POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Deep Autoencoders for Anomaly Detection in Electricity Consumption of Buildings

Tesi di Laurea Magistrale in
Computer Science and Engineering - Ingegneria Informatica

Antonio Guadagno, 10561018
Claudio Alfredo Emanuele, 10682738

Advisor:
Prof. Francesco Amigoni

Co-advisors:
Davide Azzalini
Benedetta Flammini

Academic year:
2021-2022

Abstract:
Topics of environmental pollution and energy saving are among the most discussed in the recent years. It is now clear to everybody that these issues cannot be ignored any longer as consequences would be catastrophic: from rising temperatures to the destruction of the most fragile ecosystems. According to the United Nations Environment Programme, buildings are responsible for 38% of the global energy use. This, combined with the rising costs of energy, provides us a good reason to devise effective new ways to prevent energy wastes in buildings. One of the possible approaches is to detect faults and abnormal patterns in the electricity consumption of buildings via anomaly detection algorithms. Among all the different anomaly detection algorithms, this thesis focuses on Autoencoders, semi-supervised deep learning neural networks that provide state-of-the-art detection performance. The goal of this thesis is to present an innovative network topology that outperforms the state-of-the-art methods by solving the bypassing phenomenon affecting Variational Autoencoders with Self-Attention. The bypassing problem is caused by the Self-Attention mechanism which does not perform at its best when working with real world and noisy data. To solve it, we have designed the Convolutional Variational Self-Attention mechanism, an innovative alternative to the traditional Self-Attention mechanism whose performance does not deteriorate even when working with noisy datasets. We provide an extensive experimental comparison on both a univariate and a multivariate dataset which shows that the proposed network topology offers better performance with respect to the state-of-the-art methods, and that, in case of noisy datasets, simpler models can work better than more complex ones.

Key-words: Anomaly Detection, Electricity Consumption, Deep Autoencoders, Self-Attention

1. Introduction

Topics of environmental pollution and energy saving are among the most discussed in the recent years. It is now clear to everybody that these issues cannot be ignored any longer as consequences would be catastrophic: from rising temperatures to the destruction of most fragile ecosystems. Many of the world's leading countries are

putting these issues at the core of their political agendas and 'year 2050' is becoming an increasingly concrete and worrying deadline. A solution must be found.
While studying the main causes of environmental pollution, the United Nations Environment Programme has reported that buildings are responsible for 38% of the global energy use and for around 40% of the global CO_2 emissions [1]. This, combined with the rising costs of energy due to the Russian invasion of Ukraine, provides us a good reason to devise effective new ways to prevent energy wastes.
One of the possible approaches that can be adopted to reduce energy wastes is to detect faults and abnormal patterns in the electricity consumption of buildings [2]. In this way, when a fault is detected, building managers can be timely alerted about possible problems and can take action to solve them. In order to detect faults and abnormal patterns in the electricity consumption of buildings, anomaly detection algorithms can be used.
Among all the possible anomaly detection algorithms, this thesis mainly focuses on semi-supervised learning approaches, in particular, on Autoencoders, deep learning neural networks that provide state-of-the-art detection performance and that can be adapted to work with a wide variety of input modalities.
The goal of this thesis is to design an innovative network topology that outperforms current technologies when applied to the detection of anomalies in the electricity consumption of buildings, and that will lead to a reduction in energy wastes, costs, and environmental pollution.

The main original contributions of this thesis are:

- an innovative network topology that outperforms the state-of-the-art methods by solving the bypassing phenomenon affecting Variational Autoencoders with Self-Attention,
- an extensive experimental comparison comprising many baselines and the state-of-the-art methods, considering both a univariate and a multivariate dataset.

This thesis is organized as follows. In Section 2 the related work is presented, providing an overview of different approaches to anomaly detection tasks. In Section 3 our approach to the problem is presented together with the explanation of how the proposed network topology solves the bypassing problem affecting the current state-of-the-art methods based on Variational Autoencoders with Self-Attention. In Section 5 experimental results related to the univariate dataset are presented, while, in Section 6 we present experimental result related to the multivariate dataset.
Some parts of this thesis are included in an article titled "Empirical Evaluation of Deep Autoencoders for Anomaly Detection in Electricity Consumption of Buildings", currently under major revision for IEEE Intelligent Systems.

2. Related Work

Anomaly detection [3] is the task of finding patterns in data that do not conform to the expected behavior. Thanks to anomaly detection systems for energy consumption, building managers can be promptly informed of potential issues and take action to prevent energy wastes.
When dealing with time series, anomaly detection tasks present some difficulties. The most important one is due to the natural sequentiality of data: observations are recorded at some frequency in an orderly fashion, and their correlation in time cannot be ignored.
Moreover, in the specific case of electricity consumption, information such as the hour of the day, the day of the week, or the season are fundamental to correctly interpret the data. Additionally, many other factors such as the temperature regulation via heating, ventilation, and air conditioning systems have to be considered.
To perform anomaly detection, data-driven approaches are used. Data-driven approaches require the existence of past data which will be used to train a model that identifies anomalies. When dealing with data-driven approaches, it is possible to distinguish between unsupervised, semi-supervised, and supervised learning. The main difference between the three is that unsupervised methods do not require the existence of a labeled dataset, while supervised methods do. Semi-supervised learning algorithms require labels for the nominal class only.
Although we have mainly focused on Autoencoders and Variational Autoencoders as they provide state-of-the-art performance, in the following we want to provide a more comprehensive overview of all the learning paradigms just mentioned, concentrating on anomaly detection on electricity consumption data.

2.1. Unsupervised Learning

Unsupervised learning uses machine learning algorithms on unlabeled datasets. These algorithms aim at discovering hidden patterns in data without the need for human intervention. Anomalies represent outliers that are unknown to the user during the training stage, therefore the detection of anomalous power consumptions is reduced to the modeling of the normal consumption behavior, in addition to the definition of specific metrics

used to classify data samples as anomalous or not. The main advantage of using unsupervised methods is that they can work with unlabeled datasets (the labeling process is very expensive) but, on the other hand, they usually make the implicit assumption that anomalous instances are far more rare (less than 20% [4]) and sufficiently different from nominal data, which may not accurately reflect reality, especially when it comes to electricity consumption data [2].

2.1.1 Clustering

Clustering is an unsupervised learning method used to divide samples of a dataset into a certain number of groups (or clusters) in such a way that data points belonging to the same cluster have similar characteristics while data points belonging to different clusters have different characteristics. In the specific field of anomaly detection, the goal is to divide nominal samples from anomalous ones.
In [5], authors describe a two-step clustering algorithm whose aim is to distinguish between anomalous and nominal data points in the field of electricity power consumption. In the first step, an anomaly score for each user is periodically evaluated considering only his energy consumption and its variations in the past. Then, in the second step, the anomaly score of each user is adjusted taking into account the energy consumed by his neighborhood. In [6], the Mutual K-Nearest-Neighbor algorithm and the K-means clustering algorithm are used to pre-process power consumption data (reducing the number of samples to be processed), and then the consumption patterns are analyzed to detect abnormal behaviors and malicious customers.

2.1.2 Dimensionality Reduction

Dimensionality reduction techniques are used to transform data from a high-dimensional space into a low-dimensional space while preserving the most important information. Dealing with anomaly detection tasks, dimensionality reduction techniques can be used during the pre-processing phase to perform features extraction and engineering. These features are then provided as input to a classifier that identifies anomalies.
Linear Discriminant Analysis (LDA) is one of the most famous techniques to reduce the dimensionality of a dataset, and it is used in [7] to pre-process data. The authors present a model that automatically labels power consumption patterns with reference to their corresponding categories. This has been accomplished via the use of discriminant weights to separate the hyperplanes generated by the LDA statistical learning. In [8], a variant of the Principal Component Analysis (another famous technique to reduce the dimensionality of a dataset) is used to estimate the principal components of every consumption category. Then, a classifier that detects anomalous power consumptions is created via projecting power patterns on the subsets obtained by those principal components related to the two main categories (i.e., nominal and anomalous).

2.2. Supervised Learning

Considering anomaly detection tasks, supervised learning approaches are used to train models that allow to classify new data as “nominal” or “anomalous”. To do that, annotated datasets are needed during the training phase so that the model can check it has assigned the correct class label to the samples. There are two different ways to perform anomaly detection via supervised learning methods. The first one is to train a supervised classifier that is able to identify the input sequence as anomalous or not, providing as output a class label. On the other hand, the second approach is to train a regressor that, given an input, provides as output a numerical result which, compared to a threshold, allows us to identify anomalies assigning them to the correct class.
Although supervised anomaly detection can achieve high-accuracy identification results, its adoption in the real world is still limited compared to unsupervised methods, due to the absence of power consumption annotated datasets (the labeling process can be very expensive especially for large datasets) [2, 4, 5].

2.2.1 Regression

Regression aims at finding a relationship between independent variables and a dependent one. In the context of anomaly detection in electricity consumption, regression is used to identify anomalous power consumption patterns starting from previously collected anomalous footprints.
In [9], authors adopt regression-based approaches to find anomalies in the electricity consumption of premises and to clear them in order to provide precise assessments of energy consumption patterns. A similar approach is used in [10] to find anomalous energy consumption patterns by analyzing the smart meters temporal data streams. In particular, support vector regression with radial basis function is used to perform predictions on the electricity consumption, then the difference between the actual and the predicted energy consumption is evaluated to find anomalies. In [11] a two-stage processing, namely prediction and then anomaly detection, is used to identify anomalies. In particular, a hybrid neural network ARIMA model of daily consumption is used

to make predictions while, in the second step, a two-sigma rule was adopted to localize the anomalies via the evaluation of the mismatch between real and predicted consumption.

2.2.2 Traditional Classification

Traditional classification algorithms are supervised learning techniques used to train models that will be used to classify new observations on the basis of an annotated dataset. As for regression tasks, a dataset in which both nominal and anomalous samples are labeled is needed to train a model that will be taught to classify samples never seen before as nominal or abnormal.
The K-Nearest Neighbor (KNN) algorithm is a non-parametric supervised learning classifier which uses proximity to classify new observations. In [8, 12], KNN based heuristics are proposed to detect abnormal power consumptions. SVM (Support Vector Machine) is a supervised machine learning algorithm that tries to find a hyperplane that creates a boundary between nominal and anomalous data. In [13, 14], SVM is deployed to detect abnormalities due to energy theft attacks. In the same direction, in [15], a genetic SVM model is proposed to detect abnormal consumption data and suspicious customers. Decision trees are non-parametric supervised learning algorithms whose structure consists of a root node, internal nodes, branches and leaf nodes associated to classes (dealing with anomaly detection, nominal class and anomalous class). Given a data sample, its attributes will define a path from the root of the tree to one of its leaves. The sample will be classified accordingly to the class label associated to the leaf reached. In [16], a decision tree-based solution is introduced to learn energy consumption anomalies triggered by fraud energy usage. Similarly, in [17], an improved decision tree model is developed to detect anomalous consumption data using densities of the anomaly and nominal classes. Another way to perform classification tasks is to use naïve Bayes algorithms, which analyze the attributes of a data point in order to compute the probability that a sample belongs to each of the possible classes. In [18, 19], naïve Bayes algorithms are proposed to detect the abnormalities generated by electricity theft attacks.

2.3. Semi-Supervised Learning

Semi-supervised learning algorithms require training datasets in which nominal samples only are labeled and, for this reason, they are more widely applicable than supervised methods.

2.3.1 One-Class Learning

For what concerns the specific field of anomaly detection, the goal of One-Class Learning (OCL) is to train a classifier on a dataset containing nominal samples only (anomalous samples are not present in the training set). Once trained, the model is used to classify new samples as nominal or anomalous, even though abnormal data points were never shown to the model during the training phase.
In [20], a One-Class Support Vector Machine (OCSVM) is introduced to identify the smallest hypersphere encompassing all the power observations. In [21], a kernel based One-Class Neural Network (OCNN) is proposed to detect abnormal power consumptions. It merges the capability of Deep Neural Networks (DNN) to derive progressively richer representations of power signals with OCL, building a tight envelope surrounding nominal power consumptions patterns.

