Model-centered Ensemble for Anomaly Detection in Time Series

Erick L. Trentini, Ticiana L. Coelho da Silva, Leopoldo Melo Junior and Jose F. de Macêdo
Insight Data Science Lab, Fortaleza, Brazil

Keywords: Time Series, Anomaly Detection, Ensemble.

Abstract: Time-series anomalies detection is a fast-growing area of study, due to the exponential growth of new data produced by sensors in many different contexts as the Internet of Things (IOT). Many predictive models have been proposed, and they provide promising results in differentiating normal and anomalous points in a time-series. In this paper, we aim to find and combine the best models on detecting anomalous time series, so that their different strategies or parameters can contribute to the time series analysis. We propose TSPME-AD (stands for Time Series Prediction Model Ensemble for Anomaly Detection). TSPME-AD is a model-centered based ensemble that trains some of the state-of-the-art predictive models with different hyper-parameters and combines their anomaly scores with a weighted function. The efficacy of our proposal was demonstrated in two real-world time-series datasets, power demand, and electrocardiogram.

1 INTRODUCTION

Stock market prices, sleep monitoring, trajectories of moving objects are real-world data commonly registered taking into account some notion of time. When collected together, the measurements compose what is known as a time series.

Collecting vast volumes of time series data opens up new opportunities to discover hidden patterns. As an example, doctors can be interested in searching for anomalies in the sleep patterns of a patient. In the mobility data domain, for instance, a new interest in trajectory anomaly research has occurred, which can be integrated with navigation to provide dynamic routes for drivers or travelers. Besides, this research can provide accurate real-time advisor routes compared with navigation based on static traffic information. Another application is for taxi companies that may observe drivers with traveling trajectories that are different from the popular choices of other drivers and detect fraudulent behavior.

There is a range of different approaches that address the problem of anomaly detection on time series. Several techniques can be applied to perform such tasks using predictive models, clustering-based methods, distance-based methods, among others (Meng et al., 2018). However, detecting anomalies in sequence learning tasks become challenging using standard approaches based on mathematical models that rely on stationarity (Malhotra et al., 2016). The state-of-the-art has been investigating LSTM neural networks (Hochreiter and Schmidhuber, 1997) to overcome these limitations and to model the normal behavior of a time series, then accurately detect deviations from normal behaviour without any pre-specified threshold or preprocessing phase (Malhotra et al., 2015; Malhotra et al., 2016).

In this paper, we follow a similar idea. We use some predictors, based on LSTM neural network to model normal behavior, and subsequently, use the prediction errors to identify anomalies. These network models are data-hungry techniques and require a massive amount of training data. We profit from the fact that there are a plethora of instances of normal behavior than anomalous to employ these techniques. The intuition behind is that the network model would only have seen instances of normal behavior during training and the model can reconstruct them. When given an anomalous time series, it may not be able to rebuild it properly, and it would end up with higher reconstruction errors than for non-anomalous time series.

We propose TSPME-AD (stands for Time Series Prediction Model Ensemble for Anomaly Detection). TSPME-AD combines two state-of-the-art detection models (Malhotra et al., 2016; Malhotra et al., 2015) to derive a combined decision. Various classifier combination schemes have been devised and it has been experimentally demonstrated that some of them consistently outperform a single best classifier (Kittler

Trentini, E., Coelho da Silva, T., Melo Junior, L. and F. de MacÃŀdo, J.
Model-centered Ensemble for Anomaly Detection in Time Series.
DOI: 10.5220/0008985507000707
In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 2*, pages 700-707
ISBN: 978-989-758-395-7; ISSN: 2184-433X

et al., 1996). In the experimental section, we prove that by using an ensemble of such classifiers, the final model improves in terms of F-measure on detecting anomalous behavior.

This paper investigates a challenging problem since the anomaly detection is performed on multivariate time series data. As discussed in (Wang et al., 2018), anomalies may occur in only a subset of dimensions (variables). Another drawback is the locations and lengths of anomalous sub-sequences may be different in different dimensions. Third, the anomalous time series may look normal in each dimension individually, but their combinations may be anomalous.

The remainder of the paper is structured as follows: Section 2 introduces formally the problem statement. Section 3 presents the preliminary concepts to understand our approach. Section 4 presents our proposal. Section 5 presents the related works. Section 6 discusses the experimental evaluation, and finally Section 7 draws the final conclusions.

