quantile-Long Short Term Memory: A Robust, Time Series Anomaly Detection Method

Snehanshu Saha, *Senior Member, IEEE,* Jyotirmoy Sarkar, Soma Dhavala, *Member, IEEE,* Preyank Mota, and Santonu Sarkar, *Member, IEEE*

***Abstract*—Anomalies refer to the departure of systems and devices from their normal behaviour in standard operating conditions. An anomaly in an industrial device can indicate an upcoming failure, often in the temporal direction. In this paper, we make two contributions: 1) we estimate conditional quantiles in the popular Long Short Term Memory networks (LSTM) architecture, propose a novel anomaly detection method, qLSTM, and consider three different ways to define anomalies based on the estimated quantiles. 2) we use a new learnable activation function (AF), Parametric Elliot Function (PEF), in qLSTM architecture to model temporal long-range dependency. Unlike *sigmoid* and *tanh*, the derivative of the PEF depends on the input as well as on the parameter, which help in mitigating the vanishing gradient problem and therefore facilitates in escaping early saturation. The proposed algorithms are compared with other well-known anomaly detection algorithms, such as Isolation Forest (iForest), Elliptic Envelope, Autoencoder, and modern Deep Learning models such as Deep Autoencoding Gaussian Mixture Model (DAGMM), Generative Adversarial Networks (GAN). The algorithms are evaluated using various performance metrics, such as Precision and Recall. The algorithms have been tested on multiple industrial time-series datasets such as Yahoo, AWS, GE, and machine sensors. We have found that the LSTM-based quantile algorithms are very effective and outperformed the existing algorithms in identifying anomalies.**

***Impact Statement*—Though anomalies happen rarely, when they occur, the impact is catastrophic on industrial systems. While detecting anomalies using machine learning is the only viable option, traditional supervised, semi-supervised and unsupervised approaches suffer from class imbalance, non-availability of data labeling, and fine-tuning modeling parameters for datasets across domains. Our quantile-based LSTM approach, q-LSTM, does not assume any underlying distribution of the data. q-LSTM trains a model on a normal dataset without any labeling of anomalies. q-LSTM detects singleton anomaly precisely, while some, recent state of the art algorithms require at least two anomalies to be effective. Further, a proposal of a parameterized Elliot function, PEF, as an activation function aids in understanding the patterns in data. PEF does not saturate easily, and the parameter is learned through back-propagation without manual tuning. As a result, the proposed method can handle unlabelled data, remains unaffected due to class imbalance, accommodates the variances better without manual retuning, and outperforms many state-of-the-art algorithms.**

***Index Terms*—Anomaly, LSTM, Quantile, Parametric Activation.**

I. Introduction

Anomalies indicate a departure of a system from its normal behaviour. In Industrial systems, they often lead to failures. By definition, anomalies are rare events. As a result, from a Machine Learning standpoint, collecting and classifying anomalies pose significant challenges. For example, when anomaly detection is posed as a classification problem, it leads to extreme class imbalance (data paucity problem). Though several current approaches use semi-supervised neural networks to detect anomalies [1], [2], these approaches still require some labeled data. In the recent past, there have been approaches that attempt to model normal datasets and consider any deviation from the normal as an anomaly. For instance, an autoencoder-based family of models [3] use some form of thresholds to detect anomalies. Another class of approaches relied on reconstruction errors [4], as an anomaly score. If the reconstruction error of a datapoint is higher than a threshold, then the datapoint is declared as an anomaly. However, the threshold value can be specific to the domain and the model, and deciding the threshold on the reconstruction error can be cumbersome.

In this paper, we have introduced the notion of *quantiles* in multiple versions of the LSTM-based anomaly detector. Our proposed approach is principled on:
a. training models on a normal dataset
b. modeling temporal dependency
c. proposing an adaptive solution that does not require manual tuning of the activation

Since our proposed model tries to capture the normal behavior of an industrial device, it does not require any expensive dataset labeling. Our approach also does not require re-tuning of threshold values across multiple domains and datasets. We have exhibited through empirical results later in the paper (see Table 2 of the supplementary file) that the distributional variance does not impact the prediction quality. Our contributions are three folds:
(1) Introduction of *quantiles*, free from the assumptions on data distributions, in the design of quantile-based LSTM techniques and their application in anomaly identification.
(2) Proposal of the *Parameterized Elliot* as a 'flexible-form, adaptive, learnable' activation function in LSTM, where the parameter is learned from the dataset. Therefore, it does not require any manual retuning when the nature of the dataset changes. We have shown empirically that the modified LSTM architecture with PEF performed better than the Elliot Function (EF) and showed that such behavior might be attributed to the slower saturation rate of PEF.
(3) Demonstration of *superior performance* of the proposed LSTM methods over state-of-the-art (SoTA) deep learning (Autoencoder [5], DAGMM [6], DevNet [7], Deep Quantile Regression [8]) and non-deep learning algorithms (iForest [9], Elliptic envelope [10])

The rest of the paper is organized as follows. The proposal

and discussion of various LSTM-based algorithms are presented in section II. Section III describes the LSTM structure and introduces the PEF explaining the intuition behind choosing a parameterized version of the AF and better variability. Experimental results are presented in section IV. Section V discusses relevant literature in anomaly detection. We conclude the paper in section VI.

II. Anomaly detection with Quantile LSTMs

Since *distribution independent* and *domain independent* anomaly detection are the two key motivations behind this work, we borrow the concept of quantiles from Descriptive and Inferential Statistics to address this challenge.

A. Why Quantile based approach?

Quantiles are a robust alternative to classical conditional means in Econometrics and Statistics [11]. In a previous work, Tambwekar et.al.[[12] extended the notion of conditional quantiles to the binary classification setting, allowing for quantification of the uncertainty in the predictions and providing interpretations into the functions learnt by the models via a new loss called binary quantile regression loss (sBQC). The estimated quantiles are leveraged to obtain individualized confidence scores that accurately measure a misclassified prediction. Since quantiles are a natural choice to quantify uncertainty, they are a natural candidate for anomaly detection. However, to the best of our knowledge, the quantile based method has not been used for anomaly detection, however natural it seems.

Empirically, if the data being analyzed are not actually distributed according to an assumed distribution, or if there are other potential sources for anomalies that are far removed from the mean, then quantiles may be more useful descriptive statistics than means and other moment-related statistics. Quantiles can be used to identify probabilities of the range of normal data instances, such that data lying outside the defined range can be conveniently identified as anomalies.

The important aspect of distribution-free anomaly detection is the anomaly threshold being agnostic to the data from different domains. Simply stated, once a threshold is set (in our case, 10-90), we don't need to tune the threshold in order to detect anomalous instances for different data sets. Quantiles allow the use of distributions for many practical purposes, including looking for confidence intervals. A quantile divides a probability distribution into areas of equal probability, i.e., quantiles offer us the chance to quantify chances that a given parameter is inside a specified range of values. This allows us to determine the confidence level of an event (anomaly) actually occurring.

Though the mean of a distribution is useful when it is symmetric, there is no guarantee that actual data distributions are symmetric. If there are potential sources for anomalies are far removed from the mean, then medians are more robust than means, particularly in skewed and heavy-tailed data. It is well known that quantiles minimize check loss [13], a generalized version of Mean Absolute Error (MAE) arising from medians rather than means. Thus, quantiles have less susceptibility to long-tailed distributions and outliers in comparison to mean [14].

