POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

Analysis of Machine learning methods for Anomaly detection of Power Consumption in buildings

TESI MAGISTRALE IN COMPUTER SCIENCE AND ENGINEERING – INGEGNERIA INFORMATICA

AUTHOR: FRANCESCO DI SIMONE

ADVISOR: FRANCESCO AMIGONI

ACADEMIC YEAR: 2020-2021

1. Introduction

Anomaly detection is the process of finding patterns or points that distinguish themselves from a more regular collection. Finding such patterns could be extremely important since negative effects can take place when an anomaly is present in a system. That is the case of building power consumption. When in the time series an irregularity is present, an increase in costs and in pollution will inevitably be generated. This thesis wants to analyze some machine learning methods that were claimed to perform well in the anomaly detection field.

2. Analysis of the state of the art

This field does not seem to have a clear state of the art. A big number of methods have been implemented, many arguing to be better than the others, but most of those research did not fully and clearly explain how to obtain such results. Sometimes the datasets used are not published because they are private, sometimes the methods might be hand-crafted to work particularly well on a specific dataset but not as a general method. Therefore, in this thesis an exploration of some of the most promising techniques has been performed, by using two datasets, one from a facility in the Netherlands, the other one from an office building in Bergamo.

The anomalies were hand labelled by the author of this thesis, and were either artificially added, or were the holidays of the year, when the weekly pattern was disrupted. The methods analyzed were 2 types of autoencoders (one can be considered an offline method and the other an online one) two LSTMs, a multi-layer perceptron ensemble, support vector regression and random

forest. The results of this research were rather surprising since the methods did not seem to reach the same performances claimed in the original papers.

3. Datasets

Two different datasets were used in this work, both represent the power consumption of a facility. The first one is a dataset of a research building located in the Netherlands, the second one is an Italian office building in Bergamo.

3.1 Dutch Facility

A very clean dataset from 1997 (Figure 1), where most of the seemingly anomalous behavior comes from the low power consumption of holidays. It has a granularity of 15 minutes, which means 96 points per day, and it has been used by many papers mentioned before.

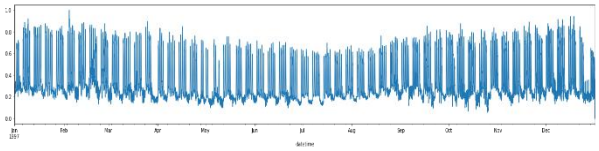

Figure 1: Dutch power consumption

3.2 Italian Building

The second dataset taken into consideration is about the power consumption of an office building in Italy over a period of three years, with a granularity of 15 minutes. Associated with it there are many other interesting features such as average temperatures and humidity but, most importantly, it came with power consumption per each floor. Every floor has a specification about its consumptions:

- First floor: it concerns energy consumption for underground parking lighting, electric vehicle charging stations, UPSs for IT, front desk, outdoor lighting, firefighting system, and mechanical workshop.

- Second floor: this floor aggregates the consumptions of lighting and power outlets of three office floors, auditorium, and cafeteria.

- Third floor: finally, this floor reports the power consumptions of the heating and cooling system, the ventilation system, elevators, kitchen, and cafeteria's refrigerators.

For this dataset there are no labeled anomalies. The analysis phase was performed in the same fashion as for the previous dataset but, since holidays are not anomalies, in this case, they are not highlighted.

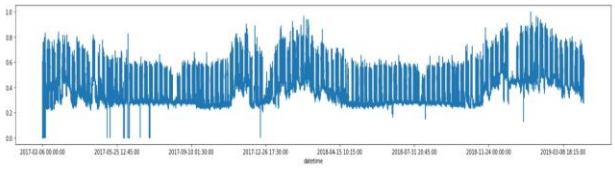

Figure 2: Italian dataset

The first two floors are very similar in power. For this reason, only the second will be considered in the following analysis shown in Figure 2.

By just looking at the time series graph of the three floors it becomes obvious that these time series are less predictable than the first dataset, especially the third floor. Furthermore, there are far less prominent outliers.

3.3 Experimental settings and exceptions

Both datasets have been used for the implementation of the papers and both had to be somehow regularized. But while the Dutch one was complete and did not have any missing values, the Italian one was filled with holes, missing points and many negative/unrealistic values probably caused by faulty sensors. Two options were available at this point:

-Try to fill the missing data by interpolation / copy data from the previous week

- Try to select time windows that only contain nominal data to feed the NN.

The first option while it may seem more correct, it does bring in a few problems. First, there are multiple sequential points missing, sometimes entire days. Obviously, an interpolation does need

a starting point and a finishing one but with this much missing data, the good theoretical effect of interpolation would be nullified.

Second, by copying the data from the previous week we might be adding fictitious anomalies, which would increase the bias and variance of the model trained on such data.

Moreover, the experimental setting had to be modified according to the needs of the paper. Sometimes it was just not possible to achieve good results in one paper by just feeding it the same dataset of another one because of missing points or incomplete sequences.

In Figure 3 we can see a table containing 96 points, collected every 15 minutes for a temporal range of 24 hours.

- AvgP: The power dissipated by the building (Second floor).

- hour: A variable containing a normalized version of the hour of the day.

- Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday: Columns containing the one hot encoding of the week.

- anomaly: tells whether that point is to be considered an anomaly.

DateTime	AvgP	hour	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	anomaly
2017-08-01 00:00:00	41	0.000000	0	1	0	0	0	0	0	0
2017-08-01 00:15:00	43	0.000000	0	1	0	0	0	0	0	0
2017-08-01 00:30:00	42	0.000000	0	1	0	0	0	0	0	0
2017-08-01 00:45:00	42	0.000000	0	1	0	0	0	0	0	0
2017-08-01 01:00:00	42	0.041667	0	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
2017-08-01 22:45:00	42	0.916667	0	1	0	0	0	0	0	0
2017-08-01 23:00:00	41	0.958333	0	1	0	0	0	0	0	0
2017-08-01 23:15:00	41	0.958333	0	1	0	0	0	0	0	0
2017-08-01 23:30:00	42	0.958333	0	1	0	0	0	0	0	0
2017-08-01 23:45:00	41	0.958333	0	1	0	0	0	0	0	0

Figure 3: Dataframe

3.4 Results

Four indices were taken in consideration when trying to evaluate the models:

- Precision
- Recall
- F1 score
- Area under the Curve

In table 1 and 2 we can see the results obtained for each category by each method in the Dutch and Italian datasets respectively.

DUTCH DS	Prec	Recall	F1	AUC
Offline AE	1	0,75	0,85	0,91
SVR	0,77	0,98	0,86	0,99
RF	0,60	0,97	0,75	0,99
Online AE	0,28	0,94	0,44	0,95
MLP	0,44	1	0,61	0,85
LSTM 1	0,2	1	0,33	0,89
LSTM 2	0,33	0,04	0,07	0,47

Table 1: Performance metrics of Dutch dataset

Bergamo DS	Prec	Recall	F1	AUC
Offline AE	0,86	1,00	0,92	0,99
SVR	0,67	0,98	0,80	0,99
RF	0,58	0,95	0,73	0,99
Online AE	0,32	0,94	0,47	0,90
MLP	0,41	0,83	0,55	0,81
LSTM 1	0,33	0,80	0,47	0,62
LSTM 2	0,01	1,00	0,01	0,57

Table 2: Performance metrics of Italian dataset

As it can be seen, the statistical methods (SVR and Random Forest) are the ones that overall obtained the best performances. The two autoencoders had drastically different performances but managed to highlight all the anomalous days even if only partially at times. All the other methods performed significantly worse, showing lack of robustness under these experimental settings.

4. Transfer Learning

It is not simple to obtain data for buildings, in general when a new structure is built, new data must be collected if we wanted to use one of the anomaly detection methods presented in this paper.

But collecting it might be expensive and time consuming, therefore a possible solution might be transfer learning. Transfer learning for deep neural networks is the process of first training a base network on a source dataset, and then transferring the layers to a second network to be applied to a target dataset. This can improve the performances of the network as shown by [3].

Transfer learning applied to time series seems to be an unexplored field, but it does seem to have a lot of potential.

Bergamo1	Prec	Recall	F1	AUC
Offline AE	0,86	1,00	0,92	0,99
Online AE	0,32	0,94	0,47	0,90
Offline AE TL	0,83	1,00	0,90	0,98
Online AE TL	0,61	0,84	0,71	0,87

Table 3: Performance metrics of first Italian test set with Transfer Learning

Bergamo2	Prec	Recall	F1	AUC
Offline AE	0,79	0,81	0,80	0,87
Online AE	0,71	0,67	0,69	0,78
Offline AE TL	0,82	0,72	0,77	0,86
Online AE TL	0,73	0,61	0,67	0,75

Table 4: Performance metrics of winter Italian test set with and without Transfer Learning

While in this case, as shown in Table 3 and Table 4, the performances did not have a significant improvement, they managed to be stable even though the two datasets used were significantly different. Both datasets deal with power consumption information, but it is easy to see that the winter period of the Bergamo dataset used as test set is extremely different from the winter period registered by the Dutch facility.

The use of the Dutch dataset for training a neural network could be applied to the anomaly detection process in new buildings in Lombardy at least for the first years when the proprietary data is being collected. This will result in a final dataset with fewer anomalies, since they can be promptly caught thanks to a momentary anomaly detection system implemented using transfer learning.

5. Conclusions and future work

This work focused on trying to understand which machine learning method performs better for detecting anomalies in the context of building energy consumption. It was not easy to draw a conclusion because, for instance, the online autoencoder performed well in the partial recognition of the anomalies and managed to detect most of the anomalous days correctly but failed to predict the precise hours of the day at which the anomalies occurred, therefore lowering the performance metrics. This does not mean that such methods are not fit for this purpose, but rather that more research should be made before applying these methods in the real word. It must be noted that the entire datasets used here were not labelled at the source, but labelling was a task performed by the author of this thesis, who is not an expert in the field of building management. There are chances that the anomalies registered by a method like the online autoencoder were actual anomalies in the system that were not recognized when hand labelling. In any case all these results are conservative: even though the precision of the autoencoder is low it just means that there are many false positives, but the main anomalies were all discovered with good precision. The same can be said about the transfer learning technique: it obtained results on par with normal training. There is not a lot of background for transfer learning in time series, more research should be done on this topic as it seems to be promising, possibly, by trying to use properly labelled datasets. This thesis was therefore an explorative research in which the best methods according to performance obtained seem to be the statistical methods of SVR and Random Forest and the offline AE. It is extremely hard to evaluate all of these methods and compare them to the results reported in the original articles, since the datasets used by them were different and not publicly available. The transfer learning technique seemed rather promising especially as a temporary anomaly detection system to be used in BEMSs during the data collection period for buildings that do not adopt any anomaly control strategies. Therefore, in the future, a further exploration of autoencoders that implement transfer learning could be useful to confirm and extend the results obtained in this work. The focus should be on the usage of datasets that have more regular features, are complete and are properly labeled by experts.

Bibliography

[1] Cheng Fan, Fu Xiao, Yang Zhao, Jiayuan Wang, Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data, Applied Energy, Volume 211, 2018, Pages 1123-1135

[2] Daniel B. Araya, Katarina Grolinger, Hany F. ElYamany, Miriam A.M. Capretz, Girma Bitsuamlak, An ensemble learning framework for anomaly detection in building energy consumption, Energy and Buildings, Volume 144, 2017, Pages 191-206,

[3] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar , Pierre-Alain Muller, "Transfer Learning for time series classification" arXiv:1811.01533

POLITECNICO
MILANO 1863

Master's degree in Computer Science and Engineering

Dipartimento di Elettronica, Informazione e Bioingegneria

Analysis of Machine Learning Methods for Anomaly Detection of Electrical Power Consumption in Buildings

Supervisor:

Prof. Francesco Amigoni

Co-Supervisor:

Eng. Davide Azzalini

Francesco Di Simone

944661

ACADEMIC YEAR 2020/2021

Abstract

Anomaly detection is the process of finding patterns or points that distinguish themselves from a more regular collection. There are different causes for this happening: it could be caused by external factors or related to faults in the system that generated the patterns or the points. Sometimes data come as time series, like for power consumption of buildings. One way to perform anomaly detection is to "teach" a machine learning method what the normal behavior of a system should be and then identify the deviations. The goal of this thesis is to investigate whether and to what degree some of the most promising machine learning methods, such as autoencoders, LSTM and statistical methods, have the potential to deal with the problem of identifying anomalies in power consumption time series in buildings. The reason for finding such points that disrupt the general trend is not only economical, since they might create unexpected costs, but also concerns the environment because an excessive consumption of resources in buildings generates more pollution. In a nutshell, the aim of this thesis is to try to find what method works best in this setting and understand why they might perform better under certain

conditions, and then try to propose a solution to the lack of datasets used for training the previously mentioned methods. This last solution is based on transfer learning.

Keywords: Anomaly detection, Autoencoders, Neural Networks, LSTM, Transfer Learning, Power Consumption Time Series, Building, Machine Learning.

Sommario

Il rilevamento di anomalie è il processo grazie al quale è possible trovere pattern o punti che si distinguono da un insieme più regolare. Ci sono svariate cause per l'emergere delle anomalie: potrebbero essere causate da fattori esterni oppure da problemi interni al sistema considerato. I dati sul consumo energetico degli edifici sono raggruppati in serie temporali nelle quali le anomalie vanno individuate. Un modo per attuare il rilevamento di anomalie è quello di "insegnare" ad un metodo di machine learning quale dovrebbe essere il comportamento normale di un sistema e poi riconoscere gli scostamenti. Quello che questo lavoro di tesi vuole vuole indagare è fino a che punto alcuni dei metodi più promettenti di machine learning, come autoencoders, LSTM e metodi statistici, hanno il potenziale di individuare anomalie nei dati di consumo energetico negli edifici. La ragione per trovare questi punti anomali che rompono il trend nominale non è solo economica, poichè potrebbero portare a costi inaspettati, ma riguarda anche l'ambiente, perchè un consumo eccessivo di risorse negli edifici può generare in incremento dell'inquinamento. In sintesi, lo scopo di questa tesi è quello di provare a capire

quale metodo funziona meglio in questo campo e capire perchè alcuni metodi potrebbero funzionare meglio sotto certe condizioni, e poi proporre una soluzione alla mancanza di dataset utilizzati per il training dei metodi mezionati precedentemente. Questa soluzione è basata sul transfer learning.

Parole chiave: Rilevamento di Anomalie, Autoencoder, Reti Neurali, LSTM, Transfer Learning, Serie Temporali, Consumo Energetico, Edificio, Machine Learning.

This thesis is dedicated to my family and to all those that have stayed by my side over the years, that have helped me and believed in me.

INDEX

Chapter 1

Introduction

With the term "anomaly detection" we mean the ability to recognize elements that behave abnormally, from a group that instead is considered to follow a certain pattern or to have certain characteristics. These elements deviate from the nominal behavior.

Anomaly detection is a field that has been widely explored during the years, and many methods have been introduced, ranging from statistical methods to machine learning methods [14]. These methods can be quite important when applied to time series. Time series are collections of points ordered by time, and they generally follow a periodic pattern, meaning they tend to show behaviors that should be somewhat predictable. Being able to recognize whenever a certain subsequence in a time series acts abnormally could have a huge impact on applications.

Anomaly detection has been widely used in various industries, such as intrusion detection in network systems, fraud detection in financial transactions, and patient health monitoring in medical treatment. For instance, looking at electrocardiograms (ECGs) as a time series of points, detecting an anomaly inside could save a person's life. Another example could be the detection of anomalous points in a time

series of the power produced by an engine over a period: it is easy to see how analyzing such a collection of points could lead to finding improvements to better optimize the work done by the engine [9].

This type of reasoning can be applied to more massive systems through the use of sensors, that are getting cheaper and more accurate. The example considered in this thesis is the detection of anomalies in the energy consumption of a building.

The real-time building operational performance can be monitored and controlled through the Building Energy Management Systems (BEMS) or Building Automation Systems (BAS). Massive amounts of building operational data are being collected and available for data analysis [7]. It is therefore very promising to develop data-driven approaches to achieve reliable and robust anomaly detection.

A BEMS must deal with a lot of changes throughout the year, for instance the internal temperature regulation through the heating, ventilation, and air conditioning system (HVAC), to maintain a high comfort level for those living or working inside. By implementing an anomaly detection system inside a building, the building managers could easily be notified about an internal problem and promptly act to solve it. This does not only decrease the costs sustained by the owners and increase the comfort of those inhabiting the complex, but it also has a huge impact on the environment since building operations are energy-intensive and contribute to approximately one third of the United States final energy consumption [1][7].

The reduction of power consumption in building environments could support the urgently needed reduction in the world-wide power consumption and the related environmental interests. BEMSs have a substantial energy saving potential considering for instance the wide presence of equipment faults and energy-wasting occupant behaviors. In buildings, an anomalous behavior of an electrical device or of the end-user could occur either because of a faulty operation of a device, end-user negligence (e.g., cold loss in a room by keeping a window open while the air conditioner is on or refrigerant leak in a fridge via maintaining the fridge door open), a theft attack, a non-technical loss, etc. From 15% to 30% of the energy waste in commercial buildings is due to the performance degradation, improper control strategy and malfunctions of HVAC systems [1]. An occurrence of anomalous behavior could lead to higher power consumption, longer operation time than necessary and could result in a permanent malfunction of the device, caused for instance by overheating. So, anomalies are not to be looked at from a cost perspective only, but also considering that faults in electrical systems could cause even more damages by ruining components, damage circuits, and cause data losses. It is therefore essential to find a way to avoid energy waste as much as possible.

An anomaly could be happening with rarity maybe during days off, making it hard to identify it without a support system. Or, even worse, it could be happening silently and be integrated and learned by the anomaly detection system as nominal data, therefore making it unrecognizable. It is therefore essential for this task to be

able to train the system on a long enough nominal dataset with only recognizable and complete weekly patterns, but generic enough so that the system manages to learn representations useful for the whole year. All the data that do not fit into the nominal category must not be fed to the algorithms responsible for the anomaly detection process.

There are several challenges inherent to anomaly detection. Some are related to the definition of anomalies, some to the evaluation of the anomaly detection algorithms and to availability of datasets.

In this thesis we used different methods for anomaly detection to understand and compare how well they can perform in context of building power consumption by applying them to two power consumption datasets belonging to a Dutch research facility and an Italian office building located in Bergamo. Statistical methods and offline autoencoder seem to be the most accurate methods, while the others performed significantly worse obtaining a high number of false positives. Furthermore, an analysis of transfer learning , which is an innovating technique that has not been widely applied to the setting of time series, is carried out, showing promising results as a temporary anomaly detection system in buildings that do not have yet proprietary data.

