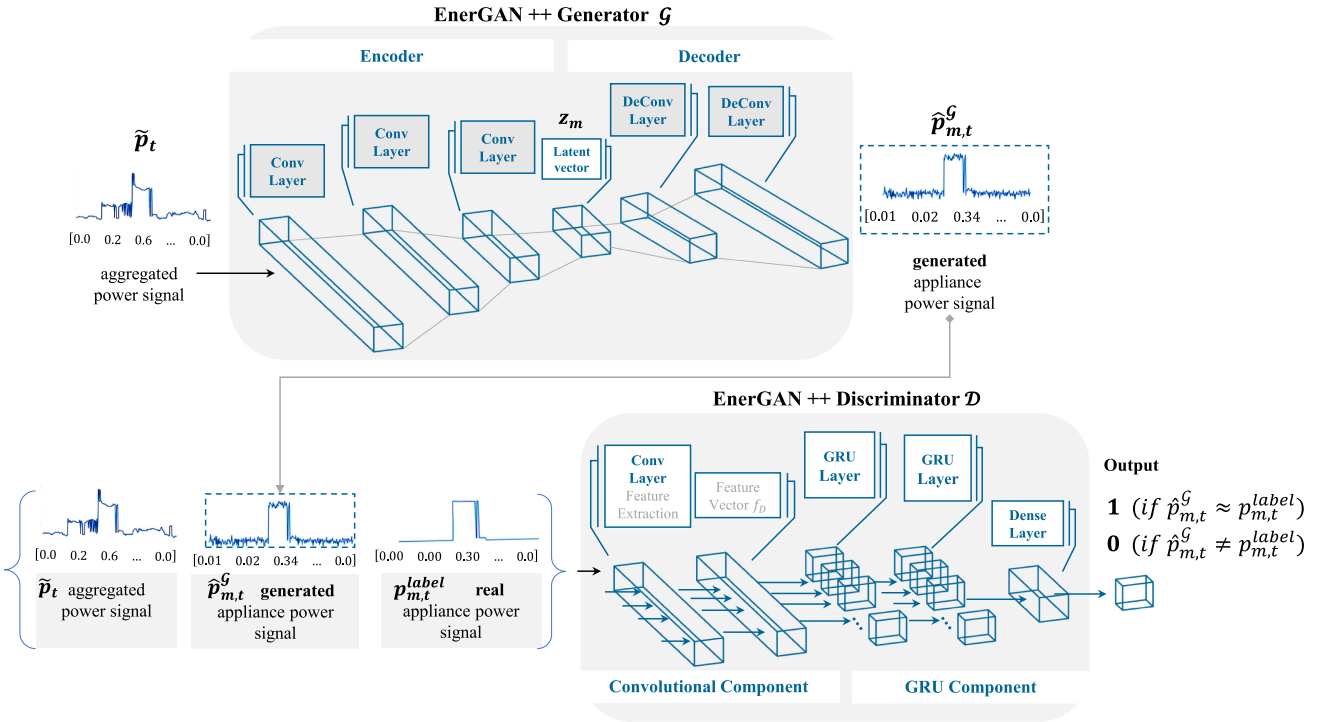

FIGURE 2. Proposed EnerGAN++ architecture. The proposed method aims to unify existing AE and GAN architectures into a single framework, in which AE achieves a non-linear total power signal source separation, and adversarial training enhances model's robustness to noise. The generator is a convolutional autoencoder and as input has the noisy version of the aggregated power signal. The autoecoder is trained adversarially with the discriminator. The discriminator is a long-term convolutional recurrent network for sequence classification, and is conditioned with the aggregated power signal.

of T duration. The signal $\{\tilde{p}_t\}_{t=t}^{t=t+T}$ is considered as a noise trigger since it is the summation of independent energy consumption signals of the appliances. In addition, during training, the ground truth data of the m-th appliance $\{p_{m,t}^{label}\}_{t=t}^{t=t+T}$ over a time window of T duration is considered in order to initiate the EnerGAN++ generator to simulate the real energy consumption data of the m-th appliance. We denote as $\mathcal{I}_{train}$ the input trigger vector of the $\mathcal{G}_{EnerGAN++}(\cdot)$ generator during the training phase,

$$\mathcal{I}_{train} \equiv \left[\{\tilde{p}_t\}_{t=t}^{t=t+T}\ \{p_{m,t}^{label}\}_{t=t}^{t=t+T}\right]^T \tag{3}$$

To handle the aggregate signal $\{\tilde{p}_t\}_{t=t}^{t=t+T}$ and the ground truth labels of $\{p_{m,t}^{label}\}_{t=t}^{t=t+T}$ as input trigger of the EnerGAN++ model, an encoder layer is added prior to the decoder. The encoder generates a compressed "noise" signal z_m (by encoding the input vector signal of $\mathcal{I}_{train}$) [10] which is used as input trigger of the decoder of the EnerGAN++ generator to produce a real appliance energy consumption time series $\mathcal{G}(z_m)$ for the m-th appliance. Thus, the pipeline of the EnerGAN+ generator during the *training phase* is the following:

$$\begin{aligned} &\textit{the pipeline of EnerGAN++ generator during training}: \\ &\mathcal{I}_{train} \rightarrow \ Encoder\,(\mathcal{I}_{train}) \rightarrow z_m \rightarrow \\ &\rightarrow Decoder\,(z_m) \rightarrow \mathcal{G}_{EnerGAN++}(z_m) \end{aligned} \tag{4}$$

Eq. (4) means that the input signal $\mathcal{I}_{train} \equiv [\{\tilde{p}_t\}_{t=t}^{t=t+T}\ \{p_{m,t}^{label}\}_{t=t}^{t=t+T}]^T$ is transformed (compressed) to a latent noise trigger z_m, through a convolutional encoder and then, the noise signal z_m is decompressed to generate a signal $\mathcal{G}_{EnerGAN++}(z_m)$ that resembles the real energy consumption of the m-th appliance. The encoder with convolutional layers, is forced to be an inverted version of the decoder (with transposed convolutional layers), where corresponding layers perform opposite mappings and share parameters [46]. The model tries to minimize the difference between the $\{\hat{p}_{m,t}^{\mathcal{G}}\}_{t=t}^{t=t+T}$ sequence values and the actual sequence values $\{p_{m,t}^{label}\}_{t=t}^{t=t+T}$, generating data sequences $\mathcal{G}_{EnerGAN++}(z_m)$ that confuse the discriminator $\mathcal{D}$.

During *the testing phase*, the generator of the EnerGAN++ model $\mathcal{G}_{EnerGAN++}(\cdot)$ receives as input only the aggregate signal $\{\tilde{p}_t\}_{t=t}^{t=t+T}$ and not the appliance ground truth data of $\{p_{m,t}^{label}\}_{t=t}^{t=t+T}$ since it has been learned during the training phase to produce almost identical energy consumption time series of the m-th appliance. Thus, the pipeline of the EnerGAN++ generator $\mathcal{G}_{EnerGAN++}(\cdot)$ during *the testing phase* is the following:

$$\begin{aligned} &\textit{the pipeline of EnerGAN++ generator during testing}: \\ &\{\tilde{p}_t\}_{t=t}^{t=t+T} \rightarrow Encoder\left(\ \{\tilde{p}_t\}_{t=t}^{t=t+T}\ \right) \rightarrow \hat{z}_m \rightarrow \\ &\rightarrow Decoder\,(\hat{z}_m) \rightarrow \mathcal{G}_{EnerGAN++}(\hat{z}_m) \end{aligned} \tag{5}$$

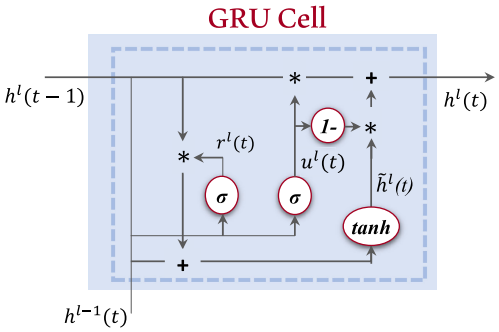

FIGURE 3. Computation of a hidden state in a GRU cell.