2 PROBLEM STATEMENT

Consider a multivariate time series $X = [x^{(1)}, x^{(2)}, \ldots, x^{(n)}]$ such that $x^{(i)} \in \mathbb{R}^m$ is a m-dimensional vector $x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}]$ at time $t = i$. Usually, the predictive models seek to predict the next point given a time series. That is, for a predictive model M and a time series $X = [x^{(1)}, x^{(2)}, \ldots, x^{(n)}]$, $M(x^{(i)}) = x^{(i+1)}$. Some models may differ from this perspective, such as predicting more than one data point, $M(x^{(i)}) = [x^{(i+1)}, x^{(i+2)}]$, or re-building the time-series backwards $M(x^{(i)}) = x^{(i-1)}$.

Given a predictive model M and a time series X, $Y = M(X)$ is the predicted sequence of X using M such that $Y = [y^{(1)}, y^{(2)}, \ldots, y^{(n)}]$, and $y^{(i)}$ is the *attempt* from M to build $x^{(i)}$. Our goal is to reconstruct the sequence X, compute the prediction errors based on the prediction $M(x^{(i)})$ compared to x_i, compute the anomaly scores (using the error distribution) and identify the anomalies on X.

3 PRELIMINARIES

This section discusses the network architectures used by our approach introduced in (Malhotra et al., 2015; Malhotra et al., 2016). Our proposal is an ensemble model that combines both strategies.

3.1 Stacked LSTM

Consider two sets of time series: s_N for training the prediction model M and v_N for validating M. Let $s_N = [s_N^{(1)}, s_N^{(2)}, \ldots, s_N^{(n)}]$ such that $s_N^{(i)} \in \mathbb{R}^m$ is a m-dimensional vector $s_N^{(i)} = [s_{N_1}^{(i)}, s_{N_2}^{(i)}, \ldots, s_{N_m}^{(i)}]$ at time $t = i$. The same applies for v_N.

For $s_N^{(i)}$, each one of the m dimensions ($s_N^{(i)} \in \mathbb{R}^m$) is taken by one unit in the input layer, and there is one unit in the output layer for each of the l future predictions for each of the m dimension. The LSTM units in a hidden layer are fully connected through recurrent connections. (Malhotra et al., 2015) stacks LSTM layers such that each unit in a lower LSTM hidden layer is fully connected to each unit in the LSTM hidden layer above it through feedforward connections. Figure 1 shows the Stacked LSTM architecture. The prediction model M is learned using the non-anomalous training sequence s_N.

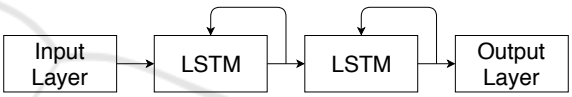

Figure 1: Stacked LSTM model as proposed by (Malhotra et al., 2015).

Consider X a set of time series, and a prediction length of l, each of the selected d dimensions of $x^{(t)} \in X$ for $l < t \leq n - l$ is predicted l times. Error vectors are computed for each $x^{(t)}$ such that $e^{(t)} = [e_{11}^{(t)}, \ldots, e_{1l}^{(t)}, \ldots, e_{d1}^{(t)}, \ldots, e_{dl}^{(t)}]$ where $e_{ij}^{(t)}$ is the difference between $x_i^{(t)}$ and the value predicted by the model M at time $t - j$. In (Malhotra et al., 2015), the prediction model trained on s_N is used to compute the error vectors for each point in the validation and test sequences. The error vectors are modelled to fit a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. The validation set is used to estimate μ and Σ using Maximum Likelihood Estimation. The anomaly score $p^{(t)}$ of an error vector $e^{(t)}$ is given by the value of $\mathcal{N}$ at $e^{(t)}$, in other words, $p^{(t)}$ is computed as $(e^{(t)} - \mu)^T \Sigma^{(-1)} (e^{(t)} - \mu)$ for an observation $x^{(t)}$. For $x^{(t)}$, the value predicted is considered as anomalous if the $p^{(t)} > \tau$, else it is classified as normal. The value of τ is learned using v_N by maximizing F1-score (considering a classification problem where that anomalous points belong to a class and normal points to another class).

3.2 Encoder Decoder Model

The network architecture discussed in this section is composed of an LSTM-based encoder that learns

fixed-length vector representation of the input time-series. And an LSTM-based decoder that uses this representation to reconstruct the time-series using the current hidden state and the value predicted at the previous time-step. The network architecture was proposed in (Malhotra et al., 2016) and it is illustrated in Figure 2.

