Unsupervised Anomaly Detection in Heterogeneous Network Time Series with Mixed Sampling Rates

Shaoyu Dou , **Kai Yang*** , **Pan Luo** and **Yang Jiao**

School of Computer Science and Technology, Tongji University, Shanghai, China
Shenzhen Institute of Artificial Intelligence and Robotics for Society, China

Abstract

Identifying anomalies in a collection of heterogeneous time series finds a variety of important applications in diverse domains, ranging from smart city to network security analysis. An underappreciated problem from these applications is how to detect and understand anomalies in heterogeneous time series with *mixed sampling rates*. In this paper we develop an unsupervised learning approach that harnesses the power of both stacked recurrent autoencoder and statistical Gaussian Mixture Model (GMM) to learn temporal anomalies from heterogeneous time series. The primary advantage of the proposed method is that it can identify subtle but statistically significant abnormal patterns in a large set of highly dynamic and unpredictable signals. In addition, a single detection model trained by the proposed method remains effective robustly for temporal sequences with different sampling rates without any model retraining. Through extensive qualitative and quantitative analysis, we demonstrate that the proposed method outperforms other state-of-the-art methods considerably, attaining a performance improvement of up to 13% in terms of area under curve (AUC) score.

1 Introduction

How can we detect anomalous patterns in a set of heterogeneous network time series with mixed sampling rates? A major application of this technology is to identify anomalous behavior in a set of network heterogeneous time series generated by Internet of Things (IoT) devices. For example, an IoT device connected to your laptop may leak out sensitive information to Internet if its security is compromised, which requires immediate attention and even emergence actions in some cases. As a matter of fact, a recent study conducted by researchers from US and UK reveals that a majority of IoT devices share sensitive information to unrelated third parties [Ren *et al.*, 2019]. More generally, temporal anomaly detection plays a central role in understanding time series IoT data from diverse application areas, including environmental monitoring, network intrusion detection, video analysis, and so on.

A key challenge in analyzing this type of time series lies in the heterogeneity, i.e., the majority of these time series, while being non-anomalous, appear to be quite different from each other. As a motivating example, Figure 1 shows three samples of non-anomalous IoT traffic time series. Apparently, the traffic patterns are highly dynamic and differ significantly from each other. In addition, IoT traffic time series often contains a few short traffic bursts and possibly a long sleep time with no data transmission, as shown in the first sub-figure. Therefore, the unpredictable dynamics and heterogeneity of IoT traffic time series pose significant challenges against effective discovery of anomalous behaviors. Apart from the heterogeneity, other challenges include: 1) such time series, especially those produced by multiple network devices often come with different sampling rates; 2) there exist no anomaly labels for supervised classification; 3) in many applications, we need to not only detect the temporal anomalies but explain their mechanism so that domain expert can understand them and offer suggestions for emergence actions. The above challenges give rise to the following questions.

How can we design an unsupervised machine learning approach to identify unusual behavior in highly dynamic and heterogeneous network time series with mixed sampling rates?

In addition, can we explain the detected anomalies by pinpointing the most anomalous time instances that do not conform to the normal pattern?

To this end, we develop a Bi-temporal compression network (Bitnet) to effectively uncover anomalous patterns within the heterogeneous IoT data stream. Bitnet takes a combined model- and data-driven approach that harnesses the power of both deep autoencoder ensembles and statistical Gaussian Mixture Model (GMM). In addition, it can provide explainable anomaly detection results and be applied to heterogeneous time series with different sampling rates universally without any model retraining. More concretely, we made contributions as listed in the sequel.

*Corresponding Author : Kai Yang(kaiyang@tongji.edu.cn). This work was supported in part by National Natural Science Foundation of China under Grant 61771013, was supported in part by funding from Shenzhen Institute of Artificial and Robotics for Society, was supported in part by the Fundamental Research Funds for the Central Universities of China, and was supported in part by Fundamental Research Funds of Shanghai Jiading District.