This thesis is organized as follows:

In Chapter 2 we give an overview of the theory and concepts that are essential for understanding the work done in rest of the project. First, some preliminary information about the basic concepts of neural networks is given, and then the most interesting methods used in the literature are presented. The different techniques are briefly explained and divided into categories, according to the nature of the method (statistical, deep learning etc.).

In Chapter 3 the reader will be introduced to the experimental settings used for this work: the datasets used, the transformations done to them, the application of the different methods for anomaly detection, the results of the experiments performed in this thesis and the corresponding comments.

Chapter 4 focuses on one technique that could improve the existing methods, known as transfer learning.

Chapter 5 presents the conclusions of this thesis and introduces possible future works.

Chapter 2

Literature Review

2.1 Preliminaries

When given a time series, which is a collection of data points ordered by timestamp, it is sometimes possible to find anomalous points when an anomalous event is recorded. It is important to define what an anomaly is inside a system, and we can therefore identify three types of anomalies [32]:

1- Point anomalies: a point is considered anomalous on its own when it behaves substantially differently from the other data points.

2- Contextual anomalies: if a data point is abnormal when viewed in a particular context but normal otherwise it is regarded as a contextual anomaly. Context is often present in the form of an additional variable. Most common examples of this kind are present in time series data when a point is within normal range but does not conform to the expected temporal pattern.

3- Collective anomalies: a subset of data points within a data set is considered anomalous if those values as a collection deviate significantly from the entire dataset, but the values of the individual data points are not themselves anomalous. For instance, when a time series deviates from its usual pattern.

Anomalies are rare events, and it is not possible to have a prior knowledge of every type of anomaly. Moreover, the definition of anomalies varies across applications. Though it is commonly assumed that anomalies and normal points are generated from different processes.

This thesis wants to analyze some pre-existing methods, mainly using a particular type of neural network known as *autoencoder*. It will be done by implementing the methods proposed in some papers (slightly modified as the context, data, and the system itself used in this thesis were different) and later a discussion on a possible improvement using transfer learning will be started.

2.1.1 Neural Networks

In recent years, deep learning has emerged as one of the most popular machine learning techniques, obtaining state-of-the-art results for a range of supervised and unsupervised tasks [33]. The primary reason for the success of deep learning is its ability to learn high-level representations which are relevant for the task at hand.

These representations are learned automatically from data with little or no need of manual feature engineering and domain expertise.

Neural networks (NNs) are a type of machine learning model inspired by the way the human brain works. The smallest unit in the neural network is called a node or more commonly a neuron. A neuron receives inputs from the incoming edges and multiplies the input by the corresponding edge weight, and then a non-linear function is applied. This is called activation function. Thanks to the activation function, an output is produced.

The working of a neuron is illustrated in Figure 1 and can be represented mathematically by the vector Equation 1, where the symbols x, w, b, dot, f, and y represent input vector, weight vector, neuron bias, dot product, activation function, and neuron output respectively.

$$\text{Eq. 1: } y(x) = f(w \cdot x + b)$$

The output of a neuron is a non-linear function of the weighted sum of its inputs. The non-linearity is introduced by the activation function.

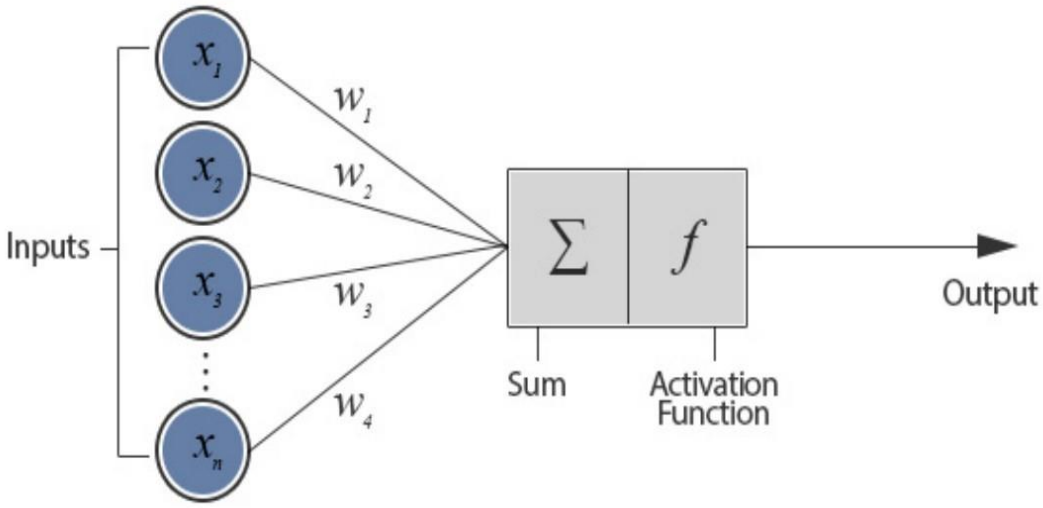

Figure 1: Basic structure of a neuron

2.1.2 Activation Function

The most used activation functions include logistic sigmoid shown in Equation 2, hyperbolic tangent in Equation 3 and the rectified linear units in Equation 4 (ReLU). For regression problems which require predicting continuous values, linear activation is used as presented in Equation 5.

Eq. 2: $\sigma(z) = \frac{1}{1+e^{-z}}$

Eq. 3: $\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$

Eq. 4: $ReLu(z) = max\,(0, z)$

Eq. 5: $a(z) = z$

The neurons are divided in layers and the neurons of a layer can be connected through edges to the previous and to the following layer. The first layer that receives an input is called the input layer. The last layer, that emits the final output is known as output layer. All the remaining layers are referred to as hidden layers.

2.1.3 Loss Function

During training, the network is presented with input data along with corresponding outputs. A loss function which measures the distance between network output and the desired output is constructed to facilitate learning. The loss function commonly used for a regression problem (i.e., predicting a continuous value) is the mean squared error (MSE). MSE is computed as shown in Equation 6, where N is the number of observations, y_i denotes the true value, and the predicted value is denoted by $\tilde{y}_i$. MSE measures the averaged squared distance between the predicted values and true values. The difference between the true value and predicted value is also referred to as the error or residual.

$$Eq.6: MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \tilde{\tilde{y}}_i\right)^2$$

Another commonly used metrics is the MAE or mean absolute error (Equation 7).

$$Eq.7: MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \tilde{y}_i|$$

The next step is to feed the data to the neural network and start the learning process. The learning process is an optimization problem, with the goal to minimize the loss function by tuning the parameters of the neural network. One of the most used optimization algorithms is called gradient descent. Gradient descent involves calculating the gradients of the loss function with respect to the network parameters i.e., weights and biases. But with big datasets, calculating the loss and gradient over the entire dataset may be too slow and computationally infeasible. Thus, in practice, a variant of gradient descent called stochastic gradient descent (SGD) is adopted. In SGD, the data is divided into subsets called batches, and the parameters are updated after calculating the loss function over one batch. Other popular variants are: RMSprop, AdaGrad, Adam [25].

The method used to compute the gradients is called back-propagation and is based on the chain rule of derivatives [2]. The gradient is a measure of the change in the loss value corresponding to a small change in a network parameter. A scalar value called the learning rate (γ) is used to update the parameters (θ) in opposite direction of the gradient. The process is done iteratively by making several passes over the training data. A pass over training data is called an epoch and after every epoch the parameters move closer to their optimum values which minimizes the loss function. In Equation 8 the formula for the gradient descent is presented.

$$Eq.8: \theta = \theta - \gamma * \frac{\partial L(\theta)}{\partial \theta}$$

A common problem in training Neural Networks is overfitting. Overfitting occurs when the model learns to fit the noise in training data and is often the result of using a more complex model than required. In the presence of overfitting, the model performs well on training data but poorly on new data as it is not able to generalize. Some methods to avoid overfitting exist, such as early stopping, where a small subset of the training data is used as a validation set. After every epoch the value of the loss function on the training set is compared to the value on the validation set. If the loss on the validation set starts increasing even though the loss on the training set is decreasing, it is an indication of overfitting, and the model training can be stopped. Dropout is another method used, where a fixed percentage of neurons are "turned off" at every epoch.

While the network parameters like weights and biases are learned through training, the hyperparameters like the number of epochs or the learning rate, must be set before training by the user. Some tweaking is usually required to obtain good results.

After the training is completed and the test set has been evaluated, what we are left with, is a prediction vector. In the vector the class for the normal state is assigned the class label 0 and the class with the abnormal state is assigned the class label 1.

2.1.4 Performance Metrics

We can compare the results obtained through the prediction with the vector containing the real anomalies. Doing so will give us information about the number of values that were correctly classified and those who weren't. With the true positives, false positives, true negatives, and false negatives we are able to calculate different indexes of performance:

-The *precision* is the ratio $\frac{tp}{tp+fp}$ where *tp* is the number of true positives and *fp* the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

-The *recall* is the ratio $\frac{tp}{tp+fn}$ where *tp* is the number of true positives and *fn* the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.

-The *F1 score* can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is $\frac{2*precision*recall}{precision+recall}$.

- The false positive rate and true positive rate are useful to obtain the so-called *area under the curve* of the ROC curve or *AUC*. The ROC curve is plotted with true positive rate(TPR) against the false positive rate(FPR) where TPR is on the y-axis and

FPR is on the x-axis. Since the TPR and FPR are both ranging from 0 to 1, the AUC will also range from 0 to 1. The AUC indicates how good a model is at separating between classes, therefore in general the higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.

2.2 State of The Art

In this subsection the state of the art on the methods used for anomaly detection is analyzed.

As it was previously specified, a wide range of methods is available, and the choice of the method depends on the context and the type of anomaly analysis.

We can divide the type of data used into two major categories:

- Power consumption time series: the most commonly available data. The methods take the time series by themselves or together with other data like exogenous variables (temperature, weather etc.) and try to learn to forecast data for that time series. After the prediction is made, it is compared to the real behavior. Then using a threshold, the system tries to understand whether there is an anomaly in the considered time window. This function can change from a simple threshold to a more complex system that integrates this step in the learning process like in variational autoencoders through the use of probabilistic measures, thus not requiring a data-

specific threshold [3]. The problem of this approach is that it approximates the whole system with a single time series and thus if anomalous behaviors of different components balance out, for example if an appliance consumes less and another more than expected, it might not be able to detect an anomaly.

- Power consumption time series per component: instead of a single time series, the dataset is composed of many different time series, one for each component of the system. In this way if the system does not show any anomalous behavior, the solution is able to understand if there are multiple anomalies that balances out over the system consumption. The issue with this approach is the significant rise in the complexity of the problem due to higher dimensionality of the data and the possible non-trivial modeling of relationships between these series.

With different contexts it is necessary to apply different solutions as there's not yet one single method able to perfectly classify all the anomalies in each context, and different algorithms perform better when applied to the right problem [31].

2.2.1 Unsupervised Detection

It aims at detecting previously unknown rare consumption observations or patterns without using any a priori knowledge of these observations. Generally, this

kind of detection assumes that the amount of anomaly patterns to the overall consumption data is small. Abnormalities in this case are unknown to the technique as the data is not labelled, therefore detecting anomalous consumption is reduced to the modeling of normal consumption behavior and to the recognition of abnormal patterns. Unsupervised techniques are mainly built on clustering, semi-supervised learning, and dimensionality reduction algorithms.

-Clustering: it is a machine learning technique used to split power consumption data into various clusters and helps in classifying them into normal or abnormal in unlabeled datasets. Many authors referred to this method in their work like [4][5][6]. In some, a version of clustering called k-means is used [4]. The k-means algorithm tries to divide the observations into *k* clusters, in which each observation belongs to the cluster with the closest mean. It is very fast (one of the fastest clustering algorithms available), but it falls in local minima. That's why it can be useful to restart it several times.

-Dimensionality Reduction: in different machine learning applications, dimensionality reduction could be used as a feature extraction to help with the final classification with a low computational cost as it can remove irrelevant power patterns and redundancy. Some of the most famous techniques belonging to this category

are Principal Component Analysis (PCA) [1] [7] and linear discriminant analysis [8].

-Generative Adversarial Networks: a deep learning solution to deal with the unbalanced property of anomaly detection datasets is generative adversarial networks (GAN) . It simultaneously trains two models: a generative model that captures the data distribution, and a discriminative model that estimates the probability that a sample came from the training data rather than the generative part [34]. By training the GAN with nominal data it is possible to apply it to the anomaly detection problem and use the same threshold method to find points that do not follow the original data patterns. It can model complex and high-dimensional data of different types like images and time series.

2.2.2 Semi-supervised Detection

Semi- supervised learning is an approach that combines a small amount of labeled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning (with no labeled training data) and supervised learning (with only labeled training data).

- Auto-Encoders: autoencoders (AE) are neural networks with a particular neuronal disposition. They are used in order to learn efficient data encoding in an unsupervised manner. The encoder maps input data $x \in \mathbb{R}^{dx}$ to a latent space (or code) $z \in \mathbb{R}^{dz}$ and the decoder maps back from latent space to input space [29]. The autoencoders training procedure is either unsupervised or semi-supervised and it consists of finding the parameters that make the reconstruction $\hat{x}$ as close as possible to the original input x, by minimizing a loss function that measures the quality of the reconstructions (e.g., mean squared error). Typically, the latent space z has a lower dimensionality than the input space x and, hence, AEs are forced to learn compressed representations of the input data. This characteristic makes them suitable for dimensionality reduction tasks, where they were proven to perform much better than other dimensionality reduction techniques, such as Principal Component Analysis [3]. Autoencoders are an unsupervised /semi-supervised learning method, which does not need anomalies labels in order to learn to find them, even though this information is still necessary in order to make performance evaluation. The variational autoencoder is a deep generative model that constrains the latent code z of the conventional AE to be a random variable distributed according to a prior distribution $p(z)$, usually a standard normal distribution, Normal(0, I).

As shown in Figure 2 taken from [26], two autoencoder layouts are commonly used based on the dimensionality of the neurons in the input and output (m) and the dimensionality of the neurons in the hidden layer (p):

1. an under-complete or a bottleneck layout where p is smaller than m.
2. an over-complete layout where p is larger than m.

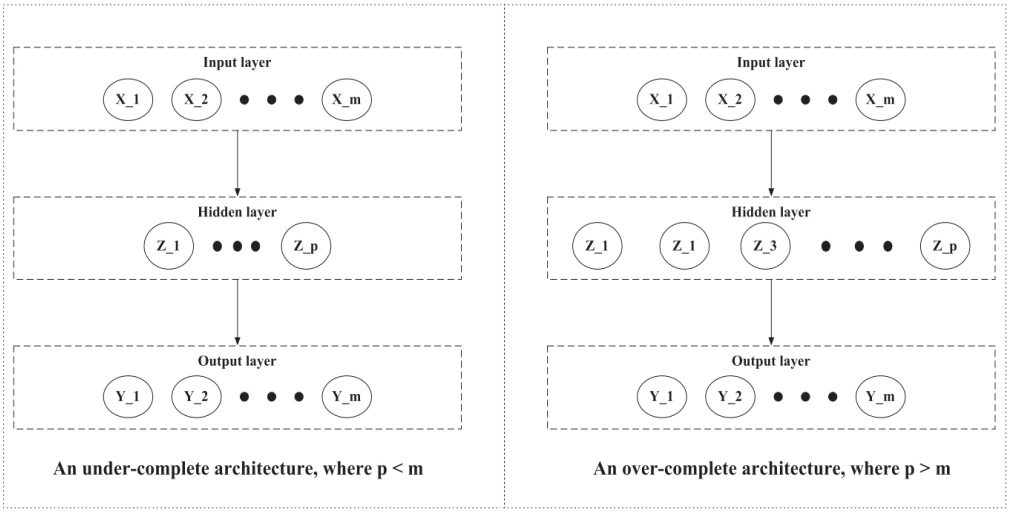

Figure 2: Structures of under-complete and over-complete autoencoder taken from [26]

The under-complete layout learns a compressed representation of X while the over-complete layout learns a sparse representation of X. The sparse over-complete representation can be regarded as a special case of under-complete representation, as the majority of hidden neurons are forced to be zeros.

2.2.3 Supervised Detection

Supervised anomaly detection in energy consumption necessitates training the machine learning classifiers using annotated datasets, where both normal and abnormal power consumptions are labeled. Although supervised anomaly detection can achieve high-accuracy identification results as demonstrated in academic frameworks, its adoption in the real world is still limited compared to unsupervised methods, due to the absence of power consumption annotated datasets.

- Long Short-Term Memory: It is a particular type of recurrent neural network, which can learn from past experience happening in a temporal order LSTMs were developed to deal with the vanishing gradient problem that afflicts the basic recurrent neural network. This method uses a particular structure over the neurons in the network in order to behave as a brain-like memory. It means that it can learn to remember knowledge from past experiences, making it more suitable to learn periodic or repetitive behaviors, typically found in time series which follow a similar pattern under certain conditions. Many different approaches have been implemented using this type of neural network in anomaly detection [9]. In time series there usually is repetitiveness, given by seasonal and weekly trends and patterns. It is therefore simple to see how a recurrent neural network that exploits LSTM with the ability to learn from these patterns can be applied to power consumption

time series. While falling in the supervised learning category, LSTMs could be also inserted as a layer inside an autoencoder, therefore they can also be applied to unsupervised/semi-supervised learning.

- Regression: It refers to identifying the relationship between two or more power variable classes in order to obtain model parameters to predict the generation of abnormal power observations [18]. Various regression models have been introduced in the literature to identify abnormalities in building energy consumption, including linear regression and support vector regression (SVR) [14], the latter is an extension of Support Vector Machines (SVM). SVMs use hyperplanes in multidimensional space to separate the different classes. While generally it is a supervised method, for the scope of this thesis it has been used in an unsupervised manner since it can also be used without labels, by setting a threshold and applying it to the reconstruction error obtained.

2.2.4 Statistical Models

A totally different approach are statistical models. These models use statistical analysis of the data in order to compute the probability of a value or even the value itself in time series. Statistical models are used due to their great accuracy over periodical patterns. They embody this last statistical assumption and assign lower

probability to data which do not follow the assumption. The most used approaches in the literature are the following:

- Seasonal Auto-Regressive Integrated Moving Average (SARIMA): ARMA, ARIMA and SARIMA are often used in this field since they usually are the way to go for time series analysis and forecasting [7]. They assume that the series maintains its stationarity or that an integration could make it stationary. This is usually the case for power consumption time series which do not show deviations from their previous behavior.