2.3.2 Autoencoders

One common way of using semi-supervised learning in anomaly detection tasks is to train deep learning models using data samples with no anomalies. In this way, we obtain models that would produce low reconstruction errors for nominal instances and high reconstruction errors for anomalous samples [2, 4, 5, 22] allowing us to easily identify abnormal data points. Among the most used deep learning models we find Autoencoders (AEs), whose popularity is due to the fact that they provide state-of-the-art detection performance and can be adapted to work with a wide variety of input modalities [23].
Autoencoders [24] are neural networks [25] trained to reconstruct the input in a self-supervised manner. They are made of an encoder followed by a decoder. The encoder produces a latent representation of the input containing meaningful information only, and the decoder, starting from the latent representation produced by the encoder, provides as output a reconstruction of the input and without noise and anomalies.
In [26], a model made of two sub-networks (a Convolutional Autoencoder for the reconstruction of unlabeled data and a classifier for the labeled ones) is presented. Data points are marked as anomalous when the joint model has low predictive variance, high predictive mean, and high reconstruction error. In [27], an LSTM AE (see Appendix A.0.2) is used to identify anomalies and reduce energy wastes in manufacturing companies. In [28], a Bi-LSTM Autoencoder detects anomalies in smart-metering data while in [29] an LSTM Autoencoder is trained to identify anomalies in a smart campus by employing an ensemble method to label data.

Autoencoders can be used not only to directly find anomalies, but also as features extractors capable of collecting most important features of a given input. These features will then be provided to a classifier that will label the data samples as anomalous or not.
In [30], an Autoencoder is used to extract the high-level electricity consumption information of residential users to improve the anomaly detection performance of the actual detector. The extracted features are provided as input to a one-class SVM model for anomaly detection. In [31], an Autoencoder is used to extract the high-level representation from massive amounts of monitoring data acquired automatically from an actual smart meter network. Then a deep neural network is used to detect anomalies and send alarm messages using web technologies.

2.3.3 Variational Autoencoders

Variational Autoencoders (VAEs) [32] differ from plain AEs in the fact that the latent representation of the input is sampled from a Gaussian distribution whose mean and variance are provided as outputs by the encoder. In [33], a Variational Bi-LSTM Autoencoder with a Variational Self-Attention mechanism to detect anomalies is proposed. A similar approach is followed in [34], with a variational LSTM Autoencoder with a fully connected attention mechanism. Both methods define an anomaly score based on reconstruction probability. Castangia et al. [35] propose a variational Convolutional Autoencoder to detect energy consumption anomalies employing a reconstruction error-based anomaly score. In [36] a VAE is used to identify anomalies in datasets that will then be used to train models for the load forecasting of smart cities.

2.4. Mathematical Background

In this section we present the mathematical background necessary to understand the design choices behind the proposed network topology.

2.4.1 Autoencoders

As stated before, Autoencoders (AEs) [24] are neural networks [25] trained to reconstruct their input in a self-supervised manner. An AE is composed of an encoder network and a decoder network. The encoder takes as input the data $\boldsymbol{x} \in \mathbb{R}^d$, where d is the dimension of the data, and compresses these data into a latent space $\boldsymbol{z} \in \mathbb{R}^h$, where h is the dimension of the encoding (latent dimension). Usually, $h < d$. Then, the decoder tries to map back the latent internal representation z to the original input space $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ through reconstruction. The encoder structure can be considered as a bottleneck, in which data pass and are compressed to extract a meaningful encoded representation. The decoder does the opposite. The two networks are characterized by f_ϕ, the encoding function, and f_θ, the decoding function, where $f_\phi : \mathbb{R}^d \to \mathbb{R}^h$ and $f_\theta : \mathbb{R}^h \to \mathbb{R}^d$. Finding weights (parameters) ϕ and θ for the two functions can be done by backpropagation, minimizing the loss function $\mathcal{L}_{AE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \| \boldsymbol{x} - \hat{\boldsymbol{x}} \|^2$, called *reconstruction error*, given input $\boldsymbol{x}$ and model output $\hat{\boldsymbol{x}}$. Autoencoders can be made of Convolutional Layers (see Appendix A.0.1) or of LSTM Layers (see Appendix A.0.2).

2.4.2 Variational Autoencoders

Variational Autoencoders (VAEs) [32] differ from plain AEs in the fact that they assume the existence of a probabilistic model parametrized by the latent variable $\boldsymbol{z} \in \mathbb{R}^h$ that generates the observed input values $\boldsymbol{x} \in \mathbb{R}^d$. Being $\boldsymbol{z}$ latent (i.e., hidden), we can infer its characteristics by computing the posterior $p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})}$; unfortunately, computing the marginalization $p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$ at the denominator is intractable when $\boldsymbol{z}$ is high-dimensional. Variational inference can be used to overcome this issue by approximating $p(\boldsymbol{z}|\boldsymbol{x})$ with a distribution $q(\boldsymbol{z}|\boldsymbol{x})$ which is tractable, and by minimizing their *KL-divergence* $D_{KL}(q(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}|\boldsymbol{x}))$ (see Appendix A.0.3) to ensure that $q(\boldsymbol{z}|\boldsymbol{x})$ is similar to $p(\boldsymbol{z}|\boldsymbol{x})$. By replacing $p(\boldsymbol{z}|\boldsymbol{x})$ in $D_{KL}(q(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}|\boldsymbol{x}))$ with $\frac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{x})}$, the minimization problem becomes equivalent to the maximization of $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}(q(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))$, where the first term represents the reconstruction likelihood (analogous to the reconstruction error in AEs), while the second term ensures that the learned distribution $q(\boldsymbol{z}|\boldsymbol{x})$ is similar to the true prior distribution $p(\boldsymbol{z})$ (with the effect of regularizing the latent variable $\boldsymbol{z}$). The distributions $q(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z}|\boldsymbol{x})$ can be parametrized by means of two artificial neural networks which can be considered as encoder (with weights ϕ), and decoder (with weights θ), respectively. Finding weights ϕ and θ can be done by backpropagation, minimizing the loss function $\mathcal{L}_{VAE}(\boldsymbol{x}) = D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p_\theta(\boldsymbol{z})) - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}(\log p_\theta(\boldsymbol{x}|\boldsymbol{z}))$, which represents the variational lower bound of the data $\boldsymbol{x}$ according to the Jensen's inequality [32]. The posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is usually assumed to be normally distributed with parameters $\mathcal{N}(\boldsymbol{\mu_z}, \boldsymbol{\Sigma_z})$, while a common choice for the prior distribution $p_\theta(\boldsymbol{z})$ is an isotropic normal distribution $\mathcal{N}(0, I)$. As AEs, VAEs can be implemented using both Convolutional (see Appendix A.0.1) and LSTM Layers (see Appendix A.0.2).

3. Proposed Method

In this section we formalize the problem of anomaly detection in the electricity consumption of buildings, we explain how we address the problem and how the proposed network topology identifies anomalies outperforming the current state-of-the-art by solving the bypassing phenomenon affecting Variational Autoencoders with Self-Attention.

3.1. Problem Definition

The goal of this thesis is to propose a new method to perform online anomaly detection on electricity consumption of buildings.
If we represent by $\mathbf{O}^o = [\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, ...]$ a possibly infinite stream of data, online anomaly detection is the task of classifying the data point at time t as nominal or anomalous by considering a portion of the stream up to t. We represent by $\mathbf{O} = [\mathbf{o}_1, ..., \mathbf{o}_T]$ a D-dimensional portion of the stream composed of T observations of electricity consumptions. Each observation $\mathbf{o}_s$ is a D-dimensional vector representing the multivariate observation collected by smart meters at time s. In our case, T is the number of samples collected in a week, and the last data point belonging to the portion of the stream $\mathbf{O} = [\mathbf{o}_1, ..., \mathbf{o}_T]$ is the one that we want to classify as nominal or anomalous. The choice of setting T equal to the number of of samples collected in a week is due to the fact that electricity consumption data usually present a weekly seasonality.
Every Q minutes, a new observation is collected by the smart meters and appended to the data stream $\mathbf{O}^o$, then a new input sequence to be analyzed is provided as input to our model.
The input sequences received by our model overlap as we work with a sliding window over the stream of data $\mathbf{O}^o$. In particular, if the last input sequence received by the model was $\mathbf{O} = [\mathbf{o}_1, ..., \mathbf{o}_T]$, the next one will be $\mathbf{O} = [\mathbf{o}_2, ..., \mathbf{o}_{T+1}]$ where $\mathbf{o}_{T+1}$ is the observation collected Q minutes after $\mathbf{o}_T$.
Looking at the related work, it is possible to notice that, in most of the cases, anomaly detection models are designed to classify the entire input sequence $\mathbf{O} = [\mathbf{o}_1, ..., \mathbf{o}_T]$ as nominal or anomalous. We want to be more specific: our goal is to classify each single data point $\mathbf{o}_s$ belonging to the data stream $\mathbf{O}^o$ as abnormal or not. In this way, we avoid having weeks classified as entirely anomalous just for a single spike in the energy consumption plot. The proposed Autoencoder topology is able to to notify the presence of anomalies with an average delay of $Q/2$ minutes + *inference time* from their occurrence.
To train the model we assume to have a training set made of nominal electricity consumptions only.
To successfully perform anomaly detection tasks, the first thing to do is to define what an anomaly is in the specific domain. According to Hawkins [37], an anomaly is "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". In other words, the observation $\mathbf{o}_s$ has to be considered an anomaly if it does not follow the nominal behavior expected at time s.

According to [3, 38, 39], we can group anomalies in three categories:

- Point anomalies: individual instances that are anomalous with respect to the majority of other individual instances.
- Sub-sequence anomalies: a sub-sequence of instances (two or more) that are anomalous with respect to the majority of other individual instances.
- Conditional anomalies: individual instances or sub-sequences of instances that are anomalous considering the specific context, otherwise nominal.

This categorization can be effectively applied to the field of electricity demand as follows:

- Point anomalies: spikes (upward or downward) in the energy consumption.
- Subsequence anomalies: overuse or underuse of energy for a more or less prolonged period of time.
- Conditional anomalies: a low energy consumption is normal on weekends or nights but not during working hours of a working day (from Monday to Friday) or a daily high energy consumption that is normal during the working days but not during the weekends or nights.

The first two categories are mutually exclusive (a point anomaly cannot be a subsequence anomaly and vice versa), but note that anomalies can be both point and conditional anomalies and also subsequence and conditional anomalies.

3.2. Proposed Network Topology

The model we propose (depicted in Figure 1) is a Variational Bi-LSTM Autoencoder with Convolutional Variational Self-Attention. It consists of five blocks:

- Encoder
- Variational Layer
- Convolutional Variational Self-Attention
- Decoder
- Post-Decoder

In the following, we analyze the five above mentioned blocks and we explain how they interact to each other to identify anomalies.

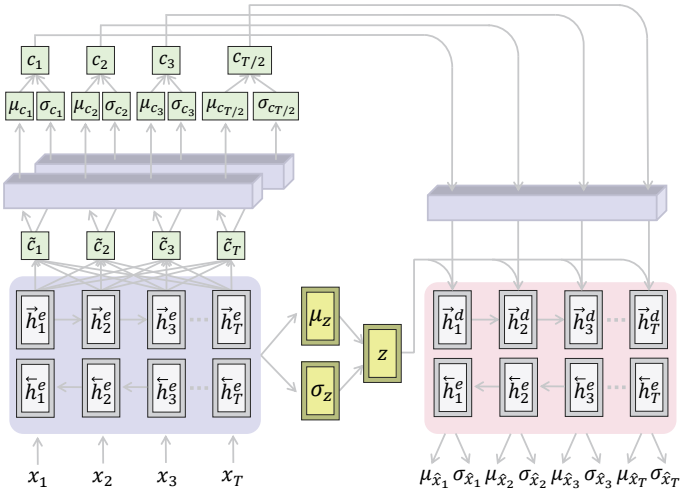

Figure 1: The proposed network topology.