Therefore, it makes practical sense to investigate the power of quantiles in detecting anomalies in data distributions. Unlike the methods for anomaly detection in the literature, our proposed quantile-based thresholds applied in the quantile-LSTM are generic and not specific to the domain or dataset. The need to isolate anomalies from the underlying distribution is significant since it allows us to detect anomalies irrespective of the assumptions on the underlying data distribution. We have introduced the notion of quantiles in multiple versions of the LSTM-based anomaly detector in this paper, namely (i) quantile-LSTM (ii) iqr-LSTM and (iii) Median-LSTM. All the LSTM versions are based on estimating the quantiles instead of the mean behaviour of an industrial device. Note that the median is 50% quantile.

B. Various quantile-LSTM Algorithms

Before we discuss quantile-based anomaly detection, we describe the data structure and processing setup, with some notations. Let us consider $x_i, i = 1, 2, .., n$ be the n time-series training datapoints. We consider $T_k = \{x_i : i = k, \cdots, k+t\}$ be the set of t datapoints, and let T_k be split into w disjoint windows with each window of integer size $m = \frac{t}{w}$ and $T_k = \{T_k^1, \cdots, T_k^w\}$. Here, $T_k^j = \{x_{k+m(j-1)}, ..., x_{k+m(j)-1}\}$. In Figure 1, we show the sliding characteristics of the proposed algorithm on a hypothetical dataset, with $t = 9, m = 3$. Let $Q_\tau(D)$ be the sample quantile of the datapoints in the set D. The training data consists of, for every T_k, $X_{k,\tau} \equiv \{Q_\tau(T_k^j)\}, j = 1, \cdots, w$ as predictors with $y_{k,\tau} \equiv Q_\tau(T_{k+1})$, sample quantile at a future time-step, as the label or response. Let $\hat{y}_{k,\tau}$ be the predicted value by an LSTM model.

A general recipe we are proposing to detect anomalies is to: (i) estimate quantile $Q_\tau(x_{k+t+1})$ with $\tau \in (0, 1)$ and (ii) define a statistic that measures the outlier-ness of the data, given the observation x_{k+t+1}. Instead of using predefined thresholds, our approach makes the thresholds adaptive i.e. they change at every time-point depending on quantiles.

1) quantile-LSTM: As the name suggests, in quantile-LSTM, we forecast two quantiles q_{low} and q_{high} to detect the anomalies present in a dataset. We assume the next quantile values of the time period after sliding the time period by one position are dependent on the quantile values of the current time period.

It is further expected that the nominal range of the data can be gleaned from q_{low} and q_{high}. Using these q_{low} and q_{high} values of the current time windows, we can forecast q_{low} and q_{high} values of the next time period after sliding by one position. Here, it is required to build two LSTM models, one for q_{low} (LSTM$_{qlow}$) and another for q_{high} (LSTM$_{qhigh}$). Let's take the hypothetical dataset as a training set from Figure 2a. It has three-time windows from time period $x_1 \cdots x_9$. Table 1 defines the three-time windows of the time period $x_1 \cdots x_9$ and the corresponding q_{low}, q_{high} values against the time window.

The size of the inputs to the LSTM depends on the number of time windows w and one output. Since three time windows

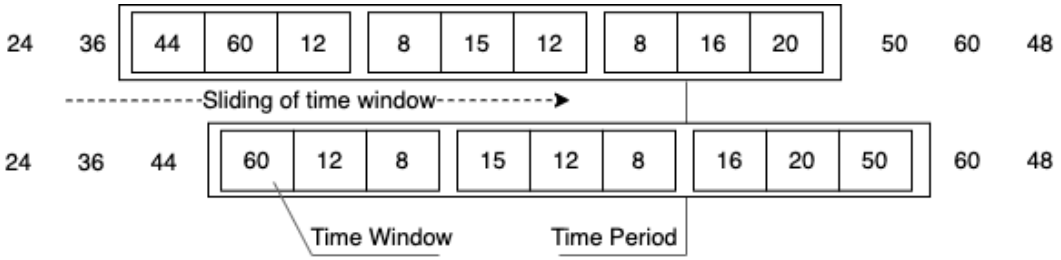
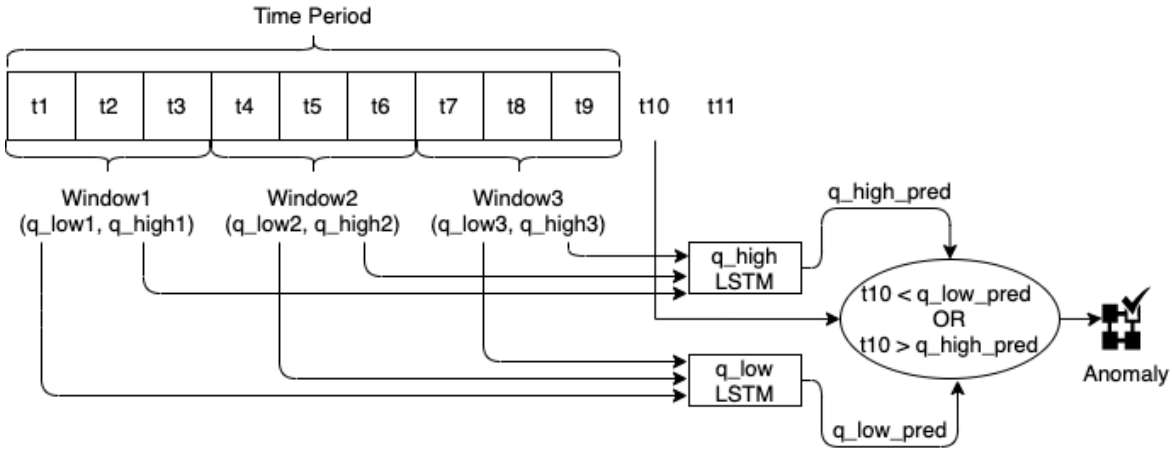
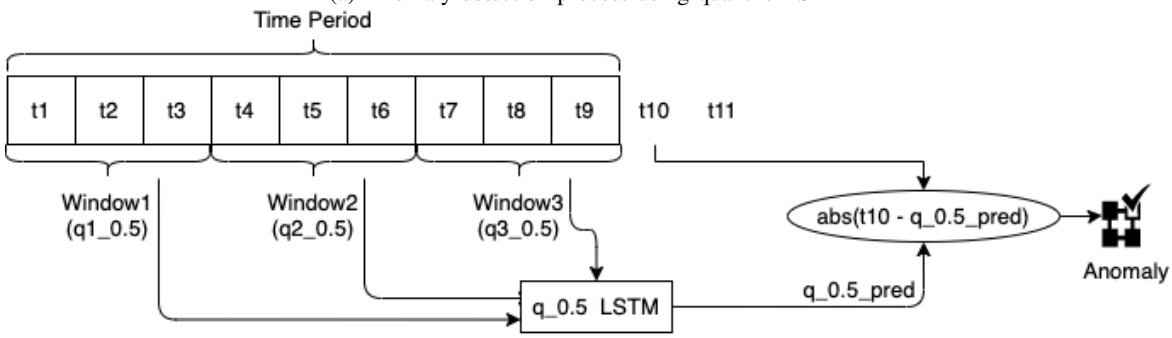


Fig. 1: Sliding movement of a time period



(a) Anomaly detection process using quantile-LSTM



(b) Anomaly detection process using median-LSTM

Fig. 2: Sigmoid function has been applied as a recurrent function, which is applied on the outcome of the forget gate ($f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$) as well as input gate ($i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$). PEF decides the information to store in cell $\acute{c}_t = PEF(W_c * [h_{t-1}, x_t] + b_c)$.