- Generalized Additive Model (GAM): this is a very versatile model able to capture non-linear relationships in the data. The literature has shown the usefulness of this method, if combined it with ARMA models and weather data, to capture non-linear forms of non-stationarity of the data [10] or combined with ARCH, a different autoregressive model, focusing more on error variance [11].

- Symbolic Aggregate approXimation (SAX): while there are at least 200 different symbolic approximations of time series in the literature , SAX is unique in that it is the only one that allows both dimensionality reduction and lower bounding of L^p norms [12]. Applications of this technique have been used to find time series, to

mine rules in health data, for anomaly detection, to extract features from a hepatitis database, for visualization, and a host of other data mining tasks [12]. This function is usually used to make time series easier to read and analyze, by transforming the data in a string of symbols. It has been shown its usefulness in anomaly detection since it can capture deviations from the normal behavior of time series and make these deviations very simple to understand.

2.2.5 Ensemble Methods

During the training process, the models are subject to errors caused by the accumulation of bias and the introduction of variance. For instance, retraining the same model might give extremely different predictions in the test phase, as the initial randomized parameters have a huge influence on the result. One way to mitigate this problem is to use ensemble methods. Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to decrease variance (bagging) or bias (boosting). The main concept is that many unstable learners' predictions are combined to obtain a more robust result. The method proposed in [13], takes advantage of Ensemble Empirical Mode Decomposition (E-EMD) to uncover the patterns of power-draw signals, thereby enabling them to estimate the intrinsic inter-device correlations.

In [14] three methods are combined to obtain a more stable result. An autoencoder, a random forest (RF) which is a widely used ensemble learning approach for both classification and regression problems, and support vector regression which is a particular implementation of a SVM applied to regression, a technique that can reach good levels of generalization. To then be able to evaluate the performance of this method, the tp, fp, tn, fn parameters are calculated following a method consisting in evaluating whether or not a sample has been classified in a positive or negative manner creating a matrix called Anomaly Classifier of dimension n x m where n is the length of the error vector found after the prediction(aka MSE, MAE etc.), and m is the length of the Unique error values vector which is the set of the error vector. This same Anomaly Classifier method has been implemented in this thesis to make an evaluation. After all the Anomaly classifiers have been found for the three methods, all the possible combinations of the three classifiers are computed using a majority voting approach, which was not implemented here because of the computational demand.

2.3 Difficulties of Anomaly Detection

2.3.1 Problems with Missing Labels

The biggest enemy common to all of the state-of-the-art approaches for anomaly detection is definitely the lack of labelled data. There is currently almost no dataset

with labels and many of the ones used in the experiments previously described are proprietary, therefore not publicly available. Others instead might add some artificially generated anomaly, which implies the impossibility to perfectly reproduce the dataset and obtain a fair comparison. This makes it almost impossible to compare the performance of algorithms. Ideally it seems easy to compute the performance of an anomaly detection algorithm, but without properly labelled data that is not feasible.

2.3.2 Problems Identifying the Boundary of the State of the Art

While in general in other fields circumscribing a state of the art might not be too challenging, in this case it is especially difficult. It is not only the lack of labelled dataset but also of the lack of shared public datasets. Most of the datasets used in the papers are not available and this makes it extremely challenging to be able to compare the qualities and flaws of different algorithms. Not only that but it is also almost impossible to find a codebase for the algorithms implemented. One must start from scratch every time.

Another problem are datasets that do not show anomalous behavior: the only public dataset used with labeled anomalies, uses holidays as anomalies. That is the case of the Dutch dataset used in this thesis and in that will be presented in Chapter 3. Although this is useful to understand the ability of the different algorithms to

learn this kind of anomalous behavior, it does not reflect the behavior of anomalies in BEMS.

And finally, even if the datasets were shared, most papers decide to inject artificial anomalies without explaining how they have been injected which makes a comparison almost impossible, since a different injected anomalous behavior could result in lower-than-expected performance of the newly implemented algorithm.

2.4 State of the Art in reality

In Table 1 and 2 we can see the papers and scientific articles previously mentioned, that have implemented the solutions mentioned from Subsection 2.2.1 to Subsection 2.2.5.

Paper Reference	Anomalies	Algorithm
[9]	Holidays (unlabeled)	LSTM
[12]	Holidays (unlabeled)	Symbolic Aggregate approXimation (SAX)
[15]	Holidays (unlabeled)	Exemplar-based model + SST (Statistical and Smoothed Trajectories)
[16]	Holidays (unlabeled)	LSTM Enc-Dec
[4]	Unspecified (Unlabeled)	Enc-Dec (identify anomalous Days) and LSTM + K-means (identify anomalous hours)
[5]	Unspecified (Unlabeled) real, Unspacefied (Labeled) Synthetic	Standard Deviation + T-Student, K-NN + DTW (Discrete Time Warping)
[6]	Unspecified (Unlabeled)	CART and K-Means + GESD (generalized extreme studentized deviate), DBSCAN and ANN Ensambling
[32]	Unspecified (Unlabeled) real, Unspacefied autogenerated	Autoencoders
[13]	Unspecified (Unlabeled) but maybe reported	E-EMD (Ensamble Empirical Mode Decomposition) + IMF agg. (Intrinsic Mode Functions aggregation)
[17]	Unspecified (Unlabeled)	PARX (Periodic Auto-Regression with eXogenous variables) + Gaussian Statistical Distribution in Lambda Architecture
[18]	Different and some are described	kNN, SVM, and ANN compared + three-sigma-rule for Anomaly Detection
[10]	Unspecified (probably Labeled)	GAM (Generalized Additive Model) + ARMA (AutoRegressive Moving Average)
[11]	Unspecified (probably Labeled)	GAM (Generalized Additive Model) + ARCH (AutoRegressive Conditional Heteroskedasticity)
[1]	Unspecified (Unlabeled)	PCA + Q-Statistic
[19]	Various (Unlabeled)	Fuzzy-ADLD (Anomaly Detection and Linguistic Description)
[20]	Unspecified (maybe labeled, at least computable with 2-sigma)	K-Menas, ANN, ARIMA, NNAR (Hybrid ANN + ARIMA)
[21]	Injected anomalous consumption patterns	Unspecified but explained in the paper
[22]	Unspecified	Frequent Pattern Mining + Pattern-based Embedding + Isolation Forest
[7]	Unspecified (Labeled and Unlabeled)	PCA + Wavelet transform
[23]	Unspecified (Unlabeled)	Nearest neighbor clustering with euclidean distance + fuzzy rule set per cluster (description)
[8]	Unspecified (Unlabeled)	linear discriminant analysis (LDA) + boundary-based discriminative subspace identification method
[24]	Unspecified (Unlabeled)	neural network pre-processed by wavelet and fractal (NNPWF)

Table 1: Papers, anomaly type and algorithms

Paper Reference	Algorithm Type	Databases	Database Type	DB Granularity
[9]	Offline (Semi-Supervised)	Power Demand, Space Shuttle, ECG + Engine-P, Engine-NP	Research Building	15 min. (Power DB)
[12]	Offline	Power Demand, Space Shuttle, ECG + many other	Research Building	15 min. (Power DB)
[15]	Offline	Power Demand, Space Shuttle, ECG	Research Building	15 min. (Power DB)
[16]	Offline (Semi-Supervised)	Power Demand, Space Shuttle, ECG	Research Building	15 min. (Power DB)
[4]	Offline (Semi-Supervised)	Power Demand for every device in the houses (67)	Houses	60 min. + 24 hour
[5]	Offline	Power Demand, Power Demand synthetic	Houses	60 min.
[6]	Offline	Power Demand Building Energy Management System	Research Building	15 min.
[32]	Offline	Power Demand of HVAC system + Injected Anomalies	School Building	5 min.
[13]	Offline	Power Demand of HVAC and lightning systems	University Building	5 min.
[17]	Online (Supervised)	Power Demand	Houses	60 min.
[18]	Offline (Supervised)	Power Demand for 6 DB (KMH, KSH, TTH, WBW, Singel, G647) + Weather DB	University Building	5 min. (KMH, WBW, Singel), 15 min. (KSH, TTH), 60 min. (G647)
[10]	Offline	Power Demand per single circuit and weather informations	Office Building	15 min. but 60 min. period is used. There are missing values
[11]	Online	Power Demand per single circuit and weather informations	Office Building	15 min. but 60 min. period is used. There are missing values
[1]	Offline	Power Demand of HVAC system	Commercial Building	Unspecified
[19]	Online (NNC -> Fuzzy rules)	Power Demand	Office Building	15 min.
[20]	Online	Power Demand	Office Building	1 min.
[21]	Offline (Unspuervised)	Power Demand	Houses	15 min.
[22]	Offline	Many different	xxx	xxx
[7]	Offline (Supervised)	Air Handling Unit	Office Building	Unspecified
[23]	Online	Temperature and CO2 readings per room per floor	Office Building	Unspecified (maybe 15 min.)
[8]	Offline (Semi-Supervised)	AHU sensors	Office Building	60 min.
[24]	Offline	AHU sensors	Unspecified	Unspecified

Table 2: References, algorithm type, database, database type and granularity

Among those papers, the first four papers shared the same Dutch Power Demand dataset used in this work. The techniques presented by them are various and not only based on NN, like the LSTMs, but also statistical methods, like SAX. The other papers instead used different datasets, proprietary to the research facility responsible for the paper.

In all papers the authors tried to find traces of anomalous behavior inside time series describing the power consumption of office buildings, university buildings or houses through the sensors installed in the facilities. The main difference between these three types of settings is the shape of the input data. While offices and universities generally follow a weekly pattern composed of five days with high power consumption and then two days with a lower power consumption, the houses behave in the opposite way. The time series generated by a house power consumption tend to have five days of low demand during the day and high in the evening, and two days of high demand at all times since most people tend to be at work during the week.

Another major difference stands in the granularities of the points in the datasets used. The Dutch Power Demand dataset was used with the same granularity of this thesis (15 minutes). In general, the granularities used range from 15 minutes to 60 minutes per point. Occasionally some datasets have a very fine granularity of a point per minute, that is the case of the paper [20] which deals with real-time anomaly detection.

Another conceptual difference between the methods implemented is the idea of online/offline algorithm. Offline algorithms need to receive the entire input to be able to compute a solution, while online algorithms can instead analyze and output a solution when the input is given to the code piece-by-piece, without the need of collecting the whole data from the beginning. This difference is evident when applied to the anomaly detection setting: the offline methods need to receive batches of data to be able to detect whether an anomaly occurred, while online methods only need the last collected point to classify it. Most of the methods implement an offline algorithm since real-time anomaly detection tends to be more computationally eager. The online methods in Table 1 are mainly statistical methods like PARX, GAM, fuzzy systems with nearest neighbor clustering and k-means but also an Artificial Neural Network-ARIMA hybrid.

The anomalies are mostly derived from holidays rather than real malfunctions in the systems, others are not specified, and others are synthetically generated and added to a test set. However, it is not clear how these anomalies have been generated and to what days they have been added.

Chapter 3

Experimental Comparison of Anomaly Detection for Power Consumption

3.1 Datasets

In this section the datasets used are going to be presented, as well as the adjustments made while cleaning the data.

Two different datasets were used in this work, both representing the power consumption of a facility. The first one is a dataset of a research building located in the Netherlands, the second one is an Italian office building in Bergamo.

3.1.1 Dutch Facility

This is a very clean dataset from 1997, where most of the seemingly anomalous behaviors come from the low power consumption of holidays. It has a granularity of 15 minutes, which means 96 points per day, and it has been used by many papers mentioned before.

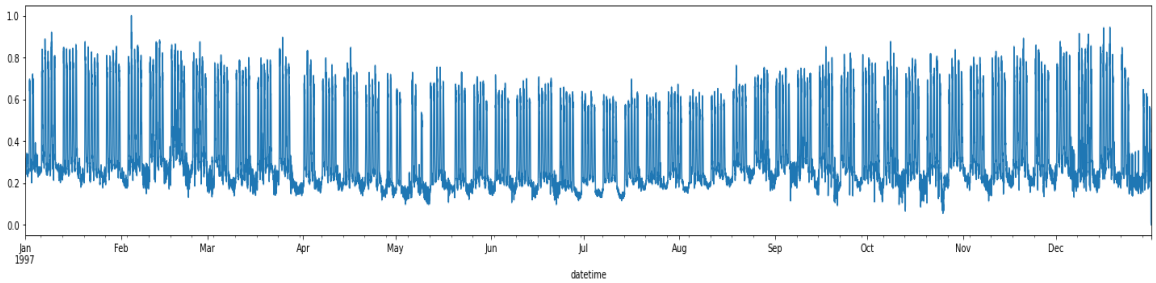

Figure 3: Complete dataset of Dutch facility.

By just looking at Figure 3 we can see the patterns given by seasonality, and a weekly behavior, where five days a week have a high-power consumption, while the last two days of the week have a low one. The dataset does not have any missing data, furthermore it also does not have any huge trend change so it is rather regular.

3.1.2 Italian Building

The second dataset taken into consideration is about the power consumption of an office building in Italy over a period of three years, with a granularity of 15 minutes. Associated with it there are other interesting features such as average temperatures and humidity but, most importantly, it came with power consumption per each floor. Every circuit breaker has a specification about its consumptions:

- First floor: it concerns energy consumption for underground parking lighting, electric vehicle charging stations, UPSs for IT, front desk, outdoor lighting, fire-fighting system, and mechanical workshop.

- Second floor: this floor aggregates the consumptions of lighting and power outlets of three office floors, auditorium, and cafeteria.

- Third floor: finally, this floor reports the power consumptions of the heating and cooling system, the ventilation system, elevators, kitchen, and cafeteria's refrigerators.

For this dataset there are no labeled anomalies. The analysis phase was performed in the same fashion as for the previous dataset but, since holidays are not anomalies, in this case, they are not highlighted.

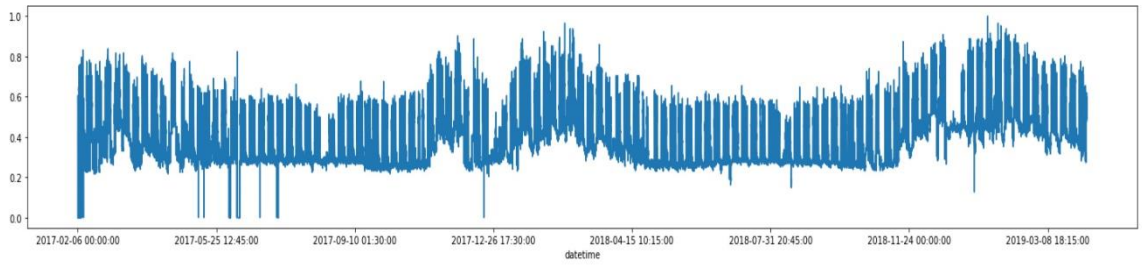

Figure 4: Complete dataset of second floor of Bergamo facility.

The first two floors are very similar in power, and they almost completely overlap. For this reason, the analysis will be carried on with the power consumption of the second floor, shown in Figure 4.

By just looking at the time series graph of the three floors it becomes obvious that these time series are less predictable than the first dataset, especially the third floor. Furthermore, there are far less prominent outliers.

3.2 Experimental Setting and Exceptions

Both datasets have been used for the implementation of the methods and both had to be regularized. The Dutch one was complete and did not have any missing values. In the Italian one instead, the data is not available for certain periods, there are missing points and many unrealistic values probably caused by faulty sensors like temperatures under 0K. Two options were available at this point:

-Try to fill the missing data by interpolation / copy data from the previous week.

- Try to select time windows that only contain the most regular data to feed the methods implemented.

The first option while it may seem more correct, it does bring in a few problems. First, there are multiple sequential points missing, sometimes entire days. Obviously, an interpolation does need a starting point and a finishing one but with this much missing data, the good theoretical effect of interpolation would be nullified.

Secondly, by copying the data from the previous week we might be adding fictitious anomalies, which would increase the bias and variance of the model trained on such data.

Moreover, the experimental setting had to be modified according to the needs of the specific methods. Sometimes it was just not possible to achieve good results in one method by just feeding it the same dataset of another one because of missing points or incomplete sequences. It is the case of [9] where an even more scarce version of the dataset had to be used. In this method there was the need to have big sliding windows, therefore all the data fed to the LSTM had to be sequential and without holes, something that other methods did not need.

It did seem therefore better to the author to simply use the original data by shattering it to smaller windows and manually selecting those that seemed to have the most regular features and were anomaly-free.

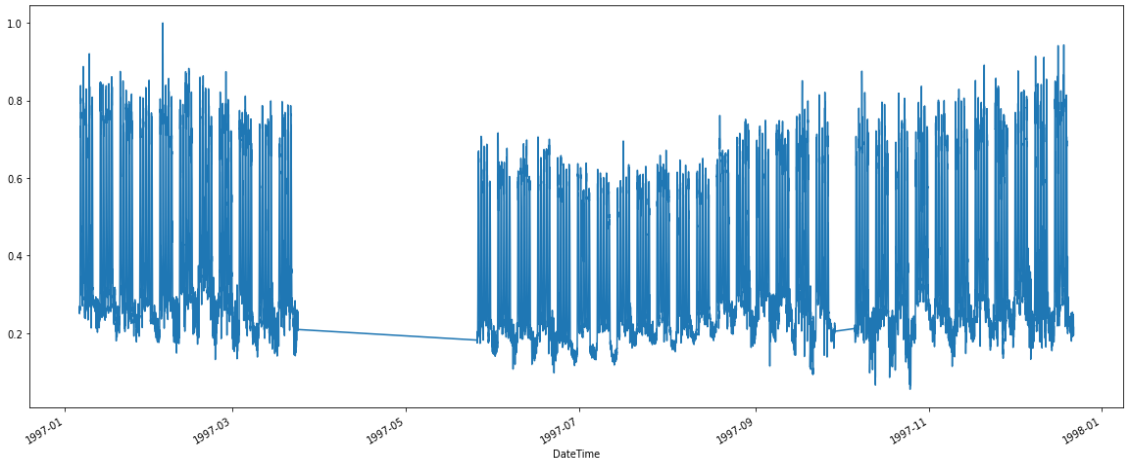

Figure 5: Cut version of the Dutch training set

In Figure 5 we can see the way the Dutch dataset has been cut. A portion of it has been removed to be used for testing since it contained most of the holidays.