It is clear that the output of the EnerGAN++ generator during the testing phase approximates its output during training since the ground truth data of the m-th appliance are available only during training. Therefore, we have that $\mathcal{G}_{EnerGAN++}(\hat{z}_m) \approx \mathcal{G}_{EnerGAN++}(z_m)$, meaning that the propoduce data time series by the generator $\{\hat{p}^{\mathcal{G}}_{m,t}\}_{t=t}^{t=t+T}$ is very close to the labeled data of $\{p^{label}_{m,t}\}_{t=t}^{t=t+T}$

IV. ENERGAN++ DISCIMINATOR

In the EnerGAN++ model there is not a loss function for training the network until convergence. Instead, adversarial learning is adopted based on a min-max two-player game between the generator $\mathcal{G}_{EnerGAN++}$ and the discriminator $\mathcal{D}_{EnerGAN++}$. The generator produces a sequence of data $\{\hat{p}^{\mathcal{G}}_{m,t}\}_{t=t}^{t=t+T}$ to cheat the discriminator, while the discriminator seeks to distinguish whether the data sequence produced by $\mathcal{G}$ is real (assigned a value of 1) or fake (assigned a value of 0). The traditional discriminator of a GAN network does not take into account the regressive temporal characteristics of the energy consumption signals. Therefore, it can be easily cheated by the generator which has been appropriately modified to produce almost identical time series for the m-th appliance (see Section III).

Here, the discriminator solves a binary sequence classification problem. Sequence classification is posed as a problem of assigning a label to a sequence of observations. Recently deep learning reccurent networks have been proposed in the literature for sequence classification. Examples are the LSTM [36] and the GRU [47] structures. These networks have an internal mechanism to balance between current and previous time steps, and thus, memorizing the temporal information flow.

A. CNN ENRICHED - GATED RECURRENT NETWORKS FOR ENERGAN++ DISCRIMINATOR

In the EnerGAN++ model, a combined CNN enriched - GRU classifier is adopted as the discriminator unit. GRU networks are appropriate for modelling the temporal auto-regressive properties of a time series (Fig. 3). However, GRU structures are not able to extract features from the input data in a way to optimize the overall classification performance. For this reason, in this paper, we adopt a combined approach by introducing a CNN model [12] prior to the GRU framework. Hereby, CNNs have been used as feature extractors for the GRUs. In other words, the combination of CNN as an efficient feature extractor with the GRU model, is capable of representing, synthesize and therefore distinguish the temporal dynamic nature of the power sequence signals.

GRU is a modern version and very similar to the traditional LSTM cell, albeit less complex. It merges the cell state and hidden state of the traditional LSTM cell, and in addition, it combines the forget and input gates into a single update gate. Thus, it is more computational efficient as it need less parameters for training and it has less complex structure. GRUs make use of a gated activation function and are designed so as to have more persistent memory thereby making it easier to capture long-term dependencies.

In the proposed EnerGAN++ dicriminator, a stack of l GRU layers is considered. In general, stacking improves discrimination performance [48]. Stacked GRU layers have two main operations of dependence; *the in-depth dependence* and *the temporal dependency*. The in-depth dependence implies that the output of the current GRU layer is related with the output of the previous layer, while temporal dependency assumes that the hidden GRU states in a layer are inter-related in the time domain. Each GRU unit has two additional control variables; the reset gate $r^l(t)$ and the update gate $u^l(t)$ (see Fig. 3). The reset gate $r^l(t)$ is responsible for determining how much of information to forget. The update gate $u^l(t)$ is responsible for determining the worth-remembering information of the previous states that should be forwarded to the next state. Therefore, the gates $r^l(t)$ and $u^l(t)$ are related with the hidden states $h^l(t)$ and $h^l(t-1)$ as follows

$$\begin{bmatrix} u^l(t) \\ r^l(t) \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \end{bmatrix} \mathbf{W}_l \begin{bmatrix} h^l(t-1) \\ h^{l-1}(t) \end{bmatrix} + \begin{bmatrix} b^l_u \\ b^l_r \end{bmatrix} \tag{6}$$

In Eq. (6) σ is the sigmoid function and b^1_u and b^1_r are the respective biases of each component for the GRU cell. Variables $\mathbf{W}$ and $\mathbf{U}$ are the transition matrices of the l-th GRU. In case of $l = 1$, the convolutional layer is considered as previous layer and the input vector is the feature vector [see Eq. (9)]. This means that $h^{l=0}(t) \equiv f_{\mathcal{D}}(t)$.

In stacked GRUs configuration a recursive approach is considered as regards the operation of each GRU cell. In particular, a new memory state, denoted as $\tilde{h}^l(t)$, is considered, acting as the consolidation of the hidden state of the previous layer $h^{l-1}(t)$ and the previous hidden state $h^l(t-1)$ of the current layer. The consolidated hidden state $\tilde{h}^l(t)$ is given by:

$$\tilde{h}^l(t) = \tanh\left(r^l(t)\mathbf{U}h^l(t-1) + \mathbf{W}h^{l-1}(t)\right) \tag{7}$$

In Eq. (7) function $tanh(\cdot)$ refers to the hyperbolic tangent relationship. As we stated previous, for $l = 0$ the value of the hidden state equals $h^{l=0}(t) \equiv f_{\mathcal{D}}(t)$ (see Fig. 4). Eq. (7) means

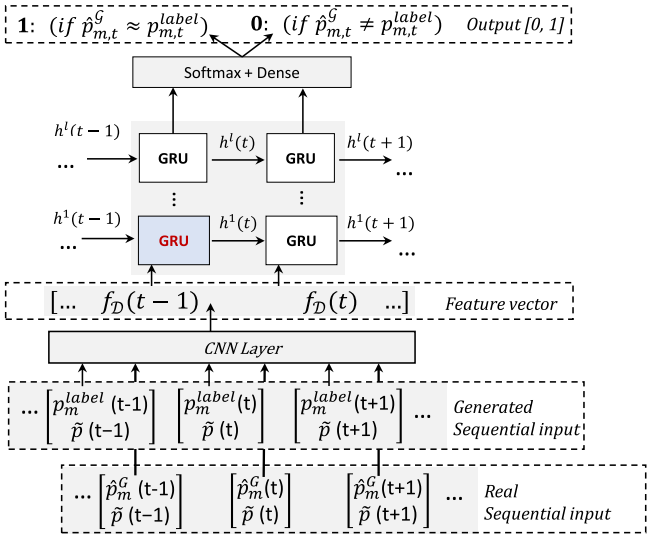

FIGURE 4. Discriminator is a long-term convolutional recurrent network for sequence classification. The proposed architecture of the discriminator leverages the strength of rapid progress in CNNs, among with the need to apply the convolutional models for temporal sequences of data. Thus, a sequential input is entered into the discriminator, a CNN layer is used for the optimal feature extraction and then, the stacked recurrent layers follow. The scope of the discriminator is to decide whether the given sequence is the generated or the real one.

that the consolidated state is related with the output of the hidden state $h^l(t-1)$ at the time instance $t-1$ and the output of the previous hidden layer $h^{l-1}(t)$ at the time instance t. Using the values of the consolidated state $\tilde{h}^l(t)$ [see Eq. (7)] and the values of the update gate $u^l(t)$ [see Eq. (6)] the value of the hidden state of the l-th GRU element is estimated

$$h^l(t) = \left(1 - u^l(t)\right)\tilde{h}^l(t) + u^l(t)h^l(t-1) \tag{8}$$

In more detail, the GRU related above mentioned operations are illustrated in Fig. 4.

B. OPERATION OF THE ENERGAN+ DISCRIMINATOR

The proposed discriminator has two main components; *the convolutional* layer and the *GRU* unit. The convolutional layer transform the input signal to a reliable feature vector $f_{\mathcal{D}}(t)$, while the GRU unit performs the discrimination. As input signal the generated power signal $\mathcal{G}_{EnerGAN++}(\hat{z}_m)$ of the m-th appliance is used [see Eq.(5)]. In addition, the labeled training samples of the respective appliance $\{p^{label}_{m,t}\}^{t=t+T}_{t=t}$ and the aggregate measurements $\{\tilde{p}_t\}^{t=t+T}_{t=t}$ is used as input triggers for classification comparisons. The discriminator has been optimized through training to distinguish the "fake" data sequences produced by the generator $\{\hat{p}^{\mathcal{G}}_{m,t}\}^{t=t+T}_{t=t}$ from the real ones.