- **Unsupervised anomaly detection:** The proposed method is unsupervised and does not require any anomaly labels. The only assumption is that the majority of the time series under investigation is non-anomalous, which holds in most cases in practice.
- **Effectiveness:** The proposed Bitnet outperforms five state-of-the-art methods in terms of the AUC score. Moreover, the proposed method performs robustly well even in the presence of contaminated training data.
- **Universality:** Due to its special structure, a single Bitnet model can be employed to detect anomalies universally across IoT temporal sequences with mixed sample rates without any model retraining.
- **Explainability:** The outcome of Bitnet can be easily visualized and explained in the latent space, which help users to understand the identified anomalies.

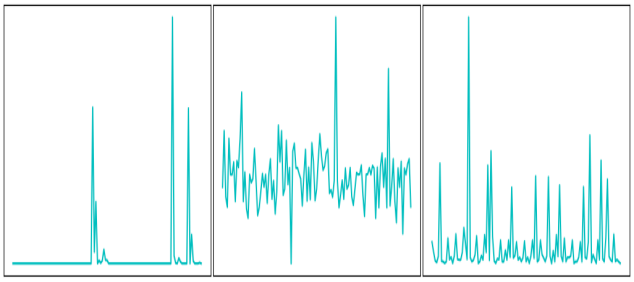

Figure 1: Exemplary heterogeneous time series from the IoT traffic dataset

2 Related Work

Time series anomaly detection can be broadly divided into two groups, i.e., supervised and unsupervised anomaly detection methods.

Supervised anomaly detection methods require a large amount of accurate labels although it can achieve better detection performance. Popular algorithms in this category include Support-Vector Machine (SVM) [Cortes and Vapnik, 1995] and Random Forest (RF) [Breiman, 2001]. However, in practice, imbalanced training data as well as inaccurate labels lead to performance degradation. More importantly, in many cases, no labels can be obtained for anomaly detection, which renders the supervised approach completely inapplicable.

Unsupervised classification-based methods, such as One Class SVM (OCSVM) [Schölkopf *et al.*, 2000] and Deep SVDD [Ruff *et al.*, 2018], train the one class classifier on normal data, and can achieve satisfactory anomaly detection accuracy. Density-based methods [Breunig *et al.*, 2000; Leung and Leckie, 2005] detect anomalies by estimating the density of data points and considering data points samples located in low-density areas as anomalies. This kind of methods are explainable but often ineffective when dealing with high-dimensional time series data. Recently, reconstruction-based method has emerged as a new means for anomaly detection for high-dimensional data, which assumes that the reconstructions of the low-dimensional projections of the anomalies will deviate greatly from the original samples. Principal Component Analysis (PCA) [Wold *et al.*, 1987] is a well-known method in this category, which reconstructs data in a linear way. Autoencoder [Baldi, 2012] detects anomalies through inspecting the reconstruction error. [Zhou *et al.*, 2019] uses multi-layer 1-D convolutional networks to reconstruct data samples, and then utilizes the Generative Adversarial Networks (GANs) to regularize the reconstruction error.

	PCA	OCSVM	AE	DAGMM	BeatGAN	**Bitnet**
Universality						✓
Non-linear		✓	✓	✓	✓	✓
Explainability	✓		✓	✓	✓	✓
Robustness	✓			✓	✓	✓

Table 1: Comparison of related work

However, the performance of reconstruction-based methods is often limited due to the fact that they only conduct anomaly detection analysis based on the reconstruction error. As a remedy, hybrid methods that harness the power of both data-drive approach, i.e., autoencoders and modeldriven approach, i.e., GMM have been proposed. More specially, this combined approach compresses the original data into the latent space and then estimate the distribution of data samples in latent space via an density estimator. Notice that in the proposed framework the reconstruction errors will be concatenated with the latent space representation to obtain the representation in another *extended* latent space. Finally, data points are deemed as anomalies if they are in low-density areas [Zong *et al.*, 2018]. We propose Bitnet to provide an end-to-end approach that combines both the advantages of reconstruction-based model and the density estimation method to detect time series anomalies. The comparison of related work is summarized in Table 1. As evident from the Table, only Bitnet meets all the desired properties.