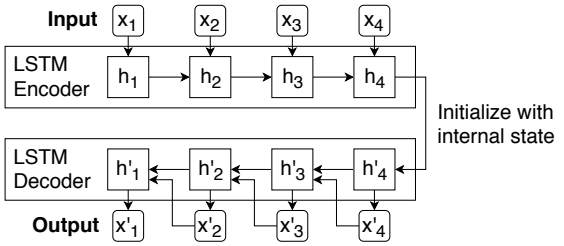

Figure 2: Encoder-Decoder model as proposed by (Malhotra et al., 2016).

In general, by using Encoder Decoder architecture, the representation is learned from the entire sequence which is then used to reconstruct the sequence. This is different from usual prediction based anomaly detection models. Given s_N for training the prediction model M, $h_E^{(i)}$ is the hidden state of encoder at time t_i for each $i \in \{1,2,...,n\}$ where $h_E^{(i)} \in \mathbb{R}^c$, c is the number of LSTM units in the hidden layer of the encoder. The encoder and decoder are jointly trained to reconstruct the time series in reverse order as $\{s_N^{(n)}, s_N^{(n-1)}, \ldots, s_N^{(1)}\}$. The final state $h_E^{(n)}$ of the encoder is used as the initial state for the decoder. A linear layer on top of the LSTM decoder layer is used to predict the target. During the decoding phase, the decoder uses $s_N^{(i)}$ and the internal state $h_D^{(i-1)}$ to predict $s_N^{(i-1)}$ corresponding to target $s_N^{(i-1)}$. Let s_N be a set of normal training sequences, the encoder decoder model is trained to minimize the objective $\sum s_N^{(i)} \in s_N \sum_{i=1}^{n} \left\| s_N^{(i)} - s_N^{(i)} \right\|^2$.

For an observation, $x^{(t)}$, the anomaly score $p^{(t)}$ in (Malhotra et al., 2016) is computed similarly as explained in the last section by modeling the error vectors to fit a Multivariate Gaussian distribution. The next section discusses our approach.

4 TSPME-AD: TIME SERIES PREDICTION MODEL ENSEMBLE FOR ANOMALY DETECTION

In this paper, we combine the models (Malhotra et al., 2015; Malhotra et al., 2016) using a model-centered ensemble technique that attempts to combine the anomaly score from both models built on the same dataset. However, there exist some challenges in the combination process. According to (Aggarwal, 2013), the main issues are normalization and combination. The former corresponds to the problem of different models may output anomaly scores not easily comparable. The latter is the problem of deciding which combination function is the best (the minimum, the maximum or the average). These are still open questions, according to (Aggarwal, 2013), the literature on outlier ensemble analysis is very sparse so the solutions for these mentioned issues are not completely known.

To address the first issue, a damping function is applied to the anomaly scores, in order to prevent it from being dominated by a few components (Aggarwal, 2013). Examples of a damping function could be the square root or the logarithm. The second issue is addressed in this paper by using a weighted average on the damped scores, that can be trained using some sort of optimization algorithm. Figure 3 gives an overview of the TSPME-AD pipe to construct the model.

To construct the ensemble, we first calculate the anomaly scores for each model in our ensemble as in Figure 4, then we use a damping function on all anomaly scores as in Figure 5, and aggregate each set of scores using the weighted average function as shown in Figure 6. In the experiments, we show that the damped weighted average function (used by TSPME-AD) performs better than the ensemble model using the logarithm as a damping function. With our new aggregated set of anomaly scores, we try to find a threshold that maximizes some desired score on the validation set as exemplified in Figure 7.

For a time series X, let the anomaly score $a_i \in \mathbb{R}$ and $b_i \in \mathbb{R}$ be computed from the prediction value outputted by the models (Malhotra et al., 2015) and (Malhotra et al., 2016), respectively. Our approach uses the combination function shown in Equation 1 to compute the anomaly score $\forall x_i \in X$:

$$A_i = \frac{w_{(1)} \times \ln a_i + w_{(2)} \times \ln b_i}{w_{(1)} + w_{(2)}} \tag{1}$$

We say that x_i is anomalous if

$$A_i > \tau$$

where τ is learned as one of the weights.

The weights $w_{(1)}$, $w_{(2)}$ and τ are learned from the validation set v_N during the training phase, with w_1, $w_2 \in [0,1]$ and $\tau \in \mathbb{R}$. And the goal is to maximize the F1-score.