TW	q_{low}	q_{high}
x_1, x_2, x_3	$X_{1,low} \equiv Q_{low}(T_1^1)$	$X_{1,high} \equiv Q_{high}(T_1^1)$
x_4, x_5, x_6	$X_{2,low} \equiv Q_{low}(T_1^2)$	$X_{2,high} \equiv Q_{high}(T_1^2)$
x_7, x_8, x_9	$X_{3,low} \equiv Q_{low}(T_1^3)$	$X_{3,high} \equiv Q_{high}(T_1^3)$

TABLE 1: The first time period and its corresponding time windows

have been considered for a time period in this example, both the LSTM models will have three inputs and one output. For example, the LSTM predicting the lower quantile would have $X_{1,low}$, $X_{2,low}$, $X_{3,low}$ as its inputs and $y_{1,low}$ as its output, for one time-period. A total of $n - t + 1$ instances will be available for training the LSTM models, assuming no missing values.

After building the LSTM models, for each time period it predicts the corresponding quantile value and slides one position to the next time period on the test dataset. quantile-LSTM applies a following anomaly identification approach. If the observed value x_{k+t+1} falls outside of the predicted (q_{low}, q_{high}), then the observation will be declared as an anomaly. For example, the observed value x_{10} will be detected as an anomaly if $x_{10} < \hat{y}_{1,low}$ or $x_{10} > \hat{y}_{1,high}$. Figure 2a illustrates the anomaly identification technique of the quantile-LSTM on a hypothetical test dataset.

2) IQR-LSTM: iqr-LSTM is a special case of quantile-LSTM where q_{low} is 0.25 and q_{high} is the 0.75 quantile. In addition, another LSTM model predicts median $q_{0.5}$ as well. Effectively, at every time index k, three predictions are made $\hat{y}_{k,0.25}, \hat{y}_{k,0.5}, \hat{y}_{k,0.75}$. Based on this, we define the

Inter Quartile Range (IQR) $\hat{y}_{k,0.75} - \hat{y}_{k,0.25}$. Using IQR, the following rule identifies an anomaly when $x_{t+k+1} > \hat{y}_{k,0.5} + \alpha(\hat{y}_{k,0.75} - \hat{y}_{k,0.25})$ or $x_{t+k+1} < \hat{y}_{k,0.5} - \alpha(\hat{y}_{k,0.75} - \hat{y}_{k,0.25})$

3) Median-LSTM: Median-LSTM, unlike quantile-LSTM, does not identify the range of the normal datapoints; rather, based on a single LSTM, the distance between the observed value and predicted median ($x_{t+k+1} - \hat{y}_{k,0.5}$) is computed, as depicted in Figure 2b, and running statistics are computed on this derived data stream. The training set preparation is similar to quantile-LSTM.

To detect the anomalies, Median-LSTM uses an implicit adaptive threshold. Having a single threshold value for the entire time series dataset is unreasonable when it exhibits seasonality and trends. We introduce some notations to make the description concrete. Adopting the same conventions introduced before, define $d_k \equiv x_{t+k+1} - Q_{0.5}(T_{k+1}), k = 1, 2, \ldots, n-t$ and partition the difference series into s sets of size t each, i.e., $D \equiv D_p, p = 1, \ldots, s$, where $D_p = \{d_i : i = (s-1)t + 1, \ldots, st\}$. After computing the differences on the entire dataset, for every window D_p, mean (μ_p) and standard deviation (σ_p) for the individual time period D_p. As a result, μ_p and σ_p will differ from one time period to another time period. Median-LSTM detects the anomalies using upper threshold and lower threshold parameters of a particular time period D_p and they are computed as follows:

$$T_{p,lower} = \mu_p + w\sigma_p; T_{p,higher} = \mu_p - w\sigma_p$$

An anomaly can be flagged for $d_k \in T_p$ when either $d_k > T_{p,higher}$ or $d_k < T_{p,lower}$ Now, what should be the probable value for w? If we consider $w = 2$, any datapoint beyond two standard deviations away from the mean on either side will be considered an anomaly. It is based on the intuition that differences of the normal datapoints should be close to the mean value, whereas the anomalous differences will be far from the mean value. Hence 95.45% datapoints are within two standard deviations distance from the mean value. It is imperative to consider $w = 2$ since there is a higher probability of the anomalies falling into the 4.55% datapoints. We can consider $w = 3$ too where 99.7% datapoints are within three standard deviations. However, it may miss the border anomalies, which are relatively close to the normal datapoints and only can detect the prominent anomalies. Therefore we have used $w = 2$ across the experiments.

C. Probability Bound

In this subsection, we analyze different datasets by computing the probability of occurrence of anomalies using the quantile approach. We have considered 0.1, 0.25, 0.75, 0.9, and 0.95 quantiles and computed the probability of anomalies beyond these values, as shown in Table 1 of supplementary file. The multivariate datasets are not considered since every feature may follow a different quantile threshold. Hence, it is impossible to derive a single quantile threshold for all the features. It is evident from Table 1 of the supplementary file that the probability of a datapoint being an anomaly is high if the datapoint's quantile value is either higher than 0.9 or lower than 0.1. However, if we increase the threshold to 0.95, the probability becomes 0 across the datasets. This emphasizes that a higher quantile threshold does not detect anomalies. It is required to identify the appropriate threshold value, and it is apparent from the table that most of the anomalies are near 0.9 and 0.1 quantile values. Table 1 of the supplementary file also demonstrates the different nature of the anomalies present in the datasets. For instance, the anomalies of Yahoo Dataset$_1$ to Yahoo Dataset$_6$ are present near the quantile value 0.9, whereas the anomalies in Yahoo Dataset$_7$ to Yahoo Dataset$_9$ are close to both quantile values 0.9 and 0.1. Therefore, it is possible to detect anomalies by two extreme quantile values. We can consider these extreme quantile values as higher and lower quantile thresholds and derive a lemma. We provide proof in the supplementary file.

Lemma 1: For an univariate dataset $\mathcal{D}$, the probability of an anomaly $\mathcal{P}(\mathcal{A}) = \mathcal{P}(\mathcal{E} > \alpha_{high}) + \mathcal{P}(\mathcal{F} < \alpha_{low})$, where $\alpha_{high}, \alpha_{low}$ are the higher and lower level quantile thresholds respectively.

The lemma entails that anomalies are trapped outside the high and low quantile threshold values. The bound is independent of data distribution as quantiles assume nominal distributional characteristics.

III. LSTM with Parameterized Elliot Activation (PEF)

We introduce the novel parameterized Elliot activation function PEF, an adaptive variant of usual activation, wherein we modify the LSTM architecture by replacing the activation function of the LSTM gates with PEF as follows.

A single LSTM block comprises four major components: an input gate, a forget gate, an output gate, and a cell state. We have applied the parameterized Elliot Function (PEF) as activation.

