

Reward Once, Penalize Once: Rectifying Time Series Anomaly Detection

Keval Doshi

Department of Electrical Engineering
University of South Florida
Tampa, USA
kevaldoshi@usf.edu

Shatha Abudalou

Department of Electrical Engineering
University of South Florida
Tampa, USA
shathaa@usf.edu

Yasin Yilmaz

Department of Electrical Engineering
University of South Florida
Tampa, USA
yasin@usf.edu

Abstract—While anomaly detection in time series has been an active area of research for several years, most recent approaches employ an inadequate evaluation criterion leading to an inflated F1 score. We show that a rudimentary Random Guess method can outperform state-of-the-art detectors in terms of this popular but faulty evaluation criterion. In this work, we propose a proper evaluation metric that measures the timeliness and precision of detecting sequential anomalies. Moreover, most existing approaches are unable to capture temporal features from long sequences. Self-attention based approaches, such as transformers, have been demonstrated to be particularly efficient in capturing long-range dependencies while being computationally efficient during training and inference. We also propose an efficient transformer approach for anomaly detection in time series and extensively evaluate our proposed approach on several popular benchmark datasets.

Index Terms—Time series, anomaly detection, sequential anomalies, self-attention, transformer

I. INTRODUCTION

Time series analysis is used to perform important tasks such as predicting the future values of a variable (e.g., stock market price) and detecting anomalies in sequential data. Time series anomaly detection methods aim to identify abnormal data patterns in temporal data. For instance, in health care, ECG signals are analyzed to determine if the patient suffers from a heart disease [1]. Similarly, in cybersecurity, the data traffic over time in a computer network is monitored to detect cyber-attacks [2]. Time series anomaly detection methods are used in various domains by companies, e.g., Extensible Generic Anomaly Detection System (EGADS) by Yahoo [3] and SR/CNN developed by Microsoft [4].

Time series anomalies are typically classified into two main categories, point anomalies and sequential anomalies. A point anomaly, also known as outlier, is a single data instance with unexpected value with respect to the nominal baseline. In many applications, anomalies continue for a duration with successive anomalous data instances, which is called a sequential anomaly. [5] presents a detailed survey about time series anomaly detection by evaluating twenty different methods based on statistical and deep learning approaches on univariate time series datasets.

Predictive models provide an intuitive way to detect anomalies [17]. The motivation is that a predictive model trained on

	Anomalous Sequence																	
Ground Truth	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
Anomaly Scores	0.5	0.3	0.1	0.1	0.4	0.6	0.2	0.3	0.1	0.2	0.3	0.3	0.9	0.1	0.1	0.2		
Instance-based Evaluation	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0		
Adjusted Instance-based Evaluation	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1		

Fig. 1. The commonly used adjusted instance-based evaluation method. A threshold of 0.5 is applied to an example sequence of anomaly scores produced by a detection algorithm. The traditional instance-based evaluation compares the anomaly/no-anomaly decision for each instance with the ground truth to determine true/false positive/negative decisions. In the recently proposed and widely used adjusted instance-based evaluation, while errors are penalized once as in the traditional evaluation approach, true detections are greatly amplified by considering all instances in an anomalous sequence as multiple true positives if an alarm is raised during the anomalous sequence. This amplification of true positives causes an artificially inflated F1 score (see Table I).

nominal data should give statistically similar prediction error for nominal test data, whereas the prediction error is expected to be larger when encountered with anomalous data. Recurrent Neural Networks (RNN) replaced the classical statistical methods such as Autoregressive (AR) and Autoregressive Moving Average (ARMA) models in many applications. While the early RNN structures could not utilize long-term dependencies in time series data due to the diminishing gradient problem, Long Short-Term Memory (LSTM) network overcome this problem by introducing a more complex memory unit [18]. Recently, the attention mechanism was proposed to improve the predictive performance of LSTM [19]. However, more recently, a completely new deep neural network structure called self-attention, also known as transformer, significantly outperformed the LSTM+attention model to become the state-of-the-art predictive model in many applications. While both the attention and self-attention mechanisms are originally proposed for Natural Language Processing (NLP), they were shown to be effective in various other time series data domains [20], [21].

Evaluating performance for detecting sequential anomalies has been traditionally done in the same way as point anomalies using instance-based detection metrics such as AUC and F1