Initially, the input vector of the discriminator, that is the data produced by the generator $\{\hat{p}^{\mathcal{G}}_{m,t}\}^{t=t+T}_{t=t}$, the real labelled data $\{p^{label}_{m,t}\}^{t=t+T}_{t=t}$ and the aggregate measurements $\{\tilde{p}_t\}^{t=t+T}_{t=t}$ are fed as inputs to a CNN structure with the main purpose of transforming them into optimized feature maps of $f_{\mathcal{D}}(t)$.

$$f_{\mathcal{D}}(t) \sim Conv_{\mathcal{D}_{EnerGAN++}}\left(\mathcal{I}_{input}\right)$$

with

$$\mathcal{I}_{input} = \left[\{p^{label}_{m,t}\}^{t=t+T}_{t=t}, \{\hat{p}^{\mathcal{G}}_{m,t}\}^{t=t+T}_{t=t}, \{\tilde{p}_t\}^{t=t+T}_{t=t}\right]^T \tag{9}$$

The features $f_{\mathcal{D}}(t)$ at the time instance t are fed to the GRU structure trained to distriguish the "fake" sequence produced by $\mathcal{G}_{EnerGAN++}$ from the real one (available in the training set). Therefore, we have that

$$\mathcal{D}_{EnerGAN++} \equiv GRU\left(f_{\mathcal{D}}(t)\right) = \begin{cases} 1 & \text{if } \{\hat{p}^{\mathcal{G}}_{m,t}\}^{t=t+T}_{t=t} \approx \{p^{label}_{m,t}\}^{t=t+T}_{t=t} \\ 0 & \text{if } \{\hat{p}^{\mathcal{G}}_{m,t}\}^{t=t+T}_{t=t} \text{ not } \approx \{p^{label}_{m,t}\}^{t=t+T}_{t=t} \end{cases} \tag{10}$$

V. PROPOSED ENERGAN++ MODEL CONFIGURATION

The min-max game refers to the minimization of generator and the maximization of the discriminator.

A. ADVERSARIAL LEARNING BETWEEN THE GENERATOR AND THE DISCRIMINATOR

Eq. (10) means that the disctiminator is trained not to be cheated by the generator $\mathcal{G}_{EnerGAN++}$. At the same time the generator is optimized to produce a sequence of the m-th appliance in a way to cheat the discriminator, that is $\mathcal{G}_{EnerGAN++}(\hat{z}_m) \approx \{p^{label}_{m,t}\}^{t=t+T}_{t=t}$. This results into a 2-player adversarial learning between the discriminator and the generator

$$\text{EnerGAN++: } \min_{\mathcal{G}}\max_{\mathcal{D}} V(\mathcal{D}_{EnerGAN++}, \mathcal{G}_{EnerGAN++}) =$$
$$= E\left[\log \mathcal{D}_{EnerGAN++}(p^{label}_m, \mathcal{G}_{EnerGAN++}(\hat{z}_m))\right] +$$
$$+ E\left[\log(1 - \mathcal{D}_{EnerGAN++}(\mathcal{G}_{EnerGAN++}(\hat{z}_m), \tilde{p}))\right] \tag{11}$$

In Eq. (11) it is held that $\mathcal{G}_{EnerGAN++}(\hat{z}_m) \equiv \{\hat{p}^{\mathcal{G}}_{m,t}\}^{t=t+T}_{t=t}$

B. PROPOSED ENERGAN++ NETWORK CONFIGURATION

Fig. 2 shows the proposed methodology and the basic structure of the proposed model. As we have stated previously two main components are included in the EnerGAN++ model; *the generator* and *the discriminator*.

EnerGAN++ generator configuration: The EnerGAN++ generator consists of an autoencoder having (i) *a convolutional encoder* and (ii) *a decoder part*, which is an inverted convolutional version of the encoder. During the training phase the encoder part takes the aggregate signal measurements $\tilde{p}(t)$ and the labelled energy consumption time series of the m-th appliance $p^{label}_m(t)$ and maps it to a latent compressed vector of $z_m(t)$. The compressed vector $z_m(t)$ feeds the decoder part of the generator with the purpose of reconstructing an approximate version of $p^{label}_m(t)$ denoted as $\hat{p}_m(t)$. During the testing phase only the aggregate measurements are used as inputs to the encoder part of the generator and therefore,

it produces an approximate latent compressed vector of $\hat{z}_m(t)$. The decoder part has been learned, during the training phase, to produce approximate energy consumoption time series of the m-th appliance. Therefore, during the test phase, the decoder part of the generator produces a data sequence of $\hat{p}_m^{\mathcal{G}}(t)$ close to $p_m^{label}(t)$ from the input latent signal $\hat{z}_m(t)$. Therefore, the loss function of the generator autoencoder will be:

$$L_{\mathcal{G}_{EnerGAN++}}: \ min \left\| p_m^{label}(t) - Decoder(\hat{z}_m(t)) \right\|_2$$

with

$$\hat{z}_m(t) = Encoder(\tilde{p}(t)) \tag{12}$$

Eq. (12) means that the encoder part of the generator has been trained to produce a latent compressed signal $\hat{z}_m$ with the capability of generating the energy time series p_m^{label} for the m-th appliance.

The encoder configuration structure has three convolutional layers: the first layer consists of 256 filters with a kernel size 1×8, the second layer consists of 128 filters of 1×16 kernel size, whereas the third layer consists of 64 filters with size 1×32. The decoder maps the latent value z_m to a higher feature space, which describes accurately the individual appliance waveform. The proposed topology layout has two deconvolution layers: the first layer consists of 64 filters with a kernel size 1×8 and the second layer consists of 128 filters of 1×16 kernel size. The latent vector is not randomly generated, on the contrary it comes from an inverted generator (the encoder) and encloses information from the aggregate signal.

EnerGAN++ discriminator configuration: The discriminator $\mathcal{D}$ comprises of a convolutional layer with 60 different kernels consisting of trainable parameters which can convolve the given input and extract the appropriate features. After the automated feature selection, a sequence classification with two stacked recurrent GRU layer follows. The first GRU layer comprises of 40 filters and the second with 30 filters, whereas the network architecture ends up with two dense layers. The first one consists of 30 neurons, whereas the second one is fully connected to the output neuron to predict whether the input pair is the actual data or the generated one. The neurons in output layer use as the activation function the sigmoid function, whereas all the remaining layers use ReLU as the activation function.

VI. EXPERIMENTS

A. DESCRIPTION OF THE DATASET

The evaluation of our generative adversarial model for energy disaggregation has been conducted on nine appliances derived from AMPds [50] and REFIT [51] datasets. These open-access energy consumption datasets provide the aggregate power measurements of the whole house and sub-metered readings (smart plugs) from individual appliances at different time resolutions; 60 s for AMPds, 8 s for REFIT. In our study, REFIT data are down sampled to 60 s resolution. The AMPds consists of a single house in Canada, whereas the REFIT consists of 20 houses located in U.K. Both the AMPds and REFIT datasets collected over a period of two years. The appliances are: clothes dryer, heat pump and oven appliances from AMPds and dishwasher, kettle, microwave, toaster, tumble dryer and washing machine appliances from REFIT dataset.

The Section V presents: (i) the evaluation performance metrics for our proposed methods in contrast to the other state of the art methods (see Section V.B), (ii) the robustness of our method to noise, which is our comparative advantage in contrast to other conventional NILM methods (see Section V.C).

The paper points out the EnerGAN++ method's comparative advantage for noisy aggregate data. Furthermore, REFIT measurements contain noise in the label ground truth data due to numerous unknown appliances and measurement errors. This introduces an additional challenges for our proposed method.

Our EnerGAN++ algorithm is implemented in Python 3.6 using the Keras API integrated into TensorFlow 2. The computer used for all of the training and testing was an Intel Core i7-8750H CPU at 2,20 GHz with 8 GB of random-access memory and an NVIDIA GeForce GTX 1050 with 4096 MB of DDR5 memory.