A. Parameterized Elliot Function PEF

PEF is represented by

$$f(x) = \frac{\alpha x}{1 + |x|} \tag{1}$$

with the first order derivative of PEF as: $f'(x) = \frac{\alpha}{(|x|+1)^2}$. The function is equal to 0, and the derivative is equal to the parameter α at the origin. After introducing the PEF, the hidden state equation is:$h_t = O_t \alpha_c PEF(C_t)$. By chain rule,

$$\frac{\partial J}{\partial \alpha_c} = \frac{\partial J}{\partial \alpha_c} = \frac{\partial J}{\partial h_t} O_t * Elliot(C_t)$$

. After each iteration, the α_c is updated by gradient descent $\alpha_c^{(n+1)} = \alpha_c^n + \delta * \frac{\partial J}{\partial \alpha_c}$ (See section 3 of the supplementary file for backpropagation of LSTM with PEF). Salient features of the PEF are:

1) The α in Eq.1 is learned during the back-propagation like other weight parameters of the LSTM model. Hence, this parameter, which controls the shape of the activation, is learned from data. Thus, if the dataset changes, so does the final form of the activation, which saves the "parameter tuning" effort.
2) The cost of saturation of standard activation functions impedes training and prediction, which is an important

barrier to overcome. While the PEF derivative also saturates as the $|x|$ increases, the saturation rate is less than other activation functions, such as $\tanh$, $sigmoid$.

3) PEF further decreases the rate of saturation in comparison to the non-parameterized Elliot function.

To the best of our knowledge, insights on 'learning' the parameters of an activation function are not available in literature except for the standard smoothness or saturation properties activation functions are supposed to possess. It is, therefore, worthwhile to investigate the possibilities of learning an activation function within a framework or architecture that uses the inherent patterns and variances from data.

B. PEF saturation

The derivative of the PEF is represented by: $= \frac{\alpha}{x^2} EF^2$. While the derivatives of the sigmoid and tanh are dependent on x, PEF is dependent on both α and x. Even if $\frac{EF^2(x)}{x^2}$ saturates, the learned parameter α will help the PEF escape saturation. The derivatives of the sigmoid, tanh, saturate when $x > 5$ or $x < -5$. However, it is not true with PEF, as evident from fig 3a. As empirical evidence, the layer values for every epoch of the model are captured using various activation functions like PEF, sigmoid and tanh. It is observed that, after about 10 epochs, the values of the layers become more or less constant for sigmoid and tanh (fig 3c and fig 3d), indicating their values have already saturated whereas, for PEF, variation can be seen till it reaches 50 epochs (fig 3b). This shows that compared to sigmoid and tanh as activation functions, PEF escapes saturation due to its learned parameter α. *The parameter α in PEF* changes its value as the model trains over the training dataset while using PEF as the activation function. Since it is a self-training parameter, it returns different values for different datasets at the end of training. These values have been documented in Table 2. Table 2 demonstrates the variations in α values across multiple datasets as these values get updated.

Dataset	α after training	α initial value
AWS Dataset$_1$	1.612	0.1
AWS Dataset$_2$	0.895	0.1
AWS Dataset$_3$	1.554	0.1
AWS DatasetSyn$_1$	1.537	0.1
AWS DatasetSyn$_2$	0.680	0.1
AWS DatasetSyn$_3$	1.516	0.1
Yahoo Dataset$_1$	1.432	0.1
Yahoo Dataset$_2$	1.470	0.1
Yahoo Dataset$_3$	1.658	0.1
Yahoo Dataset$_5$	1.686	0.1
Yahoo Dataset$_6$	1.698	0.1
Yahoo Dataset$_7$	1.725	0.1
Yahoo Dataset$_8$	1.850	0.1
Yahoo Dataset$_9$	1.640	0.1

TABLE 2: Different α values for each Dataset after the training.

IV. Experiment

In this section, we evaluate the performance of quantile-LSTM techniques on multiple datasets. We have identified multiple baseline methods, such as iForest, Elliptic envelope, Autoencoder and several deep learning based approaches for comparison purposes (See section V for more details on baseline methods). [1]

Hardware and Software Considerations: The experiments are run on Windows 10 Enterprise Edition (64 bit) on a laptop with the following configuration: 11th Gen Intel(R) Core(TM) i7 3GHz processor with 16GB RAM and NVIDIA GeForce MX130. This was sufficient to support the software environment of Pycharm community edition 2019.3.2, Python 3.8, scikit-learn 1.1.3, Torch 1.8 and Tensorflow 2.10.

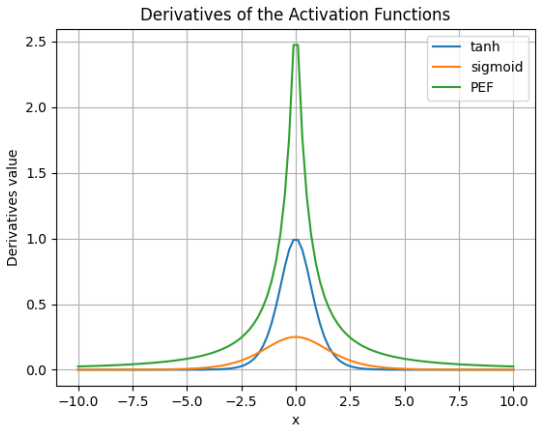
A. Datasets

The dataset properties have been shown in Table 2 of the supplementary file. A total of 29 datasets, including real industrial datasets and synthetic datasets, have been considered in the experiments. The industrial datasets include Yahoo webscope [2], AWS cloudwatch [3], GE. A couple of datasets have either one or a few anomalies, such as AWS$_1$, AWS$_2$. We have injected anomalies in AWS, Yahoo, and GE datasets to produce synthetic data for fair comparison. The datasets are univariate, unimodal, or binodal and follow mostly Weibull, Gamma, and Log normal distributions. The highest anomaly percentage is 1.47 (GE Dataset$_2$), whereas AWS Dataset$_2$ has reported the lowest percentage of anomaly i.e. 0.08 (For more details, see Table 2 of the supplementary text).

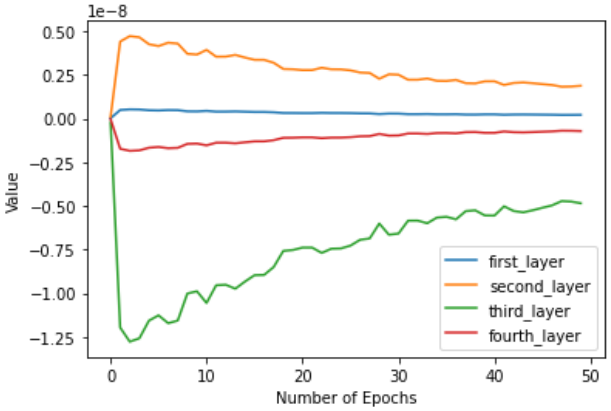
B. Results-Industrial Datasets

Table 3 and Table 4 show the performance comparison of various qLSTM techniques against different SoTA deep learning methods and non-deep learning methods using Precision and Recall scores, on industrial and synthetic benchmarks. Let us consider the industrial dataset results in Table 3. The Median-LSTM has achieved Recall 1 in most datasets (10 out of 15 datasets). Compared to existing benchmarks, LSTM methods are SOTA on most of the datasets regarding Recall. For comparison purposes, we have first compared the Recall. If the Recall is the same for two different methods, we have compared the Precision. The method with a higher Recall and Precision will be considered a better performer. In AWS datasets, most of the techniques have achieved the highest Recall apart from DAGMM and DevNet. DevNet needs a minimum of two anomalies; hence it is not applicable for AWS1 and AWS2. However, as per Precision, iqr-LSTM has performed better than other methods. In the case of GE1, DevNet has produced a better result, whereas quantile-based LSTM techniques have outperformed others on GE$_2$. Median-LSTM has demonstrated better results in Ambient temperature. In the case of Yahoo datasets, Median-LSTM has achieved the highest Recall on four datasets; however, quantile-LSTM and iqr-LSTM have produced better results on several datasets. For example, Median-LSTM and iqr-LSTM both achieved Recall 1 on Yahoo$_1$. However, if we compare the Precision, iqr-LSTM has shown better results. It is evident from Table 3 that