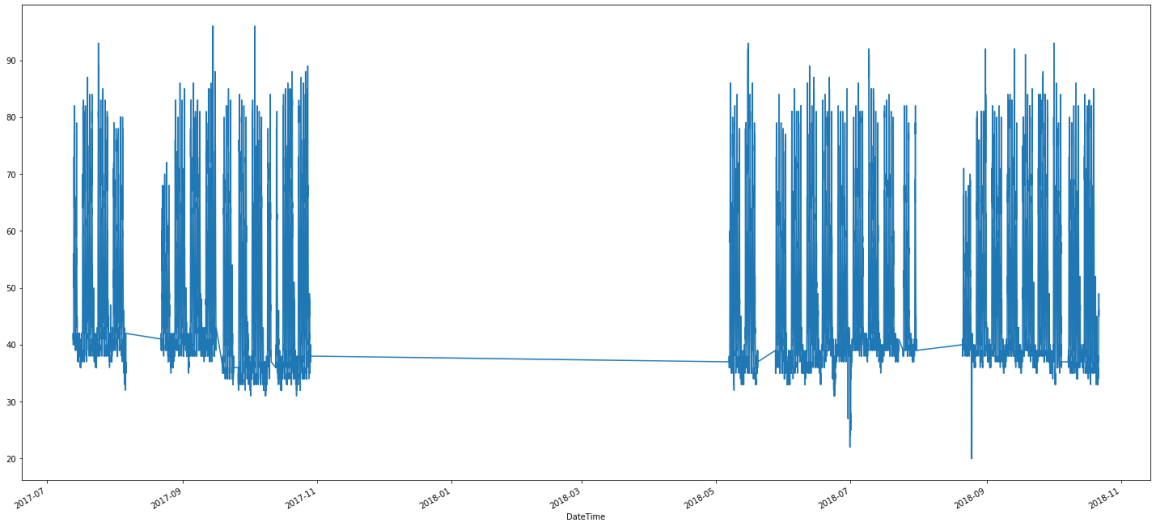

Figure 6: Cut version of the Italian training set

The Italian dataset had to be trimmed even further considering the amount of missing data as seen in Figure 6.

As for the test sets, three time slots have been selected, 1 from the Dutch dataset and 2 from the Italian dataset. Both datasets were normalized using min-max normalization.

The first, the Dutch one in Figure 7, shows a period with both normal and anomalous days. The anomalous days in this set are 4:

-30th of April – Queen's Birthday

-5th of May – Liberation Day

-8th of May – Ascension Day

-19th of May – Whit Monday

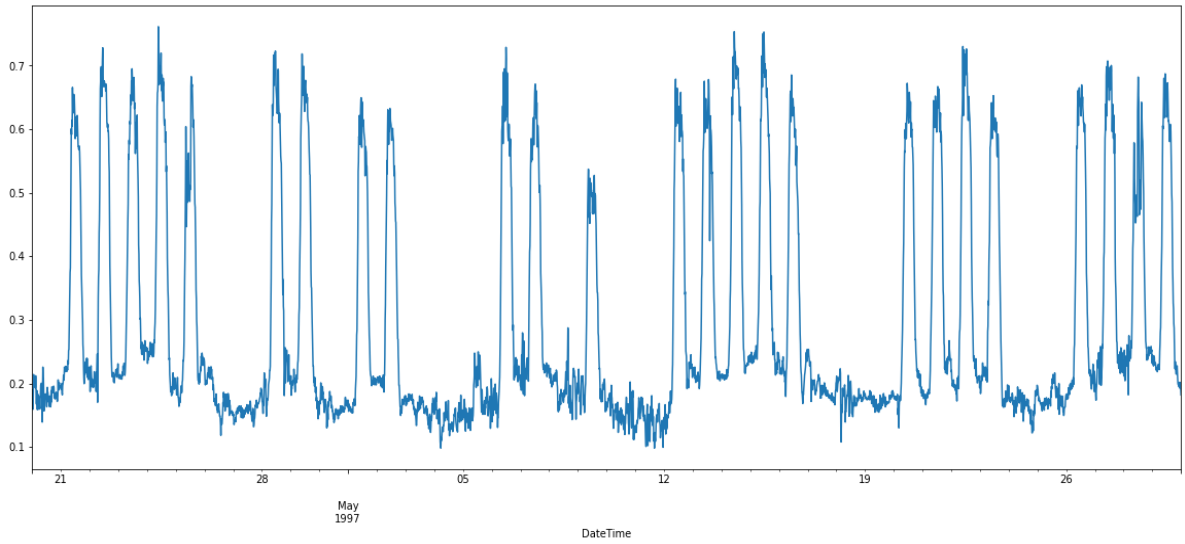

Figure 7: Dutch test set

The first Italian test set in Figure 8 includes both real and artificial anomalies for a total of 6 anomalous days:

-8th of August – Artificial Anomaly

-10th of August – Artificial Anomaly

-14th of August – Gap Day because of Holiday

-15th of August – Feast of the Assumption

-16th of August – Gap Day because of Holiday

-17th of August – Gap Day because of Holiday

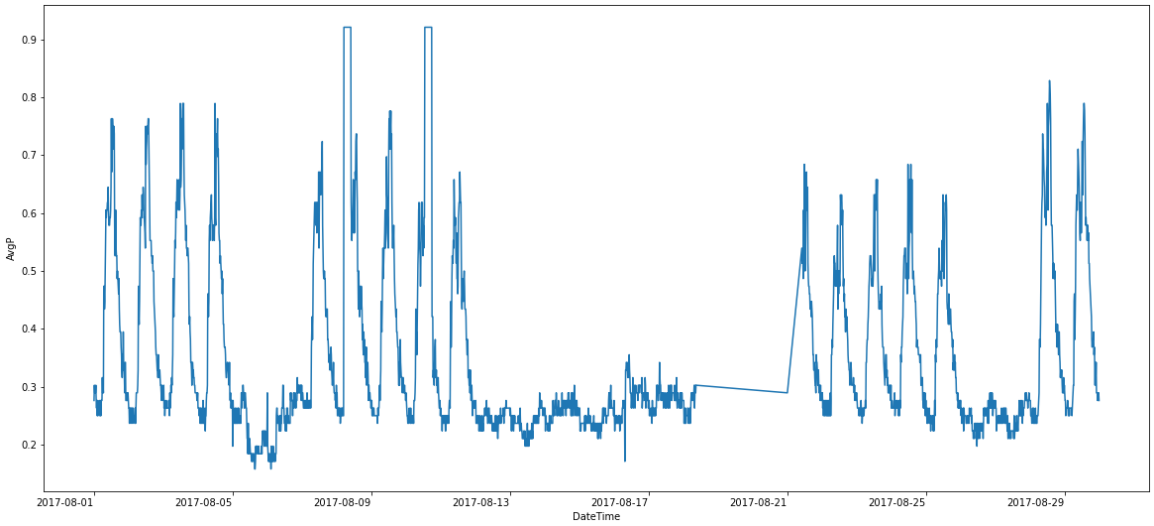

Figure 8: First Italian test set

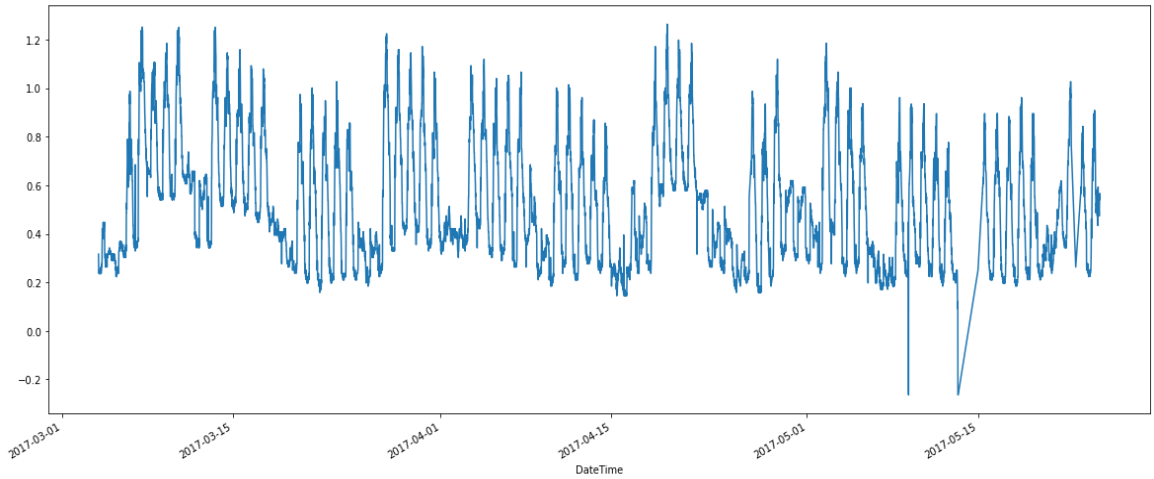

Figure 9: Second Italian test set

The second Italian test set in Figure 9 is instead a longer collection of days. Here the days that have extremely high peaks or that have an unbalanced behavior compared to the Italian training set in Figure 6 are considered anomalous.

The second Italian dataset has only been used with the two autoencoders. That is because it is going to be part of the focus of Chapter 4 and the transfer learning technique.

The jumps with missing data are given by the fact that the data has not been interpolated for the aforementioned reasons.

The data has then undergone a pre-processing phase that added, modified, and extended the features.

- Normalization: the data has been rescaled using the min-max normalization so to have values always in the range 0-1.
- Feature expansion: 8 new features w.r.t. the original dataset have been introduced to help the NN understand better the context of each value of the power consumption. Such features have been selected from the DateTime feature. The first feature to be introduced is the hour of the day normalized on a scale of 0-1 which was done to maintain the whole output in the same range as the power consumption. The other 7 features are a one-hot encoding of the days of the week, such that only one day per row contains a 1 and all the others a 0.

Another one-hot encoding of the months was also initially added but given the noisy nature of the Italian dataset, it did not yield any improvement, so it was later excluded from the model.

- Sequence creation: training and test sequences are then computed.

In one paper [26], a time window of one day or 96 points has been used, together with a stride of 96 points. Basically, the autoencoder is fed with single days and is then able to classify an entire day as correct or as an anomaly.

In paper [14], a sliding window of one day with a stride of one point (15 minutes) has been used. While in the beginning all methods were supposed to be implemented with a one point stride, the experiments performed with the NN presented in [26] showed that it was not able to recognize any of the anomalies in the test sets presented. It is very likely that this is due to the noisiness of the dataset. Initially a sliding window of one week was also tried and even though it did start showing some promising results, it was extremely inefficient, and the idea was abandoned shortly after, towards a more efficient solution of using a single day. It is not possible to have a perfectly uniform experimental setting since it might not fit the needs of one or more specific methods. However, this condition is not invalidating for the purpose of evaluating different methods. It must just be noted that different methods learn differently under different conditions and might not perform as well as expected if not implemented correctly.

Paper [4] was only able to recognize anomalously high peaks because its input contained 23 hours previous to the prediction but was not able to distinguish between working days and holidays. Therefore, the input was modified by feeding the method 7 days and predicting the 8^{th} day.

Paper [9] used a similar approach, but with a longer time window consisting of 14 days and predicting the 15th day.

Paper [6] used the same principle with 7 days of input and an output consisting of the 8th day.

In Figure 10 we can see a table containing 96 points, collected every 15 minutes for a temporal range of 24 hours.

- AvgP: The power dissipated by the building (Second floor)

- hour: A variable containing a normalized version of the hour of the day

- Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday: Columns containing the one hot encoding of the week

- anomaly: tells whether that point is to be considered an anomaly

DateTime	AvgP	hour	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	anomaly
2017-08-01 00:00:00	41	0.000000	0	1	0	0	0	0	0	0
2017-08-01 00:15:00	43	0.000000	0	1	0	0	0	0	0	0
2017-08-01 00:30:00	42	0.000000	0	1	0	0	0	0	0	0
2017-08-01 00:45:00	42	0.000000	0	1	0	0	0	0	0	0
2017-08-01 01:00:00	42	0.041667	0	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
2017-08-01 22:45:00	42	0.916667	0	1	0	0	0	0	0	0
2017-08-01 23:00:00	41	0.958333	0	1	0	0	0	0	0	0
2017-08-01 23:15:00	41	0.958333	0	1	0	0	0	0	0	0
2017-08-01 23:30:00	42	0.958333	0	1	0	0	0	0	0	0
2017-08-01 23:45:00	41	0.958333	0	1	0	0	0	0	0	0

Figure 10: Dataframe of Italian building

3.3 Methods

Firstly, three methods with some simple implementations of NNs have been considered.

3.3.1 LSTM 1

In [4] two ideas are built:

- hybrid approach with k-means-LSTM

-the use of an autoencoder

This thesis focuses only on the LSTM part, so to compare the goodness of this method with the others.

That is an LSTM with weekly inputs: instead of predicting the consumption of the following hour like specified in [4], this network wants to predict the consumptions of the following day starting from an input composed of the past seven days, so to be able to identify the weekly trends.

3.3.2 MLP

[6] trained 10 different multiple layer perceptrons to perform Ensemble + ANN. Both with the Netherlands dataset and the Italian (completely nominal training set) decent results were achieved and the prediction of the test set was mostly done

correctly. This was achieved with different granularities of 12 points per day (one point every two hours) and 96 points per day (one point every 15 minutes).

3.3.3 LSTM 2

In [9] a LSTM is trained to predict the same time series but moved of one day forward, and to use the predicted sequence and the real sequence to create a residual vector. These vectors are then used to calculate the average and covariance to study the Gaussian distribution of non-anomalous data. Then a threshold is found by separating the anomalous data from the normal one. Furthermore, it is proposed to use the L^p norm instead of the likelihood as the evaluations were obtaining better results. The training set only contains nominal data, and the test set contains both nominal and anomalous data.

But this also contains a major flaw. When giving the NN a sequence of weekly values, if the sequence is not anomalous it is not a problem since the following day will be predicted correctly. That is the case of for instance L L H H H H H, where the first two Lows represent the weekend and the Highs the weekdays. But as soon as an anomalous week is given as input like L L L H H H H it is not possible to distinguish anymore the following day. For instance, the L L L sequence could be interpreted as a Friday (holiday), Saturday, Sunday or as Saturday, Sunday, Monday (holiday).

The hyperparameters used have been found using the TPE (Tree of Parzen Estimators). The network is composed of stacked LSTM and between each level of the LSTM there is a fully connected layer with an activation function (swish).

Analyzing the residual distribution with statistical tests like Henze-Zirkler, Kolmogorov-Smirnov and D'Agostino-Pearson it was found that the distribution is not Gaussian.

3.3.4 Offline Autoencoder

In [26] the authors want to create a neural network, more specifically an autoencoder able to predict anomalies in building power consumption time series. The hardest part is always to do it knowing that there are no labels to indicate the correctness of the prediction made so other methods must be employed.

They found that the main limitations of existing unsupervised anomaly detection methods are:

1. The anomaly detection performance and computational efficiency can be degraded dramatically when applying to big data. Conventional methods like motif discovery are based on exhaustive search, which results in the computational costs increasing dramatically for long time series and is therefore not applicable. For instance, statistical methods are not scalable to large-scale data, and they are subject to stringent mathematic assumptions,

like normality of the data or independence, which may be adapted but not fulfilled by real-world high-dimensional data. Some unsupervised data mining techniques have been used to enhance the effectiveness and efficiency in analyzing big data. Nevertheless, the associated post-mining workload can be overwhelming [27].

2. The performance of existing unsupervised methods relies heavily on features used. Currently, features for anomaly detection are selected or constructed based on domain expertise or simple statistics (e.g., the mean and standard deviation of a numeric variable). There is a lack of data-driven methods to automate the feature generation process for generalization purposes. More advanced methods are desired to enhance the performance and applicability of unsupervised anomaly detection in the building field. One promising solution to these limitations is the autoencoder. An autoencoder adopts the neural network architecture to perform unsupervised learning, where the model input and output are set identical. The rapid development in the deep learning community has provided various techniques for analyzing different types of data (e.g., cross sectional or temporal data) and training models with advanced architectures (e.g., deep convolutional autoencoders) [28].

An autoencoder, as explained before, consists of an encoder and a decoder, the encoder transforms the input data into high-level features and the decoder tries to reconstruct the input starting from the high-level features.

Two types of possibilities were analyzed by giving the raw information or by adding extra information by using a vector of one hot encoding highlighting the month.

In this thesis a similar approach was implemented but adding information about the months did not bring any improvement. What improved the results was adding information about the time of the day and the day of the week (one hot encoding). The most obvious enhancement came thanks to the idea of feeding the NN sequences starting each day at 00:00 and ending at 23:45. This allowed the net to correctly classify anomalous days on the most basic test set. This means that the vector containing the label "anomaly" does not refer to the single data point but rather to an entire day. It surely might seem that the autoencoder is not able to correctly classify the single point anomalies, but it must be taken into consideration that a building manager would overview the whole process when actions to fix power consumption problems must be taken.

3.3.5 Support Vector Regression

In [14] an ensemble method is proposed. In this thesis it was instead decided to break apart the pieces composing the ensemble and evaluate them singularly to better understand the difference between autoencoders and more statistical/ensemble-based methods. Also, the optimization method to find the optimal threshold was taken from this paper using the roc curve and the anomaly classifier presented in it to calculate the vector of anomalies for each threshold. That is done by finding the point with minimum distance from the point (0,1) of the roc curve, that is the value that should optimize the true positives and the false positives. However, this method, as will be shown later in this research, did not obtain similar results to the claims made in the paper.

3.3.6 Random Forest

The second method described in [14] is the Random Forest. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time.

3.3.7 Online Autoencoder

The third method presented in [14] is an autoencoder, and the results seem promising as well. With the MAE it seems to recognize at least partially the days with an anomaly, but also adding anomalies for a period of about a third of a day right before each anomalous day. That is because the time window of each slice is 24 hours or 96 points. When an anomaly starts approaching the sliding window, the MAE/MSE of that window starts rising, but the point connected to the window is the first, therefore the anomalous points are shifted by about half a day.