Informal Problem Definition

Given a collection of heterogeneous time series generated by multiple IoT devices $\mathbf{X} = \{\mathbf{x}_i, i = 1, 2, \cdots, N\}$, where $\mathbf{x}_i$ is a time series, and most of samples are assumed to be normal. The overarching goal is to identify anomalous traffic samples $\mathbf{x}$ that deviate significantly from the normal pattern. In addition, the anomaly detection outcome needs to be interpretable and helps domain experts to understand why they are deemed as anomaly. The primary challenge of this task stems from the heterogeneity and complex dynamics of the IoT heterogeneous time series.

3 Stacked Recurrent Autoencoder Gaussian Mixture Model

3.1 General Framework

In general, the framework for detecting anomalies based on reconstruction-based method and density estimation has two components: reconstruction model and density estimator. The former is used to compress the original data and calculate reconstruction errors, and the latter is used to estimate the distribution of data in the latent space. Data points in the low-density area are then output as anomalies.

The objective function for learning this model is given in the sequel.

$$L = \|\mathbf{X} - g(\mathbf{X})\|_2 + E(\mathbf{Z}|\mathbf{X}) \tag{1}$$

where $\mathbf{X}$ is a matrix concatenating each sample in training set, and $g(\cdot)$ is reconstruction model. $\mathbf{Z}$ is the representation of $\mathbf{X}$ in the latent space. $\|\cdot\|_2$ denotes the Euclidean distance. $E(\cdot)$ is an energy function, which is inversely proportional to the density of the training samples in the latent space. Then, the anomaly score for sample $\mathbf{x}$ can be defined as:

$$Score(\mathbf{x}) = E(\mathbf{z}|\mathbf{x}). \tag{2}$$

The above framework is essentially a combined data- and model-driven approach that encompasses a variety of known anomaly detection schemes. And Bitnet falls into this category as well.

3.2 Model Overview

We leverage an architecture similar to sequence to sequence (seq2seq) [Sutskever *et al.*, 2014] to build the reconstruction model, as shown in Figure 2. In particular, the reconstruction model is composed of two components, i.e., an encoder function mapping time series in the original space to latent space and a decoder function that converts the latent space representation back to the original space, as shown below.

$$\mathbf{z_c} = g_e(\mathbf{x}) \quad \mathbf{x}' = g_d(\mathbf{z_c}), \tag{3}$$

where $g_e(\cdot)$ and $g_d(\cdot)$ denote encoder function and decoder function, respectively. $\mathbf{z_c}$ represents the latent space representation of the time series $\mathbf{x}$. We further stack the latent space representation with the reconstruction errors to obtain the *extended* latent space representation $\mathbf{z}$, as given below.

$$\mathbf{z} = [\mathbf{z_c}, d(\mathbf{x}', \mathbf{x})], \tag{4}$$

where $d(\cdot)$ is a function representing the reconstruction error, including the cosine similarity and relative distance as two components.

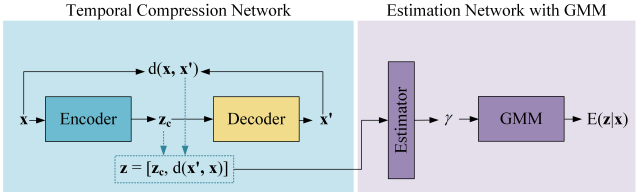

Figure 2: The architecture of basic anomaly detector

Once the extended latent space representation is obtained, it is fed into a GMM estimator for density estimation, as shown in the sequel.

$$\begin{array}{l} \gamma = g_m(\mathbf{z}) \quad \varphi_k = \sum_{i=1}^{M} \frac{\gamma_{i,k}}{M} \\ \mu_k, \Sigma_k = \eta\left(\{[\mathbf{z}_i, \gamma_i]\}_{i=1}^{M}\right) \end{array} \tag{5}$$

where K is the number of mixture components in GMM and M is the number of samples in mixture component k. $g_m(\cdot)$ is a membership estimator, and γ is a K-dimensional vector representing the probability that sample $\mathbf{z}$ belonging to the k-th mixture component. φ_k, μ_k and Σ_k are mixture probability, mean and covariance for k-th mixture component, respectively. $\eta(\cdot)$ denotes a function for computing the mean and covariance.