B. QUANTITATIVE EVALUATION METRICS

In this study, we compare the proposed EnerGAN++ method for NILM, against other state of the art methods. In particular, our method is evaluated against the previous version of our proposed GAN (EnerGAN) [9], as well as the BabiLSTM network, as presented in [2]. Furthermore, the method is compared against the traditional deep learning models, applied for NILM, such as sequence to sequence CNNs (seq2seq CNN), [16], [13], unidirectional LSTMs [11], [3] and denoising autoencoders [10]. These models have also been implemented in Python 3.6 using Keras. Finally, we compare the proposed method against additional benchmark approaches such as FHMM [49] and CO [49]. These approaches are implemented in NILMTK library [49].

Table 1 is a summary that compares the aforementioned techniques with the proposed EnerGAN++ model. The metrics used are:

i) the Mean Absolute Error (MAE) that measures the average magnitude of the errors in the set of predictions and is a commonly used metric for NILM [4]. MAE metric is used when we are interested in the error in power at every time point, and is less affected by outliers, i.e. isolated predictions that are particularly inaccurate:

$$MAE = \frac{\Sigma_{t=1}^{N} \left| \hat{\mathbf{p}}_\mathbf{j}(t) - \mathbf{p}_\mathbf{j}(t) \right|}{N} \tag{13}$$

ii) the Signal Aggregate Error (SAE) is a common metric used to estimate the total energy consumed by each appliance over a period of time [4]. This measure is

TABLE 1. Performance Metrics (MAE, SAE, RMSE) for Nine Appliance of the AMPds and REFIT Datasets. in Bolds, We Have Highlighted the Method That Succeeds the Best Performance Per Appliance

	Clothes Dryer			Heat Pump			Oven		
	MAE	SAE	RMSE	MAE	SAE	RMSE	MAE	SAE	RMSE
EnerGAN++	17.700	**0.018**	192.523	**80.032**	**0.065**	**227.704**	**8.051**	**0.011**	**108.909**
ENERGAN [9]	25.030	0.034	260.748	80.125	0.255	256.610	10.258	0.619	145.82
BabiLSTM [2]	**10.009**	0.062	**99.211**	88.215	0.354	230.155	17.645	1.420	146.459
DAE [10]	37.355	0.170	310.081	55.608	0.261	228.690	19.255	1.306	137.930
seq2seq CNN [16]	15.473	0.116	149.819	107.119	1.501	260.468	67.578	7.536	322.287
unidirectional LSTM [11]	90.200	1.104	219.296	154.947	2.181	285.355	57.685	6.094	242.572
FHMM [49]	129.568	3.422	323.225	121.689	2.641	426.662	49.383	9.047	360.852
CO [49]	120.191	2.712	468.625	249.389	2.731	459.612	267.089	50.242	433.586
	Dishwasher			**Kettle**			**Microwave**		
	MAE	SAE	RMSE	MAE	SAE	RMSE	MAE	SAE	RMSE
EnerGAN++	**20.320**	**0.079**	156.633	**7.811**	**0.232**	119.102	**8.302**	1.417	**75.840**
ENERGAN [9]	22.166	0.154	197.087	31.304	0.788	122.357	12.046	1.444	94.008
BabiLSTM [2]	29.200	0.716	**144.776**	41.286	4.835	137.877	15.263	**0.210**	89.678
DAE [10]	25.410	0.235	205.516	9.063	0.303	**114.404**	12.252	0.391	89.587
seq2seq CNN [16]	34.987	0.886	188.099	19.896	1.968	115.482	14.830	0.408	79.568
unidirectional LSTM [11]	102.171	3.432	241.053	41.120	4.332	149.885	15.922	0.168	92.983
FHMM [49]	147.701	3.902	493.499	40.856	2.433	204.101	77.342	3.599	196.579
CO [49]	138.805	3.629	492.172	40.659	2.567	203.586	51.826	2.042	165.419
	Toaster			**Tumble Dryer**			**Washing Machine**		
	MAE	SAE	RMSE	MAE	SAE	RMSE	MAE	SAE	RMSE
EnerGAN++	**2.255**	**0.249**	**32.979**	**16.992**	**0.060**	**108.256**	**7.362**	1.519	105.555
ENERGAN [9]	2.645	0.632	38.743	19.207	0.069	133.004	8.171	**0.265**	101.452
BabiLSTM [2]	12.810	11.897	45.886	48.749	1.097	121.309	17.696	0.811	**88.099**
DAE [10]	8.294	7.737	57.960	32.864	0.149	142.340	13.400	0.275	109.782
seq2seq CNN [16]	15.044	13.790	50.837	42.513	0.648	158.970	27.090	2.196	133.218
unidirectional LSTM [11]	26.708	14.471	50.413	87.832	1.966	183.102	31.859	2.617	121.371
FHMM [49]	32.457	15.443	57.786	91.554	2.456	193.771	177.015	2.747	535.198
CO [49]	35.664	12.544	67.729	91.988	3.211	201.533	210.956	2.936	458.503

useful because a method could be accurate enough for reports of daily power usage even if its per-timestep prediction is less accurate,

$$SAE = \frac{\left|\Sigma_{t=1}^{N}\hat{\mathbf{p}}_{\mathbf{j}}(t) - \Sigma_{t=1}^{N}\mathbf{p}_{\mathbf{j}}(t)\right|}{\Sigma_{t=1}^{N}\mathbf{p}_{\mathbf{j}}(t)} \tag{14}$$

iii) the Root Mean Squared Error (RMSE), that is more sensitive to large errors, occurring either due to time delays or even fault detection (outliers),

$$RMSE = \sqrt{\frac{\Sigma_{t=1}^{N}(\hat{\mathbf{p}}_{\mathbf{j}}(t) - \mathbf{p}_{\mathbf{j}}(t))^2}{N}} \tag{15}$$

From the above, we consider that the smaller the MAE, SAE and RMSE errors are, the better the performance of the examined method is.

C. EXPERIMENTAL RESULTS

In Table 1, our proposed method has the lower values for MAE metric. This indicates that the proposed EnerGAN++ has the best performance compared to the other methods, in terms of MAE. The exception is the clothes dryer appliance, where MAE as well as RMSE values seem to be higher, while SAE error succeeds the minimum value. This is probably occurs due to the "jagged edges" appeared in clothes dryer appliance pattern, that successfully cashed by the bidirectional-LSTM scheme, but only as outline in the proposed scheme (also see Fig. 5). SAE and RMSE metric values show also good performance for the majority of appliances. In terms of RMSE error, however, the "failures" observed refereed to time delays occurred between the generated and ground truth data. Furthermore, from previous studies in NILM, has been observed that architectures such as LSTM and CNN, underestimate power values during the training process. As a result, such schemes, produce patterns with lower power values, leading to small RMSE values, but, on the contrary, to increased MAE values. Furthermore, it should be pointed out that the proposed network has been significantly improved compared to our former EnerGAN network [9]. The improvement is due to the different discriminator configuration, i.e. the recurrent convolutional network for sequence classification that is trained to discriminate samples from data, as well as the different loss functions selected for our proposed GAN network.