Dataset	SMD			MSL			SMAP			SWaT			PSM		
Metric	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
OCSVM [6]	44.34	76.72	56.19	59.78	86.87	70.82	53.85	59.07	56.34	45.39	49.22	47.23	62.75	80.89	70.67
IsolationForest [7]	42.31	73.29	53.64	53.94	86.54	66.45	52.39	59.07	55.53	49.29	44.95	47.02	76.09	92.45	83.48
LOF [8]	56.34	39.86	46.68	47.72	85.25	61.18	58.93	56.33	57.60	72.15	65.43	68.62	57.89	90.49	70.61
Deep-SVDD [9]	78.54	79.67	79.10	91.92	76.63	83.58	89.93	56.02	69.04	80.42	84.45	82.39	95.41	86.49	90.73
DAGMM [10]	67.30	49.89	57.30	89.60	63.93	74.62	86.45	56.73	68.51	89.92	57.84	70.40	93.49	70.03	80.08
LSTM-VAE [11]	75.76	90.08	82.30	85.49	79.94	82.62	92.20	67.75	78.10	76.00	89.50	82.20	73.62	89.92	80.96
BeatGAN [12]	72.90	84.09	78.10	89.75	85.42	87.53	92.38	55.85	69.61	64.01	87.46	73.92	90.30	93.84	92.04
OmniAnomaly [13]	83.68	86.82	85.22	89.02	86.37	87.67	92.49	81.99	86.92	81.42	84.30	82.83	88.39	74.46	80.83
InterFusion [14]	87.02	85.43	86.22	81.28	92.70	86.62	89.77	88.52	89.14	80.59	85.58	83.01	83.61	83.45	83.52
THOC [15]	79.76	90.95	84.99	88.45	90.97	89.69	92.06	89.34	90.68	83.94	86.36	85.13	88.14	90.99	89.54
Anomaly Transformer [16]	89.40	95.45	92.33	92.09	95.15	93.59	94.13	99.40	96.69	91.55	96.73	94.07	96.91	98.90	97.89
<i>Random Guess</i>	90.54	96.43	93.39	94.95	98.13	96.51	95.43	97.74	96.48	93.21	99.47	96.24	98.83	98.14	98.48

TABLE I

PERFORMANCE OF *Random Guess* WITH PROBABILITY OF ALARM $p = 0.01$ DOMINATES THE STATE-OF-THE-ART METHODS IN TERMS OF THE COMMONLY USED ADJUSTED INSTANCE-BASED EVALUATION, DEMONSTRATING ITS INHERENT FLAW.

score. However, consecutive anomalous instances are typically caused by the same event, and in real-world applications raising an alarm for such anomalous event is what matters. For instance, after successfully raising alarm for an anomalous event, it is not important in practice to label every anomalous instance. Hence, recent state-of-the-art methods in time series anomaly detection [13]–[16], [22]–[24] used a different performance evaluation system instead of the traditional instance-based evaluation, which is suitable for point anomalies. In this new evaluation system, all instances in an anomalous segment are considered as true positives if a single alarm is raised in the entire segment, as shown in Fig. 1. While this evaluation system rightfully focuses on the detection of anomalous events/sequences, it fails to provide a meaningful metric for sequential anomalies since it still uses the instance-based F1 score for performance evaluation. In this *adjusted instance-based evaluation*, since errors are penalized once, but detection is rewarded generously, it leads to an inflated F1 metric.

To illustrate the serious flaw of the adjusted instance-based F1 metric, let us consider using a *Random Guess* method which randomly (e.g., from a uniform distribution) raises an alarm for each instance with probability p . Note that this *Random Guess* method makes arbitrary decisions independent of the dataset. The probability of raising a true alarm increases with the duration of anomalous sequence. The exact probabilities are given in Section III. Since its each random detection is amplified by the anomalous sequence duration while its errors are penalized once, such a rudimentary approach is able to achieve very high adjusted instance-based F1 score on the popular benchmark datasets and even outperform the state-of-the-art models, as shown in Table I.

Motivated by this limitation, we propose a proper performance metric for sequential anomalies, which also evaluates the timeliness of alarm. Leveraging the high potential of transformer architectures in capturing the long-term dependencies in time series data, we also propose a novel transformer-based

time series anomaly detector. Our contributions in this paper can be summarized as follows:

- A thorough analysis of the inherent flaw in the existing evaluation metric and a proper metric that measures the timeliness and precision of raised alarms.
- A novel end-to-end trained transformer algorithm for time series anomaly detection with asymptotic false alarm rate analysis and closed-form expression for detection threshold.

After discussing related works in Section II and limitations of the existing evaluation system in Section III, we present the proposed performance metric and the detector in Section IV and the experimental results in Section V. The paper is concluded in Section VI.

II. RELATED WORK

The detection of anomalies in time series has been extensively investigated for a long time and remains an active subject due to the great need for more robust methods in complex real-world scenarios [25], [26]. Popular approaches for detecting anomalies in time series data include CNN models [27]–[29], RNN models [30]–[33], and spectral residuals [4], [34]. The most recent approaches use the attention-based mechanisms for time series forecasting [35]–[37].

Self-attention, which is also known as transformer, is an attention-based neural network architecture without the sequential structure. Current state-of-the results in the NLP domain are obtained by transformer models, e.g., GPT-3 [38]. The transformer encoder in [39] is used for unsupervised representation learning of multivariate time series data. It outperforms the state-of-the-art time series classification and regression methods.

The Temporal Fusion Transformer (TFT) presented in [40] is an attention-based model used for multi-horizon forecasting. TFT employs recurrent layers for local processing and interpretable self-attention layers for long-term dependency to learn temporal correlations at various scales. In addition, TFT

curbs any pointless components by using specialized elements that pick the critical characteristics and a succession of gating layers to get significant performance in various applications.