Once we obtain the parameters of the GMM model, the sample energy function can be calculated as follows,

$$E(\mathbf{z}|\mathbf{x}) = -\log\left(\sum_{k=1}^{K} \varphi_k \frac{\exp(-\frac{1}{2}[\mathbf{z}-\mu_k]^T \Sigma_k^{-1}[\mathbf{z}-\mu_k])}{\sqrt{2\pi\Sigma_k}}\right). \tag{6}$$

The loss function is given by,

$$L = \sum_{i=1}^{N} \|\mathbf{x}_i', \mathbf{x}_i\|_2 + \lambda \sum_{i=1}^{N} E(\mathbf{z}_i|\mathbf{x}_i), \tag{7}$$

where N is the number of training samples, and λ is a weighting parameter that governs the tradeoff between two individual objective functions. $\|\cdot\|_2$ denotes the Euclidean distance between two vectors.

More specifically, the proposed Bitnet is composed of two parallel branches of encoder-decoder structures with GMM, i.e., the **P**arallel compression network (P-compression network) and the **H**ierarchical compression network (H-compression network), as shown in Figure 3. The output of both branches will then be combined to derive the final anomaly score. Such an architecture allows parallel processing that can speed up the training.

P-compression Network

The IoT heterogeneous time series is highly dynamic and there may exist both short-term and long-term temporal dependencies. We thus make use a collection of parallel sparely connected RNNs (S-RNNs) [Kieu *et al.*, 2019] to capture the temporal dynamics at different scales for outlier detection.

Figure 4 shows the architecture of P-compression network. Reconstructed sequence output by i-th decoder $\mathbf{x}'_{(i)}$ is computed as follows.

$$\mathbf{z_c}_i = g_{ei}(\mathbf{x}) \quad \mathbf{x}'_{(i)} = g_{di}(\mathbf{z_c}), \tag{8}$$

where $\mathbf{z_c}_i$ is the final state of i-th encoder. The final state of all encoder branches are used to generate the latent space representation $\mathbf{z_c}$.

$$\mathbf{z_c} = \mathbf{W}^{(P)} \cdot [\mathbf{z_c}_1, \cdots, \mathbf{z_c}_{N_E}]^T + \mathbf{b}^{(P)} \tag{9}$$

where N_E is the number of encoders/decoders. $\mathbf{W}^{(P)}$ and $\mathbf{b}^{(P)}$ denote a trainable weight matrix and a bias vector, respectively.