Fig. 5 shows the aggregate signal (grey line) and the generated power timeseries from our GAN network (purple line) at a given time period. Also, with orange line the ground truth data are depicted. As shown, the operation of each appliance is detected at an adequate level. In Fig. 5, the generated timeseries of power data are identical with the actual operation (ground truth) of clothes dryer, oven, kettle, microwave, toaster, tumble dryer and washing machine appliances. However, during the snapshot in time as illustrated in Fig. 5, for the heat pump appliance, a false positive is appeared, since the appliance is detected in operation five times, whereas 4

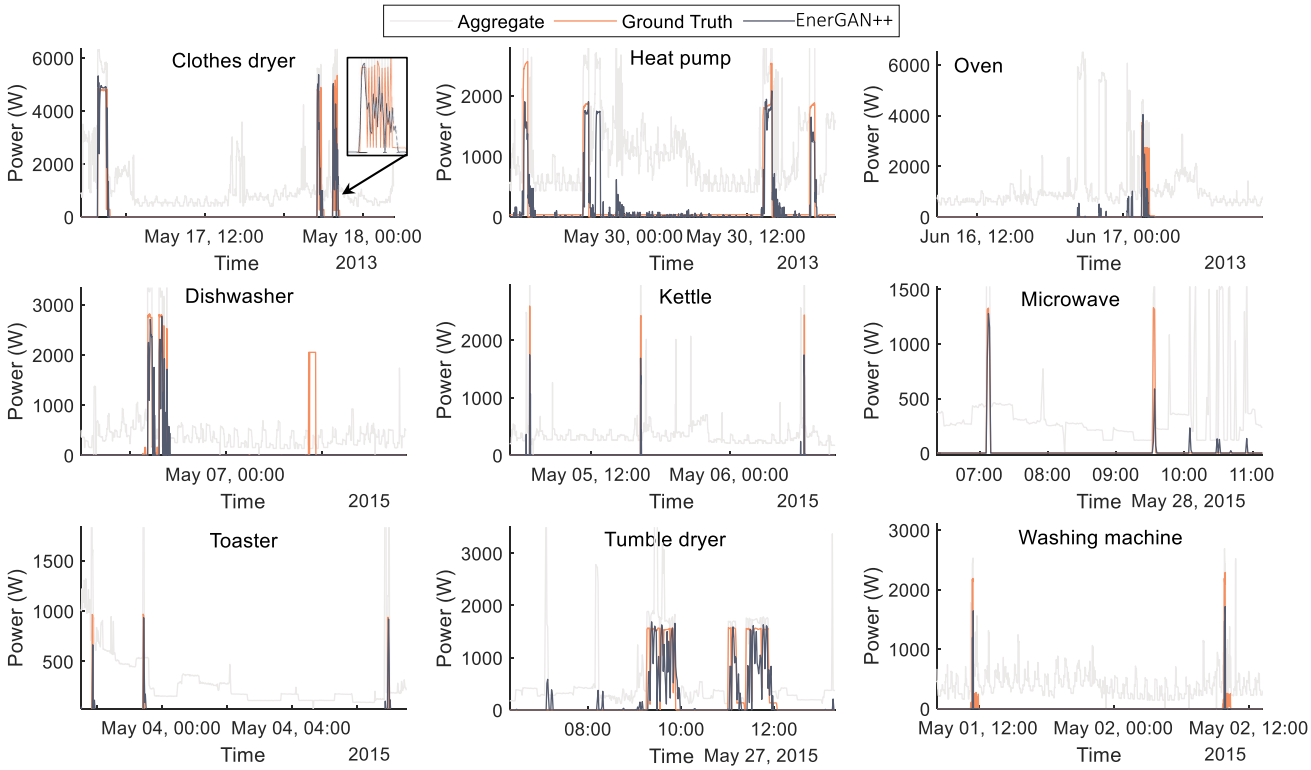

FIGURE 5. **Comparison of the proposed method (purple line) with ground truth (in orange) for selected appliances in AMPds and REFIT dataset. Furthermore, the aggregated data are also illustrated.**

times is actually ON. On the contrary, a false negative is detected, at first, for dishwasher appliance, but actually the orange undetected signal for this case indicates noise and not actual presence of the dishwasher appliance.

D. ROBUSTNESS TO NOISE

Hereby we compare the results of our proposed method after having applied additive white Gaussian noise in the aggregate signal in varying percentages (5%, 10%, 20%, 30% and 40%). The $n\%$ percentage values means that the power measurements of the aggregated signal (in W) have a deviation of $n\%$ with 68%. Fig. 6 shows the generated values and their respective ground truth, assuming clear aggregated signal (as illustrated in the up left diagram) and aggregated signal, corrupted with 5%, 10%, 20%, 30% and 40% Gaussian noise, respectively. As noticed, the aggregated power values are differentiated from the real ones, according to the percentage of the Gaussian noise, and gradually "new" peak values in the aggregate timeseries are appeared, that could lead to an erroneously detection of an appliance in operation or to degrade the quality of the generated appliance power signal. In Fig. 6, the aggregated signal distortion, is observed, for the different percentages of the Gaussian noise. For example, on the 6th of May, at 6:00 AM, the aggregated signal has its maximum, with peak values reaching the 4000 W, while, after adding the Gaussian noise (40%), the same maximum is now higher (6000 W). Despite of the additive Gaussian noise, it appears that the GAN's performance isn't affected.

Dishwasher appliance has a duration in operation of about a half hour or even 1 h. The Gaussian noise, distorts the aggregated signal, however, this distortion is not enough to fool the EnerGAN++ and force it to fault detection. In the next step, the study examines the behaviour of the EnerGAN++ model in detecting appliances in operation for short time, since the peak values from the Gaussian noise act as a catalyst for techniques that detect the appliance pattern into the aggregated signal. Toaster, an appliance that remains ON for a short period of time (some minutes), has been selected. Fig. 7 compares the generated appliances power values (dark green line) using as input an aggregate signal corrupted with Gaussian noise (gray filled color), with the respective estimations after applying the traditional LSTM (yellow line) and CNN (orange line) methods. In contrast to EnerGAN++ method that performs learning in an adversarial framework (between generator and discriminator), CNN and LSTM methods try to catch the appliance signal from the aggregated signal using non-linear regression relationships. Thus, a possible aggregated signal distortion, leads to erroneous appliance signal detection and the opposite one. This is obvious, in Fig. 7, where the CNN and LSTM architectures underestimate the toaster appliance power values during operation at 8:00 AM, as the Gaussian noise increases from 5% to 40%. In addition, fault detection seems to be appeared for LSTM and CNN models. The right

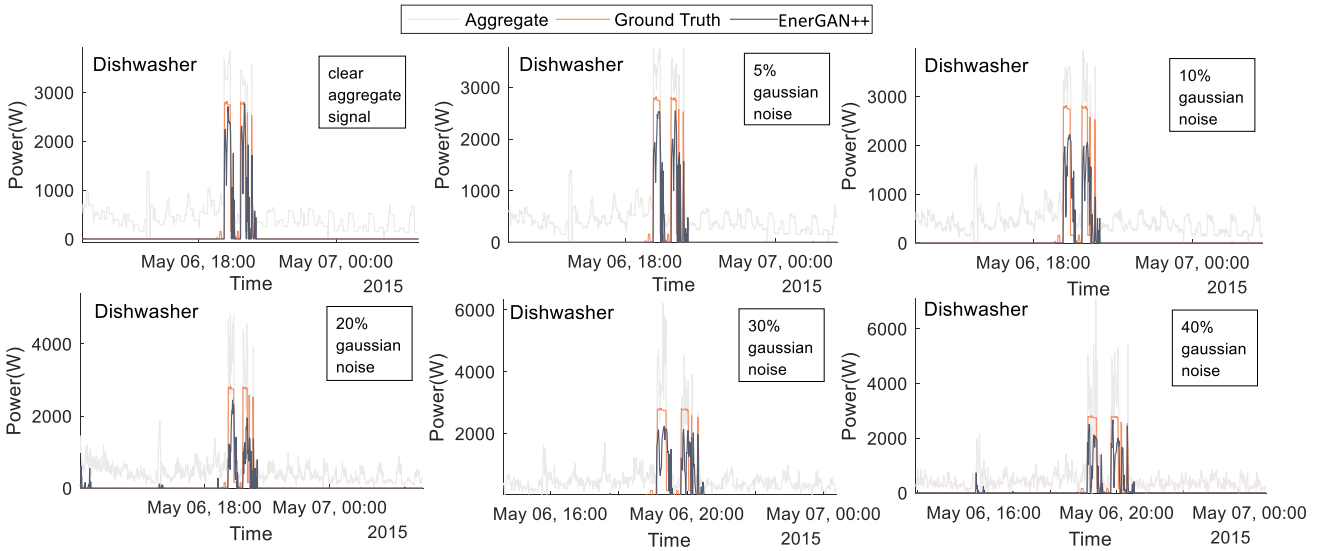

FIGURE 6. **Generated power appliance samples for dishwasher appliance (REFIT). The diagrams show the GAN's model robustness to noise, for all the cases (adding Gaussian noise in 5%, 10%, 20%, 30%, and 40%).**