Based on Generative Adversarial Network (GAN), in [41], the authors develop the Adversarial Sparse Transformer, which acts as a generator for learning sparse attention mappings for specific time steps to enhance time series forecasting. Using graph learning with the transformer-based network [42] proposes learning the graph structure for IoT systems to learn sensor dependencies in multivariate time series datasets automatically. Informer is another time series forecasting model based on self-attention, which can be used in anomaly detection [43], [44].

III. PROBLEM FORMULATION

Sequential Anomaly Detection: Consider a time series $\{X_1, \dots, X_t, \dots\}$, which may include anomalous sequences starting and ending at unknown times. Denote the unknown starting time of the i th anomalous sequence with τ_i . Since in real-world applications, such as cybersecurity, surveillance, etc. such anomalous sequences are caused by potentially hazardous anomalous events, it is critical to detect such sequences in a timely manner. Controlling the number of false alarms is also crucial to ensure the reliability of the detection system. Instead of the traditional instance-based evaluation of performance, which is commonly used for point anomalies and other standard machine learning tasks (e.g., classification, regression), the performance of sequential anomaly detection methods should be evaluated in terms of true/false detection of anomalous sequences.

Flaw of Adjusted Instance-based Evaluation: The adjusted instance-based evaluation method has been extensively used in the recent literature (e.g., [13]–[16], [22]–[24]) to compare the state-of-the-art deep learning methods. However, this evaluation method is severely flawed, as illustrated by the high performance of the *Random Guess* algorithm (Table I). Assuming *Random Guess* raises an alarm with probability p , the expected number of false alarms is equal to Np , where N is the number of nominal instances. The probability of raising a true alarm within the i th anomaly sequence of length M_i is given by $1 - \text{Binom}(M_i, 0, p)$, where $\text{Binom}(M_i, 0, p) = (1 - p)^{M_i}$ is the binomial probability mass function for zero success with M_i trials and p success probability. With the adjusted labeling of the entire sequence as true positive in case of a true alarm, the expected number of true positives is equal to $\sum_{i=1} M_i [1 - (1 - p)^{M_i}]$. Consequently, the expected number of false negatives is given by $\sum_{i=1} M_i (1 - p)^{M_i}$. Hence, the expected precision and recall are

$$\begin{aligned} \text{Precision}_{\text{RandomGuess}} &= \frac{\sum_{i=1} M_i [1 - (1 - p)^{M_i}]}{\sum_{i=1} M_i [1 - (1 - p)^{M_i}] + Np} \\ \text{Recall}_{\text{RandomGuess}} &= \frac{\sum_{i=1} M_i [1 - (1 - p)^{M_i}]}{\sum_{i=1} M_i} \end{aligned} \quad (1)$$

As the duration M_i of anomalous sequences increase, the expected number of true positives increases significantly, making both precision and recall approach to 1. Note the insignificant

effect of false positives (Np) since they are penalized once. In the popular benchmark datasets, there are long anomalous sequences lasting thousands of instances, and thus we observe the high precision, recall, and F1 scores in Table I with $p = 0.01$.

IV. PROPOSED APPROACH

In this section, we first present a proper performance metric for sequential anomalies and then our transformer based anomaly detection approach.

A. Performance Evaluation

Sequence Detection Delay: Given τ_i as the starting time of an anomalous event i and $T_i \geq \tau_i$ as the alarm time, we can empirically formulate the average detection delay as

$$\text{ADD} = \frac{1}{S} \sum_{i=1}^S (T_i - \tau_i), \quad (2)$$

where S denotes the number of anomalous events. Since most anomalies indicate critical incidents, it might be essential to detect an anomalous event within a certain time period. Hence, if no alarm is raised within the duration $[\tau_i, \tau_i + \delta_{\max}]$ after anomalous activity i happens, we set the delay to the maximum tolerable delay δ_{\max} . Here, it is important to note that minimizing the detection delay is analogous to the more commonly used objective of maximizing the true positive rate, except it assigns a more specific cost of detection delay $\delta_i = T_i - \tau_i$.

Sequence Alarm Precision: Our second objective emphasizes on maximizing the number of anomalous events being detected with respect to the total number of alarms, similar to the well-known precision metric. However, in contrast to the instance-based precision metric, our metric focuses on the detection of true anomalous *sequences*, and hence only focuses on detecting the anomalous event onset accurately. If an alarm is raised before an alarm even begins, i.e., $T^j \leq \tau_i$, then it is considered as a false alarm.

Empirically, the alarm precision is computed as

$$P = \frac{1}{\hat{S}} \sum_{j=1}^{\hat{S}} 1_{\{T_j \in \cup [\tau_i, \tau_i + \delta_{\max}]\}}, \quad (3)$$

where $1_{\{\cdot\}}$ denotes the indicator function, $\hat{S} = |\{T_j\}|$ is the number of all alarms, and $|\cdot|$ denotes the cardinality of a set.