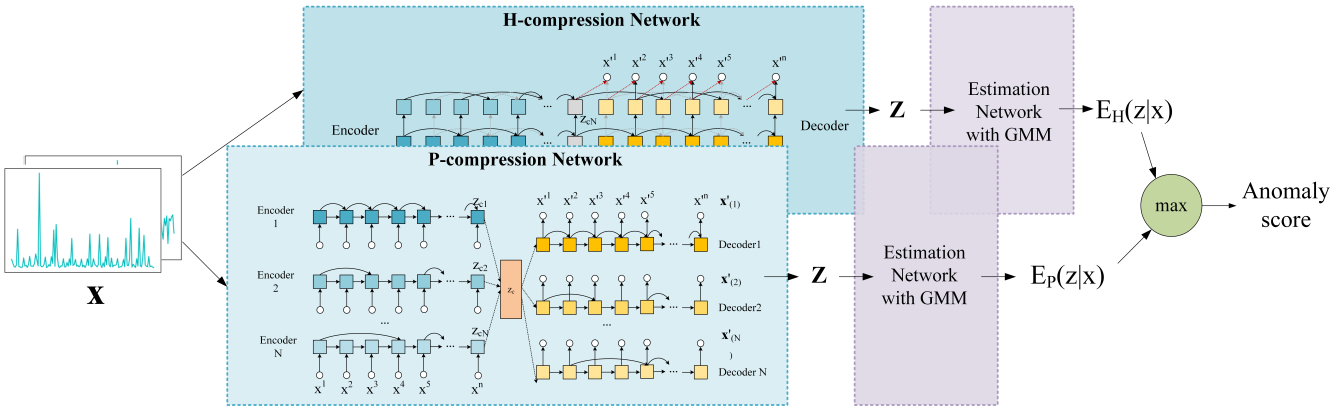

Figure 3: The architecture of Bitnet

Different from the traditional RNN, in which each recurrent unit updates its hidden state as $h^t = f(h^{t-1}, x^t)$, where x^t is the value of $\mathbf{x}$ at time instance t, and $f(\cdot)$ denotes a nonlinear function, typically Long-Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] or Gated Recurrent Unit (GRU) [Chung *et al.*, 2014] is selected. The hidden state of the recurrent unit in S-RNN is updated as follows [Kieu *et al.*, 2019],

$$\begin{aligned} h^t &= w_1 \cdot f(h^{t-1}, x^t) + w_2 \cdot f(h^{t-s}, x^t) \\ s.t. \quad & w_1, w_2 \in \{0, 1\}, w_1 + w_2 \neq 0, \end{aligned} \tag{10}$$

where w_1 and w_2 are randomly generated weights. s is the parameter that controls the memory ability of S-RNN. When s is small, S-RNN tends to learn short-term dependencies. Otherwise, it will learn long-term dependencies. We set the parameter s of each encoder/decoder to a different value, so that it tends to extract dependencies at a particular scale of the time series.

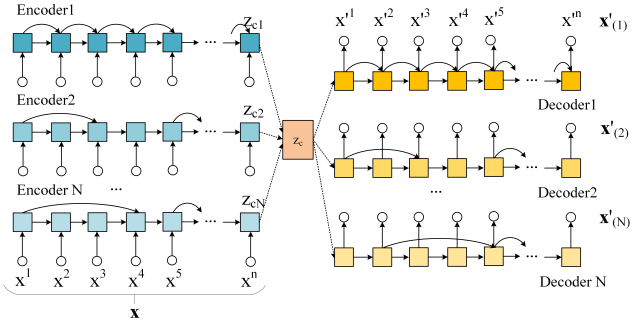

Figure 4: The structure of P-compression network

The extended latent space representation in (4) is then given by $\mathbf{z} = [\mathbf{z_c}, \min_{1 \leq i \leq N_E} d(\mathbf{x}, \mathbf{x}'_{(i)})]$ in this anomaly detector branch. Consequently, we obtain the following loss function for the entire P-compression network.

$$L = \sum_{i=1}^{N} \min_{1 \leq j \leq N_E} \left\| \mathbf{x}_i, \mathbf{x}_i{}'_{(j)} \right\|_2 + \lambda \sum_{i=1}^{N} E(\mathbf{z}_i | \mathbf{x}_i), \tag{11}$$

H-compression Network

We next present H-compression network based on the dilated RNN [Chang *et al.*, 2017], which is essentially a multi-timescale architecture that is capable of learning long-term dependencies among data points in the time series. Figure 5 shows the architecture of H-compression network, where $\mathbf{x}$ and $\mathbf{x}'$ are the original time series and reconstructed time series, respectively. The latent space representation $\mathbf{z_c}$ is computed as follows,

$$\mathbf{z_c} = \mathbf{W}^{(H)} \cdot [\mathbf{z_{c1}}, \cdots, \mathbf{z_{cN_L}}]^T + \mathbf{b}^{(H)}, \tag{12}$$

where z_{ci} represents the final state of i-th layer. N_L is the number of layers in encoder and decoder. $\mathbf{W}^{(H)}$ and $\mathbf{b}^{(H)}$ denote a trainable weight matrix and a bias vector, respectively.