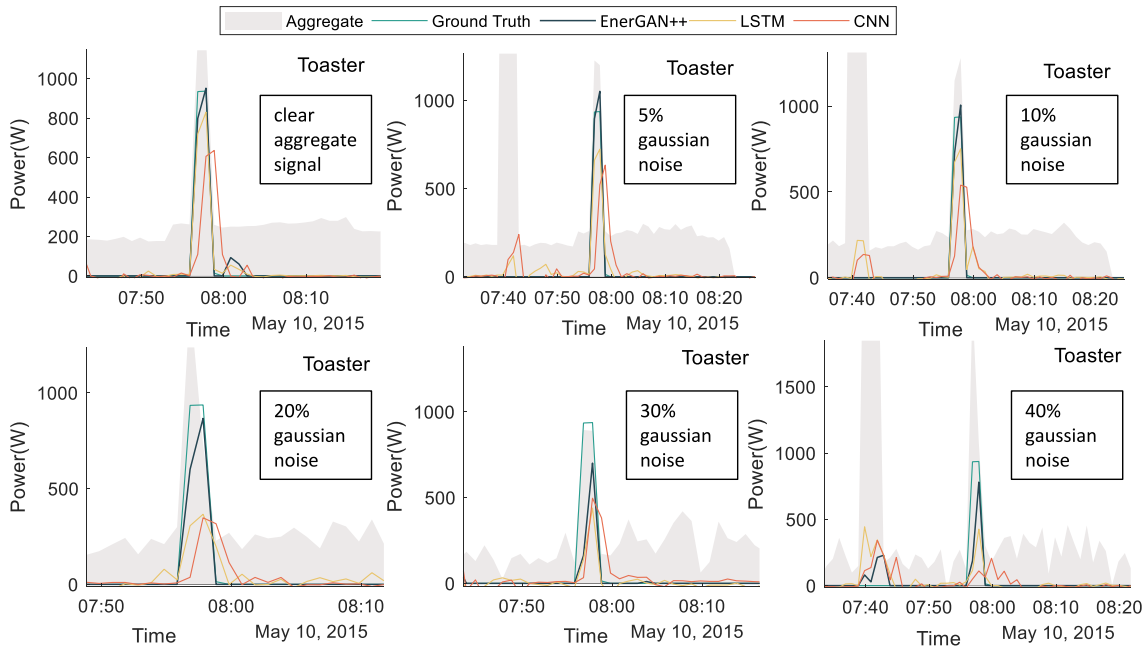

FIGURE 7. **Generated power values for toaster appliance, The diagrams indicate that the proposed model is robust to noise, whereas the baselines are weak for the cases with noise.**

down diagram (40%) is indicative of the CNN method's "confusion". In particular, it appears that the CNN model detects the toaster appliance in operation, two times in the time interval between 7:45 and 8:00 AM, and in addition, the peak power values for the appliance in operation reach almost the 400 W, whereas the actual power values should be of about 900 W.

Fig. 8 shows the MAE metric results for noise in the aggregate signal in all cases (5%, 10%, 20%, 30%, 40%) between the proposed approach and the LSTM and CNN methods, per appliance. The quantitative results on AMPds and REFIT datasets, shown that the EnerGAN++ approach is robust to noise in the aggregate signal. Here, it is nevertheless necessary to remind that the smaller MAE error values provide a better performance. It is indicative that in most appliances, the rate of change of the MAE is lower than the increase in Gaussian error (for further details see Table 2).

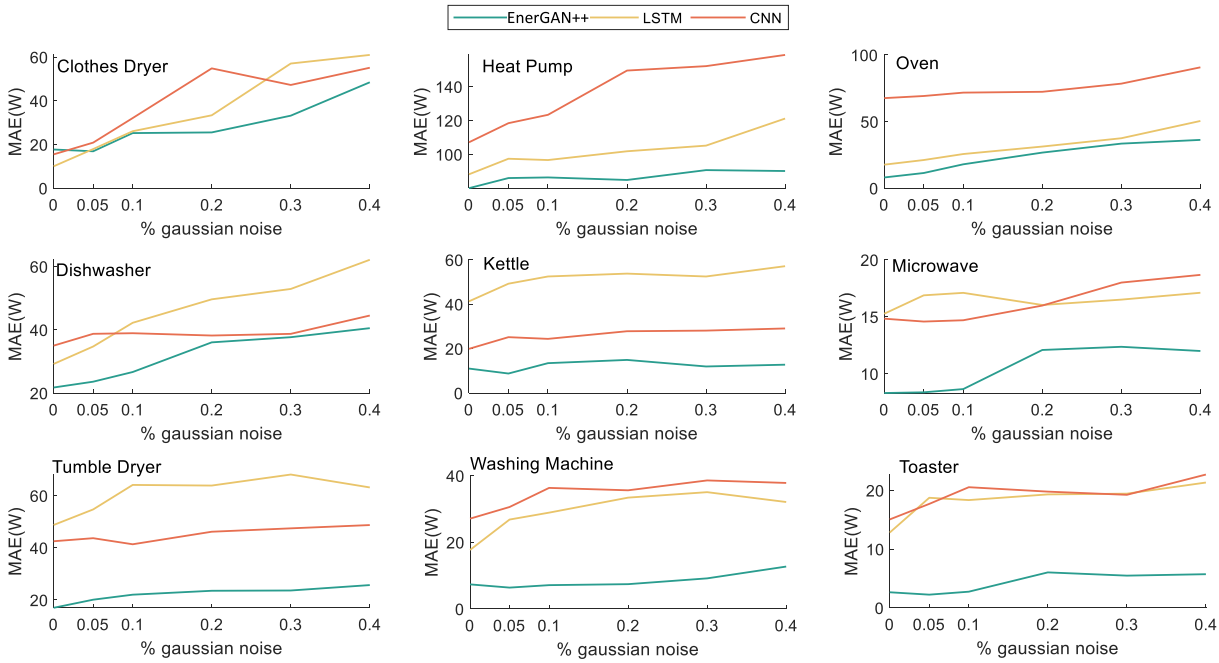

FIGURE 8. Quantitative results on AMPds and REFIT datasets. The small MAE error values provide better performance.

TABLE 2. Rate of Change in MAE Error With an Increase on Gaussian Noise. the Comparison is Between the Proposed EnerGAN++ Method and the BabiLSTM As Well As Sequence to Sequence CNN Methods

	Appliance	**Clothes dryer**			**Heat pump**			**Oven**		
	Method	*EnerGAN++*	*BabiLSTM*	*seq2seqCNN*	*EnerGAN++*	*BabiLSTM*	*seq2seqCNN*	*Proposed*	*BabiLSTM*	*seq2seqCNN*
Δn	$(0-5)\%$	-0.16	1.58	1.07	1.20	1.84	2.26	0.67	0.70	0.32
	$(5-10)\%$	1.67	1.63	2.25	0.07	-0.15	0.99	1.31	0.90	0.51
	$(10-20)\%$	0.03	0.73	2.27	-0.15	0.52	2.61	0.88	0.56	0.06
	$(20-30)\%$	0.77	2.37	-0.76	0.58	0.33	0.26	0.67	0.62	0.61
	$(30-40)\%$	1.53	0.39	0.79	-0.05	1.60	0.66	0.28	1.30	1.22
	$\overline{\Delta r}$	**0.83**	1.34	1.43	**0.41**	0.89	1.36	0.76	0.82	**0.55**
	Appliance	**Dishwasher**			**Kettle**			**Microwave**		
	Method	*EnerGAN++*	*BabiLSTM*	*seq2seqCNN*	*EnerGAN++*	*BabiLSTM*	*seq2seqCNN*	*Proposed*	*BabiLSTM*	*seq2seqCNN*
Δn	$(0-5)\%$	0.37	1.10	0.75	0.19	1.59	1.05	0.01	0.32	-0.05
	$(5-10)\%$	0.61	1.50	0.04	0.93	0.65	-0.15	0.06	0.04	0.02
	$(10-20)\%$	0.94	0.74	-0.07	0.15	0.13	0.33	0.34	-0.11	0.13
	$(20-30)\%$	0.16	0.33	0.05	-0.29	-0.13	0.03	0.03	0.05	0.20
	$(30-40)\%$	0.29	0.92	0.58	0.08	0.46	0.10	-0.04	0.06	0.07
	$\overline{\Delta r}$	0.47	0.92	**0.30**	**0.33**	0.59	0.34	**0.09**	0.12	0.09
	Appliance	**Tumble dryer**			**Washing machine**			**Toaster**		
	Method	*EnerGAN++*	*BabiLSTM*	*seq2seqCNN*	*EnerGAN++*	*BabiLSTM*	*seq2seqCNN*	*Proposed*	*BabiLSTM*	*seq2seqCNN*
Δn	$(0-5)\%$	0.62	1.19	0.23	-0.19	1.82	0.70	-0.08	1.18	0.53
	$(5-10)\%$	0.38	1.88	-0.47	0.15	0.41	1.15	0.10	-0.08	0.57
	$(10-20)\%$	0.15	-0.02	0.48	0.03	0.45	-0.07	0.33	0.10	-0.08
	$(20-30)\%$	0.01	0.42	0.13	0.17	0.16	0.30	-0.06	0.01	-0.05
	$(30-40)\%$	0.21	-0.49	0.13	0.35	-0.30	-0.08	0.02	0.19	0.35
	$\overline{\Delta r}$	**0.27**	0.80	0.29	**0.18**	0.63	0.46	**0.12**	0.31	0.31