3.4 Results

In this section the results obtained are going to be presented. Four indices, that have been introduced in Subsection 2.1.4, were taken in consideration when trying to evaluate the models:

- Precision
- Recall
- F1 score
- Area under the Curve

3.4.1 Results LSTM 1

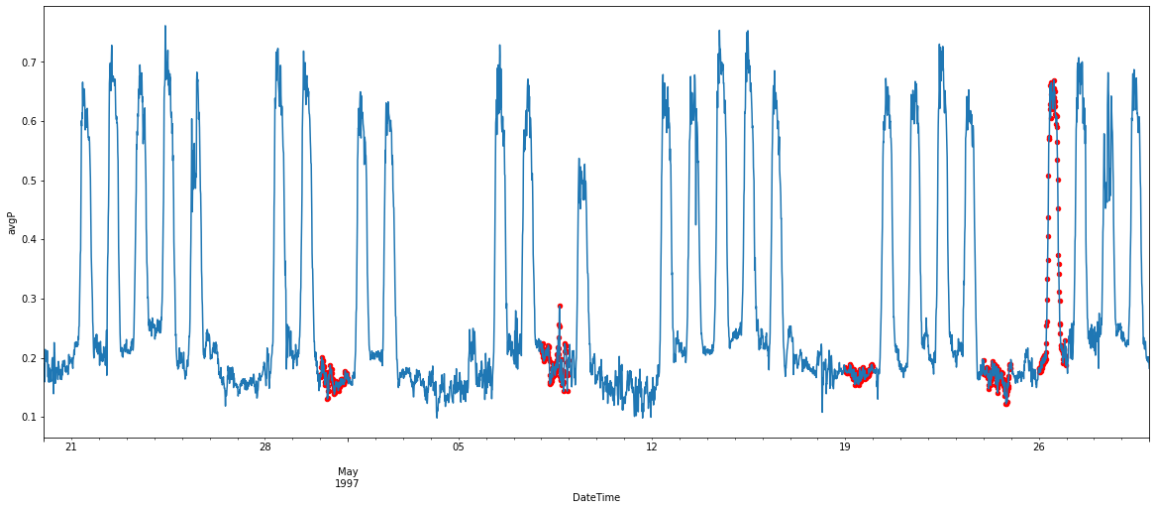

Figure 11: Prediction for Dutch test set with LSTM 1

As it can be seen in Figure 11 the model managed to detect some of the anomalous days in the Dutch dataset. The first anomalous day was correctly classified (1st of May), but in the third week we can see that it missed the low Monday (5th of May), correctly detected the anomalous Thursday (8th of May). As for the fifth week, the Monday (19th of May) was correctly considered anomalous but gave two false positives on the 24th and 26th.

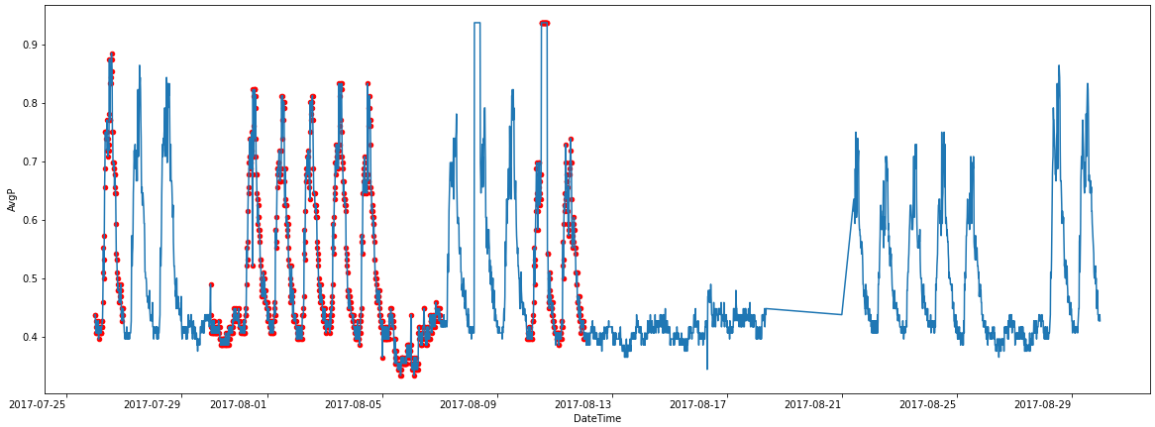

Figure 12: Prediction for Italian test set with LSTM 1

In the Italian dataset results instead, most of the days are incorrectly labelled (Figure 12). The first week is considered completely anomalous, and only the second artificial anomaly has been categorized as such out of all the anomalies presented in the Section 3.2.

3.4.2 Results MLP

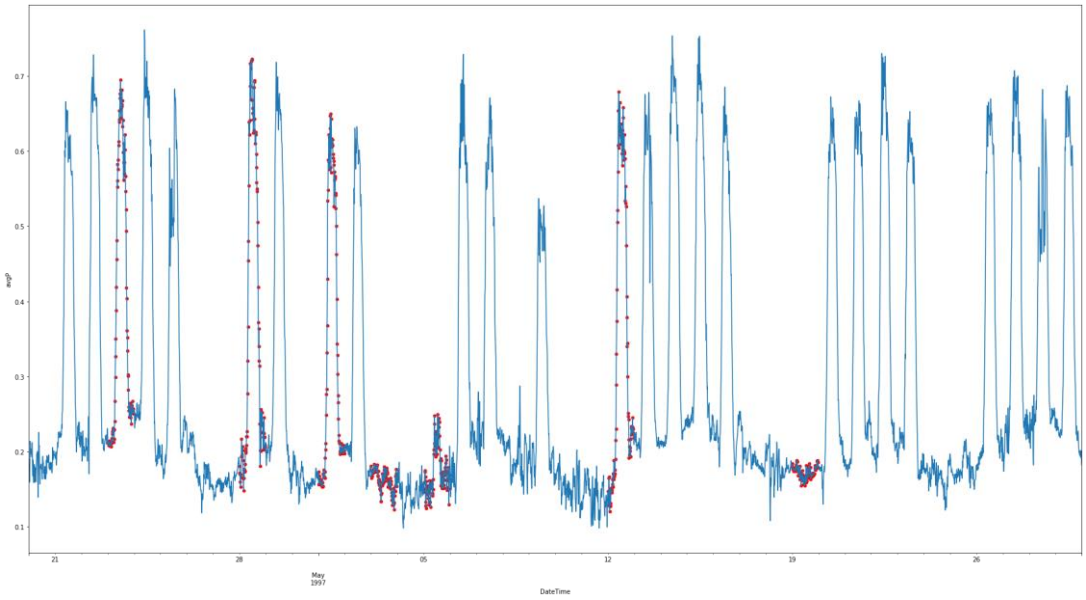

Figure 13: Prediction for Dutch test set with MLP

The Dutch test set had many normal days labelled as anomalous which lowered the precision in the final metrics (Figure 13). The anomalies that were correctly classified were on the 5th, 8th, and 19th of May.

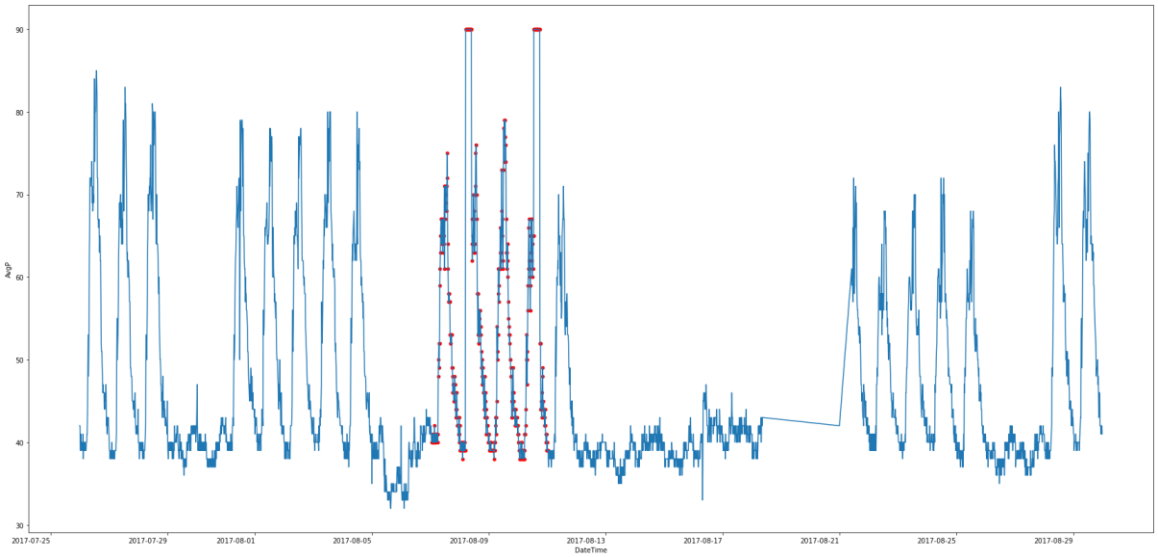

Figure 14: Prediction for Italian test set with MLP

As for the Italian test set (Figure 14), only the two artificial anomalies were correctly put into the anomalous category, together with the false positives in the same week. The anomalous days from the 14th to the 17th have been categorized as normal while obviously being erroneous because of too many low days in a row.

In both datasets some anomalous days were highlighted but there is a tendency to also label normal days incorrectly.

3.4.3 Results LSTM 2

The second LSTM performed worse than the first one. The training fit was accurate in both datasets, but this did not show in the test sets. It is therefore very likely that the model ended up overfitting. In Figure 15 it is shown that many days were incorrectly classified as anomalous, almost as if the network couldn't recognize the weekly pattern anymore.

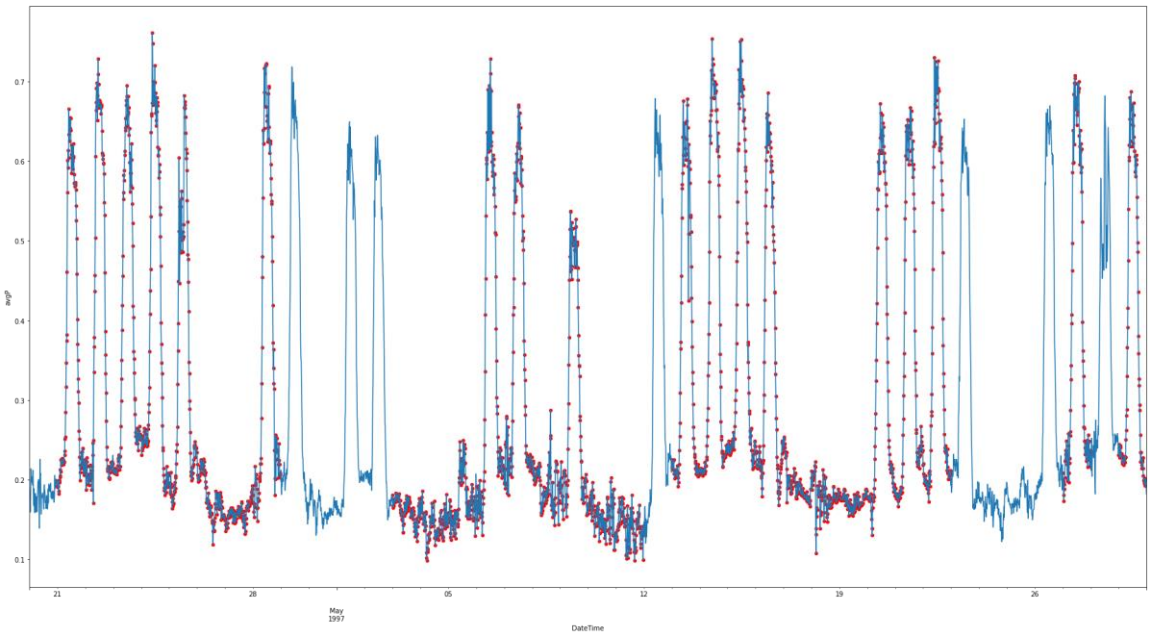

Figure 15: Prediction for Dutch test set with LSTM 2

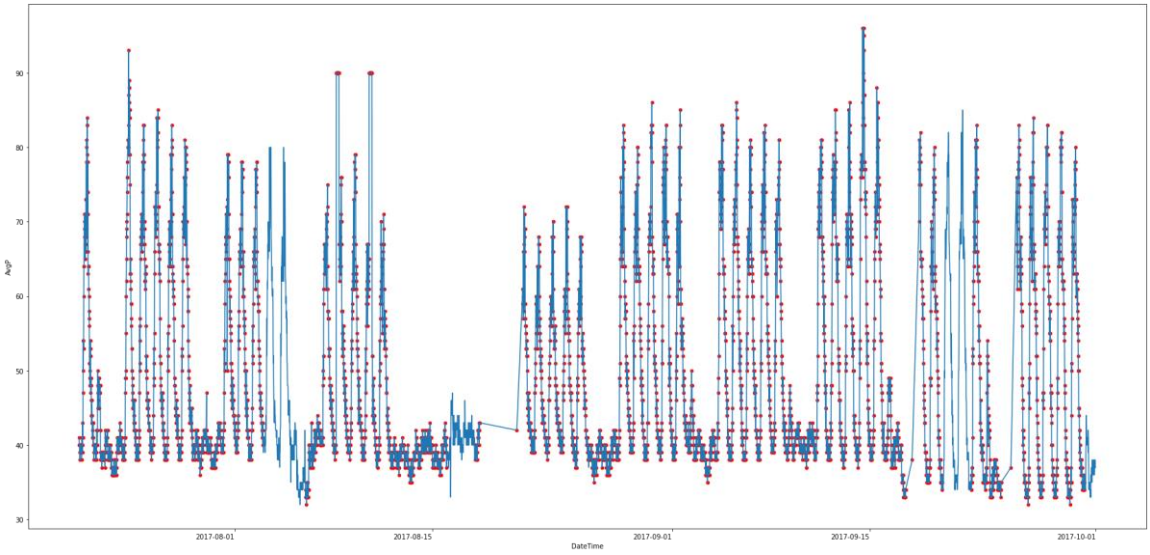

Figure 16: Prediction for Italian test set with LSTM 2

Both Figure 15 and Figure 16 show that many false positives were found. The Italian dataset basically only contains false positives, which explains the 0.01 obtained in the precision metric.

3.4.4 Results Offline Autoencoder

The first model was fed with the Dutch facility data, the cleaner of the two datasets available, and the results seem promising. All the anomalies were found, and they were all days with anomalous low power consumption (Figure 17).

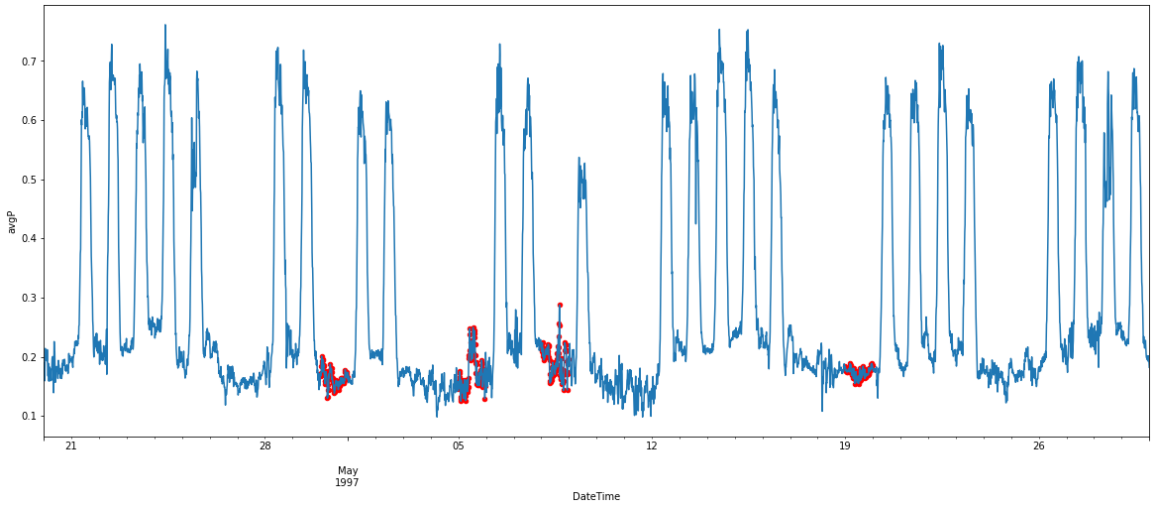

Figure 17: Prediction for Dutch test set with offline AE

Then the Italian dataset was given as input to the autoencoder for training and testing.

From the artificial peaks inserted on the 8th and 10th day of the test set we can realize how the same type of anomaly can have different impact on this solution. Since the NN learned a representation for each day of the week, there might be days with significantly more noise than others, and the prediction for those days becomes less reliable (Figure 18).

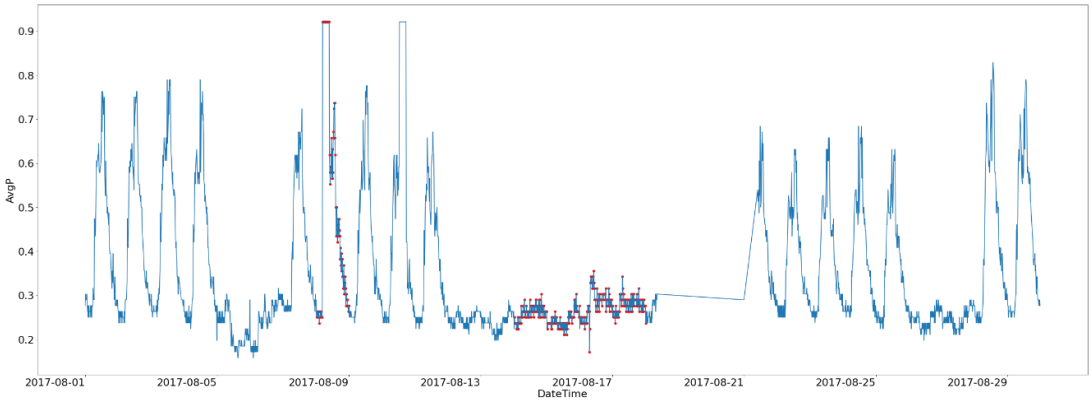

Figure 18: Prediction for Italian test set with offline AE

In Figure 19 we can see that the application of the same model to the winter part of the Italian test set considered anomalous almost all days, but mainly because the test set included extremely unregular days since the winter season was noticeably different from the training set that included only nominal summer data. Therefore 38 days with anomalies were found in this run, although it must be noticed that this value ranged between 35 and 53 in different runs. This show also an extremely unstable behavior probably induced from a training set that was filtered excessively and the neural network is now not able to recognize similar patterns but with different offsets. Days with similar consumption are classified in different manners, showing an extreme sensitivity to small differences.