Sequence Precision Delay: Finally, we present a new metric called *Sequence Precision Delay* that combines the sequence based average detection delay with sequence alarm precision in order to achieve a single metric for easily comparing time series algorithms. The SPD statistic quantifies the area under the Precision vs. normalised ADD (NADD) curve, much like the common AUC metric does for TPR and FPR. To map ADD into $[0, 1]$, we normalize it by the maximum delay, i.e., $\text{NADD} = \text{ADD} / \delta_{\max}$. Mathematically, SPD is given by

$$\text{SPD} = \int_0^1 P(\alpha) d\alpha, \quad (4)$$

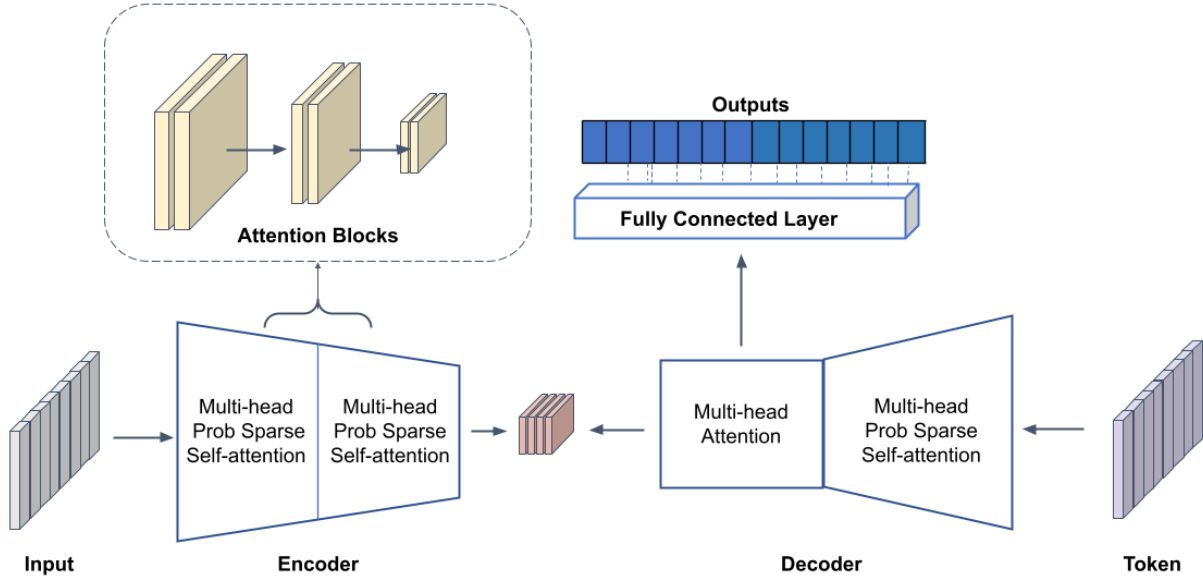


Fig. 2. Proposed TiSAT architecture

where α denotes NADD, and P denotes the precision. A highly successful algorithm with an SPD value close to 1 must have high precision and low delay in its alarms.

Most existing approaches leverage an RNN based model for time series forecasting, and compute the *residuals*, i.e. the prediction or reconstruction error to determine if an observation is anomalous or not. However, it is shown in [43] that RNN based approaches suffer in long sequence time series forecasting. To this end, we propose a novel transformer based approach called *Time Series Anomaly Transformer (TiSAT)*, which is superior in capturing long range temporal dependencies. We next discuss our proposed approach in detail.

B. Time Series Anomaly Transformer (TiSAT)

The overall structure of TiSAT is shown in Fig. 2. The proposed approach utilizes the the ProbSparse mechanism discussed in the Informer architecture [43] as compared to the self-attention mechanism proposed in the Vanilla Transformer [45] for reducing computational complexity. Traditionally, the self attention mechanism for a d -dimensional input is defined as

$$A(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (5)$$

where Q , K and V represent the query, key and value respectively. However, it was recently observed that only a few key and value pairs contribute to the attention score, rendering a majority of the computed dot products as worthless. Hence, we propose using a probabilistic attention mechanism, since it reduces the computational complexity from $\mathcal{O}(L^2)$ to $\mathcal{O}(L \log L)$. Particularly, since only a subset of the query/value tensors require costly operations, ProbSparse attention allows

each key to focus on the most important queries rather than all of them. The ProbSparse attention is given by

$$A(Q, K, V) = \text{Softmax}\left(\frac{\bar{Q}K^T}{\sqrt{d}}\right)V \quad (6)$$

where \bar{Q} is a sparse matrix consisting of the top queries. As shown in [43], the measurement of sparsity between a query q_i and all keys K is given by

$$M(q_i, K) = \log\left(\sum_{j=1}^{L_K} e^{\frac{q_i k_j^T}{\sqrt{d}}}\right) - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j^T}{\sqrt{d}} \quad (7)$$

Encoder: The input to the proposed architecture at a time instance t is a matrix representation of the d -dimensional time series over a sequence of L instances given by \mathcal{Z}^t . Following the dilated convolution approach proposed in [46], [47], we employ a distilling procedure to extract a focused self-attention feature map. The distilling function from the j^{th} layer towards the $(j+1)^{th}$ layer is given by

$$\mathcal{Z}_{j+1}^t = \text{MaxPool ELU}(\text{Conv1d}([\mathcal{Z}_j])) \quad (8)$$

where $[\cdot]$ represents the attention block, which is followed by a 1-D convolution filter with a kernel width of 3. The number of self-attention blocks are progressively decreased and then finally concatenated to form the final representation of the encoder.