3.2.1 Encoder

The Encoder (depicted in Figure 2) is made of a Bi-LSTM layer (see Appendix A.0.2): this choice is motivated by the fact that Bi-LSTM layers have been specifically designed to work effectively with time series, and to understand how past data points affect the future ones and vice versa.
The model takes as input a tensor having shape (T, D), where T is the number of samples that we collect in a week and D is the number of dimensions we are considering. A random Gaussian noise is then added to the input in order to teach the model how to remove noise. The noisy input is provided to the Encoder which is made of a Bi-LSTM layer having L memory elements. The Bi-LSTM layer generates a sequence of hidden states for each direction: forward and backward. The two sequences of hidden states are then concatenated and provided as input to the Convolutional Variational Self-Attention, while the final hidden state is provided as input to the Variational Layer.

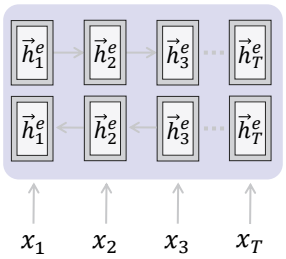

Figure 2: Encoder.

3.2.2 Variational Layer

The Variational Layer takes as input the final hidden state of the Encoder and returns as output a latent representation of the input.
The Variational Layer is made of two Dense layers, one used to compute the mean and one used to compute the logarithm of the variance of an isotropic Gaussian distribution from which the latent representation of the input will be sampled. The Dense layer of the mean is characterized by a linear activation function while the Dense layer of the logarithm of the variance is characterized by a SoftPlus activation function.
In Figure 3 it is shown a partial representation of the proposed network topology reporting the Encoder and the Variational Layer only.

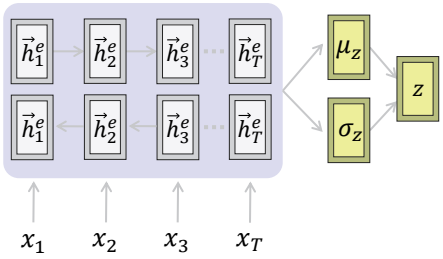

Figure 3: Encoder + Variational Layer.

3.2.3 Convolutional Variational Self-Attention

The Self-Attention mechanism is used to point out which are the relationships between different time stamps of the input, a fundamental aspect when dealing with time series. Thanks to the Self-Attention mechanism, the model is able to understand how input data points are correlated to each other. This information, together with the latent representation of the input, is used to output more accurate reconstructions.
Generally speaking, the Self-Attention mechanism receives as input a sequence of T encoded hidden states and outputs a sequence of T context vectors, all having the same length. Each of these context vectors is computed as the weighted sum of all the encoded hidden states. To go deep in the details, the Self-Attention mechanism works as follow.
First, the relevance of every pair of encoded hidden states h_i^e and h_j^e is scored using the scaled dot product similarity [33]:

$$s_{i,j} = score(h_i^e; h_j^e) = \frac{(h_i^e)^T h_j^e}{\sqrt{L}} \tag{1}$$

where L is the size of the encoder Bi-LSTM state.
Second, the attention weights a_{ij} are computed by normalizing the scores:

$$\mathbf{a}_t = softmax(\mathbf{s}_t) \tag{2}$$

where $\mathbf{a}_t = (a_{t1}, a_{t2}, ..., a_{tT})$ and $\mathbf{s}_t = (s_{t1}, s_{t2}, ..., s_{tT})$.
Finally the context vector at timestamp t, c_t, is obtained as a sum of the encoded hidden states weighted by the attention weights a_{ij}:

$$\mathbf{c}_t = \mathbf{a}_t \cdot \mathbf{h}_t^e \tag{3}$$

Dealing with Autoencoders, a variant of the Self-Attention mechanism, the Variational Self-Attention mechanism, is more often used. The reason is that the Variational Self-Attention mechanism allows to avoid the bypassing phenomenon pointed out by Bahuleyan et al. [40]. In fact, if we consider the standard Self-Attention mechanism, we can notice that the decoder has a direct and deterministic access to the encoder hidden states through the Self-Attention. This means that the latent representation z may not be forced to learn expressive representations of the input, since the Self-Attention mechanism could bypass most of the information to the decoder. The Variational Self-Attention mechanism solves the problem by applying to the context vectors the same constraint applied to the latent variables of the VAE, by modelling them as random variables.
The Variational Self-Attention model receives as input a sequence of T encoded hidden states from which a first group of T context vectors is computed as the weighted sum of all the encoded hidden states. Starting from this first group of hidden states, T vectors of means and T vectors of logarithm of variances are computed

to characterize T Gaussian distributions from which a group of T new context vectors will be sampled. The sampling process avoids the direct connection between the encoder and the decoder. Also in this case, the goal of using the Variational Self-Attention mechanism is to point out which are the correlations between different timestamps of the input in order to obtain better reconstructions.
Instead of using the basic version of the Variational Self-Attention mechanism, we have designed an innovative alternative called "Convolutional Variational Self-Attention mechanism", which differs from the Variational Self-Attention mechanism in the presence of a Convolutional block that allows to structurally avoid any direct connection between the encoder and the decoder, which causes the bypassing phenomenon pointed out in [40]. The need to design an alternative to the Variational Self-Attention mechanism arises from the fact that, in case of complex loss functions containing many terms, the Variational Self-Attention mechanism fails to avoid the bypassing phenomenon above mentioned. The reason is that, to avoid the bypassing phenomenon pointed out in [40], the standard version of the Variational Self-Attention mechanism completely relies on the proper minimization of a $KL\ loss$ term in the loss function. However, when the loss function is complex, not all of its many terms succeed in being properly minimized during the training, and therefore latent spaces result to be not properly formed. If the $KL\ loss$ term relative to bypassing is not properly minimized, the bypassing phenomenon is not properly solved.
Thanks to our innovation, we are able to avoid the bypassing phenomenon affecting the Variational Self-Attention mechanism by acting on the model architecture and not only relying on the minimization of a complex loss function during the training phase (over which we have little or no control).
The Convolutional Variational Self-Attention mechanism takes as input the concatenation of the two sequences of hidden states of the Encoder which is provided as input to a fully connected Dense layer to obtain a first tensor of context vectors C_{det} having shape $(T, 2L)$. The tensor C_{det} is then provided as input to two Convolutional layers, one returning a tensor of means having shape $(\frac{T}{2}, A)$ and one returning a tensor of standard deviations having shape $(\frac{T}{2}, A)$, where A is the latent dimension of the Convolutional Variational Self-Attention mechanism. These means and standard deviations characterize a multivariate Gaussian distribution from which a new tensor of context vectors of shape $(\frac{T}{2}, A)$ is sampled. The two Convolutional layers are both characterized by a linear activation function. The tensor of context vectors is then provided as input to a transpose convolution layers that take its shape from $(\frac{T}{2}, A)$ to (T, A). Each of the T context vectors just obtained is then concatenated with a copy of the latent representation of the input z (a copy for each timestamp) and provided as input to the decoder.
Figure 4 displays a partial representation of the proposed network topology reporting the Encoder, the Variational Layer, and the Convolutional Variational Self-Attention mechanism.

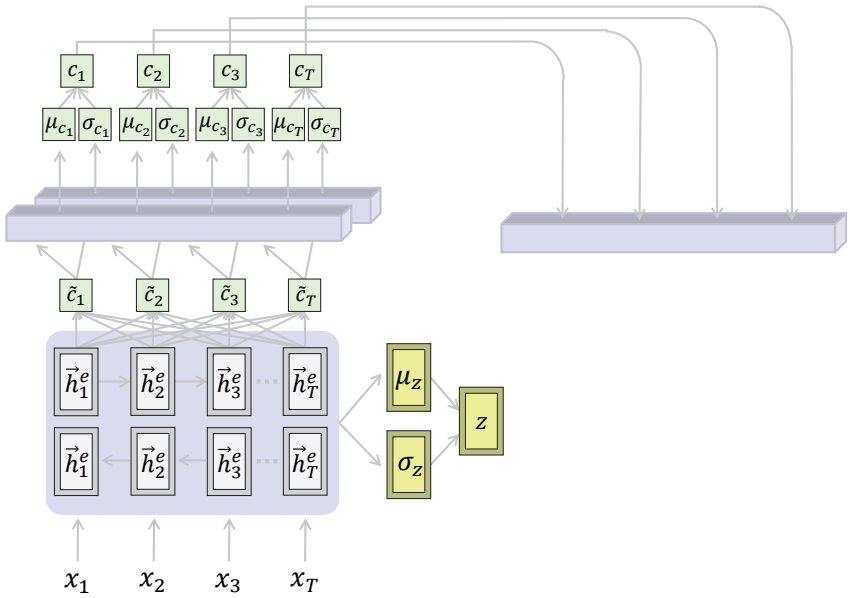

Figure 4: Encoder + Variational Layer + Convolutional Variational Self-Attention.

3.2.4 Decoder

As the Encoder, the Decoder is made of a Bi-LSTM layer. It takes as input the T context vectors computed by the Convolutional Variational Self-Attention mechanism, each one concatenated with a copy of the latent representation of the input z. The decoder output is a reconstruction of the input, therefore it has a shape of (T, D).
In Figure 5 you can see a partial representation of the proposed network topology reporting the Encoder, the Variational Layer, the Convolutional Variational Self-Attention mechanism, and the Decoder.

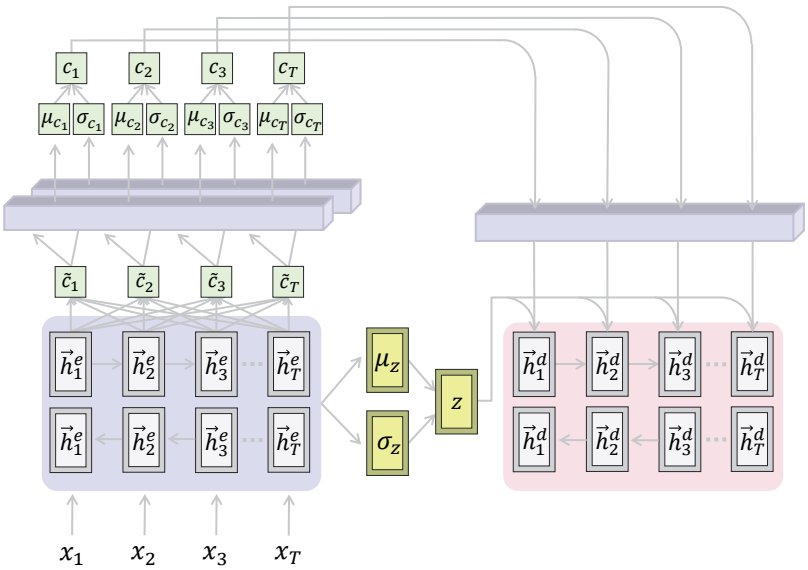

Figure 5: Encoder + Variational Layer + Convolutional Variational Self-Attention + Decoder.