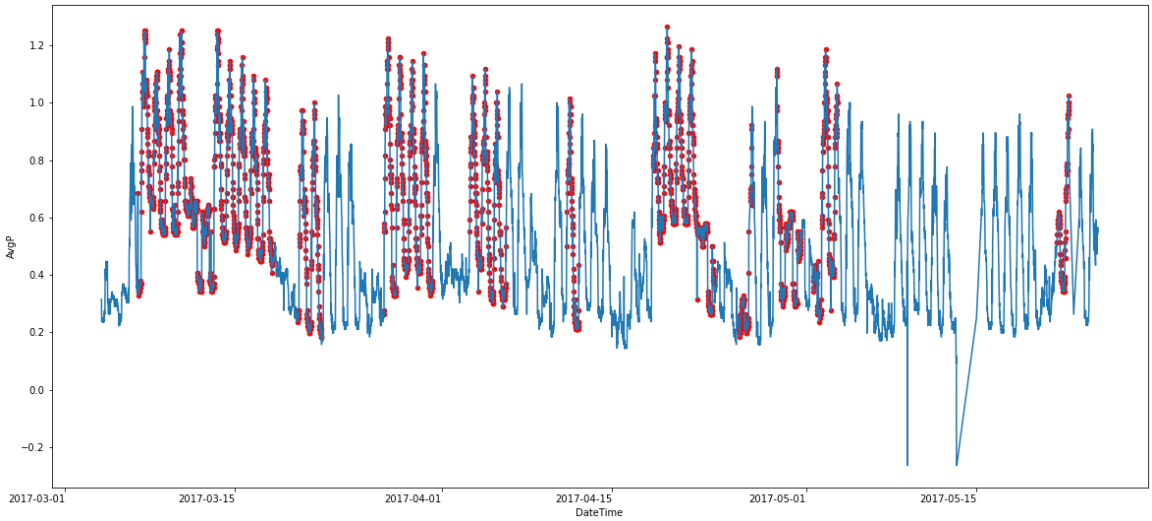

Figure 19: Prediction for Italian winter test set with offline AE

3.4.5 Results Support Vector Regression

The results obtained in the prediction were good enough for both the Dutch (Figure 20) and the Italian (Figure 21) test sets.

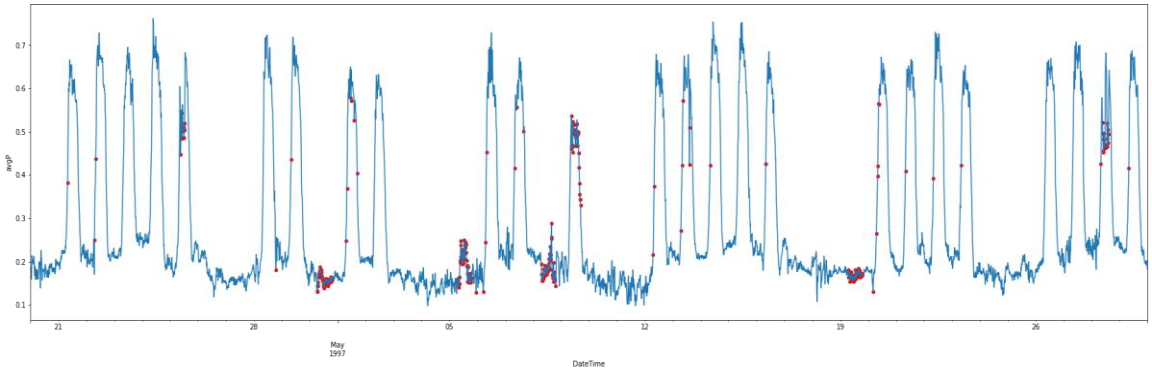

Figure 20: Prediction for Dutch test set with SVR

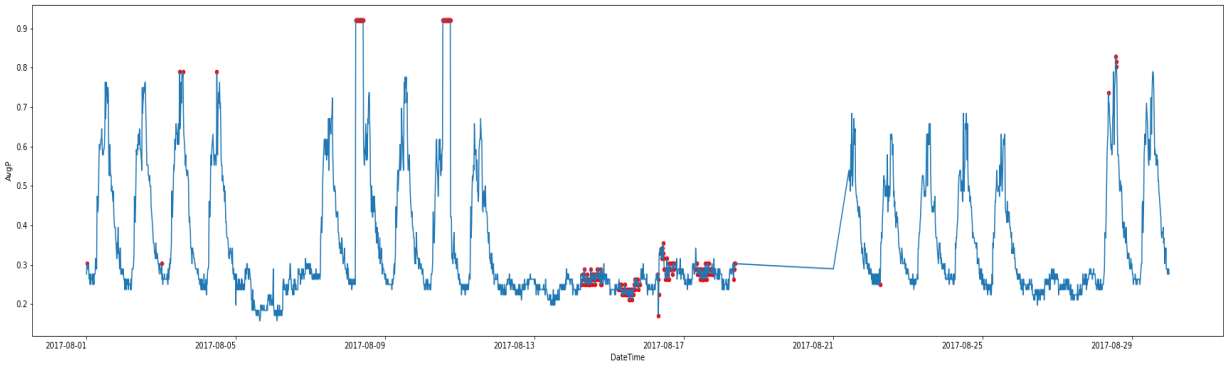

Figure 21: Prediction for Italian test set with SVR

It is easy to notice that the predictions about the anomaly of a day were mostly correct in both datasets. Some false positives were found, especially in the Dutch dataset.

3.4.6 Results Random Forest

Just like the SVR, it managed to capture most of the anomalies correctly in both Dutch (Figure 22) and Italian (Figure 23) datasets, but just like the SVR retained a few false positives.

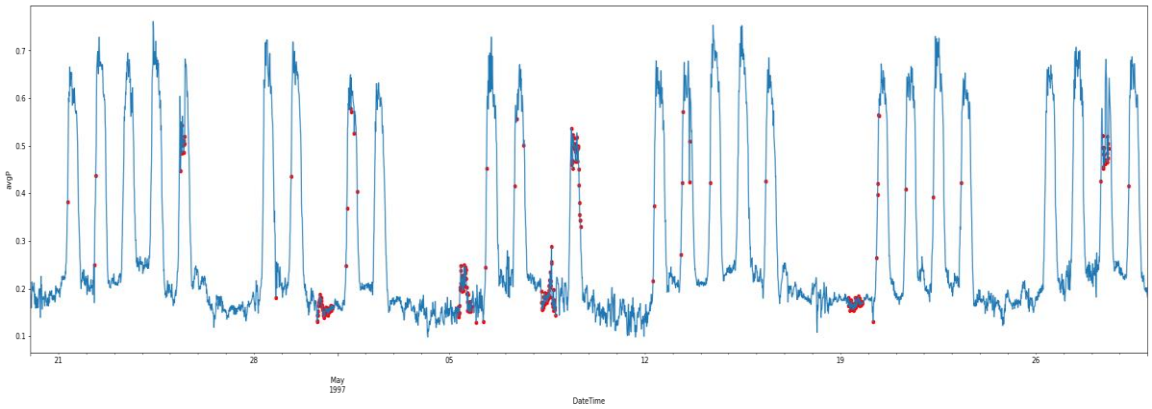

Figure 22: Prediction for Dutch test set with RF

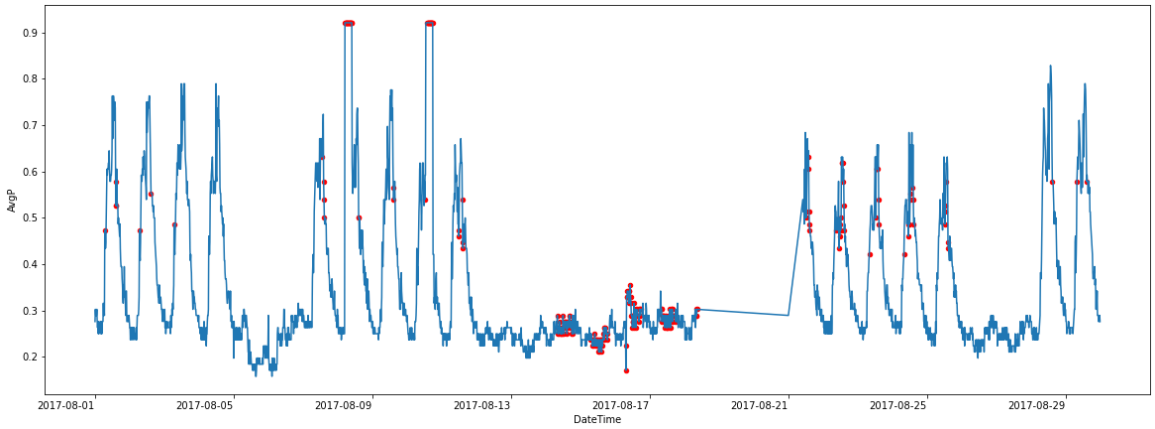

Figure 23: Prediction for Italian test set with RF

But it also managed, contrary to the SVR, to obtain a prediction much closer to the real values. That can be observed when comparing the graph of mean squared errors of both methods.

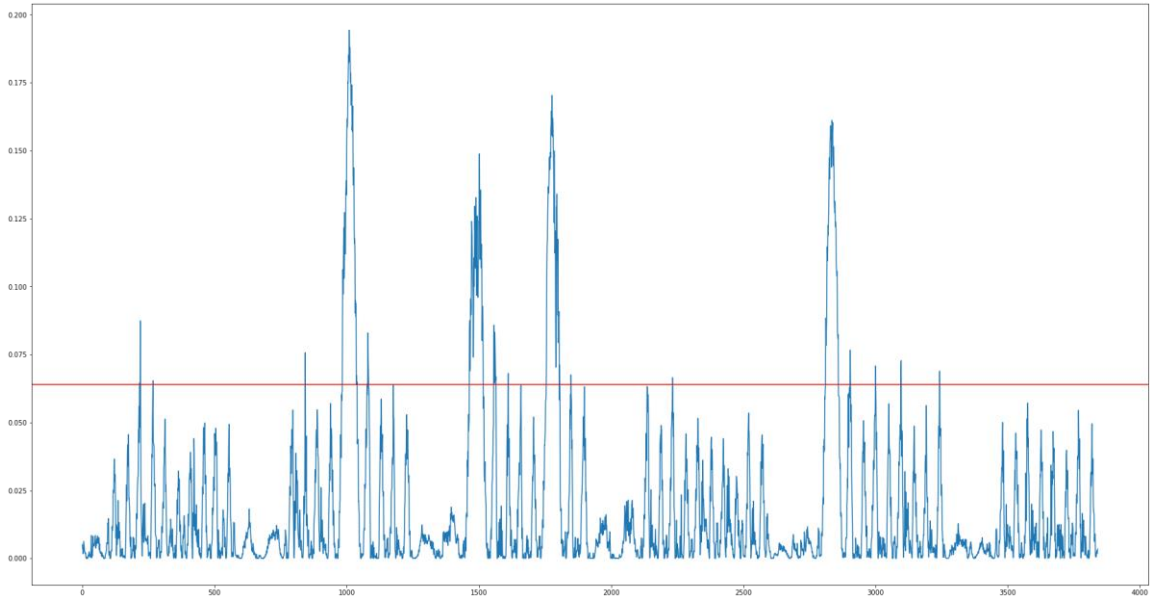

Figure 24: MSE of Dutch test set predicted by SVR

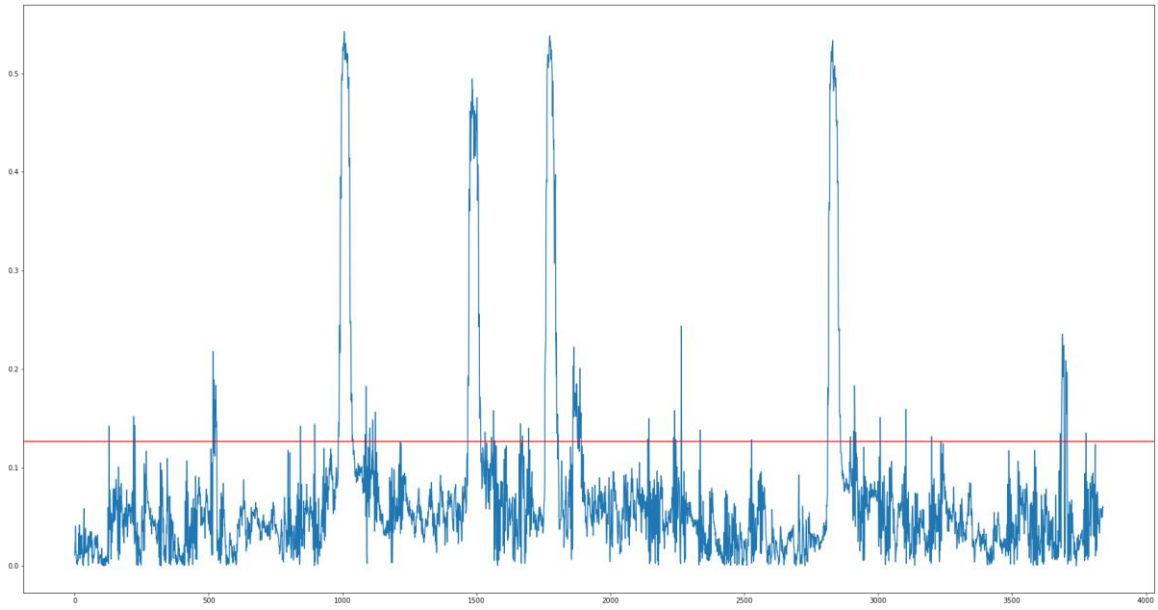

Figure 25: MSE of Dutch test set predicted by RF

The first graph in Figure 24 shows a much noisier prediction, and the anomalies are barely recognized from the threshold found (red line), while Figure 25, shows

a much higher ratio between the MSE of anomalous points and the MSE of nominal ones in the RF prediction. And the same applies to the Italian dataset MSE shown in Figure 26 and Figure 27 .

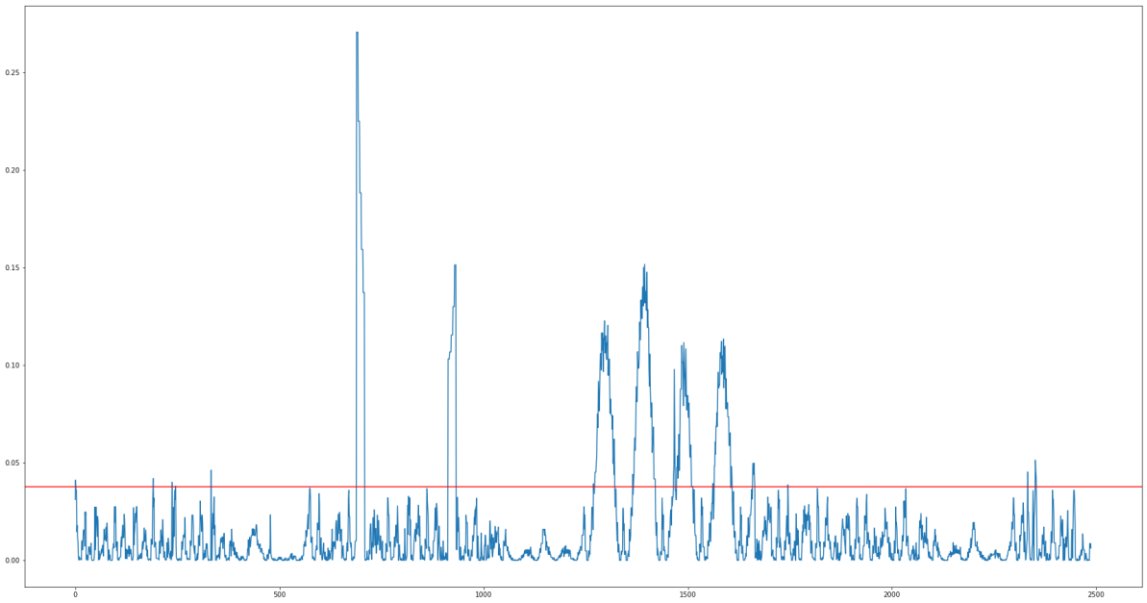

Figure 26: MSE of Italian test set predicted by SVR

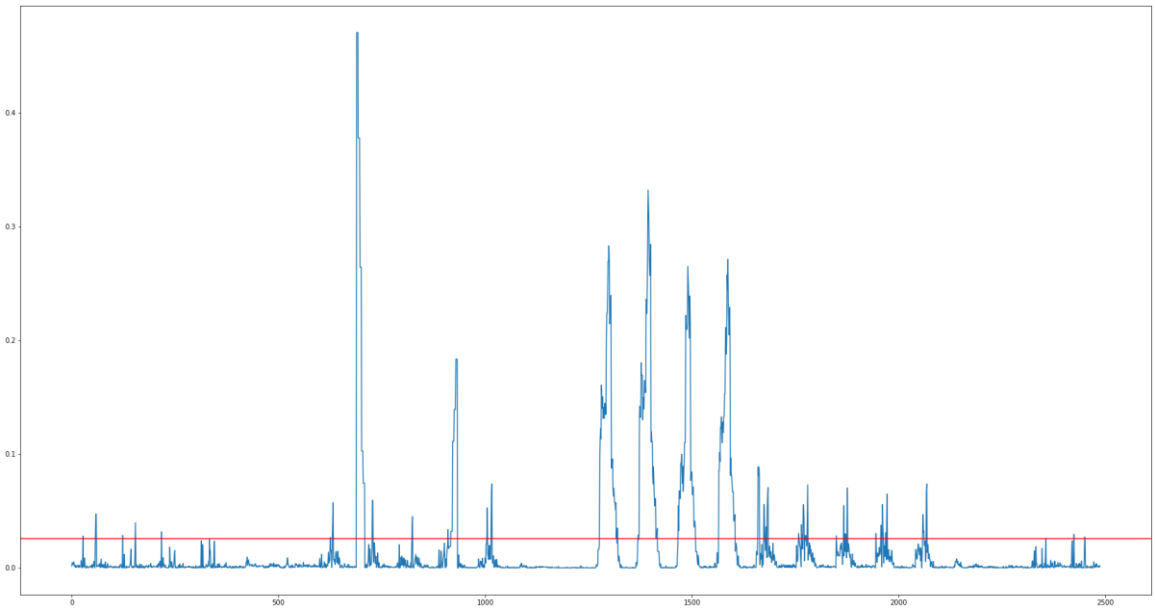

Figure 27: MSE of Italian test set predicted by RF

3.4.7 Results Online Autoencoder

Different sizes of the time windows were tested while implementing this autoencoder. The 40 and 20 points sliding windows did obtain slightly better results, increasing the precision but at the cost of decreasing the recall. In Figure 28 we can see that, at least partially, all the anomalous days were highlighted.