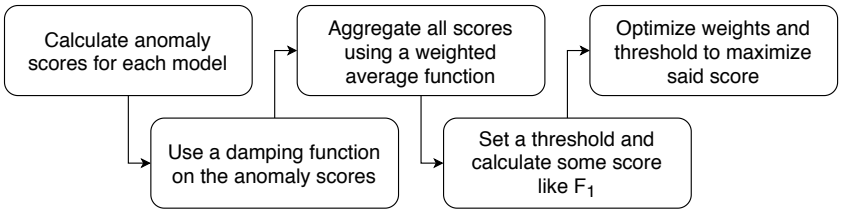

Figure 3: The full pipe of TSPME-AD.

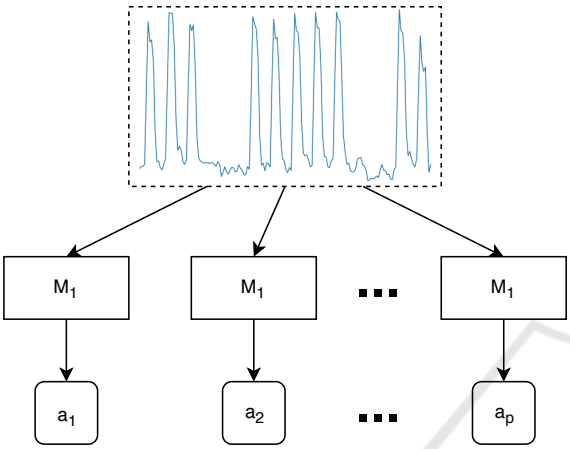

Figure 4: First, all models try to reconstruct the time series, and we calculate the anomaly scores.

In this paper, we focus on the combination of the models proposed by (Malhotra et al., 2015) and (Malhotra et al., 2016), but the combination function can be extended to any number of models with their respective variations of hyper-parameters, and can be generalized as in Equation 2.

$$A_i = \frac{w_{(1)} \times \ln a_i^{(1)} + w_{(2)} \times \ln a_i^{(2)} + \cdots + w_{(p)} \times \ln a_i^{(p)}}{w_{(1)} + w_{(2)} + \cdots + w_{(p)}} \tag{2}$$

In Equation 2, p is the number of different models used to reconstruct the time series. From the authors' knowledge, none of the previous work that proposes model-centered outlier ensemble models uses this function or an LSTM based approach for outliers detection (Aggarwal, 2013; Liu et al., 2012).

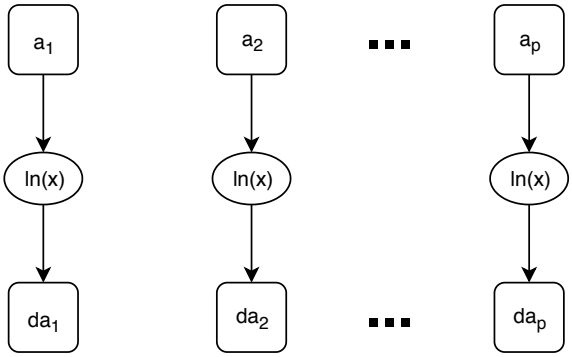

Figure 5: Applying a damping function to all anomaly scores. e.g. the natural log function.

5 RELATED WORK

Anomaly detection models in time series have been investigated by using machine learning and statistical approaches as discussed in (Chandola et al., 2009).

Besides the aforementioned techniques, a new one which is recently gaining momentum is deep learning and generally used to deal with non-linear models. However, only a few studies consider deep neural networks for resolving outlier detection. (Kieu et al., 2018) proposes an outlier detection framework to identify an anomaly in multidimensional time-series data. The framework incorporates several deep neural network-based autoencoders. The idea behind using autoencoders is they likely to fail to reconstruct outliers using small feature space. Therefore, deviations between the original input data and the reconstructed data can be taken as indicators of outliers. The paper (Malhotra et al., 2016) proposes an LSTM encoder-decoder architecture that is trained to reconstruct instances of normal behavior. When given an anomalous time series, it may not be able to rebuild it properly. Another paper that follows the same idea is (Malhotra et al., 2015), however, the model proposed stacks LSTM networks. Both paper (Malhotra et al., 2015; Malhotra et al., 2016) solves the same problem than this approach, however, we gather the best of both papers by combining them to derive a combined decision.