3.2.5 Post-Decoder

The Post-Decoder takes as input the reconstruction of the Decoder. It is made of two fully connected Dense layers that, for each timestamp, provide us with the mean and the logarithm of the scale of a Laplace distribution from which a reconstruction of the input could be sampled. The Dense layer of the means is characterized by a linear activation function, while the Dense layer of the logarithm of the scales is characterized by a SoftPlus activation function. These means and logarithm of the scales will then be used to compute the reconstruction probability of each data point of the input.
Reconstruction probability [41] is a probabilistic measure that takes into account the variability of the distribution of variables and that is often used as anomaly score when dealing with VAEs (replacing the reconstruction error).
It essentially is the probability of the data being generated from a given latent variable drawn from the approximate posterior distribution [41]. Because a number of samples are drawn from the latent variable distribution, this allows the reconstruction probability to take into account the variability of the latent variable space.
The reconstruction probability is computed as:

$$p_\theta(x^{(i)}|\mu_{\hat{x}}(i,t), \sigma_{\hat{x}}(i,t)) \tag{4}$$

The formula basically says that for each of the T data points $x^{(i)}$ provided as input, we have to compute the probability p_θ for it to be sampled from a distribution having parameters $\mu_{\hat{x}}(i,t)$ and $\sigma_{\hat{x}}(i,t)$. To provide some examples, if we consider a Gaussian distribution, parameters $\mu_{\hat{x}}(i,t)$ and $\sigma_{\hat{x}}(i,t)$ would be respectively the mean and the standard deviation, while, if we consider a Laplace distribution (see Appendix A.0.4), parameters $\mu_{\hat{x}}(i,t)$ and $\sigma_{\hat{x}}(i,t)$ would be the mean and the scale. Whichever distribution is chosen, the higher the reconstruction probability, the more likely the sample is not anomalous.
Figure 1 shows the entire proposed network topology.

3.3. How the Proposed Network Topology is Trained

To train the model, we minimize the following loss function:

$$loss\ function = \alpha * likelihood + \beta * (KL\ loss\ z + \delta * KL\ loss\ c) + \gamma * reconstruction\ loss$$

where the term "*KL loss z*" refers to the KL divergence computed considering the latent space associated to the latent representation of the input, the term "*KL loss c*" is the KL divergence computed considering the latent space associated to the Convolutional Variational Self-Attention, and the term "*reconstruction loss*" is computed considering the input and a reconstruction of it obtained by sampling from a Laplace distribution characterized by the means and the variances that the models returns as outputs.
For what concerns the "*likelihood*" term, it is the negative log likelihood for each data point of the input sequence to be sampled from a Laplace distribution having the mean and the scale provided as output by the model.
The likelihood function provides us with the joint probability of the observed data viewed as a function of the parameters of the chosen statistical model. An high likelihood means that there is an high probability that the data are generated by the given model; vice versa, a low likelihood means that there is a low probability that the data are generated by the given model.
In the field of anomaly detection in the electricity consumption of buildings, mainly the Gaussian distribution and the Laplace distribution are used. The parameters needed to define the statistical models are provided as outputs by the designed network and are associated to a specific input.
Instead of maximizing the likelihood, a more frequently used solution is to minimize the negative log likelihood. The reason behind this choice is that the logarithm helps us avoiding the phenomenon of the underflow (if we multiply many probabilities between 0 and 1 the result will vanish, while, thanks to the logarithm's properties, a chain of multiplications becomes a chain of sums that doesn't vanish). The sign minus is used to work with positive values that allow us to minimize instead of maximize (that is much easier while working with neural networks).

3.4. How the Proposed Network Topology Finds Anomalies

To find anomalies, our model computes the reconstruction probabilities of input data points and compare it to a threshold to choose if a data point has to be classified as nominal or anomalous.
Given an input sequence having shape (T, D), where T is the number of timestamps that we collect in a week and D is the number of dimensions we're considering, our model provides as outputs a vector of means having shape (T, D) and a vector of scales having shape (T, D). These two vectors are used to compute the probability that the input we're considering has been sampled from a Laplace distribution having the means and the scales that the model returns. To do that, a variation of the reconstruction probability formula (Formula (4)) is used:

$$-log(\ p_\theta(x^{(i)}|\mu_{\hat{x}}(i,t), \sigma_{\hat{x}}(i,t))\ ) \tag{5}$$

The choice of using the "negative log reconstruction probability" formula can be motivated in the same way in which we have motivated the choice of using the "negative log likelihood" in Section 3.3: the logarithm helps us avoiding the underflow phenomenon and the sign minus is used to work positive values that allow us to minimize instead of maximize (that is much easier while working with neural networks).
The Monte Carlo method is used to obtain more stable results: given a certain input sequence, we encode it only once, and then we pass the obtained latent representation to the decoder for M times. In this way, we obtain M values of the "negative log Reconstruction Probabilities" for each timestamp. Finally, the average of these values has to be computed:

$$\frac{1}{M}\sum_{m=1}^{M} -log(p_\theta(x^{(i)}|\mu_{\hat{x}}(i,t), \sigma_{\hat{x}}(i,t))) \tag{6}$$

Once that the "negative log Reconstruction Probabilities" are calculated, the optimal value of the threshold has to be found. Considering that we are working with the "negative log reconstruction probability" formula, our approach will be the following: "if the negative log reconstruction probability associated to a certain time stamp is higher than the threshold, the data point is anomalous, otherwise it is not".
To find the right threshold to be used we have opted for a grid search approach: we start from a threshold that is equal to the minimum "negative log reconstruction probability" computed on the validation set, and we try all the values up to the maximum "negative log reconstruction probability" computed on the validation set advancing by 0.01 at a time. We return the value of the threshold that maximizes the F1 score on the validation set. Once the optimal threshold is found, it will be used for the final model that we will deploy.

4. Experimental Setting

The experimental setting includes baseline models, competitors' models and the metric used to compare them.

4.1. Competitors' Models

In this section we present and compare the different network topologies we have taken into account for our extensive experimental comparison.

4.1.1 Convolutional Autoencoder + Reconstruction Error

As shown in Figure 6, the model we present is an Autoencoder made of Convolutional layers. The encoder is made of Convolutional layers alternated with MaxPool layers used to reduce the input's dimensionality. After the Convolutional part of the encoder, a Flatten layer is used to obtain a 1-dimension tensor starting from the result of the convolution. A Dense layer having a number of output neurons equal to the latent dimension is then used to obtain the latent representation of the input. The decoder is made of a Dense layer and a Reshape layer (both useful to pass from the 1-dimension tensor of the latent representation to a multi-dimension tensor) and of a number of Transpose Convolutional layers adequate to re-obtain the shape of the input. The loss function used to train the model is the following:

$$loss\ function = reconstruction\ loss,$$

where the term "reconstruction loss" refers to the Mean Squared Error (MSE) computed considering the input and its reconstruction provided as output.

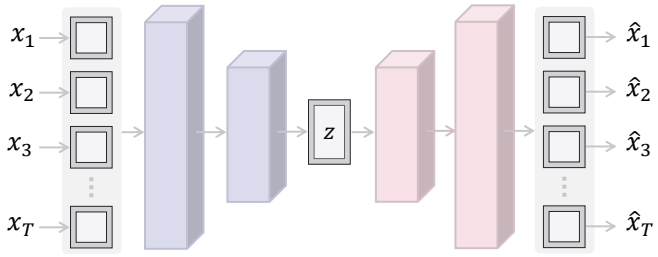

Figure 6: Convolutional AE with Reconstruction Error.

4.1.2 Convolutional Variational Autoencoder + Reconstruction Error

In Section 4.1.1 we have presented an Autoencoder made of Convolutional layers. Here we use Convolutional layers to build a Variational Autoencoder. As shown in Figure 7, the encoder's structure is identical to the one described for the "Convolutional Autoencoder + Reconstruction Error" with the difference that, after the final Dense layer, there are other two parallel Dense layers, each one having a number of output neurons equal to the chosen latent dimension, one to represent the means and one to represent the logarithm of the variances of a multivariate Gaussian distribution. The latent representation that the decoder receives as input is obtained by sampling from the multivariate Gaussian distribution whose means and variances are the ones computed by the encoder. The output of the decoder is again a reconstruction of the input, therefore has the same shape. The loss function used to train the model is the following:

$$loss\ function = reconstruction\ loss + KL\ loss,$$

where the term "reconstruction loss" refers to the mean squared error computed considering the input and its reconstruction provided as output and the term "KL loss" is used as a regularizer for the multivariate Gaussian distribution from which the latent representation of the input will be sampled (see Appendix A.0.3).

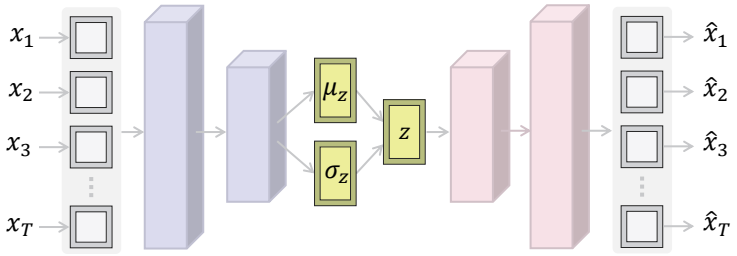

Figure 7: Convolutional VAE with Reconstruction Error.

4.1.3 Convolutional Variational Autoencoder + Reconstruction Probability

The model we present here is similar to the one presented in Section 4.1.2, but this time reconstruction probability is used instead of reconstruction error. As shown in Figure 8, the encoder's structure is identical to the one described for the "Convolutional Variational Autoencoder + Reconstruction Error", while the decoder is enriched with two final parallel Dense layers that, starting from the reconstruction of the input, allow us to obtain the means and the logarithm of the variances of a multivariate Gaussian distribution. These means and variances are then used to compute the reconstruction probability associated to the given input.
The loss function used to train the model is the following:

$$loss\ function = reconstruction\ loss + KL\ loss + likelihood,$$

where the term "reconstruction loss" refers to the mean squared error computed considering the input and a reconstruction of it obtained by sampling from a Gaussian distribution characterized by the means and the variances that the models returns as outputs, the term "KL loss" is used as a regularizer for the multivariate Gaussian distribution from which the latent representation of the input will be sampled, and the term "likelihood" is the negative log likelihood computed considering the probability for the input to be sampled from a multivariate Gaussian distribution characterized by the means and the variances that the model provides as output.

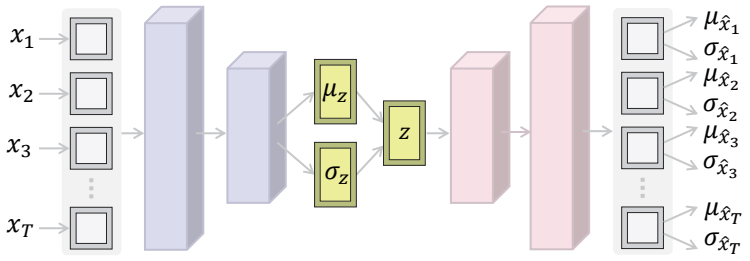

Figure 8: Convolutional VAE with Reconstruction Probability.

4.1.4 LSTM Autoencoder + Reconstruction Error

As shown in Figure 9, the model we present is an Autoencoder made of LSTM layers. The encoder is characterized by consecutive LSTM layers, the last of which has a number of memory elements equal to the chosen latent dimension. First LSTM layers return the entire sequence of hidden states while the last LSTM layer returns the final hidden state only, and therefore a 1-dimensional latent representation of the input. The decoder is characterized by a RepeatVector layer followed by a sequence of one or more LSTM layers. Finally, a TimeDistributed layer encapsulates a Dense layer having a number of output neurons equal to the number of the dimensions of the dataset. The output of the decoder is a reconstruction of the input, therefore has the same shape.
The loss function used to train the model is the following:

$$loss\ function = reconstruction\ loss,$$

where the term "reconstruction loss" refers to the mean squared error computed considering the input and its reconstruction provided as output.

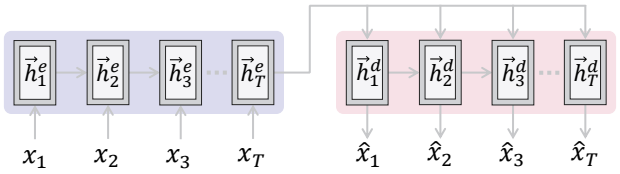

Figure 9: LSTM AE with Reconstruction Error.