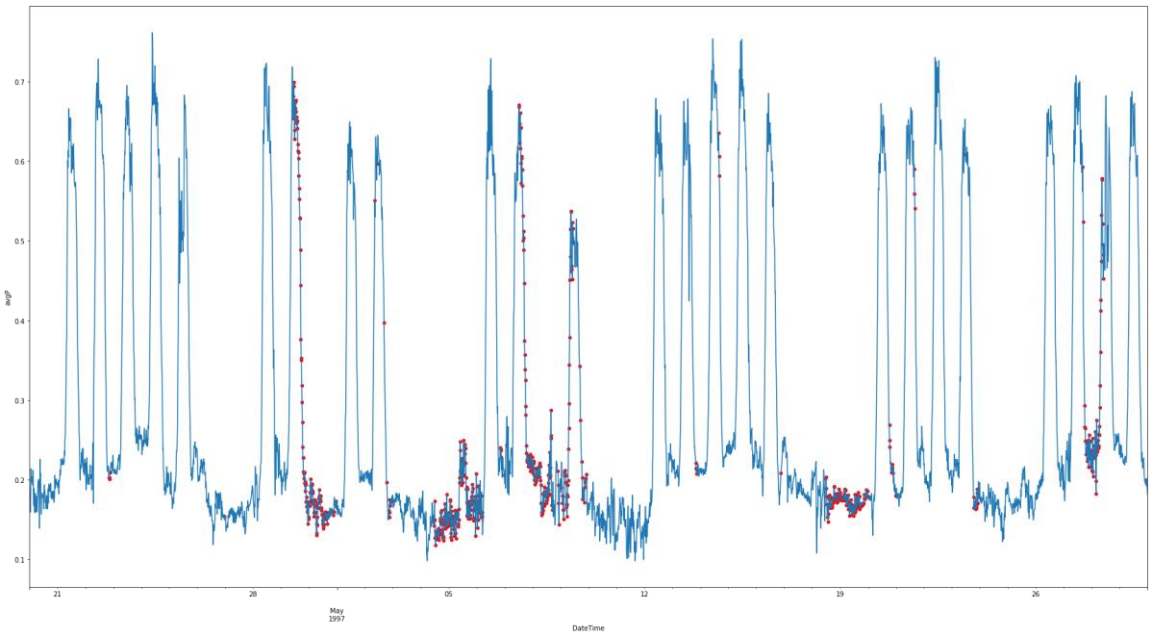

Figure 28: Prediction for Dutch test set with online AE

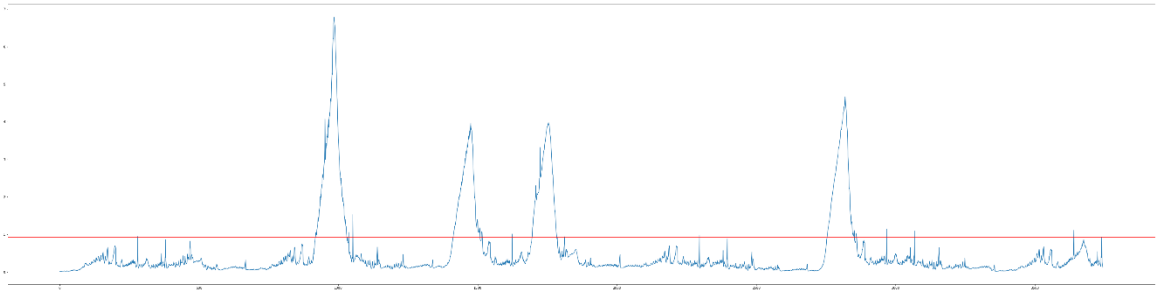

Figure 29: MSE of Dutch test set predicted by online AE

A good fit was found here. A very clean MSE graph in Figure 29 comes out with little noise and a high ratio between the height of anomalies and the nominal points.

The autoencoder had as input all the information about the day, hour, and power and by compressing and reconstructing the input it managed to obtain a good representation of each time window. Similarly, when applied to the Italian dataset, the autoencoder manages to show a decent number of anomalies as shown in Figure 30 while still presenting some false positives.

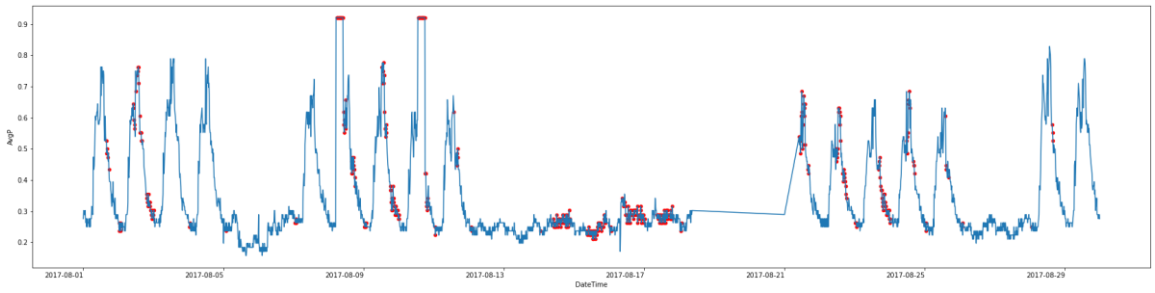

Figure 30: Prediction for Italian test set with online AE

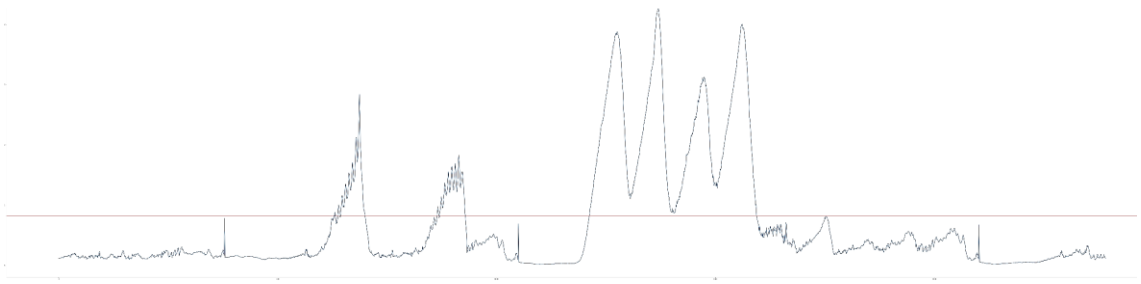

Figure 31: MSE of Italian test set predicted by online AE

In Figure 31 we can see that the MSE graph of the Italian dataset is also extremely clean with high peaks where the anomalies have been identified.

As for the winter dataset in Figure 32, many days were highlighted because of the difference in the power consumption offset with the summer dataset used for training, therefore labeling them as anomalous.

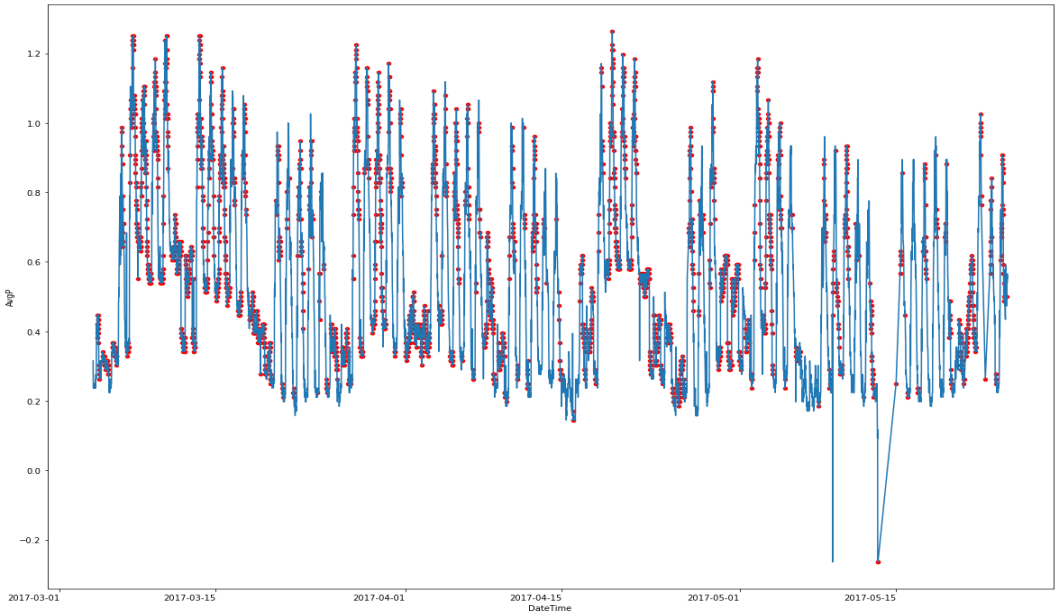

Figure 32: Prediction for Italian winter test set with online AE

3.4.8 Metrics Analysis

In Table 3 we have the results concerning the Dutch dataset, the most regular of the two.

DUTCH DS	Prec	Recall	F1	AUC
Offline AE	1	0,75	0,85	0,91
SVR	0,77	0,98	0,86	0,99
RF	0,60	0,97	0,75	0,99
Online AE	0,28	0,94	0,44	0,95
MLP	0,44	1	0,61	0,85
LSTM 1	0,2	1	0,33	0,89
LSTM 2	0,33	0,04	0,07	0,47

Table 3: Performance metrics of Dutch dataset

The results, when analyzed, do not seem to show the complete win of a method over the others. Surely the "offline" autoencoder managed to get the highest score of all but it must be recalled that it also had not the single points to check for anomalies, but the entire day was classified as anomalous. With a precision of 1, it did not obtain any false positives, but the 0.75 of recall means that there were some false negatives.

There seems to be a good fit with support vector regression and random forest. They have achieved 77% and 60% in precision, respectively, and an almost perfect recall, showing the sturdiness of the statistical methods.

As for the online autoencoder, while it did show a good recall, the same cannot be said for the precision. With a 0.28 it found several false positives decreasing a lot the precision. That seems to be because the anomalies are all shifted by about half a day and creates a lot of false positives. Aside from that, few false negatives were found as the recall scored a 0.94.

The MLP or multilayer perceptron ensemble, did score higher than the online autoencoder, with a 0.44 of precision and a perfect recall. But similarly, to the offline autoencoder, it did have the advantage of classifying only days and not the single data points. It therefore means that it still obtained too many false positives.

The LSTM 1 is the one that performed the worse in precision. While it managed to get a good fit to the training set, the test did not seem to show the same pattern and while it did not classify any false negatives, lots of false positives were found.

The LSTM 2 showed a precision of 0.33 and a recall of 0.04 meaning that a lot of both false positives and negatives were found.

In Table 4 we have the results gathered with the Italian dataset, the noisier one:

Bergamo DS	Prec	Recall	F1	AUC
Offline AE	0,86	1,00	0,92	0,99
SVR	0,67	0,98	0,80	0,99
RF	0,58	0,95	0,73	0,99
Online AE	0,32	0,94	0,47	0,90
MLP	0,41	0,83	0,55	0,81
LSTM 1	0,33	0,80	0,47	0,62
LSTM 2	0,01	1,00	0,01	0,57

Table 4: Performance metrics of Italian dataset

The results for this second test set (Table 4) seem to follow qualitatively the ones presented above.

The offline AE had the best overall performance, obtaining 0.86 in precision and 1 in recall. Very few false positives were found.

The statistical methods showed a good stability here as well, decreasing only by a few percentage points in precision, given the noisiness of the dataset, but held a good recall of >94% in both cases.

The online autoencoder behaved similarly to the other dataset as well: low precision, high recall but by looking at the graph all the points were just shifted and in general the anomalous days were all marked correctly, at least partially.

The LSTM 1 did not move too far away from the previous prediction, with a low precision and a 0.8 of recall.

The LSTM 2, instead, performed significantly worse. While it obtained 100% in recall, the precision dropped to 0.01. It is curious to see this as the training fit seemed precise, but the test set was fitted poorly, almost as if it was not able to recognize the patterns anymore.

3.5 Comments

There does not seem to be a clearly winning method here, all performed differently under different conditions. The offline autoencoder seems to be the one performing better in most situations, but at the cost of losing some information about the exact time at which the anomalous event happened, since it is only able to express if an anomaly occurred during the day. The statistical methods instead performed well in both situations, by not losing too much in precision and being able to correctly highlight most of the anomalous points.

The online autoencoder surprisingly performed worse than expected. But even though there was a shift of the anomalous data points and that while the performances were not as good as anticipated, the days that were labelled as anomalous were still highlighted. A possible explanation is that the optimization method to find the threshold through the roc curve for the MSE does not perform as well as it was suggested in the paper from which it was taken from [14]. More work should be done in this regard to further investigate the cause.

Chapter 4

Transfer Learning

As previously mentioned, one of the major problems afflicting anomaly detection for electrical power consumption in buildings, especially when done with neural networks, is the lack of data.

It is not simple to obtain data for buildings, in general when a new structure is built, new data must be collected if we want to use one of the anomaly detection methods presented in this paper. To collect such data is not only time consuming, since a few years of data must be gathered to be able to distinguish among the different seasons and obtain a decent generalization of the underlying patterns and trends, but it is also economically expensive. That is because if an anomaly detection system is not used during the data collection process, the system faults are not detected, and the nominal data collected are not clean, obtaining in the end a barely usable dataset like the Bergamo dataset used in this thesis. It seems therefore necessary to find a viable solution to obtain better data and reduce effort.

4.1 Basics of Transfer Learning

The solution that is proposed in this thesis is the use of transfer learning.

Transfer learning for deep neural networks is the process of first training a base network on a source dataset, and then transferring a small part of its layers to a second network to be trained on a smaller target dataset. This idea has been shown to improve deep neural network's generalization capabilities in many computer vision tasks such as image recognition and object localization [31].

Despite its recent success in computer vision, transfer learning has been rarely applied to deep learning models for time series data. The intuition behind the transfer learning approach for time series data is also partially inspired by the observation of Cui et al. [30], where the authors showed that shapelets (or subsequences) [13] are related to the filters (or kernels) learned by CNNs. In [31] the authors evaluate their developed framework thoroughly on the largest publicly available benchmark for time series analysis: the UCR archive[1], which consists of 85 datasets selected from various real-world domains.

For each pair of datasets (D1 and D2) in the UCR archive two experiments are performed in [31]:

• D1 is the source dataset and D2 is the target dataset.

• D1 is the target dataset and D2 is the source dataset.

Which makes it in total 7140 experiments for the 85 datasets in the archive.

[1] https://www.cs.ucr.edu/~eamonn/time_series_data/

These experiments yielded interesting yet hard-to-understand results. Here we first present the result of the 85×84 experiments in a form of a matrix in Figure 33.

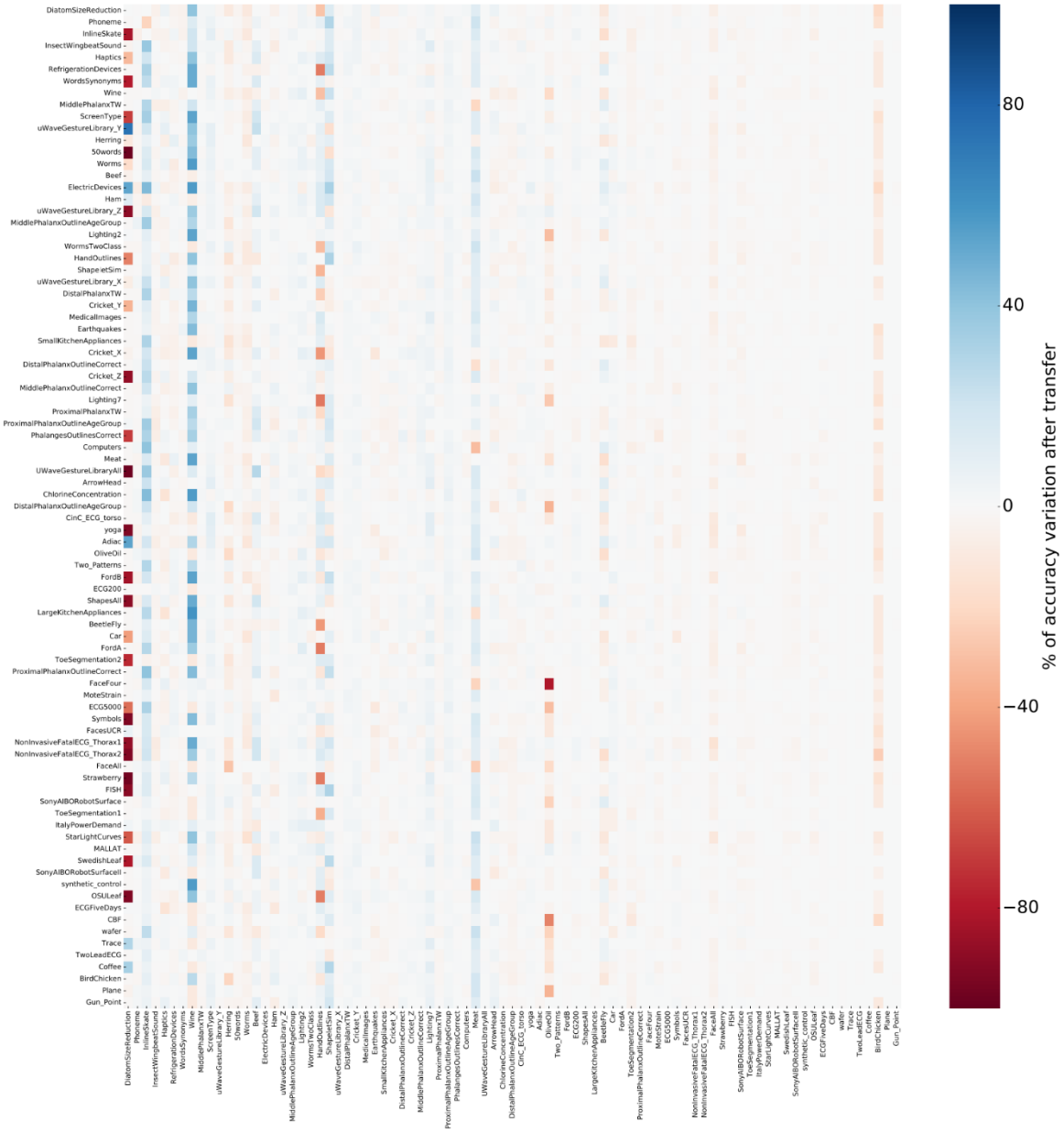

Figure 33: Matrix showing the accuracy variation of cross training in multiple datasets taken from [31]

Figure 33 displays the variation in percentage over the original accuracy when fine tuning a pre-trained model. The rows' indexes correspond to the source datasets and the columns' indexes correspond to the target datasets. The red color shows the extreme case where the chosen pair of datasets (source and target) deteriorates the network's performance. Where on the other hand, the blue color identifies the improvement in accuracy when transferring the model from a certain source dataset and fine-tuning on another target dataset. The white color means that no change in accuracy has been identified when using the transfer learning method for two datasets.