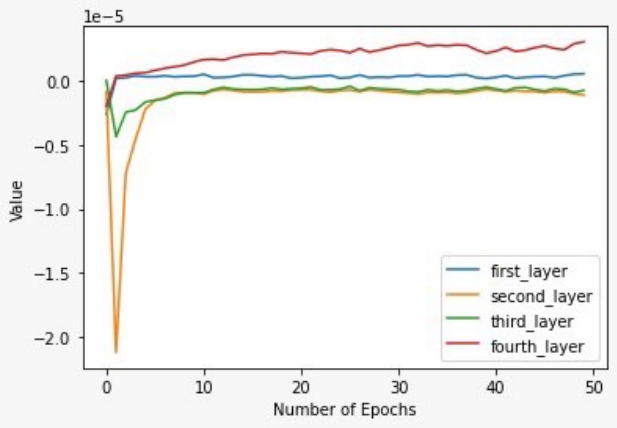
[1] LSTM code: https://github.com/PreyankM/Quantile-LSTM
[2] https://webscope.sandbox.yahoo.com/
[3] https://github.com/numenta/NAB/tree/master/data



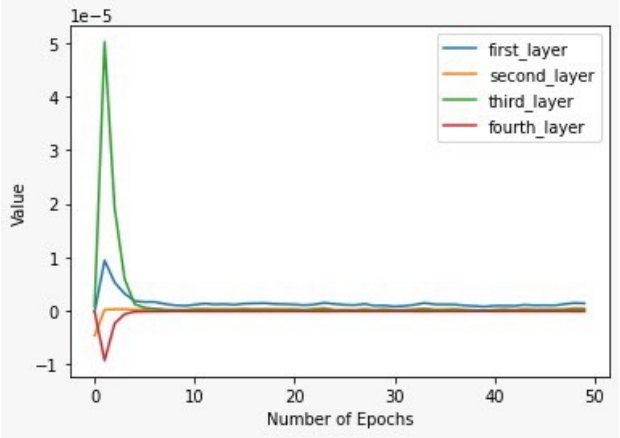
(a) Derivatives comparisons of various activation functions.



(b) LSTM values for 4 layers and 50 epochs using PEF as activation function using AWS2.



(c) LSTM values for 4 layers and 50 epochs using Sigmoid as activation function using AWS2.



(d) LSTM values for 4 layers and 50 epochs using Tanh as activation function using AWS2.

Fig. 3: Slow saturation rate as well as behavioral comparison of the different layers of LSTM model after the introduction of PEF with other activation functions. It also shows the final value of the learned parameter α on various datasets.

all these LSTM versions are performing very well on these industrial datasets. We compared our method with a recent anomaly detection method based on Graph Neural Network (GNN) [15] We observe that GNN has not shown superior performance compared to the quantile-based technique. For example, GNN's recall value is less than that of 1 quantile-based technique produced on AWS2, AWS3, Yahoo1, Yahoo2, and Yahoo9. Regarding precision, GNN produced better results than quantile LSTM only on two datasets, Yahoo1 and Yahoo9. Proposed LSTM based algorithms have outperformed another quantile-based approach as well, namely the Deep Quantile Regression [8], on all the datasets.

Now, let us consider the synthetic dataset result in Table 4. As per the Precision and Recall metrics, quantile-based approaches have outperformed iForest and other deep learning based algorithms, including Deep Quantile Regression on 7 out of 13 datasets. If we consider the Precision alone, the quantile LSTM based techniques have demonstrated better performance on 10 synthetic datasets. There are multiple reasons for the better performance demonstrated by the quantile-based LSTM approaches. Median-LSTM has detected the anomalies for each time period utilizing mean and standard deviation. The thresholds of proposed approaches are generic (distribution agnostic quantile thresholds) and therefore establish empirical robustness of the method. Additionally, the flexibility of the parameter α in determining the shape of the activation also helped. This is evident from Table 2 which represents the variation in α values of the PEF function across the datasets. α has been initialized to 0.1 for all the datasets.

C. F1 comparison on benchmark datasets

F1 Score indicates the overall performance of the anomaly detection model by combining both Recall and Precision. Table 5 demonstrates the performance comparison of the quantile based LSTM techniques with other baseline algorithms in terms of F1 scores. It is evident from Table that quantile based LSTM approaches have outperformed others in most of the datasets.

D. Robustness of quantile-LSTM Algorithms

We have experimented each of the algorithms on benchmark datasets 5 times and mean(μ) and std deviation(σ) of the results have been computed and shown in Table 6. Iqr-LSTM

Dataset	Anomaly	iqr-LSTM		Median-LSTM		quantile-LSTM		Autoencoder		GAN		DAGMM		DevNet		iForest		Envelope		Deep Quantile Regression	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
AWS1	1	**0.5**	**1**	0.052	1	0.041	1	0.045	1	0.047	1	0.125	1	NA	NA	0.0087	1	0.009	1	0.432	1
AWS2	2	0.13	1	**0.22**	**1**	0.0042	1	0.1	0.5	0.18	1	0.11	1	NA	NA	0.0062	1	0.04	1	0.21	1
AWS3	1	**1**	**1**	0.37	1	0.0181	1	0.0344	1	0.055	1	0	0	NA	NA	0.005	1	0.006	1	0.31	1
Ambient temperature	1	0.03	1	**0.0769**	**1**	0.02	1	0.055	1	0	0	0	0	NA	NA	0.01	1	0.02	1	0	0
GE1	3	0.019	1	0.048	1	0.0357	1	**0.093**	**1**	0.041	0.33	0	0	0.12	1	0.004	1	0.2	1	0	0
GE2	8	**1**	**1**	0.66	1	**1**	**1**	**1**	**1**	0	0	0.8	1	0.8	1	0.16	1	0.034	1	0.014	1
Yahoo1	2	**0.076**	**1**	0.0363	1	0.0465	1	1	0.5	0.066	1	0.07	0.5	0	0	0.005	1	0.009	1	0.02	0.5
Yahoo2	8	0.75	0.375	**0.8**	**1**	1	0.375	1	0.25	0.19	0.625	0.10	0.25	0	0	0.04	0.875	0.055	1	1	0.25
Yahoo3	8	**0.615**	**1**	0.114	0.675	0.088	1	0.023	0.25	0.11	0.875	0.15	0.62	0.39	0.5	0.04	0.875	0.032	0.875	0.036	0.5
Yahoo5	9	0.048	0.33	0.1	0.33	0.022	0.66	0.05	0.33	0	0	0.23	0.33	**0.67**	**1**	0.029	0.66	0.029	0.66	0.04	0.23
Yahoo6	4	0.12	1	0.222	1	0.0275	1	0.048	1	0	0	0.041	1	**1**	**1**	0.0073	1	0.0075	1	0.05	0.75
Yahoo7	11	0.096	0.54	**0.16**	**0.63**	0.066	0.54	0.083	0.45	0.035	0.54	0.058	0.09	0.33	0.29	0.0082	0.33	0.017	0.54	0.093	0.63
Yahoo8	10	0.053	0.7	**0.142**	**0.8**	0.028	0.3	0	0	0	0	0	0	0.063	0.11	0.01	0.6	0.010	0.6	0.056	0.3
Yahoo9	8	1	0.75	**0.333**	**1**	0.0208	0.75	1	0.37	0	0	0.5	0.375	0.07	0.8	0.04	1	0.047	1	0.072	0.375