The hidden state of the recurrent unit in dilated RNN is updated as follows,

$$h^t = f(h^{t-d_i}, x^t), \tag{13}$$

where d_i denotes the dilation size in i-th layer. The hidden state at time instance t only depends on the state at $t-d$. Thus, d governs the time scale of the dependency that the network aims to mine. In addition, a multi-layer dilated RNNs are used further extract dependencies at different scales by stacking multiple layers with different dilations. In practice, we set 3 as the dilations in first layer, and an exponential growth strategy is used to set the dilation in subsequent layer, that is, $d_i = 3^i$.

We then generate the extended latent space representation according to (4). The loss function for H-compression network remains the same to (7).

Finally, the anomaly score of the sample x is defined as $\max(E_P(\mathbf{z}|\mathbf{x}), E_H(\mathbf{z}|\mathbf{x}))$, where $E_P(\cdot)$ and $E_H(\cdot)$ denote the energy functions of anomaly detectors containing P-/H-compression network, respectively.

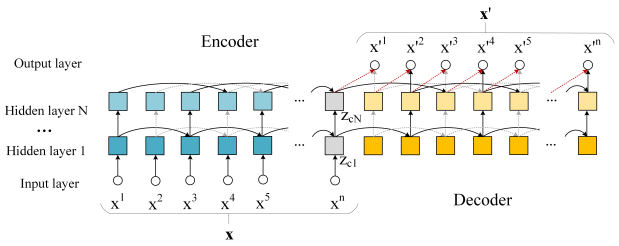

Figure 5: The structure of H-compression network

4 Experiments

4.1 Experimental Setup

Datasets

We use two public real-world traffic datasets, i.e., IoT traffic dataset (IoT Dataset)[1] [Sivanathan *et al.*, 2018] and the cellular traffic dataset (cell dataset)[2]. In IoT dataset, packets generated by all devices are captured and recorded, and each device is identified by a unique MAC address. We split the entire heterogeneous time series into non-overlapping windows, each spanning a time interval of one hundred and twenty minutes. Each data point within the time window represents the number of packets collected within one minute. We label the traffic generated by non-IoT devices, e.g., smart phones and laptops as anomaly [Ortiz *et al.*, 2019; Sivanathan *et al.*, 2017].

The cell dataset is generated from the Call Detail Record (CDR) of a tier-1 operator. The traffic data from a cell is picked to test the generality of the proposed Bitnet structure. Traffic data from another cell is injected into that of the selected cell as anomalies. For both datasets, we apportion them into training and testing datasets with a 40-60 split.

Implementation Details

We conduct the experiments on a work station with 48-core Xeon E5 CPUs, 64GB RAM and 2 Nvidia Titan V GPUs. TensorFlow 1.10 is employed to implement the proposed Bitnet framework.

Bitnet contains four types of parameters, 1) the number of branches in P-compression network; 2) the number of layers in P-compression network; 3) number of neurons in each layer of RNN; 4) number of mixture components in GMM. LSTM cell and GRU cell are adopted in P-compression network and P-compression network respectively. Finally, we use the Adam optimization algorithm [Kingma and Ba, 2014] to train the proposed model, and the learning rate is set to 1e-3. The hyperparameter settings of the two datasets are shown in Table 2. For other algorithms, we follow the default settings.

Performance Metric

AUC (Area under the Receiver Operating Curve) is employed to assess the performance of the proposed algorithm.