Decoder: We leverage the canonical decoder structure proposed by [45] and consists of two similar self-attention modules. The input to the decoder architecture is given by concatenating the start token (\mathcal{Z}_{token}) and a placeholder for the target sequence (\mathcal{Z}_{target}) as follows

$$\mathcal{Z}_{de}^t = \text{Concat}(\mathcal{Z}_{token}, \mathcal{Z}_{target}) \quad (9)$$

This is then passed to a dense fully connected layer. The network is trained using the mean squared error loss by propagating it through the decoder and encoder.

C. Anomaly Detection Framework

We propose an online and non-parametric detection approach for detecting persistent and abrupt anomalies using the transformer output. Due to the sequential (persistent) nature of time series anomalies, we need an approach which accumulates the evidence over time and then makes a decision instead of a hard threshold on the anomaly score for each instant. To this end, we propose using a nonparametric sequential algorithm based on k nearest neighbors (k NN). First, the algorithm trains on a set of nominal historic observations in an offline fashion and then tests the incoming observations until it detects a change in the observations with respect to the nominal baseline. In the training phase, assuming a training set \mathcal{X}_N consisting of N nominal data instances, it randomly partitions \mathcal{X}_N into two sets \mathcal{X}_{N_1} and \mathcal{X}_{N_2} , where $N_1 + N_2 = N$. Then, for each point in \mathcal{X}_{N_1} it finds the Euclidean distance to each point in \mathcal{X}_{N_2} as $\{d_1, d_2, \dots, d_{N_1}\}$. The $(1-\alpha)$ th percentile d_α is used as a baseline statistic during the testing phase, where α is the statistical significance level (e.g., 0.05). During testing, we compute the anomaly evidence for each instance as

$$D_t = d_t^m - d_\alpha^m, \quad (10)$$

where d_t is the k NN distance between the test point at time instance t and \mathcal{X}_{N_2} and m is the dimensionality of the transformer output.

The anomaly evidences are accumulated over time

$$s_t = \max\{s_t + D_t, 0\}, s_0 = 0. \quad (11)$$

and an alarm is raised when the anomaly statistic s_t exceeds a threshold h , i.e., at time

$$T = \min\{t : s_t^k \leq h\}. \quad (12)$$

Here, T is the minimum time required for the accumulated evidence s_t^k to be sufficiently high to raise an alarm based on a detection threshold h . The detection threshold h manifests a trade-off between minimizing the detection delay and minimizing the false alarm.

For an anomaly detection algorithm to be implemented in a practical setting, a clear procedure is necessary for selecting the decision threshold such that it satisfies a desired false alarm rate. The reliability of an algorithm in terms of false alarm rate is crucial for minimizing human involvement. To provide such a performance guarantee for the false alarm rate, we derive an asymptotic upper bound on the average false alarm rate of the proposed algorithm.

Theorem 1: The false alarm rate of the proposed algorithm is asymptotically (as $N_2 \rightarrow \infty$) upper bounded by

$$FAR \leq e^{-\omega_0 h}, \quad (13)$$

where h is the decision threshold, and $\omega_0 > 0$ is given by

$$\omega_0 = v_m - \theta - \frac{1}{\phi} \mathcal{W}(-\phi \theta e^{-\phi \theta}), \quad (14)$$

$$\theta = \frac{v_m}{e^{v_m d_\alpha^m}}.$$

In (14), $\mathcal{W}(\cdot)$ is the Lambert-W function, $v_m = \frac{\pi^{m/2}}{\Gamma(m/2+1)}$ is the constant for the m -dimensional Lebesgue measure (i.e., $v_m d_\alpha^m$ is the m -dimensional volume of the hyperball with radius d_α), and ϕ is the upper bound for D_t .

Proof. In [48][page 177], for CUSUM-like algorithms with independent increments, such as TiSAT with independent D_t , a lower bound on the average false alarm period is given as follows

$$E_\infty[T] \geq e^{\omega_0 h},$$

where h is the detection threshold, and $\omega_0 \geq 0$ is the solution to $E[e^{\omega_0 D_t}] = 1$.

To analyze the false alarm period, we need to consider the nominal case. The anomaly evidence in the nominal case does not necessarily depend on the previous selections due to lack of an anomaly which could correlate the evidences. Hence, in the nominal case, it is safe to assume that D_t is independent over time.