4.1.5 LSTM Variational Autoencoder + Reconstruction Error

In Section 4.1.4 we have presented an Autoencoder made of LSTM layers. Here we use LSTM layers to build a Variational Autoencoder. As shown in Figure 10, the encoder is made of consecutive LSTM layers. The output of the chain of LSTM layers is provided as input to a Dense layer having a number of output neurons equal to the chosen latent dimension. This Dense layer is then connected to two other parallel Dense layers, each one having a number of output neurons equal to the chosen latent dimension, one to represent the means and one to represent the logarithm of the variances of a multivariate Gaussian distribution. The latent representation that the decoder receives in input is obtained by sampling from the multivariate Gaussian distribution whose means and variances are the ones computed by the encoder. The decoder's structure is identical to the one described for the "LSTM Autoencoder + Reconstruction Error" model.
The loss function used to train the model is the following:

$$loss\ function = reconstruction\ loss + KL\ loss,$$

where the term "reconstruction loss" refers to the mean squared error computed considering the input and its reconstruction provided as output, and the term "KL loss" is used as a regularizer for the multivariate Gaussian distribution from which the latent representation of the input will be sampled.

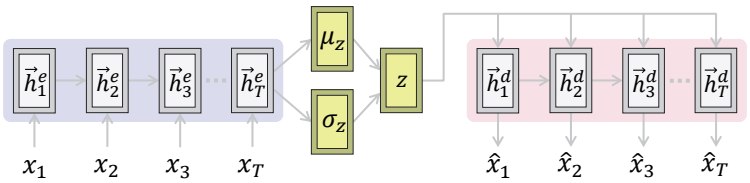

Figure 10: LSTM VAE with Reconstruction Error.

4.1.6 LSTM Variational Autoencoder + Reconstruction Probability

The model we present here is similar to the one presented in Section 4.1.5, but this time reconstruction probability is used instead of reconstruction error. As shown in Figure 11, the encoder's structure is identical to the one described for the "LSTM Variational Autoencoder + Reconstruction Error". The decoder is enriched with two final parallel Convolutional layers that, starting from the reconstruction of the input, allow us to obtain the means and the logarithm of the variances of a multivariate Gaussian distribution. These means and variances are then used to compute the reconstruction probability associated to a certain input.
The loss function used to train the model is the following:

$$loss\ function = reconstruction\ loss + KL\ loss + likelihood,$$

where the term "reconstruction loss" refers to the mean squared error computed considering the input and a reconstruction of it obtained by sampling from a Gaussian distribution characterized by the means and the variances that the models returns as outputs, the term "KL loss" is used as a regularizer for the multivariate Gaussian distribution from which the latent representation of the input will be sampled, and the term "likelihood" is the negative log likelihood computed considering the probability for the input to be sampled from a multivariate Gaussian distribution characterized by the means and the variances that the model provides as output.

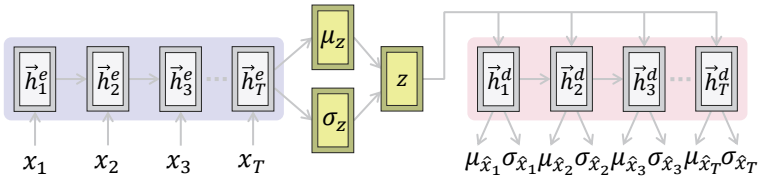

Figure 11: LSTM Variational Autoencoder with Reconstruction Probability.

4.1.7 Bi-LSTM VAE with Variational Self-Attention and Reconstruction Probability

Here we present a state-of-the-art model that stands out from its predecessors for the use of the Variational Self-Attention mechanism and of Bi-LSTM layers. As shown in Figure 12, the encoder is made of a Bi-LSTM layer that generates a sequence of hidden states for each direction, forward and backward. The two sequences of hidden states are concatenated and then provided as input to the Variational layer, which is made of two Dense layers, one used to compute the means and one used to compute the logarithm of the variances of an isotropic Gaussian distribution from which the latent representation of the input will be sampled. Furthermore, a Variational Self-Attention mechanism is implemented: it receives as input a sequence of encoded hidden states (from the Bi-LSTM layer of the encoder) and outputs a sequence of context vectors, each of which will be concatenated with a copy of the latent representation of the input (a copy for each timestamp) and provided as input to the decoder, which is made of a Bi-LSTM layer as the encoder. The decoder is then followed by two fully connected Dense layers that, for each timestamp, which provide us the mean and the logarithm of the scale of a Laplace distribution from which a reconstruction of the input could be sampled. These means and logarithm of the scales will then be used to compute the reconstruction probability of the input data point. The Bi-LSTM layers are characterized by a tanh activation function, the Dense layers used for the means are characterized by a linear activation function, and the Dense layers of the logarithm of the scales and of the standard deviations are characterized by a SoftPlus activation function. A random Gaussian noise is added to the input before it is provided to the encoder in order to teach to the model how to remove noise (denoising Autoencoder).
The loss function used to train the model is the following:

$$loss\ function = reconstruction\ loss + KL\ loss\ c + KL\ loss\ z + likelihood,$$

where the term "reconstruction loss" refers to the mean squared error computed considering the input and a reconstruction of it obtained by sampling from a Laplace distribution characterized by the means and the variances that the models returns as outputs, the term "KL loss c" is used as a regularizer for the multivariate Gaussian distribution from which the latent representation of the input will be sampled, "KL loss z" is used as a regularizer for the multivariate Gaussian distribution from which the context vectors for the Variational Self-Attention mechanism will be sampled and the term "likelihood" is the negative log likelihood computed considering the probability for the input to be sampled from a Laplace distribution characterized by the means and the scales that the model provides as output.

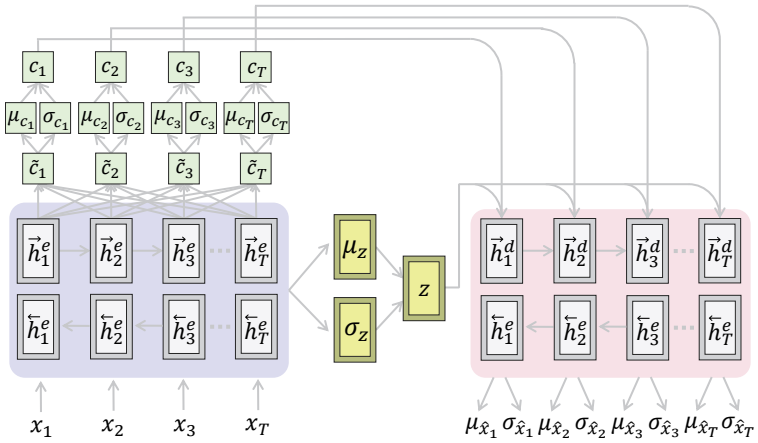

Figure 12: Bi-LSTM VAE with Variational Self-Attention and Reconstruction Probability.

4.2. F1 Score

In order to compare different models, we need to choose which metric to consider. The metric we have chosen is the F1 score, as it is the most used metric when dealing with anomaly detection tasks. It is computed as:

$$F1\ score = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

$$Precision = \frac{True\ Positives}{True\ Positives + FalsePositives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

In other words, the precision is given by the number of real anomalies correctly identified divided by the number of data points classified as anomalous, while the recall is given by the number of real anomalies correctly identified divided by the total number of anomalous data points.

5. Experimental Results - Simpler Dataset

In this section we present the univariate dataset, together with the experimental results obtained.

5.1. Univariate Dataset

The univariate dataset is a publicly available benchmark dataset for anomaly detection in the electricity consumption of buildings. It contains electrical power consumption data of a research facility in the Netherlands over the year 1997 (from the 1^{st} of January to the 31^{st} of December), recorded with a granularity of 15 minutes. When dealing with anomaly detection tasks, a crucial aspect is to determine which behaviors have to be considered nominal and which ones have to be considered anomalous. Working with buildings power consumption is not an easy task: let's say that the 25^{th} of December is on Tuesday. If we consider that people do not work on Christmas Day, it is okay to register a low energy consumption on that day: no anomaly has to be notified. On the other hand, if we consider that Tuesday is a working day, the fact that the energy consumption is low is anomalous.
The univariate dataset presents a daily and weekly seasonality and we assume that nominal behavior follows this pattern: the energy consumption is high during the day and low during the night from Monday to Friday, while, during the weekends, the energy consumption is low both during day and night as people don't work on Saturday and Sunday. Going back to the previous example, if Christmas Day is on Tuesday we will register an anomaly since Tuesday is a working day.

In agreement with what we have just said, the dataset contains 34944 data points, and 371 of them are anomalous ($\sim 1\%$). We have assigned 20160 data points ($\sim 58\%$) to the training set, 7392 data points ($\sim 21\%$) to the validation set while the test set contains 7392 ($\sim 21\%$) data points, 371 of which are anomalous ($\sim 5\%$). Hence, the training is done in a semi-supervised fashion.

The univariate dataset presents three categories of anomalies (examples are shown in Figure 13):

- Working days with a low energy consumption (a)
- Non-working days with a high energy consumption (b)
- Upward or downward spikes (c) in the energy consumption (spikes may consist of 1 or more anomalous data points)

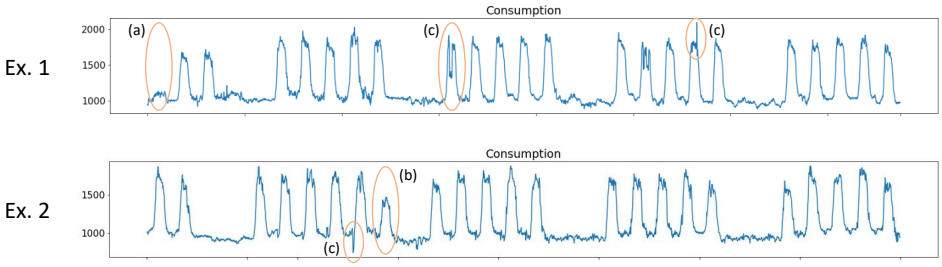

Figure 13: Examples of anomalies in the univariate dataset.

More specifically, we consider the following anomalies:

- January 1997
  - 01 January: working day with a low energy consumption
  - 13 January: downward spike in the energy consumption
  - 21 January: downward spike in the energy consumption
  - 23 January: upward spike in the energy consumption
- February 1997
  - 05 February: upward spike in the energy consumption
- March 1997
  - 26 March: downward spike in the energy consumption
  - 28 March: working day with a low energy consumption
  - 31 March: working day with a low energy consumption
- April 1997
  - 30 April: working day with a low energy consumption
- May 1997
  - 05 May: working day with a low energy consumption
  - 08 May: working day with a low energy consumption
  - 13 May: downward spike in the energy consumption
  - 19 May: working day with a low energy consumption
- June 1997: no anomalies
- July 1997: no anomalies
- August 1997: no anomalies
- September 1997: no anomalies
- October 1997
  - 03 October: downward spike in the energy consumption
  - 04 October: non-working day with a high energy consumption
- November 1997: no anomalies
- December 1997
  - 25 December: working day with a low energy consumption
  - 26 December: working day with a low energy consumption

By using this dataset, we place ourselves in the same conditions as our competitors who have used synthetic or low-noisy datasets to test their models. Looking at [33] we notice that, despite the fact that the authors have used real-world data, their time series are very similar to a perfect sinusoidal function (see (A) in Figure 14). Similarly, in [42], authors have used time series that are smooth and extremely regular (see (B) and (C) in Figure 14). The univariate dataset we consider is not very complex, however, it is still noisier than the ones of our competitors (see (D) in Figure 14).