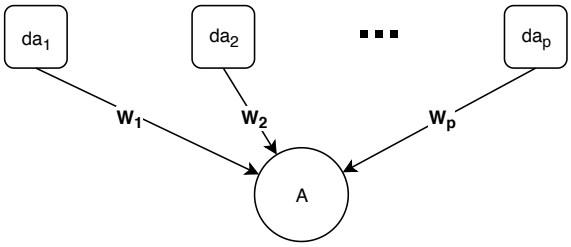

Figure 6: Aggregating all sets of anomaly scores using a weighted average function.

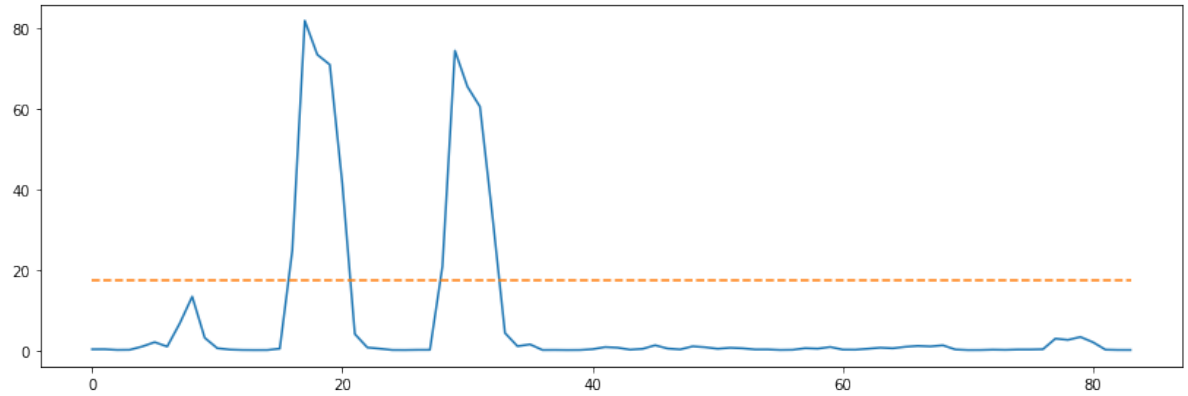

Figure 7: Setting a threshold to discriminate between 'normal' and anomalous points.

The approach proposed in (Kong et al., 2018) can detect long-term traffic anomaly with crowdsourced bus trajectory data. The time series segments are extracted from bus trajectory data to describe the whole city traffic situation from both temporal and spatial aspects. (Kong et al., 2018) extracts the average velocity and average stop time which can describe traffic conditions and travel demand respectively. Then (Kong et al., 2018) excavates poor segments which are the bottleneck of traveling in one line by calculating their anomaly index. The approach (Tariq et al., 2019) proposes an anomaly detector for a satellite system using a multivariate Convolutional LSTM combined with a complementary Mixtures of Probabilistic Principal Component Analyzer. The proposed model learns from a large amount of normal telemetry data, it predicts between normal and abnormal telemetry sequence.

Another type of anomaly detection algorithms uses clustering techniques. Paper (Wang et al., 2018) proposes a clustering algorithm that discretizes the time series data into time windows, and clusters all subsequences within each window. Univariate subsequences in the same cluster within a window are similar to each other. The behavior patterns of objects are obtained by the cluster centers, and if a time series does not follow such behavior it is anomalous. For multivariate time series, the algorithm transforms the original time series into a new feature space in which each feature is the distance to a pattern. The smaller the distance, the more similar the data is to the pattern. (Wang et al., 2018) performs clustering on the transformed data, and assign anomaly score to each time series based on the clustering results and distances to normal cluster. Other clustered based approaches for anomaly detection are (Gao et al., 2012; Iverson, 2004).

The main difference between TSPME-AD and the previous one is a model-centered based approach that uses a damped averaging function to combine the best of the state-of-the-art detection models to derive a combined decision. There exist few similar approaches that propose an ensemble model for anomaly detection (Aggarwal, 2013) as (Liu et al., 2012; Gao and Tan, 2006). However, none of these approaches models normal behavior by profiting of LSTM neural networks for multidimensional time series.

6 EXPERIMENTS AND RESULTS

In this section, we conduct some experiments with two real-world datasets and report the precision, recall, F_1 and $F_{0.1}$ scores for TSPME-AD and the baseline models (Malhotra et al., 2015; Malhotra et al., 2016). We also study different combination functions to ensemble the baseline models.