Fig. 9 compares the results of EnerGAN++ model with the corresponding results provided by the sequence to subsequence conditional GAN model [31], given noisy aggregate signal values, for the REFIT dataset. EnerGAN++ is a robust to noise model and achieves good performance, regardless of noise, in contrast to the SEQ2SUB model. SEQ2SUB model achieves good results in normal cases as shown in [31], however, given noise aggregate signal input, its performance is degraded (Fig. 9). Fig. 10 compares the result of EnerGAN model and WaveNILM [14] model given noisy aggregated inputs, for the AMPds dataset. The results using WaveNILM method are significantly degraded with the noisy inputs.

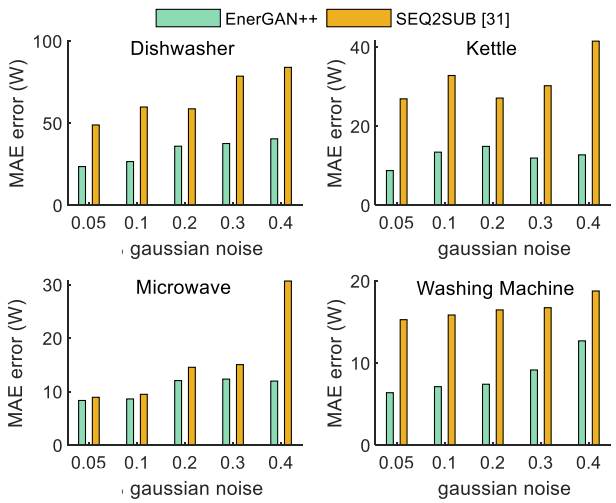

FIGURE 9. Comparison between EnerGAN++ model results provided after applying Gaussian noise to the aggregate signal and the respective results of the disaggregation using the sequence to sub-sequence conditional GAN model of [31]. The small MAE error values provide better performance.

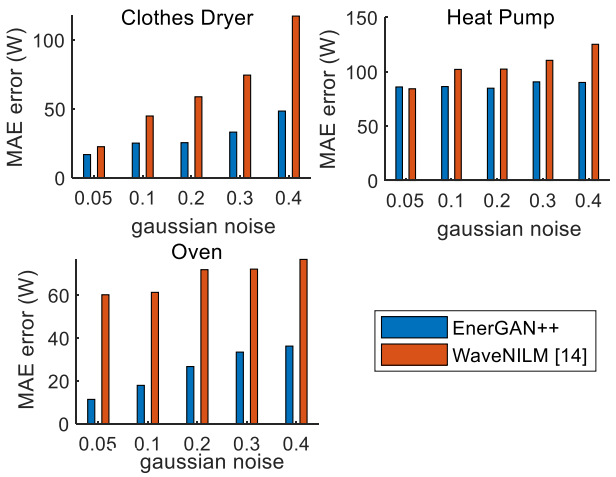

FIGURE 10. Comparison between EnerGAN++ model results provided after applying Gaussian noise to the aggregate signal and the respective results of the disaggregation using the WaveNILM model of [14]. The small MAE error values provide better performance.

Table 2 shows the MAE error rate of change Δr for the respective increase of Gaussian noise ((0−5)%, (5−10)%, (10−20)%, (20−30)%, (30−40)%), as well as the average rate of change $\overline{\Delta r}$ for all the above mentioned cases.

$$\Delta r = \frac{MAE_{n_j\%} - MAE_{n_i\%}}{n_j - n_i}, \quad n_j > n_i \tag{16}$$

$$\overline{\Delta r} = \frac{\sum_1^N |\Delta r|}{N} \tag{17}$$

As observed, the change of MAE by an additional one percentage point, referred as $\overline{\Delta r}$, is lower in our proposed case, rather than in the two other methods (BaBiLSTM and seq2seqCNN).

VII. CONCLUSION

In this paper, we attempt to incorporate the autoencoder architecture under a generative adversarial network, to achieve the inverse mapping and extract meaningful appliance power consumption signal from the generator. Furthermore, we propose a novel discriminator for sequence classification, that successfully distinguishes the generated appliance sequences from the real ones. A hybrid CNN-GRU model ensures that the discriminator creates proper features and exploits the temporal information of the timeseries. In the discriminator, the appliance power signal is paired with the aid of the aggregated signal, to assist the discriminator to properly separate the generated from the real values. Experimental results indicate the proposed method's superiority compared to the current state of the art.

REFERENCES

[1] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context aware energy disaggregation using adaptive bidirectional LSTM models," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3054–3067, Jul. 2020.

[2] M. Kaselimi, N. Doulamis, A. Doulamis, A. Voulodimos, and E. Protopapadakis, "Bayesian-optimized bidirectional LSTM regression model for non-intrusive load monitoring," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 2747–2751.

[3] D. Murray, L. Stankovic, V. Stankovic, S. Lulic, and S. Sladojevic, "Transferability of neural network approaches for low-rate energy disaggregation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 8330–8334.

[4] M. D'Incecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1419–1429, Mar. 2020.

[5] K. Barsim, R. Streubel, and B. Yang, "An approach for unsupervised non-intrusive load monitoring of residential appliances," in *Proc. 2nd Int. Workshop Non-Intrusive Load Monit.*, 2014, pp. 1–5.

[6] I. Goodfellow *et al.*, "Generative adversarial nets," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[7] T. Kaneko, Y. Ushiku, and T. Harada, "Label-noise robust generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2467–2476.

[8] A. Creswell and A. A. Bharath, "Inverting the generator of a generative adversarial network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1967–1974, Jul. 2018.

[9] M. Kaselimi, A. Voulodimos, E. Protopapadakis, N. Doulamis, and A. Doulamis, "Energan: A generative adversarial network for energy disaggregation," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2020, pp. 1578–1582.

[10] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy-Efficient Built Environ.*, 2015, pp. 55–64.

[11] J. Kim, T. Le, and H. Kim, "Nonintrusive load monitoring based on advanced deep learning and novel signature," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–22, 2017.

[12] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, 2018.

[13] M. Kaselimi, E. Protopapadakis, A. Voulodimos, N. Doulamis, and A. Doulamis, "Multi-channel recurrent convolutional neural networks for energy disaggregation," *IEEE Access.*, vol. 7, pp. 81047–81056, 2019.

[14] A. Harell, S. Makonin, and I. V. Bajić, "Wavenilm: A causal neural network for power disaggregation from the complex power signal," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 8335–8339.

[15] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2604–2611.