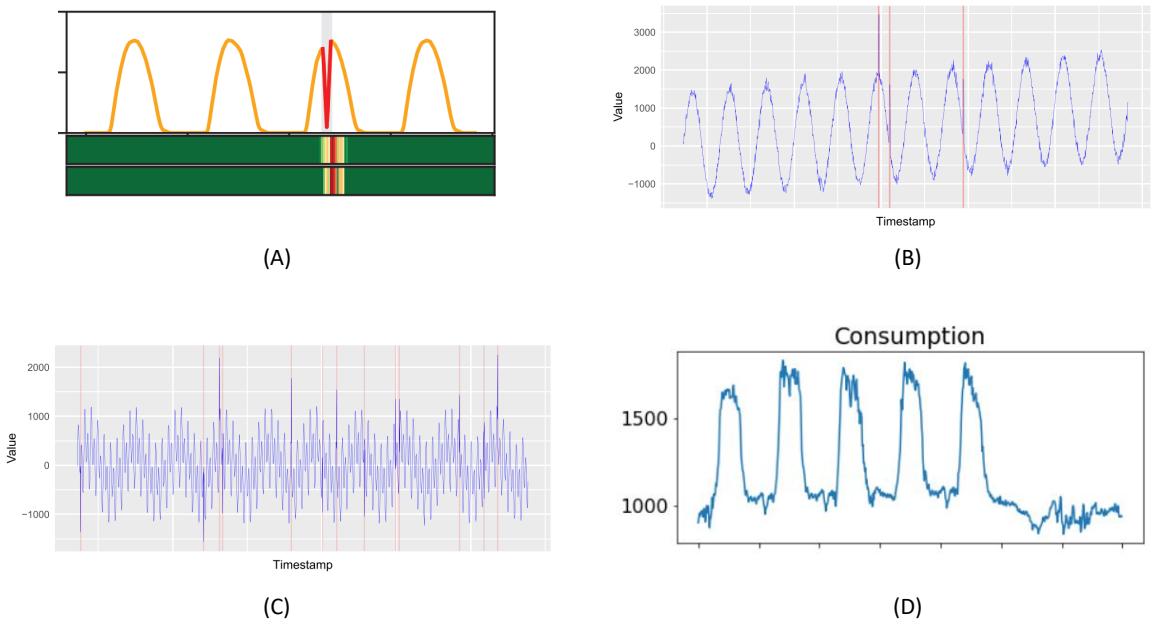

Figure 14: (A): example of time series taken from the dataset used by authors of [33], (B) and (C): examples of time series taken from the datasets used by authors of [42], (D): example of time series taken from our univariate dataset.

5.2. Discussion on Experimental Results

As shown in Table 1, models that use LSTM layers perform better than models using Convolutional layers. This is a coherent result, since LSTM networks have been specifically designed to deal with time series and to store information over arbitrarily long intervals of time steps, while Convolutional layers are meant for general purposes and not specifically to deal with temporal correlation of data. The reconstruction probability turns out to be a better metric than reconstruction error as it allows to achieve better results. Also in this case, results are consistent with theoretical studies considering that the reconstruction probability has been specifically introduced to be a better alternative to the reconstruction error with respect to anomaly detection tasks. Our results are also coherent with the fact that VAEs have been specifically developed to outperform AEs. Finally, we note that VAEs produce better F1 scores than the their non-variational counterparts, at the expense of a lower recall but an higher precision.

To explain why our proposed network topology outperforms the "Bi-LSTM VAE with Variational Self-Attention and Reconstruction Probability", we have to consider that the main improvement proposed by the authors of [33] is the introduction of the Variational Self-Attention mechanism, which helps to obtain more accurate reconstructions of the input. As noted by the authors, if the standard Self-Attention mechanism is used, the decoder has a direct and deterministic access to the encoder hidden states, therefore, the latent code z is not forced to learn expressive representations since the Self-Attention mechanism forwards most of the information from the encoder to the decoder. To solve this problem, the authors have applied to the context vectors the same constraint applied to the latent variables of the VAE, by modelling them as random variables. This helps to reduce the bypassing phenomenon but, as we will show in the following, does not eliminate it completely. By modifying the network topology, it is possible to achieve better results.

Looking at the reconstructions provided by the "Bi-LSTM VAE with Variational Self-Attention and Reconstruction Probability", we have noticed that the bypassing phenomenon was not completely fixed: in fact also anomalies were well reconstructed. This resulted in a significant lowering of the F1 score. To understand how

to solve the problem, we have made several attempts: we have tried to replace Bi-LSTM layers with LSTM layer, we have tried to replace the Laplace distribution with a Gaussian distribution. We started to obtain some improvements only through the tuning of the Variational Self-Attention latent dimension. This helped us in understanding what the cause of the bypass was.
The "Bi-LSTM Variational Autoencoder with Variational Self-Attention + Reconstruction Probability" model takes as input a tensor having a shape of (T, D), where T is the number of timestamps of the input and D is the number of considered time series (dimensions). The Bi-LSTM layer takes this input and returns as output a tensor of hidden states having shape $(T, 2L)$, where L is the number of memory elements. The tensor of hidden states is used to obtain a first tensor of context vectors C_{det} having shape $(T, 2L)$. At this point, to avoid the bypassing phenomenon pointed out in [40], authors of [33] implements the Variational Self-Attention mechanism: starting from the tensor C_{det} they obtain, through the use of fully connected Dense layers, a tensor of means having shape (T, A) and a tensor of logarithm of standard deviations having shape (T, A). These two tensors characterize a multivariate Gaussian distribution from which a final tensor of context vectors having shape (T, A) is randomly sampled, with A equal to the Variational Self-Attention latent dimension.
The problem here is that the only reduction of dimensionality from the tensor C_{det} having shape $(T, 2L)$ to the final tensor of context vectors having shape (T, A) is offered by the Variational Self-Attention latent dimension: we do not have a way to change the value of T from a tensor to the other one. As a consequence, unlike in the variational latent space of the VAE, the time series is never compressed in the time dimension, hence the bypassing.
This is a problem, since we face a trade-off: the lower the Variational Self-Attention latent dimension, the less we can benefit of the Self-Attention mechanism, the bigger the Variational Self-Attention latent dimension, the higher the bypassing phenomenon. In particular, if the Variational Self-Attention latent dimension is equal to 0, we cannot benefit of the Variational Self-Attention mechanism at all, and if the Variational Self-Attention latent dimension is grater or equal to 2L we have a complete bypass. To understand why we would have a complete bypass, remember that we are trying to "compress" a tensor having shape $(T, 2L)$ using two tensors (those of means and variances) each one having shape $(T, 2L)$. What the model learns is to ignore the vector of variances that is useless since all the information can be easily forwarded using the vectors of means only.
To avoid the complete forwarding, authors of [33] introduce a dedicated $KL\ loss$ term in their loss function. As previously explained, the KL divergence term is a regularizer that tries to avoid the bypass by constraining the variances to be as close to 1 as possible (see Appendix A.0.3). The problem is that in this way they are trying to solve the issue relying on a good construction of the latent spaces only. To build proper latent spaces, the relative terms of the loss function have to be minimized during the training. Here is the loss function:

$$lossfunction = \alpha * likelihood + \beta * (KL\ loss\ z + \delta * KL\ loss\ c) + \gamma * reconstructionloss\ ,$$

where the term "likelihood" refers to the probability that the input could be sampled from a Laplace distribution characterized by the means and the scales that the model return as output, the term "KL loss z" refers to the KL divergence computed considering the latent space associated to the latent representation of the input, the term "KL loss c" is the KL divergence computed considering the latent space associated to the Variational Self-Attention and the term "reconstruction loss" is computed considering the input and a reconstruction of it obtained by sampling from a Laplace distribution characterized by the means and the variances that the models returns as outputs.
We thus have four different terms to be minimized during the training and four parameters to be tuned. This makes a good construction of the latent spaces very difficult, if not totally impossible. We have spent countless hours to tune α, β, δ, and γ and the best result that we managed to obtain is worse than the result that we get on the first try with our proposed network topology. Moreover, in addition to the problems related to the parameters tuning, the fact that the loss function contains so many terms makes it difficult to minimize all of them all together during the training phase (on which we have little or no control).
Our idea is to address the forwarding problem without relying on the loss function, but by modifying the model architecture in order to make the bypassing phenomenon structurally impossible.
To do that, we have designed a new version of the Self-Attention mechanism called "Convolutional Variational Self-Attention mechanism". It is characterized by Convolutional layers that take as input the first tensor of context vectors C_{det} having shape $(T, 2L)$, and return as output a tensor of means having shape $(\frac{T}{2}, A)$ and a tensor of standard deviations having shape $(\frac{T}{2}, A)$ that characterize a multivariate Gaussian distribution from which a final tensor of context vectors of shape $(\frac{T}{2}, A)$ is sampled. In this way, we structurally avoid the complete bypassing from tensor C_{det} to the final tensor of context vectors, because they would never have the same shape ($(T, 2L)$ vs. $(\frac{T}{2}, A)$), whatever the value of the Variational Self-Attention latent dimension is. After the sampling, the final tensor of context vectors is provided as input to a Transpose Convolutional layer that provides as output tensors having a shape of (T, A), the right shape to be provided as input to the decoder. It is important to highlight that our model also allows further reductions of dimensions (from T to $\frac{T}{4}$, $\frac{T}{8}$ and so on) with some minor changes (an higher value of the stride, for instance).

In this way the bypassing phenomenon is structurally avoided and the model is much more robust to the tuning of α, β, δ and γ.
So, consistent with the above, with a F1 score of 0.9148 the proposed network topology achieves better results both with respect to the "LSTM Variational Autoencoder + Reconstruction Probability" model and to the "Bi-LSTM VAE with Variational Self-Attention and Reconstruction Probability" one, as is benefits from the use of the Variational Self-Attention mechanism but also avoids the bypassing phenomenon pointed out in [40].

Univariate dataset

	F1 score	Precision	Recall
CNN AE + Rec Error	0.8232	0.7896	0.8598
CNN VAE + Rec. Error	0.8544	0.8626	0.8464
CNN VAE + Rec. Prob.	0.8678	0.8598	0.8760
LSTM AE + Rec. Error	0.8774	0.8681	0.8868
LSTM VAE + Rec. Error	0.8821	0.9086	0.8571
LSTM VAE + Rec. Prob.	0.9013	0.9604	0.8491
Bi-LSTM VAE w/ Var. Self. Att. + Rec. Prob.	0.8955	0.9016	0.8895
Bi-LSTM VAE w/ Con. Var. Self. Att. + Rec. Prob	0.9148	0.8903	0.9407

Table 1: F1 scores, precisions and recalls of the topologies considered the univariate dataset.

6. Experimental Results - Noisy Dataset

In this section we present the multivariate dataset, together with the experimental results obtained.

6.1. Multivariate Dataset

The multivariate dataset contains electrical power consumption data of an office building in the North of Italy over a period of three years (from the 9^{th} of February 2017 to the 10^{th} of April 2019), recorded with a granularity of 15 minutes. The building is equipped with several smart meters that collect information on power consumption relative to different groups of devices. Hence, the Italian dataset, unlike the Dutch one, is multivariate. The dataset contains 75840 data points, and 550 of them are anomalous ($\sim 0.7\%$). We have assigned 43680 data points ($\sim 58\%$) to the training set, 16128 data points ($\sim 21\%$) to the validation set, while the test set contains 16032 ($\sim 21\%$) data points, 550 of which are anomalous ($\sim 3.4\%$). The distinction between what should be considered anomalous and what should be considered nominal follows the same rules reported in Section 5.1 for the univariate dataset.

The multivariate dataset presents three categories of anomalies (examples are shown in Figure 15):

- Working days with a low energy consumption as on weekends and holidays (as in the univariate case)
- Sudden and prolonged drops in the energy consumption (a)
- Upward or downward spikes (as in the univariate case) in the energy consumption (spikes may consist of 1 or more anomalous data points)

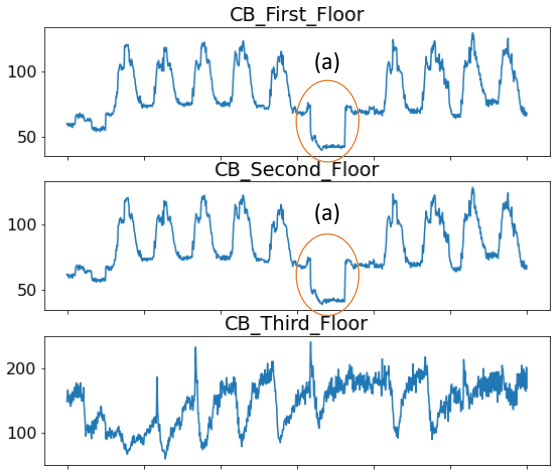

Figure 15: Examples of anomalies in the multivariate dataset.