6.1 Experimental Setup

We split the dataset into 4 groups, s_n, v_n, v_a, t_a, where s_n and v_n consist of a set of the time-series without anomalies, and the other two groups (v_a and t_a) are the remaining with at least one anomaly each.

We trained the models with s_n using v_n for early stopping. Also, the training set (s_n and v_n) were used to generate the error distribution, and then to calculate the anomaly scores of the models.

We used the set v_a to train the weights of the combination function and the threshold τ. Finally, with the set t_a we calculated the anomaly scores and compared TSPME-AD with the baseline models.

6.1.1 Competitors

For the stacked LSTM model, we implemented an LSTM network with 30 and 20 LSTM nodes on the

first and second hidden layers respectively, using the Sigmoid activation function for the hidden layers, and a linear activation for the output.

Since the Stacked LSTM presents as a hyper-parameter, the number of points ahead to predict at each step. In these experiments, we varied as 2, 4, 8 and 16 the number of points ahead to predict.

The Encoder-Decoder model only needs to reconstruct the original time-series, however there is a hyper-parameter that is the number of hidden LSTM units. We used the same number for the encoder and the decoder. In the experiments, we varied the number of hidden nodes as 16, 32, 64 and 128.

6.2 Datasets

In what follows, we provide a brief overview of each used dataset.

6.2.1 Power Demand

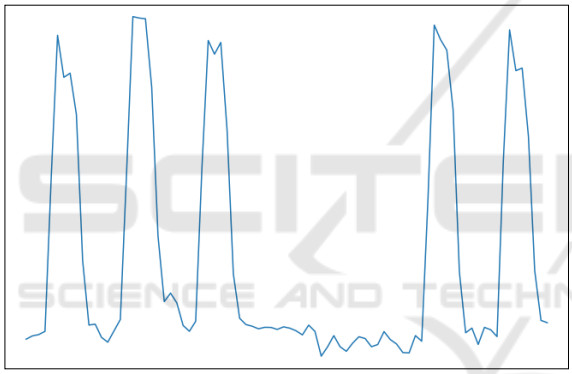

Figure 8: A normal week of power demand, starting at Wednesday.

The power demand dataset provided by (Keogh et al., 2007) registered the demand for energy supply for one year. The normal behavior is high demand during the weekdays, and low during the weekends. Then, the high demand on the weekends or low demand on weekdays indicate for us anomalies not annotated.

The dataset is then sub-sampled by a factor of 8, and broken in non-overlapping windows of 84 points, that represent exactly one week of data. This was also performed in (Malhotra et al., 2016). Figure 8 shows a normal behavior of power demand starting on Wednesday.

The sets s_n, v_n and v_a were built using the first 40% of dataset of the year, and the remaining to the set t_a. So we can better calculate the scores, as we will have more anomalous points to test.

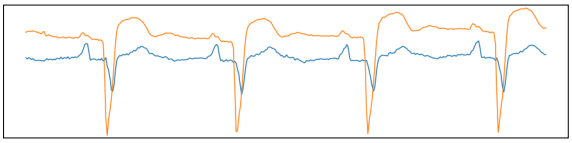

Figure 9: 4 heartbeats of the electrocardiogram dataset.

6.2.2 Electrocardiogram

The other real-world dataset is the mitdbx_108 also provided by (Keogh et al., 2007), that depicts an electrocardiogram with three distinct anomalies. This dataset is not so well behaved like the power demand since the heartbeat can occur at a different pace in different parts of the time series. This dataset is particularly harder to train the encoder-decoder model since it's harder to find the right size and skip for the sliding window.

This dataset is sub-sampled by a factor of 4, and broken in non-overlapping windows of 93 points, which represents on average one cycle of a heartbeat of the patient's electrocardiogram. Figure 9 shows four cycles of a patient's heartbeat.

As we applied in the power demand dataset, we divided the first 40% of the e.c. to create s_n, v_n and v_a and the remaining we allocated to t_a.

6.3 Results

In this section, we present the results outputted by TSPME-AD and its competitors. We compare our proposal with' Stacked LSTM (SL), and Encoder-Decoder (ED) anomaly detection techniques. We also evaluate the combination of these techniques using the following ensemble strategies: Simple average Ensemble (SA), Damped Average Ensemble (DA), and Simple Weighted Average Ensemble (SWA).