TABLE 3: Performance comparison of various quantile LSTM techniques on industrial datasets

Dataset	Anomaly	iqr-LSTM		Median-LSTM		quantile-LSTM		iForest		Envelope		Autoencoder		GAN		DAGMM		DevNet		Deep Quantile Regression	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
AWS_syn1	11	0.769	0.909	0.687	1	1	0.909	0.034	1	0.10	1	1	0.63	**0.84**	**1**	0.71	0.90	0.09	0.73	0.023	0.090
AWS_syn2	22	0.7	1	**0.733**	**1**	0.6875	1	0.065	1	0.33	1	0.5	0.63	0.70	1	0.56	1	0.44	0.27	0.22	0.090
AWS_syn3	11	1	0.9	0.47	1	**1**	**1**	0.025	1	0.072	1	0.64	0.5	0.68	1	0	0	0.2	0.45	0.031	0.090
GE_syn1	13	0.0093	1	0.203	1	0.071	0.769	0.0208	1	0.135	1	0.23	0.11	0.25	0.61	0	0	**0.33**	**1**	0	0
GE_syn2	18	0.0446	1	**1**	**1**	**1**	**1**	0.3	1	0.409	1	1	0.38	0.9	1	0.9	1	0.9	1	0.032	1
Yahoo_syn1	12	**1**	**1**	0.217	0.833	0.375	1	0.027	1	0.056	1	1	0.83	0.31	1	0.29	0.41	0	0	0.067	0.67
Yahoo_syn2	18	0.181	0.55	0.653	0.944	1	0.611	**0.233**	**1**	0.124	1	1	0.42	1	0.61	0.55	0.61	0	0	0.093	0.61
Yahoo_syn3	18	0.89	0.94	0.3333	0.555	**0.6**	**1**	0.0410	1	0.0762	0.944	1	0.88	0.81	0.71	0.3	0.66	0.17	0.63	0.052	0.73
Yahoo_syn5	19	0.081	0.52	0.521	0.631	0.0625	0.578	0.03125	0.842	0.0784	0.842	0.15	0.47	0.42	0.53	0.52	0.52	**0.73**	**0.92**	0.15	0.68
Yahoo_syn6	14	0.065	0.85	0.65	0.928	0.764	0.928	**0.01825**	**1**	0.00761	0.285	0.05	0.28	0.8	0.29	0.041	0.28	0	0	0.08	0.93
Yahoo_syn7	21	0.61	0.59	0.375	0.714	0.411	0.66	**0.032**	**0.952**	0.052	0.85	0.18	0.42	0.14	0.38	0.058	0.047	0.11	0.64	0.42	0.59
Yahoo_syn8	20	0.32	0.65	**0.482**	**0.823**	0.197	0.7	0.0192	0.75	0.023	0.7	0.009	0.05	0.25	0.1	0	0	0.23	0.64	0.62	0.75
Yahoo_syn9	18	1	0.77	**1**	**1**	1	0.94	0.053	1	0.048	1	0.875	0.388	0.72	1	0.57	0.22	0.03	0.29	0.31	0.77

TABLE 4: Performance comparison of various quantile LSTM techniques on synthetic datasets

Dataset	iqr-LSTM	median-LSTM	quantile-LSTM	GAN	DAGMM	Autoencoder	LSTM-Autoencoders
AWS Dataset$_1$	0.66	0.1	0.8	0.09	0.22	0.86	0.057
AWS Dataset$_2$	0.23	0.044	0.008	0.03	0.21	0.16	0.166
AWS Dataset$_3$	1	0.071	0.035	0.1	0	0.066	0.035
Yahoo Dataset$_1$	0.14	0.07	0.088	0.125	0.013	0.66	0.16
Yahoo Dataset$_2$	0.5	0.88	0.54	0.29	0.14	0.4	0.4
Yahoo Dataset$_3$	0.76	0.20	0.16	0.2	0.24	0.42	0.875
Yahoo Dataset$_5$	0.084	0.15	0.042	0	0.27	0.086	0.125
Yahoo Dataset$_6$	0.21	0.36	0.053	0	0.08	0.091	0.170
Yahoo Dataset$_7$	0.16	0.25	0.11	0.066	0.071	0.14	0.101
Yahoo Dataset$_8$	0.099	0.24	0.05	0	0	0	0.023
Yahoo Dataset$_9$	0.85	0.5	0.04	0	0.42	0.54	0.533
GE Dataset$_1$	0.038	0.09	0.06	0.074	0	0.17	0.20
GE Dataset$_2$	1	0.8	1	0	0.088	1	1

TABLE 5: The F1 measure comparison of quantile based LSTM techniques with other standard anomaly identifier algorithms.

(a variant of quantile-LSTM) has outperformed others in most of the datasets. There is no evidence of significant variation in std deviation(σ) in the performance metrics. This establishes robustness of the performance of qLSTM based methods.

We have performed ANOVA. We initialized the PEF with different parameter values (α) and run the experiments several times for each α value.

A one way ANOVA comprises of null and alternative hypothesis as outlined below.

- H0 : $\mu_1 = \mu_2 \ldots \mu_k$ (It implies the means of all the populations are equal)
- H1: It signifies that there will be at least one population mean that differs from the other populations.

We have considered three different α value (0.1, 0.4 and 0.7). The quantile-LSTM was made to run 5 times for each of the α value. The F statistic and p-value for precision turn out to be equal to 1.50 and 0.296 respectively. Similarly, F statistic and p-value are 1 and 0.42 for recall. Since, for all cases, the p-values are higher than 0.05, we can not reject the null hypothesis. This indicates that the performance of the quantile-LSTM does not differ in case of PEF initialization with different α values.

E. Results-Non-Industrial Datasets

We have tested our approach on non-industrial datasets shown in Table 7. The quantile-based technique is better in three of the seven datasets, while Autoencoder is better for two of the seven datasets. Deviation Networks give NA because it does not work for single anomaly-containing datasets.

Dataset	iqr-LSTM		GAN		Autoencoder		DAGMM	
	Precision($\mu \pm \sigma$)	Recall($\mu \pm \sigma$)	Precision($\mu \pm \sigma$)	Recall($\mu \pm \sigma$)	Precision($\mu \pm \sigma$)	Recall($\mu \pm \sigma$)	Precision($\mu \pm \sigma$)	Recall($\mu \pm \sigma$)
AWS Dataset$_1$	**0.875±0.216**	**1±0**	0.061±0.008	1±0	0.026±0.0111	1±0	0.031±0.054	0.25±0.43
AWS Dataset$_2$	0.126±0.067	1±0	**0.16±0.0275**	**1±0**	0.14±0.076	0.62±0.21	0.0425±0.038	1±0
AWS Dataset$_3$	**1±0**	**1±0**	0.039±0.008	1±0	0.017±0.01	1±0	0±0	0±0
Yahoo Dataset$_1$	**0.076±0.002**	**1±0**	0.066±0.003	1±0	1±0.01	0.5±0.15	0.07 ± 0.025	0.5 ± 0
Yahoo Dataset$_2$	0.75 ± 0.008	0.375 ± 0	**0.19±0.05**	**0.625±0.01**	1 ± 0.01	0.25 ± 0.05	0.10 ± 0.0065	0.25 ± 0
Yahoo Dataset$_3$	**0.615±0.04**	**1±0**	0.11 ± 0.014	0.875 ± 0.025	0.023 ± 0.01	0.25 ± 0.008	0.15 ± 0.001	0.62 ± 0
Yahoo Dataset$_5$	0.048 ± 0.0012	0.33 ± 0	0 ± 0	0 ± 0	0.05 ± 0.018	0.33 ± 0.06	**0.23 ± 0.011**	**0.33 ± 0**
Yahoo Dataset$_6$	0.12 ± 0.0052	1 ± 0	0 ± 0	0 ± 0	**0.48 ± 0.056**	**1 ± 0.02**	0.041 ± 0.034	1 ± 0.14
Yahoo Dataset$_7$	**0.096 ± 0.0061**	**0.54 ± 0**	0.035 ± 0.003	0.54 ± 0.012	0.083 ± 0.016	0.45 ± 0.021	0.058 ± 0.008	0.09 ± 0.002
Yahoo Dataset$_8$	**0.053 ± 0.015**	**0.7 ± 0**	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Yahoo Dataset$_9$	**1 ± 0.005**	**0.75 ± 0**	0 ± 0	0 ± 0	1 ± 0.01	0.37 ± 0	0.5 ± 0.018	0.375 ± 0.15