More specifically, we consider the following anomalies:

- From February 2017 to October 2018: no anomalies
- November 2018
  - 01 November: working day with a low energy consumption on floor 1 and 2
  - 02 November: working day with a low energy consumption on floor 1 and 2
  - 19 November: upward spike in the energy consumption on floor 3
  - 28 November: upward spike in the energy consumption on floor 3
  - 29 November: upward spike in the energy consumption on floor 3
- December 2018
  - 15 December: sudden prolonged drop in the energy consumption on floor 1 and 2
  - 16 December: sudden prolonged drop in the energy consumption on floor 1 and 2
  - 21 December: sudden prolonged drop in the energy consumption on floor 1 and 2
  - 22 December: sudden prolonged drop in the energy consumption on floor 1 and 2
  - 24 December: working day with a low energy consumption on floor 1 and 2
  - 25 December: working day with a low energy consumption on floor 1 and 2
  - 26 December: working day with a low energy consumption on floor 1 and 2

- 27 December: working day with a low energy consumption on floor 1 and 2
- 28 December: working day with a low energy consumption on floor 1 and 2
- 31 December: working day with a low energy consumption on floor 1 and 2

- January 2019
  - 01 January: working day with a low energy consumption on floor 1 and 2
  - 05 January: upward spike in the energy consumption on floor 3
  - 12 January: downward spike in the energy consumption on floor 1, 2 and 3
  - 19 January: downward spike in the energy consumption on floor 1 and 2
  - 24 January: working day with a low energy consumption on floor 1
- February 2019
  - 15 February: working day with a low energy consumption on floor 2
- March 2019:
  - 09 March: upward spike in the energy consumption on floor 3
- April 2019:
  - 07 April: downward spike in the energy consumption on floor 3

The multivariate dataset is definitely more complex and noisy than the univariate one (as shown by the examples in Figure 16). Our decision of testing all the previously presented network topologies also on this noisy dataset is due to the fact that our direct competitors [33, 42] have tested their models on simple datasets only, obtaining very good results which were mainly attributed to the models, as if the problem and the dataset taken into account were of secondary importance. Please notice that when we talk about "simple datasets", we mean "pretty regular and smooth time series". We want to draw the attention on the fact that we often refer to the "state-of-the-art", as if it existed an "absolute best solution", regardless of the specific problem and data involved. Our goal is to check whether this approach is valid or not and to prove that goodness should be attributed not "to the model" only, but "to the model with respect to a specific problem and dataset". As for the univariate dataset, the F1 score will be used to compare different network topologies.

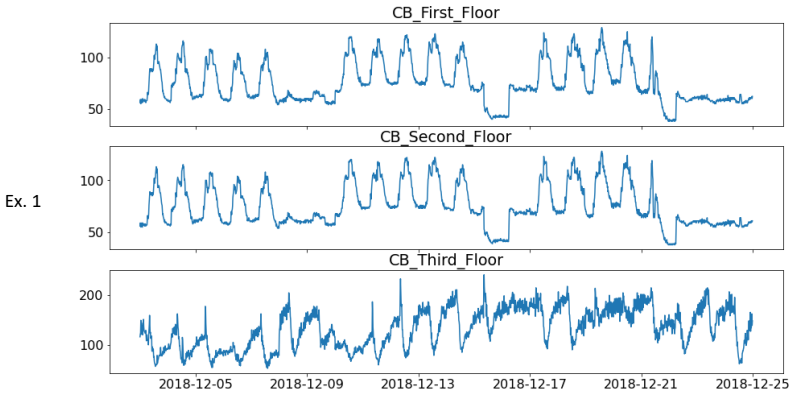

Figure 16: Examples of time series taken from our multivariate dataset.

6.2. Discussion on Experimental Results

As you can notice looking at the experimental results reported in Table 2, when dealing with the multivariate dataset, simpler models perform better than complex ones. To understand why this happens, we have carefully analyzed the outputs provided by the models and discovered that, if the dataset is very noisy, it is extremely more difficult to think of data points in terms of "mean + standard deviation". As a consequence, the calculation of reconstruction probabilities and the samplings from Gaussian and Laplace distributions are less accurate, and this leads to a deterioration in performance.
To support our intuition, we have examined the variances provided as output by the models. The "Convolutional Variational Autoencoder + Reconstruction Probability" model provides variances in the order of magnitude of 10^{-2} while working with the multivariate dataset and variances in the order of magnitude of 10^{-3} while working with the univariate dataset. The exact same thing also happens with the "LSTM Variational Autoencoder + Reconstruction Probability" model, with the "Bi-LSTM VAE with Variational Self-Attention and Reconstruction Probability", and with our proposed network topology.
Working with higher variances leads to many important consequences, mainly affecting the calculation of reconstruction probabilities. More specifically, having higher variances entail that:

- The value of the reconstruction probability associated to anomalous data points is higher than it should (as the probability of having sampled that anomalous data point from a distribution having a certain mean and an higher variance is higher)
- The value of the reconstruction probability associated to nominal data points is lower than it should (as the probability of having sampled that nominal data point from a distribution having a certain mean and a higher variance is lower)

This means that the values of the reconstruction probability associated to nominal samples will not be too far from the values of the reconstruction probability associated to anomalous ones (while in the univariate case the distinction is much sharper). The fact that it is harder to distinguish between nominal and abnormal samples by looking at their reconstruction probabilities leads to higher false negative and false positive rates, and therefore, to a lower F1 score. Given the above, it is not surprising that models using the reconstruction probability perform worse than their counterparts using reconstruction error.
It is also not surprising that the Variational Self-Attention mechanism does not lead to much improvement: the number of inaccurate variances we are working with is even higher, as now we have to consider also the ones used to sample the context vectors (notice that these variances should be constrained by the dedicated $KL\ loss$ term in the loss function, however, having too many terms in the loss function leads to a sub-optimal formation of the latent spaces).
Another result is that LSTM units seem to deliver better results when used to build AEs, while convolutional units perform better when coupled with variational architectures. The reason is probably that the increased expressive power of VAEs makes up for some of the limitations of a fully convolutional architecture.
The reason why the proposed network topology performs better than the "Bi-LSTM VAE with Variational Self-Attention and Reconstruction Probability" is that, while the "Bi-LSTM VAE with Variational Self-Attention and Reconstruction Probability" tries to solve the bypassing phenomenon presented before by only relying on the minimization of the loss function (whose terms are computed using variances) and on a good construction of latent spaces (that is not easy to obtain if we are dealing with complex loss functions), we structurally solve the bypassing phenomenon by modifying the model architecture, and therefore we are more robust to high variances.
This reasoning leads us to question the concept of "state-of-the-art". We have discovered that it is not fully correct to say that the reconstruction probability is the "state-of-the-art" for the detection of anomalies, the only thing we can say is that it performs better than the reconstruction error under some specific conditions (a sufficiently clean dataset). Similarly, it is not fully correct to talk about a model that is the "state-of-the-art" without also specifying what is the problem we want to address and without placing emphasis on the data we want to work with.
Often, when we read or write an article, we suppose that the obtained results are due to the proposed network topology only, as if the problem at hand and the dataset were of second importance. Instead, it is important to realize that there is no solution that indiscriminately provides the best possible result regardless of the data and the problem we want to solve.
We should not speak of "state-of-the-art" in absolute terms, but only of "state-of-the-art with respect to that specific problem and that specific dataset".

Multivariate dataset

	F1 score	Precision	Recall
CNN AE + Rec Error	0.3976	0.3917	0.4037
CNN VAE + Rec. Error	0.4893	0.4888	0.4898
CNN VAE + Rec. Prob.	0.4799	0.4789	0.4809
LSTM AE + Rec. Error	0.6531	0.6466	0.6598
LSTM VAE + Rec. Error	0.6115	0.8081	0.4918
LSTM VAE + Rec. Prob.	0.5797	0.6371	0.5316
Bi-LSTM VAE w/ Var. Self. Att. + Rec. Prob.	0.4013	0.4116	0.3914
Bi-LSTM VAE w/ Con. Var. Self. Att. + Rec. Prob	0.4479	0.4600	0.4365

Table 2: F1 scores, precisions and recalls of the topologies considered the multivariate dataset.

7. Conclusions

When dealing with buildings power consumption, online anomaly detection is a fundamental task as it allows to identify faults and abnormal behaviours in the electricity consumption. This allows the timely intervention of building managers and in the reduction of energy wastes, environmental pollution and costs.
In this thesis, we have proposed an innovative deep learning online anomaly detection network topology that outperforms the current state-of-the-art methods by solving the bypassing phenomenon more effectively. Our novel model allows anomalies to be identified quickly and more accurately, possibly bringing to further reductions in energy wastes.
In order to solve the bypassing phenomenon affecting the state-of-the-art methods, we have designed an alternative to the Self-Attention mechanism, the Convolutional Variational Self-Attention mechanism, which is characterized by a Convolutional block that structurally avoids the forwarding of information between the encoder and the decoder. Unlike authors of [33], we avoid the bypassing phenomenon focusing on the structure of the model, and not by relying on the minimization of complex loss functions during the training phase (on which we have little or no control).
In addition to the proposed network topology, we have performed an extensive experimental comparison. We have compared all major semi-supervised deep learning anomaly detection models, showing that the proposed network topology performs better than the state-of-the-art methods, finally we have shown that, in order to correctly choose which model to use, we have to pay attention to the specific problem we are trying to solve and to the data we are working with, as simpler models offer better performance when working with very noisy datasets since they are less sensitive to high variances.
For what concerns future works, it would be interesting to test the proposed network topology on data streams instead of a static dataset in order to simulate sensors readings with a certain time frequency. To do that, Apache Kafka and Apache Spark can be used. Kafka is a publish-subscribe event store manager from which Spark can read information that can then be analyzed. By loading the dataset in a Python notebook, we can ask Kafka to publish a new message every Q minutes as to simulate sensor readings. Spark will then read new information every Q minutes receiving new data do be analyzed. Spark will update the window of data to be provided as input to the model by inserting newly received sensors readings according to a "first-in first-out" logic. Once the window has been updated, it can be provided to our model that will classify new data points as anomalous or not.

References

[1] United Nations Environmental Programme (UNEP). 2021 Global Status Report for Buildings and Construction: Towards a Zero-emission, Efficient and Resilient Buildings and Construction Sector, 2021.

[2] Davide Azzalini, Benedetta Flammini, Claudio Alfredo Emanuele, Antonio Guadagno, and Francesco Amigoni. Empirical evaluation of deep autoencoders for anomaly detection in electricity consumption of buildings.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):1–58, 2009.

[4] Yassine Himeur, Khalida Ghanem, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. *Appl. Energy*, 287:1–26, 2021.

[5] Pandarasamy Arjunan, Harshad D Khadilkar, Tanuja Ganu, Zainul M Charbiwala, Amarjeet Singh, and Pushpendra Singh. Multi-user energy consumption monitoring and anomaly detection with partial context information. In *Proc. BuildSys*, pages 35–44, 2015.

[6] Jaime Yeckle and Bo Tang. Detection of electricity theft in customer consumption using outlier detection algorithms. In *Proc. ICDIS*, pages 135–140, 2018.

[7] Yassine Himeur, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. Smart power consumption abnormality detection in buildings using micromoments and improved k-nearest neighbors. *Int. J. Intell. Syst.*, 36(6):2865–2894, 2021.

[8] Ankur Sial, Amarjeet Singh, and Aniket Mahanti. Detecting anomalous energy consumption using contextual analysis of smart meter data. *Wirel. Netw.*, 27(6):4275–4292, 2021.