As we mentioned before, to evaluate these anomaly detection strategies, we use Power Demand and Electrocardiogram datasets. We measure the performance in terms of f-measure, studying two levels of weighting between precision and recall. First, we compute F1-score, using $\beta = 1$, which gives a balanced weight to both measures. After, we compute $F_{0.1}$, F-measure with $\beta = 0.1$, which provides higher weight to precision than to recall in the F-measure formula. The reason to evaluate $F_{0.1}$ is the extremely imbalanced behaviour of the time series data. As the number of normal data is higher than the number of anomalies, an approach that produces a higher number of false positives may not be feasible in an anomaly detection problem. Additionally, we also analyze the precision and recall separately.

6.3.1 Evaluation Results from Power Demand Dataset

Power Demand dataset is a periodic time series data. It means that the number of points per cycle is constant in time. This characteristic aids pattern recognition, and consequently, the anomaly prediction.

Table 1 shows the precision, recall, $F_{0.1}$ and F_1 for all the baseline models and the ensemble models evaluated, trained and tested with Power Demand. As we can see, the Simple Average (SA) and Damped Average (DA) ensembles achieved the best results in terms of $f_{0.1}$ and F_1, respectively. In this experiment, TSPME-AD achieved the second-best result in terms of F_1 and $F_{0.1}$.

This experiment shows that the weighted damped average strategy of TSPME-AD achieves quality results, but it does not outperform the simple damped average approach (DA). The great performance of these damping strategies can be explained by the output of the anomaly detection base models. A damping average attenuates the input values before averaging. As the output range of these models is wide, a damping strategy helps to standardize the model's outputs.

Table 1: Test results for the Power Demand dataset.

MODELS[a]	Precision	Recall	$F_{0.1}$	F_1
SL [K = 2]	4.42%	77.78%	0.04	0.08
SL [K = 4]	5.49%	77.78%	0.05	0.10
SL [K = 8]	22.86%	44.44%	0.22	0.30
SL [K = 16]	12.77%	66.67%	0.12	0.21
ED [H = 16]	47.06%	44.44%	0.47	0.45
ED [H = 32]	3.39%	22.22%	0.03	0.05
ED [H = 64]	59.09%	72.22%	0.59	0.65
ED [H = 128]	18.52%	27.78%	0.18	0.22
SA	100.0%	44.44%	0.98	0.61
DA	76.19%	88.89%	0.76	0.82
SWA	25.00%	72.22%	0.25	0.37
TSPME-AD	76.47%	72.22%	0.76	0.74

[a] **SL**: Stacked LSTM, **ED**: Encoder Decoder, **SA**: Simple Average Ensemble, **DA**: Damped Average ensemble, **SWA**: Simple Weighted Average Ensemble, **TSPME-AD**: Time Series Prediction Model Ensemble for Anomaly Detection.

6.3.2 Evaluation Results from Electrocardiogram Dataset

As the duration of a cyclic in an electrocardiogram varies from one instance to another, this data is called as quasi-periodic time-series. This class of time-series is challenging to build a prediction model because we also need to discover an average duration of a cyclic, as done by (Malhotra et al., 2016).

As in the previous subsection, Table 2 shows the precision, recall, $F_{0.1}$ and F_1 for all baseline models and the ensemble model trained and tested using this dataset. However, the TSPME-AD achieved the best results regarding F_1, $F_{0.1}$, and precision, which is different from the results obtained using the Power Demand dataset. The Encoder-Decoder-based models achieve the best recall results, but the precision score of these models indicates that almost all normal data are classified as anomaly data.

In this experiment, the standard ensemble fusion strategies, such as SA, DA, and SWA, are not able to combine the baseline models (Malhotra et al., 2015; Malhotra et al., 2016) properly. The low performance of these ensembles can be explained by the low performance of encoder-decoder base models. The standard ensemble functions can not attenuate the poor anomaly detection ability of Encoder-Decoder models.

We can conclude that the TSPME-AD fusion strategy (using a weighted damped average function) can compensate poor results of some anomaly detection baseline models and produce an ensemble with better quality results than the other fusion approaches and baseline models individually.

It is worth to mention that TPSME-AD, in general, outperforms the detection anomaly models from the state-of-the-art techniques (SL and ED) as already expected since our ensemble model combines the best of models (Malhotra et al., 2015; Malhotra et al., 2016) on detecting anomalous time series.

Table 2: Test results for the Electrocardiogram.