The proposed Bitnet is compared with five state-of-the-art unsupervised anomaly detection methods. 1) *OCSVM*; 2) *GRU-ED* [Malhotra *et al.*, 2016]; 3) *Shared-SRNN* [Kieu *et al.*, 2019]; 4) *DAGMM* [Zong *et al.*, 2018]; 5) *BeatGAN* [Zhou *et al.*, 2019].

[1]https://iotanalytics.unsw.edu.au/iottraces.html

[2]https://dandelion.eu/datagems/SpazioDati/telecom-sms-call-internet-mi

	IoT dataset	Cell dataset
# of branches	3	3
# of layers	2	4
# of neurons	18	18
# of GMM components	4	1

Table 2: Hyperparameter Settings

Method	IoT dataset	Cell dataset
OCSVM	0.7947	0.3241
GRU-ED	0.9099	0.4259
Shared-S-RNN	0.8279	0.6944
DAGMM	0.8314	0.5000
BeatGAN	0.5571	0.5029
Bitnet-P	0.9356	0.3982
Bitnet-H	0.8696	0.8333
Bitnet	**0.9503**	**0.8333**
Improvement	**4.04%**	**13.89%**

Table 3: AUC scores

4.2 Experiment Results

We assess the performance of the proposed Bitnet architecture through four aspects, including the accuracy (AUC), the robustness against contaminated training data, the explainability, and the universality, i.e., whether a single Bitnet model can be used to detect anomalous patterns in time series with different sampling rates.

Anomaly Detection

The performance of Bitnet and five existing methods on two real-world datasets are listed in Table 3. Bitnet-P and Bitnet-H represent respectively P-compression network with GMM and H-compression network with GMM.

It is evident from the experiments that the proposed Bitnet outperforms other competing methods considerably on multiple performance metrics. In particular, the parallel architecture of Bitnet allows it to capture the inherent heterogeneous dynamics of IoT heterogeneous time series which consequently gives rise to considerable performance improvement over other state-of-the-art anomaly detection methods. Bitnet also performs best among all methods for the cell dataset, acquiring an AUC improvement of up to 13.89%.

Robustness

In the previous study, we assume all the training dataset constitutes non-anomalous temporal sequences. However, such an assumption does not always hold in practice, since a small portion of the training data might be anomalies. As a remedy, we artificially inject anomalies into the training dataset to check whether we can still obtain an effective anomaly detector in the presence of contaminated training dataset. Please note that the injected anomalies are sampled

IoT dataset		Cell dataset	
Proportions	AUC Score	Proportions	AUC Score
0%	0.9503	0%	0.8333
5%	0.9177	5%	0.8229
10%	0.9071	10%	0.7779
loss	**4.32%**	**loss**	**5.54%**

Table 4: AUC scores with injected anomalies in training set

Sampling Intervals	AUC Score
60 seconds	0.9503
90 seconds	0.8909
120 seconds	0.9078
150 seconds	0.8897

Table 5: AUC scores for IoT time series with different sampling rates

from the real-world dataset, and do not include any anomalies in the testing set [Zong *et al.*, 2018; Zhou *et al.*, 2019; Ruff *et al.*, 2019]. As shown in Table 4, it is seen that the proposed Bitnet architecture remains effective even in the presence 10% of anomalies in the training dataset.

Universality

The diversity of IoT devices makes the traffic patterns extremely heterogeneous. The sampling rates may vary from one IoT device from others. We are naturally led to the following question: can a Bitnet trained by temporal sequences with one sampling rate be applied universally to time series with other sampling rates? To answer this question, we train a Bitnet model based on a training dataset with sampling rate equals to 60 seconds and then apply it to detect anomalous behaviors in time series with sampling rates varying from 60 seconds to 150 seconds. As shown in Table 5, the model remains effective even the sampling rates varies, demonstrating its universality.

5 Discussion on Explainability

The outcome of Bitnet can be visualized and interpreted via the obtained data distribution in the latent space, as elucidated in Figure 6. The latent space representation of time series in the IoT test dataset is projected onto a lower dimensional space for visualization via selecting two or three representative entries from the entire vector representation. Blue and red dots represent respectively normal and anomaly time series. Each cross represents the center of a component distribution of GMM learned from the training dataset. It is seen that anomaly time series are distant from the centers of clusters and thus can be effectively identified.