We firstly derive the asymptotic distribution of the instance-level anomaly evidence D_t in the absence of anomalies. Its cumulative distribution function is given by

$$P(D_t \leq y) = P(d_t^m \leq d_\alpha^m + y).$$

It is sufficient to find the probability distribution of d_t^m , the m th power of the k NN distance at time t . As discussed above, we have independent m -dimensional vectors $\{X_t\}$ over time, which form a Poisson point process. The nearest neighbor ($k = 1$) distribution for a Poisson point process is given by

$$P(d_t \leq r) = 1 - \exp(-\Lambda(b(X_t, r)))$$

where $\Lambda(b(X_t, r))$ is the arrival intensity (i.e., Poisson rate measure) in the m -dimensional hypersphere $b(X_t, r)$ centered at X_t with radius r [49]. Asymptotically, for a large number of training instances as $N_2 \rightarrow \infty$, under the null (nominal) hypothesis, d_t of X_t takes small values, defining an infinitesimal hyperball with homogeneous intensity $\lambda = 1$ around X_t . Since for a homogeneous Poisson process the intensity is written as $\Lambda(b(X_t, r)) = \lambda |b(X_t, r)|$ [49], where $|b(X_t, r)| = \frac{\pi^{m/2}}{\Gamma(m/2+1)} r^m = v_m r^m$ is the Lebesgue measure (i.e., m -dimensional volume) of the hyperball $b(X_t, r)$, we rewrite the nearest neighbor distribution as

$$P(d_t \leq r) = 1 - \exp(-v_m r^m),$$

where $v_m = \frac{\pi^{m/2}}{\Gamma(m/2+1)}$ is the constant for the m -dimensional Lebesgue measure. Now, applying a change of variables we

can write the probability density of d_t^m and D_t as

$$f_{d_t^m}(y) = \frac{\partial}{\partial y} [1 - \exp(-v_m y)], \quad (15)$$

$$= v_m \exp(-v_m y), \quad (16)$$

$$f_{D_t}(y) = v_m \exp(-v_m d_\alpha^m) \exp(-v_m y) \quad (17)$$

Using the probability density derived in (17), $E[e^{\omega_0 D_t}] = 1$ can be written as

$$1 = \int_{-d_\alpha^m}^{\phi} e^{\omega_0 y} v_m e^{-v_m d_\alpha^m} e^{-v_m y} dy, \quad (18)$$

$$\frac{e^{v_m d_\alpha^m}}{v_m} = \int_{-d_\alpha^m}^{\phi} e^{(\omega_0 - v_m)y} dy, \quad (19)$$

$$= \frac{e^{(\omega_0 - v_m)y}}{\omega_0 - v_m} \Big|_{-d_\alpha^m}^{\phi}, \quad (20)$$

$$= \frac{e^{(\omega_0 - v_m)\phi} - e^{(\omega_0 - v_m)(-d_\alpha^m)}}{\omega_0 - v_m}, \quad (21)$$

where $-d_\alpha^m$ and ϕ are the lower and upper bounds for $D_t = d_t^m - d_\alpha^m$. The upper bound ϕ is obtained from the training set.

As $N_2 \rightarrow \infty$, since the m th power of $(1 - \alpha)$ th percentile of nearest neighbor distances in training set goes to zero, i.e., $d_\alpha^m \rightarrow 0$, we have

$$e^{(\omega_0 - v_m)\phi} = \frac{e^{v_m d_\alpha^m}}{v_m} (\omega_0 - v_m) + 1. \quad (22)$$

We next rearrange the terms to obtain the form of $e^{\phi x} = a_0(x + \theta)$ where $x = \omega_0 - v_m$, $a_0 = \frac{e^{v_m d_\alpha^m}}{v_m}$, and $\theta = \frac{v_m}{e^{v_m d_\alpha^m}}$. The solution for x is given by the Lambert-W function [50] as $x = -\theta - \frac{1}{\phi} \mathcal{W}(-\phi e^{-\phi\theta}/a_0)$, hence

$$\omega_0 = v_m - \theta - \frac{1}{\phi} \mathcal{W}(-\phi\theta e^{-\phi\theta}). \quad (23)$$

Finally, since the false alarm rate (i.e., frequency) is the inverse of false alarm period $E_\infty[T]$, we have

$$FAR \leq e^{-\omega_0 h},$$

where h is the detection threshold, and ω_0 is given above.

Specifically, v_m is directly computed using the dimensionality m , d_α comes from the training phase, ϕ is also found in training, and finally there is a built-in Lambert-W function in popular programming languages such as Python and Matlab. Hence, given the training data, ω_0 can be easily computed, and based on Theorem 1, the threshold h can be chosen to asymptotically achieve the desired false alarm period as follows

$$h = \frac{-\log(FAR)}{\omega_0} \quad (24)$$

We finally present the comparison between the bound for false alarm rate derived in Theorem 1 and the empirical false alarm period in Fig. 3. The figure depicts the logarithm of false alarm period, which is the inverse of false alarm rate, for clarity. Hence, the upper bound on the false alarm rate

becomes the lower bound on the false alarm period in this scenario.