MODELS[a]	Precision	Recall	$F_{0.1}$	F_1
SL [K = 2]	22.37%	47.80%	0.22	0.30
SL [K = 4]	18.37%	57.07%	0.18	0.28
SL [K = 8]	20.97%	48.29%	0.21	0.29
SL [K = 16]	42.28%	30.73%	0.42	0.36
ED [H = 16]	7.02%	100%	0.07	0.13
ED [H = 32]	7.36%	100%	0.07	0.14
ED [H = 64]	7.37%	100%	0.07	0.14
ED [H = 128]	7.37%	100%	0.07	0.14
SA	30.84%	48.29%	0.31	0.38
DA	11.33%	60.00%	0.11	0.19
SWA	34.05%	46.34%	0.34	0.39
TSPME-AD	41.00%	47.80%	0.41	0.44

[a] **SL**: Stacked LSTM, **ED**: Encoder Decoder, **SA**: Simple Average Ensemble, **DA**: Damped Average ensemble, **SWA**: Simple Weighted Average Ensemble, **TSPME-AD**: Time Series Prediction Model Ensemble for Anomaly Detection.

7 CONCLUSION AND FUTURE WORK

In this paper, we provide an approach for anomaly detection which combines two state-of-the-art detection models, one based on stacked LSTM and another one encoder-decoder based. TPSME-AD, in general, outperforms the detection anomaly models from the state-of-the-art techniques as already expected since our ensemble model combines the best of models (Malhotra et al., 2015; Malhotra et al., 2016) on detecting anomalous time series. In the experiments, we also show that, for a quasi-periodic time series data, our model can outperform also standard ensemble fusion approaches, such as simple average, damped average, and simple weighted average.

As a future direction, we aim at evaluating our proposal with other datasets like the electrocardiogram, and the space-shuttle valve time-series (Keogh et al., 2007). Another future improvement can be added to a regularization of the combination function so that we can mitigate the overfitting in the validation dataset.

ACKNOWLEDGMENTS

This work is partially supported by the FUNCAP SPU 8789771/2017, and the UFC-FASTEF 31/2019.

REFERENCES

Aggarwal, C. C. (2013). Outlier ensembles: position paper. *ACM SIGKDD Explorations Newsletter*, 14(2):49–58.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.

Gao, J. and Tan, P.-N. (2006). Converting output scores from outlier detection algorithms into probability estimates. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 212–221. IEEE.

Gao, Y., Yang, T., Xu, M., and Xing, N. (2012). An unsupervised anomaly detection approach for spacecraft based on normal behavior clustering. In *2012 Fifth International Conference on Intelligent Computation Technology and Automation*, pages 478–481. IEEE.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Iverson, D. L. (2004). Inductive system health monitoring.

Keogh, E., Lin, J., Lee, S.-H., and Van Herle, H. (2007). Finding the most unusual time series subsequence: algorithms and applications. *Knowledge and Information Systems*, 11(1):1–27.

Kieu, T., Yang, B., and Jensen, C. S. (2018). Outlier detection for multidimensional time series using deep neural networks. In *2018 19th IEEE International Conference on Mobile Data Management (MDM)*, pages 125–134. IEEE.

Kittler, J., Hater, M., and Duin, R. P. (1996). Combining classifiers. In *Proceedings of 13th international conference on pattern recognition*, volume 2, pages 897–901. IEEE.

Kong, X., Song, X., Xia, F., Guo, H., Wang, J., and Tolba, A. (2018). Lotad: Long-term traffic anomaly detection based on crowdsourced bus trajectory data. *World Wide Web*, 21(3):825–847.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):3.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff, G. (2016). Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*.

Malhotra, P., Vig, L., Shroff, G., and Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. In *Proceedings*, page 89. Presses universitaires de Louvain.

Meng, F., Yuan, G., Lv, S., Wang, Z., and Xia, S. (2018). An overview on trajectory outlier detection. *Artificial Intelligence Review*.

Tariq, S., Lee, S., Shin, Y., Lee, M. S., Jung, O., Chung, D., and Woo, S. S. (2019). Detecting anomalies in space using multivariate convolutional lstm with mixtures of probabilistic pca. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2123–2133. ACM.

Wang, X., Lin, J., Patel, N., and Braun, M. (2018). Exact variable-length anomaly detection algorithm for univariate and multivariate time series. *Data Mining and Knowledge Discovery*, 32(6):1806–1844.