Figure 34 is used to visualize the worst- and best-case scenarios when fine-tuning a model against training from scratch, they plotted a pairwise comparison of three aggregated accuracies {minimum, median, maximum}.

By taking the minimum, it is easy to understand that one can find a bad source dataset for a given target dataset and decrease the model's original accuracy when fine-tuning a pre-trained network. On the other hand, the maximum accuracy (blue dots) shows that there are also cases where a source dataset increases the accuracy when using the transfer learning approach. As for the median (yellow dots), it shows that on average, pre-training and then fine-tuning a model on a target dataset improves without significantly hurting the model's performance.

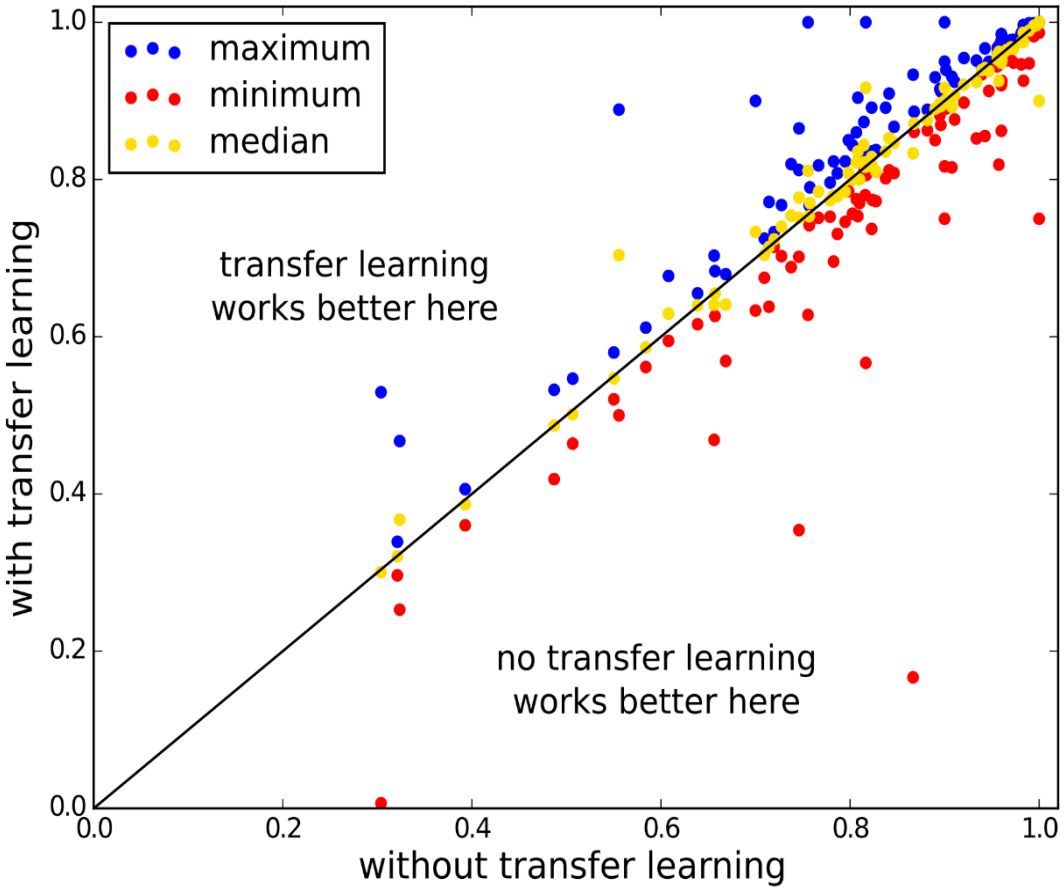

Figure 34: Aggregated accuracies with or without transfer learning taken from [31]

This analysis showed us that blindly and naively using the transfer learning approach could decrease the model's performance. This is largely due to the fact that the initial weights of the network have a significant impact on the training. This problem has been identified as negative transfer learning in the literature, where there still exists a need to quantify the amount of relatedness between the source and target datasets and whether an attempt to transfer knowledge from the source to the target domain should be made.

In the end, the authors of [31] concluded that transferring deep CNNs on a target dataset works best when fine-tuning a network that was pre-trained on a similar source dataset.

From this paper we pursue the idea of extending the research in the transfer learning field to the anomaly detection problem.

4.2 Transfer Learning in Building Electricity Consumption Anomaly Detection

Training the autoencoders presented in Section 3.3.4 and Section 3.3.7 on the Dutch dataset, which is more realistic and less prone to anomalies compared to the Italian smoothed dataset, and then performing a few epochs of fine tuning with the Italian one, seems to have improved stability. A much more stable prediction of the winter test set has been achieved over different runs and the anomalous days count was ranging between 30-35 over multiple runs (contrary to the 35 to 53 range obtained in the normal training discussed in Section 3.4.4), even though autoencoders are considered instable learners.

Moreover, the anomalies that were caught by the network were mostly days with a higher power consumption (both weekdays and weekends) but also some days with a lower power consumption than usual.

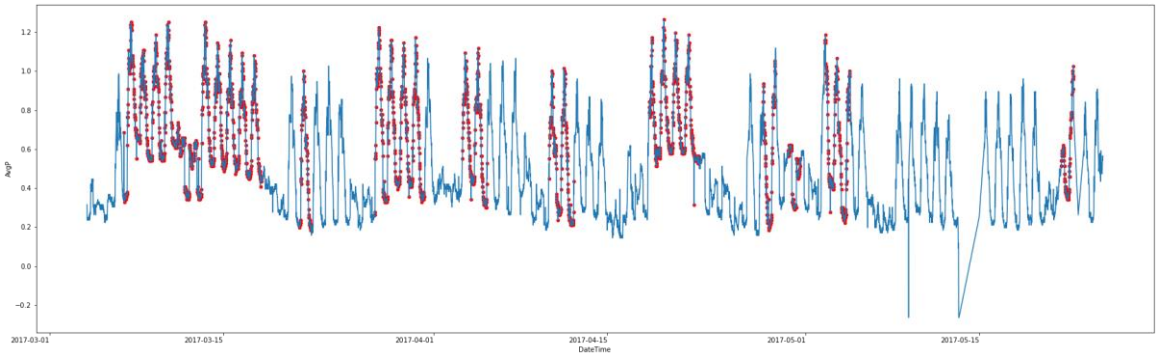

Figure 35: Prediction of Italian winter test set of offline AE with transfer learning

The results of the metrics used to compare the different models, presented in Table 5 and Table 6, did not show a drastic improvement that one could wish to expect, aside from a slight enhancement in the precision in the online AE in the first Bergamo test set going from 0.32 with normal training, to 0.61 with transfer learning at the cost of slightly reducing the recall. At the same time, the results did not show any degradation in performance.

Bergamo1	Prec	Recall	F1	AUC
Offline AE	0,86	1,00	0,92	0,99
Online AE	0,32	0,94	0,47	0,90
Offline AE TL	0,83	1,00	0,90	0,98
Online AE TL	0,61	0,84	0,71	0,87

Table 5: Performance metrics of first Italian test set with and without transfer learning

Bergamo2	Prec	Recall	F1	AUC
Offline AE	0,79	0,81	0,80	0,87
Online AE	0,71	0,67	0,69	0,78
Offline AE TL	0,82	0,72	0,77	0,86
Online AE TL	0,73	0,61	0,67	0,75

Table 6: Performance metrics of winter Italian test set with and without transfer learning

The higher overall precision shown in the second dataset is due to the fact that entire days were labelled as anomalous during the annotation period. That is because during the winter period from which this data was collected, there seemed to be anomalies also during the nighttime, where higher consumptions were registered.

Thus, it is possible to obtain similar or better results by performing transfer learning in anomaly detection using these two datasets. Even though both datasets deal with power consumption information, it is easy to see that the winter period of the Bergamo dataset used as test set is extremely different from the winter period registered by the Dutch facility. But even with that much difference, the models were able to still obtain around 80% and 70% of precision of positive samples respectively, although losing some points in recall, therefore reducing the number of errors in the samples classified as positive (higher precision) but increasing the number of false negatives (lower recall). Similarly, the harmonic mean of precision and recall (F1 score) and the area under the curve (AUC score) show that the transfer learning in this case loses some percentage points compared to the model trained

and tested with the same dataset. That should not be a surprise since there is a certain amount of bias and variance introduced when training a model with data from two buildings that are geographically distant and that have different insulation systems. This latter fact can be assumed given that the Dutch dataset follows much more stable trends and patterns all year round. Nonetheless it is easy to see how the use of the Dutch dataset for training a neural network could be applied to the anomaly detection process in new buildings in Lombardy at least for the first years when the proprietary data is being collected. This will result in a final dataset with fewer anomalies, since they can be promptly caught thanks to a momentary anomaly detection system implemented using transfer learning.

Chapter 5

Conclusion and Future Work

This work focused on trying to understand which machine learning method performs better for detecting anomalies in the context of building energy consumption. It was not easy to draw a conclusion because, for instance, the online autoencoder performed well in the partial recognition of the anomalies and managed to detect most of the anomalous days correctly but failed to predict the precise hours of the day at which the anomalies occurred, therefore lowering the performance metrics. This does not mean that such methods are not fit for this purpose, but rather that more research should be made before applying these methods in the real word. It must be noted that the entire datasets used here were not labelled at the source, but labelling was a task performed by the author of this thesis, who is not an expert in the field of building management. There are chances that the anomalies registered by a method like the online autoencoder were actual anomalies in the system that were not recognized when hand labelling. In any case all these results are conservative: even though the precision of the autoencoder is low it just means that there are many false positives, but the main anomalies were all discovered with good precision. The same can be said about the transfer learning

technique: it obtained results on par with normal training. There is not a lot of background for transfer learning in time series, more research should be done on this topic as it seems to be promising, possibly, by trying to use properly labelled datasets.

This thesis was therefore an explorative research in which the best methods according to performance obtained seem to be the statistical methods of SVR and Random Forest and the offline AE. It is extremely hard to evaluate all of these methods and compare them to the results reported in the original articles, since the datasets used by them were different and not publicly available.

The transfer learning technique seemed rather promising especially as a temporary anomaly detection system to be used in BEMSs during the data collection period for buildings that do not adopt any anomaly control strategies. Therefore, in the future, a further exploration of autoencoders that implement transfer learning could be useful to confirm and extend the results obtained in this work. The focus should be on the usage of datasets that have more regular features, are complete and are properly labeled by experts.

The discovery of a stable enough method to deal with anomaly detection could theoretically have an astonishing impact on the quality of life of people inhabiting the buildings and even more on the life of all the people on Earth, thanks to the reduced CO2 emission.

BIBLIOGRAPHY

[1]: S. Wang, Q. Zhou, Fu Xiao (2010). A system-level fault detection and diagnosis strategy for HVAC systems involving sensor faults, Energy and Buildings, Volume 42, Issue 4, pp. 477-490.

[2] D. Rumelhart , G. Hinton, R. Williams (1986). Learning representations by back-propagating errors. Nature 323, pp. 533–536 .

[3] J. Pereira, M. Silveira (2018). Unsupervised Anomaly Detection in Energy Time Series Data Using Variational Recurrent Autoencoders with Attention, 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1275-1282.

[4] C. Chahla, H. Snoussi, L. Merghem, M. Esseghir (2019). A Novel Approach for Anomaly Detection in Power Consumption Data, ICPRAM, pp. 483-490.

[5] V. Jakkula, D. Cook (2010). Outlier Detection in Smart Environment Structured Power Datasets. pp 29 - 33. 10.1109/IE.2010.13.

[6] A. Capozzoli, F. Lauro, I. Khan (2015). Fault detection analysis using data mining techniques for a cluster of smart office buildings, Expert Systems with Applications, Volume 42, Issue 9, pp. 4324-4338.

[7] S. Li, J. Wen (2014). A model-based fault detection and diagnostic methodology based on PCA method and wavelet transform, Energy and Buildings, Volume 68, Part A, pp. 63-71.

[8] K. Yoshida, M. Inui, T. Yairi, K. Machida, M. Shioya ,Y. Masukawa (2008). Identification of Causal Variables for Building Energy Fault Detection by Semi-supervised LDA and Decision Boundary Analysis," 2008 IEEE International Conference on Data Mining Workshops, pp. 164-173.

[9] P. Malhotra, L. Vig, G. Shroff, P. Agarwal (2015). Long Short Term Memory Networks for Anomaly Detection in Time Series, 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, pp. 89-94.

[10] J. Ploennigs, B. Chen, A. Schumann, N. Brady (2013). Exploiting generalized additive models for diagnosing abnormal energy use in buildings, Proceedings of the 5th ACM Workshop on Embedded Systems for Energy-Efficient Buildings, pp. 1-8.

[11] B. Chen, M. Sinn, J. Ploennings, A. Schumann (2014). Statistical anomaly detection in mean and variation of energy consumption, IEEE 22nd International Conference on Pattern Recognition pp. 3570-3575.

[12] E. Keogh, J. Lin, A. Fu (2005). HOT SAX: efficiently finding the most unusual time series subsequence, Fifth IEEE International Conference on Data Mining (ICDM'05), pp. 8-18.

[13] R. Fontugne, N. Tremblay, P. Borgnat, P. Flandrin, H. Esaki (2013). Mining anomalous electricity consumption using Ensemble Empirical Mode Decomposition, Acoustics, Speech, and Signal Processing, 1988. ICASSP-88, pp. 5238-5242.

[14] D. Araya, K. Grolinger, H. ElYamany, M. Capretz, G. Bitsuamlak (2017). An ensemble learning framework for anomaly detection in building energy consumption, Energy and Buildings, Volume 144, pp. 191-206.

[15] M. Jones, D. Nikovski, M.Imamura, T. Hirata (2014). Anomaly Detection in Real-Valued Multidimensional Time Series, ASE IGDATA/SOCIALCOM/CYBERSECURITY Conference, pp. 1-9.

[16] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff (2016). LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection, https://arxiv.org/pdf/1607.00148 last visited November 16, 2021.

[17] X. Liu, N. Iftikhar, P. Nielsen, A. Heller (2016). Online Anomaly Energy Consumption Detection Using Lambda Architecture, Big Data Analytics and Knowledge Discovery, DaWaK 2016. Lecture Notes in Computer Science, vol. 9829, pp. 193-209.

[18] Y. Himeur, K. Ghanem, A. Alsalemi, F. Bensaali, A. Amira (2021). Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives, Applied Energy, Volume 287, pp. 1-26.

[19] D. Wijayasekara, O. Linda, M. Manic, C. Rieger (2014). Mining Building Energy Management System Data Using Fuzzy Anomaly Detection and Linguistic Descriptions, IEEE Transactions on Industrial Informatics, Volume 10, Issue 3, pp. 1829-1840.

[20] J. Chou, A. Telaga (2014). Real-time detection of anomalous power consumption, Renewable and Sustainable Energy Reviews, Volume 33, pp. 400-411.

[21] N. Sisworahardjo, A. Saad (2017). Spatio-Temporal Context Anomaly Detection for Residential Power Consumption, International Journal on Electrical Engineering and Informatics, Volume 9, pp. 776-785.

[22] L. Feremans, V. Vercruyssen, B. Cule,W. Meert, B. Goethals (2019). Pattern-Based Anomaly Detection in Mixed-Type Time Series, Machine Learning and Knowledge Discovery in Databases, pp. 240-256.

[23] O. Linda, D. Wijayasekara, M. Manic, C. Rieger (2012). Computational intelligence-based anomaly detection for Building Energy Management Systems, 2012 5th International Symposium on Resilient Control Systems, pp. 77-82.

[24] Y. Zhu, X. Jin, Z. .Du (2011). Fault diagnosis for sensors in air handling unit based on neural network pre-processed by wavelet and fractal, Energy and Buildings, Volume 44, pp. 7-16.

[25] Keras, https://keras.io/api/optimizers/ last visited November 16, 2021.

[26] C. Fan, F. Xiao, Y. Zhao, J. Wang (2018). Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data, Applied Energy, Volume 211, 2018, pp. 1123-1135.

[27] Fan, C., Xiao, F., Madsen, H., & Wang, D. (2015). Temporal knowledge discovery in big BAS data for building energy management, Energy and Buildings, Volume 109, pp. 75-89.

[28] Y. LeCun, Y. Bengio, G. Hinton (2015). Deep learning. Nature, Volume 521, pp. 436–444.

[29] D. Berthelot, C. Raffel, A. Roy, I. Goodfellow (2018). Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer, ICLR 2019 Conference, Paper 73, pp. 1-20.

[30] Z. Cui, W. Chen, Y. Chen (2016). Multi-Scale Convolutional Neural Networks for Time Series Classification, https://arxiv.org/abs/1811.01533 last visited November 16, 2021.

[31] H. Fawaz, G. Forestier, J. Weber, L. Idoumghar , P. Muller (2018), Transfer Learning for time series classification, https://arxiv.org/abs/1811.01533 last visited November 16, 2021.

[32] D. Araya, K. Grolinger, H. ElYamany, M. Capretz , G. Bitsuamlak (2016). Collective contextual anomaly detection framework for smart buildings, International Joint Conference on Neural Networks (IJCNN), pp. 511-518.

[33] Sarker, I.H (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN COMPUT. SCI. 2, Article number 420 (2021).

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio (2014). Generative Adversarial Networks, 27th International Conference on Neural Information Processing Systems, Volume 2, pp. 2672–2680.