V. EXPERIMENTS

Datasets: Most of the existing works evaluate their performance on the following datasets:

- **SMD (Server Machine Dataset):** The SMD dataset is collected from a large internet company and consists of data collected over 5 weeks with 38 dimensions.
- **PSM (Pooled Server Metrics):** The PSM dataset is proposed by eBay and consists of 26 dimensional data captured internally from application server nodes.
- **MSL (Mars Science Laboratory rover) and SMAP (Soil Moisture Active Passive satellite):** The MSL and SMAP datasets are provided by NASA and consists of telemetry data and anomalies featuring 55 and 25 dimensions respectively. Since most of the dimensions are categorical, we only focus on the telemetry values.
- **SWaT:** The SWaT dataset is collected in an industrial setting and features data collected from a sewage water treatment facility. The dataset is collected over an entire week and consists of 51 dimensions, where the anomalies are caused due to cyberattacks.

Each dataset includes training, validation and testing subsets. Anomalies are only labeled in the testing subset.

Implementation Details: Following the sliding window approach used in existing works, the input to the proposed TiSAT model is a sub-series with a window size of 100. The TiSAT model encoder consists of a 3-layer stack followed by a 1-layer stack and the decoder consists of a 2-layer stack. We train the model using Adam optimizer, and the learning rate is set as $1e^{-4}$. We train the model for 4 epochs with a batch size of 64. We normalize all the datasets between [0,1] using minmax normalization.

Results: We extensively evaluate the performance of the proposed approach on the five publicly available benchmark datasets using the proposed SPD metric. We compare our approaches with classical algorithms such as OC-SVM, IsolationForest and LOF, as well as time series forecasting based approaches such as ARIMA and LSTM. As shown in Table II, the proposed TiSAT model significantly outperforms all other approaches. While there have been several neural network based approaches [13]–[16], [22]–[24] proposed recently, all of them present their results using the faulty adjusted instance-based metric shown in Fig. 1 (See Table I for the inherent flaw of this metric). Since most of them do not have their codes available, it is not feasible for us to compare with them.

VI. CONCLUSION

In this work we identified a crucial shortcoming in the existing evaluation criterion used by most recent approaches for time series anomaly detection. To rectify the evaluation method, we presented a novel performance metric which measures the timeliness and precision of detection methods. Moreover, we proposed a novel transformer based approach called TiSAT for unsupervised time series anomaly detection

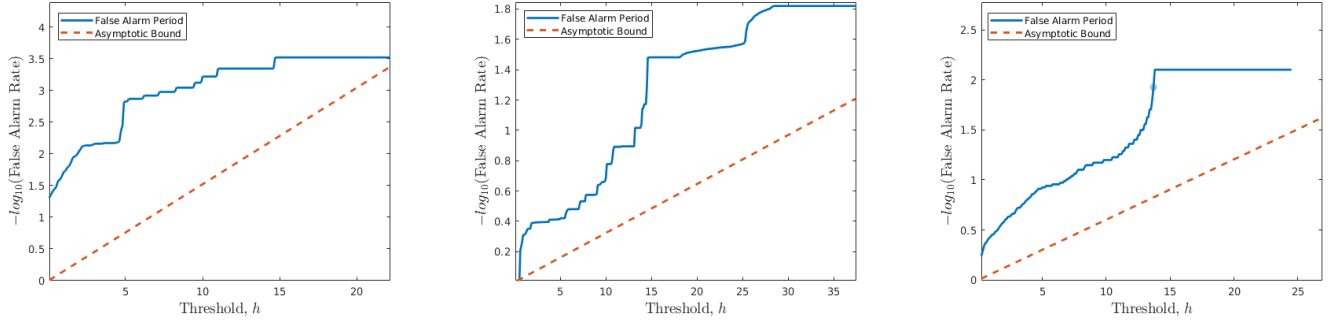


Fig. 3. Actual false alarm periods vs. derived lower bounds for the PSM, SMAP and SWAT datasets respectively.

Dataset Metric (SPD)	SMD	MSL	SMAP	SWaT	PSM
OCSVM [6]	21.08	20.15	12.82	17.21	19.44
IsolationForest [7]	19.38	16.50	9.26	12.83	18.98
LOF [8]	17.25	14.29	18.21	13.68	20.14
ARIMA	27.64	24.13	22.87	29.35	26.47
LSTM [33]	28.31	23.85	27.54	32.97	28.41
Ours	52.70	33.40	29.35	46.23	38.58

TABLE II

COMPARISON WITH EXISTING APPROACHES USING THE PROPOSED SPD METRIC.

and provided an asymptotic false alarm rate analysis for TiSAT. This analysis leads to a closed-form expression for the detection threshold, which was empirically corroborated on benchmark datasets. We comprehensively evaluated the proposed approach and showed that TiSAT is able to achieve state-of-the-art performance on benchmark datasets.