TABLE 6: Statistical robustness of qLSTM vs. baseline DL algorithms on multiple datasets.Multiple runs do not impact qLSTM performance.

Dataset	Anomaly	quantile-LSTM		Autoencoder		GAN		DevNet		iForest		Envelope	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
TravelTime$_{387}$	3	**0.011**	**0.67**	1	0.33	0.024	0.33	0.01	0.33	0.0039	0.6667	0.0107	0.6667
TravelTime$_{451}$	1	0.006	1	0	0	**0.016**	**1**	NA	NA	0.0028	1	0.0062	1
Occupancy$_{6005}$	1	**0.03**	**1**	0	0	0.007	1	NA	NA	0.0019	1	0.0042	1
Occupancy$_{t4013}$	2	**0.06**	**1**	0.438	0.5	0.014	0.5	0.02	1	0.0038	1	0.0078	1
Speed$_{6005}$	1	0.014	1	**0.103**	**1**	0.009	1	NA	NA	0.002	1	0.0038	1
Speed$_{7578}$	4	0.086	1	**0.792**	**1**	0.2	0.9	0.16	0.75	0.0153	1	0.0247	1
Speed$_{t4013}$	2	0.053	1	0.75	0.5	0.043	1	**0.1**	**1**	0.0036	1	0.007	1

TABLE 7: Performance comparison of quantile LSTM techniques on various non-industrial datasets.

F. Comparison between Elliot Function and PEF

In order to compare the performance of Sigmoid, tanh, Elliot function and PEF as activation functions, we experimented with them by using them as activation functions in the LSTM layer of the models and comparing the results after they run on multiple datasets. The results are shown in Table 8. PEF has shown superior results on 11 datasets. If we compare PEF with Elliot Function alone, PEF has shown better results on 17 datasets. PEF against tanh comparison shows that PEF has demonstrated superior results on 16 datasets and identical performance on one dataset. The same comparison in PEF vs Sigmoid demonstrates that PEF has superior performance on 17 datasets and Sigmoid has superior results on 5 datasets. Therefore PEF with quantile-LSTM is more effective in comparison to other activation functions.

Dataset	Elliot Function		Parameterized Elliot Function		Sigmoid		tanh	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
AWS Dataset$_1$	0	0	**0.041**	**1**	0.027	1	0.0357	1
AWS Dataset$_2$	0.002	1	0.0042	1	**0.0147**	**1**	0.0046	1
AWS Dataset$_3$	**0.04**	**1**	0.0181	1	0.008	1	0.0061	1
AWS DatasetSyn$_1$	0.02	0.73	1	0.909	0.084	0.63	**0.224**	**1**
AWS DatasetSyn$_2$	0.39	0.77	**0.6875**	**1**	0.042	0.863	0.048	1
AWS DatasetSyn$_3$	0.06	0.73	**1**	**1**	0.060	0.63	1	1
Yahoo Dataset$_1$	0.006	0.25	0.0465	1	0.021	1	**0.08**	**1**
Yahoo Dataset$_2$	**0.02**	**1**	1	0.375	0.011	0.625	0.014	0.25
Yahoo Dataset$_3$	0.05	1	0.088	1	0.77	0.875	**0.25**	**1**
Yahoo Dataset$_5$	0.001	0.33	**0.022**	**0.66**	0.009	0.33	0.019	0.66
Yahoo Dataset$_6$	0.002	0.17	**0.0275**	**1**	0.013	1	0.027	1
Yahoo Dataset$_7$	0.03	0.09	0.066	0.54	**0.073**	**0.54**	0.068	0.36
Yahoo Dataset$_8$	**0.017**	**0.4**	0.028	0.3	0.026	0.3	0.0288	0.3
Yahoo Dataset$_9$	**0.43**	**0.75**	0.0208	0.75	1	0.75	0.018	0.625
Yahoo DatasetSyn$_1$	0.14	0.86	**0.375**	**1**	0.081	1	0.086	1
Yahoo DatasetSyn$_2$	**0.04**	**0.72**	1	0.611	0.024	0.61	1	0.5
Yahoo DatasetSyn$_3$	0.1	0.78	**0.6**	**1**	1	0.44	0.0611	0.611
Yahoo DatasetSyn$_5$	0.004	0.31	0.0625	0.578	0.039	0.73	**0.054**	**0.73**
Yahoo DatasetSyn$_6$	0.015	0.69	**0.764**	**0.928**	0.0606	0.857	0.043	0.857
Yahoo DatasetSyn$_7$	0.35	0.43	**0.411**	**0.66**	0.26	0.61	0.2	0.61
Yahoo DatasetSyn$_8$	0.024	0.5	**0.197**	**0.7**	0.071	0.4	0.0701	0.4
Yahoo DatasetSyn$_9$	0.27	0.67	**1**	**0.94**	1	0.72	0.116	0.277

TABLE 8: Comparison of Precision and Recall score for qLSTM with Elliot Function, tanh, Sigmoid and PEF as Activation Function.

G. Generalizability of PEF

We have evaluated the performance of PEF in other neural networks, such as LSTM, GAN and Deep Quantile Regression, as shown in Table 9. This experiment is performed to understand the effectiveness of PEF on other neural networks. It is evident that PEF has improved the performance in many instances. After applying PEF, the performance of LSTM has improved for 3 datasets and remained unchanged for 2 datasets. PEF has enhanced the performance of Deep Quantile Regression on 5 datasets. In the case of GAN, the improvement is visible on 3 datasets and there is no change in performance on 5 datasets. Therefore, it is palpable from Table 9 is that PEF is not limited to qLSTM only, an evidence of generalization ability of PEF.