Please note that the outcome of Bitnet can also be explained by a variety of gradient-based methods, such as gradients $\times$ features [Baehrens *et al.*, 2010; Simonyan *et al.*, 2013], integrated gradients [Sundararajan *et al.*, 2017]. These methods can pinpoint the portions of time series that are instrumental in determining whether it is abnormal.

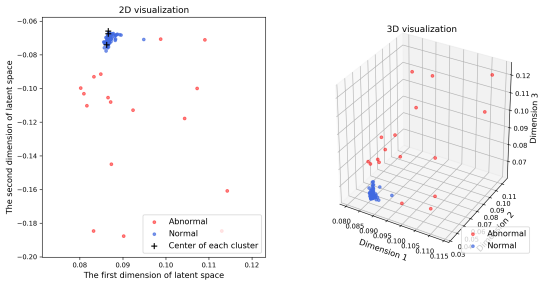

Figure 6: Visualization of the latent space representations of the IoT test dataset, where blue and red dots represent the normal and abnormal time series respectively, each cross represents the center of a cluster.

6 Conclusion

We present Bitnet, an unsupervised machine learning approach to detect anomalous patterns in heterogeneous time series with mixed sampling rates. Through extensive qualitative and quantitative studies, it is revealed that proposed model can successfully identify abnormal patterns within the heterogeneous time series. In addition, it provides a visualization interface to interprete the obtained outcome. Finally, a single Bitnet model can be universally applied to detect time series with a range of different sampling rates.

References

[Baehrens *et al.*, 2010] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

[Baldi, 2012] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49, 2012.

[Breiman, 2001] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Breunig *et al.*, 2000] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. LOF: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[Chang *et al.*, 2017] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 77–87, 2017.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

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

[Kieu *et al.*, 2019] Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S Jensen. Outlier detection for time series with recurrent autoencoder ensembles. In *28th international joint conference on artificial intelligence*, 2019.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Leung and Leckie, 2005] Kingsly Leung and Christopher Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 333–342. Australian Computer Society, Inc., 2005.

[Malhotra *et al.*, 2016] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. LSTM-based encoder-decoder for multi-sensor anomaly detection. In *Proceedings of ICML 2016 Anomaly Detection Workshop, New York, NY, USA*, 2016.

[Ortiz *et al.*, 2019] Jorge Ortiz, Catherine Crawford, and Franck Le. Devicemien: network device behavior modeling for identifying unknown IoT devices. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, pages 106–117, 2019.

[Ren *et al.*, 2019] Jingjing Ren, Daniel J Dubois, David Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. Information exposure from consumer IoT devices: A multidimensional, network-informed measurement approach. In *Proceedings of the Internet Measurement Conference*, pages 267–279, 2019.

[Ruff *et al.*, 2018] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402, 2018.

[Ruff *et al.*, 2019] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694*, 2019.

[Schölkopf *et al.*, 2000] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

[Simonyan *et al.*, 2013] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[Sivanathan *et al.*, 2017] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Characterizing and classifying iot traffic in smart cities and campuses. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WORKSHOPS)*, pages 559–564. IEEE, 2017.

[Sivanathan *et al.*, 2018] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, 2018.

[Sundararajan *et al.*, 2017] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328, 2017.

[Sutskever *et al.*, 2014] I Sutskever, O Vinyals, and QV Le. Sequence to sequence learning with neural networks. *Advances in NIPS*, 2014.

[Wold *et al.*, 1987] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[Zhou *et al.*, 2019] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. BeatGAN: anomalous rhythm detection using adversarially generated time series. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4433–4439. AAAI Press, 2019.

[Zong *et al.*, 2018] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *Proceedings of the International Conference on Learning Representations(ICLR).*, 2018.