[9] Yi Zhang, Weiwei Chen, and Jason Black. Anomaly detection in premise energy consumption data. In *Proc. PES-GM*, pages 1–8, 2011.

[10] Muhammad Fahim and Alberto Sillitti. An anomaly detection model for enhancing energy management in smart buildings. In *Proc. SmartGridComm)*, pages 1–6, 2018.

[11] Jui-Sheng Chou and Abdi Suryadinata Telaga. Real-time detection of anomalous power consumption. *Renewable Sustainable Energy Rev.*, 33:400–411, 2014.

[12] Vikramaditya Jakkula and Diane Cook. Outlier detection in smart environment structured power datasets. In *Proc. IE*, pages 29–33, 2010.

[13] Soma Shekara Sreenadh Reddy Depuru, Lingfeng Wang, and Vijay Devabhaktuni. Support vector machine based data classification for detection of electricity theft. In *Proc. PSCE*, pages 1–8, 2011.

[14] A Amara Korba and N El Islem Karabadji. Smart grid energy fraud detection using svm. In *Proc. ICNAS*, pages 1–6, 2019.

[15] Jea Nagi, Keem Siah Yap, Sieh Kiong Tiong, Syed Khaleel Ahmed, and AM Mohammad. Detection of abnormalities and electricity theft using genetic support vector machines. In *Proc. TENCON*, pages 1–6, 2008.

[16] Christa Cody, Vitaly Ford, and Ambareen Siraj. Decision tree learning for fraud detection in consumer energy consumption. In *Proc. ICMLA*, pages 1175–1179, 2015.

[17] Matthias Reif, Markus Goldstein, Armin Stahl, and Thomas M Breuel. Anomaly detection by combining decision trees and parametric densities. In *Proc. ICPR*, pages 1–4, 2008.

[18] Denis Hock, Martin Kappes, and Bogdan Ghita. Using multiple data sources to detect manipulated electricity meter by an entropy-inspired metric. *SEGAN*, 21:100290, 2020.

[19] Bernat Coma-Puig, Josep Carmona, Ricard Gavalda, Santiago Alcoverro, and Victor Martin. Fraud detection in energy consumption: A supervised approach. In *Proc. DSAA*, pages 120–129, 2016.

[20] Vikramaditya Jakkula and Diane Cook. Detecting anomalous sensor events in smart home data for enhancing the living experience. In *Proc. AAAI workshops*, pages 33–37, 2011.

[21] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[22] João Henriques, Filipe Caldeira, Tiago Cruz, and Paulo Simões. Combining k-means and xgboost models for anomaly detection using log datasets. *Electronics*, 9(7):1–16, 2020.

[23] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[24] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[25] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*. MIT press Cambridge, MA, USA, 2017.

[26] Norman L Tasfi, Wilson A Higashino, Katarina Grolinger, and Miriam AM Capretz. Deep neural networks with confidence sampling for electrical anomaly detection. In *Proc. iThings, GreenCom, CPSCom, SmartData*, pages 1038–1045, 2017.

[27] Can Kaymakci, Simon Wenninger, and Alexander Sauer. Energy anomaly detection in industrial applications with long short-term memory-based autoencoders. *Procedia CIRP*, 104:182–187, 2021.

[28] Sangkeum Lee, Hojun Jin, Sarvar Hussain Nengroo, Yoonmee Doh, Chungho Lee, Taewook Heo, and Dongsoo Har. Smart metering system capable of anomaly detection by bi-directional lstm autoencoder. In *Proc. ICCE*, pages 1–6, 2022.

[29] Yu Weng, Ning Zhang, and Chunlei Xia. Multi-agent-based unsupervised detection of energy consumption anomalies on smart campus. *IEEE Access*, 7:2169–2178, 2018.

[30] Chia-Wei Tsai, Kuei-Chun Chiang, Hsin-Yuan Hsieh, Chun-Wei Yang, Jason Lin, and Yao-Chung Chang. Feature extraction of anomaly electricity usage behavior in residence using autoencoder. *Electronics*, 11(9):1–16, 2022.

[31] Ye Yuan and Kebin Jia. A distributed anomaly detection method of operation energy consumption using smart meter data. In *Proc. IIH-MSP*, pages 310–313, 2015.

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[33] Joao Pereira and Margarida Silveira. Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention. In *Proc. ICMLA*, pages 1275–1282, 2018.

[34] Wenjing Dai, Xiufeng Liu, Alfred Heller, and Per Sieverts Nielsen. Smart meter data anomaly detection using variational recurrent autoencoders with attention. In *Proc. INTAP*, pages 311–324, 2022.

[35] Marco Castangia, Riccardo Sappa, Awet Abraha Girmay, Christian Camarda, Enrico Macii, and Edoardo Patti. Anomaly detection on household appliances based on variational autoencoders. *SEGAN*, 32:1–11, 2022.

[36] Sungwoo Park, Seungmin Jung, Eenjun Hwang, and Seungmin Rho. Variational autoencoder-based anomaly detection scheme for load forecasting. In *Proc. ICAI*, pages 833–839. 2021.

[37] Douglas M Hawkins. *Identification of outliers*. Springer, 1980.

[38] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2):1–38, 2021.

[39] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. *ACM Comput. Surv.*, 54(3):1–33, 2021.

[40] Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. Variational attention for sequence-to-sequence models. *arXiv preprint arXiv:1712.08207*, 2017.

[41] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[42] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7:1991–2005, 2018.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[44] Victor Prokhorov, Ehsan Shareghi, Yingzhen Li, Mohammad Taher Pilehvar, and Nigel Collier. On the importance of the kullback-leibler divergence term in variational autoencoders for text generation. *arXiv preprint arXiv:1909.13668*, 2019.

A. Appendix A

A.0.1 Convolutional and Transpose Convolutional Layers

Convolutional layers [25] apply filters of a given size over an input in order to reduce its dimensions. The repeated application of the same filters over the input allows the extraction of meaningful information from data points and their close neighbors and the removal of any possible noise. To extract meaningful information from an input sequence, Convolutional layers reduce its width while increasing its depth. If Convolutional layers are meant to reduce the dimensionality of the input to extract important information from each data point and its neighbours, Transpose Convolutional layers do the opposite: they are used to increase the size of the compressed input until it restores its original shape. This category of layers is very useful to move from the latent representation of the input to its reconstruction that will be provided as output. The fact that meaningful pieces of information are retrieved considering not only the data point of our interest but also its neighbours is very useful when dealing with time series in which the correlation between adjacent time stamps is a fundamental aspect. However, Convolutional and Transpose Convolutional layers are not specifically meant to work with long sequences of data as they are not able to memorize time-related information.

A.0.2 Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) networks [43] are a specific type of Recurrent Neural Network (RNN) able to learn order dependencies in sequence prediction problems. They have been specifically designed to work effectively with time series and to understand how past data points affects the future ones. LSTM networks are widely employed due to their resilience to noise and for their ability to store information over arbitrarily long intervals of time steps. To increase the expressive power of LSTM layers, Bi-LSTM layers have been introduced. While LSTM layers receive the input once, from left to right, and so are able to focus on the relationship between past and future only, Bi-LSTM layers receive the input twice: once from left to right (forward layer) and once reversed, from right to left (backward layer), to focus both on the relationship between the past and the future and the relationship between the future and the past. The two inputs provide us with two outputs that are combined providing both of them to an activation layer which returns a single final output. Both LSTM and Bi-LSTM layers can provide us with a single output (the final hidden state obtained considering the entire input) or with an output sequence (a sequence of hidden states, one for each time stamp).

A.0.3 Kullback–Leibler Divergence

The Kullback–Leibler divergence (KL divergence) is a statistical distance that measures how a probability distribution is different from a second one. The loss functions of VAEs include a term that measures the KL divergence between the posterior distribution of the latent variable z and its prior p(z) (i.e., N (0, I)). The formula used is as follows:

$$KL\ loss = -0.5 \cdot (1 + \log(\sigma^2) - \mu^2 - \sigma^2)$$

Where μ and σ^2 respectively are the means and the variances computed by the encoder and that will characterize, for instance, an isotropic Gaussian distribution from which a latent representation of the input will be sampled. The KL divergence term can be interpreted as a regularizer that prevents the network from copying the input x into the latent variable z [44]. The implementation of the Variational Self-Attention mechanism requires the addition of a KL divergence term related to the sampling of the context vectors to the loss function formula. The introduction of this regularization term helps preventing the bypassing phenomenon we have discussed in paragraph 3.2.

A.0.4 Laplace Distribution

The Laplace distribution belongs to the location-scale family of functions and its probability density function is

$$f(x|\mu, b) = \frac{1}{2b} exp(-\frac{|x - \mu|}{b})$$

where μ is the mean (also called "location") of the Laplace distribution while b is its scale (also called "diversity"). Instead of returning as outputs the means and the variances of a Gaussian distribution, models can return as outputs the means and the scales of a Laplace distribution. This choice is motivated by the fact that the Laplace distribution minimize an $\ell 1$ reconstruction loss (while the Gaussian distribution minimize an $\ell 2$ reconstruction loss) that promotes sparse reconstruction errors, which helps the model to recognize anomalies more easily (under the assumption that anomalous observations are rare and sparse).

Abstract in Lingua Italiana

I temi dell'inquinamento ambientale e dello spreco energetico sono tra i più discussi degli ultimi anni. È ormai chiaro a tutti che questi problemi non possono più essere ignorati poiché le conseguenze sarebbero catastrofiche: dall'aumento delle temperature fino alla distruzione degli ecosistemi più fragili. Secondo lo United Nations Environment Programme, gli edifici sono responsabili per il 38% del consumo energetico globale. Questo, unito ai crescenti costi dell'energia, ci fornisce un'ottima ragione per studiare nuovi ed efficaci metodi di prevenzione degli sprechi energetici negli edifici. Uno degli approcci che è possibile adottare è quello di rilevare guasti ed anomalie nel consumo elettrico degli edifici tramite appositi algoritmi. Tra tutti i possibili algoritmi, questa tesi si concentra sugli Autoencoders, reti neurali ad apprendimento semi-supervisionato che forniscono eccellenti prestazioni di rilevamento delle anomalie. L'obiettivo di questa tesi è quello di presentare una nuova topologia di rete in grado di superare l'attuale stato dell'arte risolvendo il problema di bypass che affligge i Variational Autoencoders con Self-Attention. Il problema di bypass è causato proprio dal meccanismo di Self-Attention, il quale non performa al meglio quando si lavora con dati reali e dunque particolarmente rumorosi. Per risolvere il problema abbiamo progettato il meccanismo di Convolutional Variational Self-Attention, un'alternativa alla tradizionale Self-Attention in grado di mantenere alte le proprie prestazioni anche quando si lavora con dataset particolarmente rumorosi. Presentiamo inoltre al lettore un ampio confronto sperimentale, sia su un dataset univariato che su uno multivariato, dimostrando che la topologia di rete da noi proposta offre prestazioni migliori rispetto allo stato dell'arte e che, in caso di dataset particolarmente rumorosi, modelli più semplici possono offrire prestazioni migliori di quelli più complessi.

Parole chiave: Identificazione di anomalie, Consumi elettrici, Deep Autoencoders, Self-Attention

Acknowledgements

Ringraziamo il Professor Francesco Amigoni, il Dottor Davide Azzalini e la Dottoressa Benedetta Flammini per la disponibilità e per averci seguito meravigliosamente durante tutto il progetto di tesi.
Ringraziamo inoltre tutti i ragazzi del TIS Lab, l'ambiente più stimolante e divertente che abbiamo frequentato durante i nostri anni universitari. Siete capaci di accogliere di far sentire le persone a casa, di insegnare molto e di strappare sempre una risata. Siamo sicuri che questo laboratorio e le persone che lo frequentano riusciranno ad influenzare positivamente decine di futuri ingegneri.