REFERENCES

- [1] S. Chauhan and L. Vig, "Anomaly detection in ecg time signals via deep long short-term memory networks," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–7, IEEE, 2015.
- [2] K. Doshi, Y. Yilmaz, and S. Uludag, "Timely detection and mitigation of stealthy ddos attacks via iot networks," *IEEE Transactions on Dependable and Secure Computing*, 2021.
- [3] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1939–1947, 2015.
- [4] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at microsoft," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3009–3017, 2019.
- [5] M. Braei and S. Wagner, "Anomaly detection in univariate time-series: A survey on the state-of-the-art," *arXiv preprint arXiv:2004.00433*, 2020.
- [6] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.
- [7] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth IEEE international conference on data mining*, pp. 413–422, IEEE, 2008.
- [8] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.
- [9] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*, pp. 4393–4402, PMLR, 2018.
- [10] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.
- [11] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.
- [12] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "Beatgan: Anomalous rhythm detection using adversarially generated time series," in *IJCAI*, pp. 4433–4439, 2019.
- [13] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [14] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, and D. Pei, "Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3220–3230, 2021.
- [15] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13016–13026, 2020.
- [16] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," *arXiv preprint arXiv:2110.02642*, 2021.
- [17] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018.
- [18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.
- [20] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An attentive survey of attention models," *arXiv preprint arXiv:1904.02874*, 2019.
- [21] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Thirty-second AAAI conference on artificial intelligence*, 2018.
- [22] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at microsoft," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3009–3017, 2019.
- [23] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

- [24] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, *et al.*, “Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications,” in *Proceedings of the 2018 world wide web conference*, pp. 187–196, 2018.
- [25] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.
- [26] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.
- [27] T. Wen and R. Keyes, “Time series anomaly detection using convolutional neural networks and transfer learning,” *arXiv preprint arXiv:1905.13628*, 2019.
- [28] M. Canizo, I. Triguero, A. Conde, and E. Onieva, “Multi-head cnn-rnn for multi-time series anomaly detection: An industrial case study,” *Neurocomputing*, vol. 363, pp. 246–260, 2019.
- [29] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, “Deepant: A deep learning approach for unsupervised anomaly detection in time series,” *Ieee Access*, vol. 7, pp. 1991–2005, 2018.
- [30] L. Bontemps, V. L. Cao, J. McDermott, and N.-A. Le-Khac, “Collective anomaly detection based on long short-term memory recurrent neural networks,” in *International conference on future data and security engineering*, pp. 141–152, Springer, 2016.
- [31] L. Zhu and N. Laptev, “Deep and confident prediction for time series at uber,” in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 103–110, IEEE, 2017.
- [32] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, “Time-series extreme event forecasting with neural networks at uber,” in *International conference on machine learning*, vol. 34, pp. 1–5, 2017.
- [33] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, “Lstm-based encoder-decoder for multi-sensor anomaly detection,” *arXiv preprint arXiv:1607.00148*, 2016.
- [34] S.-V. Oprea, A. Băra, F. C. Puican, and I. C. Radu, “Anomaly detection with machine learning algorithms and big data in electricity consumption,” *Sustainability*, vol. 13, no. 19, p. 10963, 2021.
- [35] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, “Geoman: Multi-level attention networks for geo-sensory time series prediction,” in *IJCAI*, vol. 2018, pp. 3428–3434, 2018.
- [36] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction,” *arXiv preprint arXiv:1704.02971*, 2017.
- [37] K. Zhou, W. Wang, T. Hu, and K. Deng, “Time series forecasting and classification models based on recurrent with attention mechanism and generative adversarial networks,” *Sensors*, vol. 20, no. 24, p. 7211, 2020.
- [38] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165*, 2020.
- [39] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, “A transformer-based framework for multivariate time series representation learning,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2114–2124, 2021.
- [40] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *International Journal of Forecasting*, 2021.
- [41] S. Wu, X. Xiao, Q. Ding, P. Zhao, W. Ying, and J. Huang, “Adversarial sparse transformer for time series forecasting,” 2020.
- [42] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, “Learning graph structures with transformer for multivariate time series anomaly detection in iot,” *IEEE Internet of Things Journal*, 2021.
- [43] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in *Proceedings of AAAI*, 2021.
- [44] L. Guo, R. Li, and B. Jiang, “A data-driven long time-series electrical line trip fault prediction method using an improved stacked-informer network,” *Sensors*, vol. 21, no. 13, p. 4466, 2021.
- [45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, pp. 5998–6008, 2017.
- [46] F. Yu, V. Koltun, and T. Funkhouser, “Dilated residual networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 472–480, 2017.
- [47] A. Gupta and A. M. Rush, “Dilated convolutions for modeling long-distance genomic dependencies,” *arXiv preprint arXiv:1710.01278*, 2017.
- [48] M. Basseville and I. Nikiforov, *Detection of abrupt changes: theory and application*, vol. 104. Prentice Hall, Englewood Cliffs, 1993.
- [49] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic geometry and its applications*. John Wiley & Sons, 2013.
- [50] T. C. Scott, G. Fee, and J. Grotendorst, “Asymptotic series of generalized lambert w function,” *ACM Communications in Computer Algebra*, vol. 47, no. 3/4, pp. 75–83, 2014.