[16] K. Chen, Q. Wang, Z. He, K. Chen, J. Hu, and J. He, "Convolutional sequence to sequence non-intrusive load monitoring," *J. Eng.*, vol. 2018, no. 17, pp. 1860–1864, 2018.

[17] R. Bonfigli, A. Felicetti, E. Principi, M. Fagiani, S. Squartini, and F. Piazza, "Denoising autoencoders for non-intrusive load monitoring: Improvements and comparative evaluation,"*Energy Buildings*, vol. 158, pp. 1461–1474, 2018.

[18] A. Rahimpour, H. Qi, D. Fugate, and T. Kuruganti, "Non-intrusive energy disaggregation using non-negative matrix factorization with sum-to-k constraint," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4430–4441, Nov. 2017.

[19] Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial HMMS with application to energy disaggregation," in *Artif. Intell. Statist.*, 2012, pp. 1472–1482.

[20] N. Batra, Y. Jia, H. Wang, and K. Whitehouse, "Transferring decomposed tensors for scalable energy breakdown across regions," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 740–747.

[21] K. Makantasis, A. Doulamis, N. Doulamis, and A. Nikitakis, "Tensor-based classification models for hyperspectral data analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 6884–6898, Dec. 2018.

[22] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," 2014, *arXiv:1401.4082*.

[23] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.

[24] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5505–5514.

[25] R. Scott, A. Zeynep, Y. Xinchen, L. Lajanugen, S. Bernt, and L. Honglak, "Generative adversarial text-to-image synthesis," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1060–1069.

[26] K. Thekumparampil, A. Khetan, Z. Lin, and S. Oh, "Robustness of conditional gans to noisy labels," in *Adv. Neural Inf. Process. Syst.*, 2018, pp. 10271–10282.

[27] Gr. Chrysos, J. Kossaifi, and S. Zafeiriou, "RoCGAN: Robust conditional GAN," *Int. J. Comput. Vis.*, vol. 128, pp. 2665–2683, 2020.

[28] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2016, *arXiv:1605.09782*.

[29] J. Luo, Y. Xu, C. Tang, and J. Lv, "Learning inverse mapping by autoencoder based generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process.*, 2017, pp. 207–216.

[30] K. Bao, K. Ibrahimov, M. Wagner, and H. Schmeck, "Enhancing neural non-intrusive load monitoring with generative adversarial networks," *Energy Informat.*, vol. 1, no. 1, pp. 295–302, 2018.

[31] Y. Pan, K. Liu, Z. Shen, X. Cai, and Z. Jia, "Sequence-to-subsequence learning with conditional GAN for power disaggregation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 3202–3206.

[32] K. Chen, Y. Zhang, Q. Wang, J. Hu, H. Fan, and J. He, "Scale- and context-aware convolutional non-intrusive load monitoring," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2362–2373, May 2019.

[33] M. N. Fekri, A. M. Ghosh, and K. Grolinger, "Generating energy data for machine learning with recurrent generative adversarial networks," *Energies*, vol. 13, no. 1, p. 130, 2020.

[34] A. Harell, R. Jones, S. Makonin, and I. V. Bajic, "PowerGAN: Synthesizing appliance power signatures using generative adversarial networks," 2020, *arXiv:2007.13645*.

[35] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2625–2634.

[38] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE IRE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[39] Y. Zhu and S. Lu, "Load profile disaggregation by blind source separation: A wavelets-assisted independent component analysis approach," in *Proc. 2014 IEEE PES Gen. Meeting—Conf. Expo.*, 2014, pp. 1–5.

[40] L. Pandey, A. Kumar, and V. Namboodiri, "Monoaural audio source separation using variational autoencoders," in *INTERSPEECH*, 2018, pp. 3489–3493.

[41] E. M. Grais and M. D. Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," in *Proc. 2017 IEEE Glob. Conf. Signal Inf. Process.*, 2017, pp. 1265–1269.

[42] A. Doulamis and N. Doulamis, "Generalized nonlinear relevance feedback for interactive content-based retrieval and organization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 656–671, May 2004.

[43] N. Doulamis, A. Doulamis, and T. Varvarigou, "Adaptive algorithms for interactive multimedia," *IEEE MultiMedia*, vol. 10, no. 4, pp. 38–47, Oct.–Dec. 2003.

[44] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," 2020, *arXiv:2001.06937*.

[45] Q. Lei, A. Jalal, I. Dhillon, and A. Dimakis, "Inverting deep generative models, one layer at a time," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13910–13919.

[46] Y. Teng and A. Choromanska, "Invertible autoencoder for domain adaptation," *Computation*, vol. 7, no. 2, p. 20, 2019.

[47] W. Pei, T. Baltrusaitis, D. Tax, and L. Morency, "Temporal attention-gated model for robust sequence classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6730–6739.

[48] E. Protopapadakis, A. Voulodimos, A. Doulamis, N. Doulamis, D. Dres, and M. Bimpas, "Stacked autoencoders for outlier detection in over-the-horizon radar signals," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–12, 2017.

[49] N. Batra *et al.*, "NILMTK: An open source toolkit for non-intrusive load monitoring," in *Proc. 5th Int. Conf. Future Energy Syst.*, 2014, pp. 265–276.

[50] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. Bajić, "AMPds: A public dataset for load disaggregation and eco-feedback research," in *Proc. 2013 IEEE Elect. Power Energy Conf.*, 2013, pp. 1–6.

[51] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study," *Sci. Data*, vol. 4, no. 1, pp. 1–12, 2017.

MARIA KASELIMI received the Diploma from the National Technical University of Athens (NTUA), Greece, in 2015 and the M.Sc. degree from NTUA in the field of geoinformatics, in 2017. She is the Ph.D. candidate with NTUA, in 2018 and she is working as Junior Researcher in several european projects focusing on machine learning, signal processing techniques, data analysis and modeling with applications in the fields of energy, geosciences and environment.

NIKOLAOS DOULAMIS (Member, IEEE) received the Diploma and Ph.D. degree in electrical and computer engineering from the National Technical University of Athens (NTUA) both with the highest honor. He is currently an Associate Professor with the NTUA. He has received many awards (e.g., Best Student among all Engineers, Best Paper Awards). He was an Organizer and/or TPC in major IEEE conferences. He has authored more than 75 (240) journals (conference) papers in the field of signal processing and machine learning and received more than 5900 citations. He is involved in large scale European projects such as H2020 Stop-It and H2020 Beneffice.

ATHANASIOS VOULODIMOS (Member, IEEE) received the Dipl.-Ing., M.Sc., and Ph.D. degrees from the School of Electrical and Computer Engineering of the National Technical University of Athens (NTUA) ranking at the top of his class. He is currently an Assistant Professor with the Department of Informatics and Computer Engineering of the University of West Attica. He has been involved in several European research projects, as a Senior Researcher and a Technical Manager. He was recipient of the awards for his academic performance and scientific achievements and has coauthored more than 90 papers in international journals, conference proceedings and books in the research areas of machine learning and signal processing receiving more than 1900 citations.

ANASTASIOS DOULAMIS (Member, IEEE) received the Diploma and Ph.D. degree in electrical and computer engineering from the National Technical University of Athens (NTUA) with highest honor. Until January 2014, he was an Associate Professor with the Technical University of Crete and currently is an Assistant Professor with NTUA. He was the receipient of the several awards in his studies, including the Best Greek Student Engineer, Best Graduate Thesis Award. He has also served as program committee in several major conferences of IEEE and ACM. He has authored more than 350 papers in leading journals and conferences receiving more than 5800 citations.

EFTYCHIOS PROTOPAPADAKIS studied Production Engineering and Management with the Technical University of Crete. He received the M.S. degree in management and business administration and the Ph.D. degree in decision systems both from the same university. He has worked as an Engineer in European (4D-CH-World, BENEFFICE, eVACUATE, ROBO-SPECT, Terpsichore, WaterSpy) and Intereg (e-Park, Poseidon) projects since 2010. He has coauthored more than forty publications receiving more than 1400. His research interests include machine learning applications. He has explored the applicability of semi-supervised techniques in maritime surveillance, energy applications, elder people support, industrial workflow monitoring, structural assessment of tunnel infrastructures, and cultural heritage applications.