Dataset	**LSTM**		**LSTM(PEF)**		**Deep Quantile Regression**		**Deep Quantile Regression(PEF)**		**GAN**		**GAN(PEF)**	
	Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
yahoo1	0.007	0.5	0.007	0.5	0	0	**0.0014**	**1**	0.066	1	0.066	1
yahoo2	0.0411	0.75	0.0411	0.75	0.0228	0.625	**0.055**	**1**	0.19	0.625	**0.22**	**0.625**
yahoo3	0.0417	0.75	**0.0486**	**0.875**	**0.0109**	**0.625**	0.0169	0.125	0.11	0.875	**0.14**	**0.875**
yahoo5	0.0141	0.2222	**0.0352**	**0.5556**	0	0	**0.0143**	**0.444**	0	0	0	0
yahoo6	0.0141	0.5	**0.0211**	**0.75**	**0.0028**	**1**	0.0032	0.75	0	0	0	0
yahoo7	**0.0119**	**0.1818**	0.006	0.0909	0	0	**0.0104**	**0.1818**	0.035	0.54	0.035	0.54
yahoo8	**0.0119**	**0.2**	0.006	0.1	0	0	**0.008**	**0.8**	0	0	0	0
yahoo9	**0.0119**	**0.25**	0	0	**0.0049**	**0.875**	0	0	0	0	**0.03**	**0.63**

TABLE 9: A comparative Study of DL methods (default activation function) versus DL methods (with PEF Activation Function) on Yahoo datasets. Marked improvement is observed on several datasets via PEF on other architectures. Effectiveness of PEF is not limited to qLSTM.

H. Impact of Varying Thresholds

Deep-learning based algorithms such as Autoencoder [4], GAN [16], Deep Quantile Regression [8] DAGMM [17] and DevNet [7] consider upper threshold and lower thresholds on reconstruction errors or predicted values. To understand the impact of different thresholds on performance, we considered three baseline algorithms: GAN, Autoencoder, and Devnet. The baseline methods have considered three different sets of threshold values for upper and lower thresholds. The sets are shown in column head of tables 10, 11 and 12, where the first threshold is the upper percentile and the second threshold is the lower percentile. In contrast, q-LSTM is robust against thresholds as data sets vary i.e. it captures all anomalies successfully within the 0.1 and 0.9 quantile thresholds.

GAN	99.25 and 0.75		99.75 and 0.25		99.9 and 0.1	
Dataset	Precision	Recall	Precision	Recall	Precision	Recall
Yahoo Dataset_1	0.09	1	0.25	1	0.5	1
Yahoo Dataset_2	0.348	1	0.333	0.375	0.4	0.25
Yahoo Dataset_3	0.28	0.5	0.444	0.286	0.28	0.5
Yahoo Dataset_5	0	0	0.375	0.333	0.6	0.333
Yahoo Dataset_6	0.5	0.5	0.5	1	0.182	1
Yahoo Dataset_7	0.154	0.364	0.3	0.273	0.5	0.182
Yahoo Dataset_8	0.038	0.1	0.1	0.1	0.25	0.1
Yahoo Dataset_9	0.192	0.625	0.5	0.625	0.5	0.25

TABLE 10: Comparison of Precision and Recall score for GAN with varying thresholds for anomaly Upper Bound and Lower Bound

It is evident from the above tables that performance varies significantly based on the thresholds decided by the algorithm. Therefore, deciding on a correct threshold that can identify all the probable anomalies from the dataset is very important.

Autoencoders	99.25 and 0.75		99.75 and 0.25		99.9 and 0.1	
Dataset	Precision	Recall	Precision	Recall	Precision	Recall
Yahoo Dataset_1	0.5	0.07	0.5	0.036	0.5	0.019
Yahoo Dataset_2	0.5	0.4	0.333	0.5	0.2	0.5
Yahoo Dataset_3	0.44	0.5	0.4	0.5	0.25	0.333
Yahoo Dataset_5	0.5	0.5	0.5	0.5	0.5	0.5
Yahoo Dataset_6	0.5	1	1	1	0.25	1
Yahoo Dataset_7	0.5	0.5	0.5	0.5	0.5	0.5
Yahoo Dataset_8	0.875	0.875	0.375	0.375	0.5	0.75
Yahoo Dataset_9	0.75	0.5	0.25	0.5	0.5	0.5

TABLE 11: Comparison of Precision and Recall score for Autoencoders with varying thresholds for anomaly Upper Bound and Lower Bound

Devnet	99.25 and 0.75		99.75 and 0.25		99.9 and 0.1	
Dataset	Precision	Recall	Precision	Recall	Precision	Recall
Yahoo Dataset_1	0.002	1	0.002	1	0.001	1
Yahoo Dataset_2	0.005	1	0.005	1	0.005	1
Yahoo Dataset_3	0.0078	1	0.0078	1	0.0078	1
Yahoo Dataset_5	0.111	0.5	0.333	0.5	0.333	0.5
Yahoo Dataset_6	0.167	1	0.5	1	0.5	0.667
Yahoo Dataset_7	0.054	0.2	0.125	0.2	0.25	0.2
Yahoo Dataset_8	0	0	0	0	0	0
Yahoo Dataset_9	0	0	0	0	0	0

TABLE 12: Comparison of Precision and Recall score for Devnet with varying thresholds for anomaly Upper Bound and Lower Bound

I. Experiments on Normal Instances

A relevant question to ask is: how would the anomaly detection methods perform on normal data instances that do not have any anomaly? We investigate this by removing anomalies from some data sets. We observe that on these data sets (AWS1, AWS2, AWS3, Yahoo1, Yahoo2, Yahoo3), q-LSTM and its variants reported very negligible false alarms (Average 40 false alarms) while other state-of-the-art methods, such as iForest, Elliptic Envelope produce higher flag false positives. Elliptic envelope has reported, on average, 137 false alarms, whereas iForest reported an average of 209 false alarms across the datasets. Autoencoder and GAN both have reported average false alarms of 46 and 123, respectively, which is higher than the false positive rate of q-LSTM. This establishes the robustness of the proposed method.

V. Related Work

Well-known supervised machine learning approaches such as Linear Support Vector Machines (SVM), Random Forest (RF), and Random Survival Forest (RSF) [18], [19] have been explored for fault diagnosis and the lifetime prediction of industrial systems. [20] have explored SVM and RF to detect intrusion based on the anomaly in industrial data. Popular unsupervised approaches, such as Anomaly Detection Forest [21] and K-means based Isolation Forest [22], try to isolate the anomalies from the normal dataset. [22] considered K-means based anomaly isolation, but the approach is tightly coupled with a clustering algorithm. Anomaly Detection Forest like k-means based iForest requires a training phase with a subsample of the dataset. A wrong selection of the training subsample can cause too many false alarms. The notion of 'likely invariants' uses operational data to identify a set of invariants to characterize the normal behavior of a system, which is similar to our strategy. Such as an approach has been attempted to discover anomalies of cloud-based systems [23]. However, this requires labeling of data and re-tuning of parameters. Recently, Deep Learning (DL) models based on auto-encoders, long-short term memory [24], [25] are used for anomaly detection. [5] have proposed an integrated model of Convolutional Neural Network (CNN) and LSTM based auto-encoder for anomaly detection. For reasons unknown, [5] tested only one Yahoo Webscope data to demonstrate their approach's efficacy. The DeepAnT [26] approach employs DL methods and uses unlabeled data for training. However, the approach is meant for UV time-series data sets A stacked LSTM [27] is used for time series anomaly prediction in unsupervised setting. The hierarchical Temporal Memory (HTM) method has been applied recently on sequential streamed data and compared with other time series forecasting models [28]. The authors in [29] have performed online time-series anomaly detection using deep RNN. The incremental retraining of the neural network allows the adoption of concept drift across multiple datasets. There are various works [1], [2], that attempt to address the data imbalance issue of the anomaly datasets since anomalies are very rare and occur occasionally. Hence, they propose semi-supervised approaches. However, the semi-supervised approach cannot avoid the expensive dataset labeling. Some approaches [6] apply predefined thresholds, such as fixed percentile values, to detect the anomalies. However, a fixed threshold value may not be equally effective on different domain datasets. Deep Autoencoding Gaussian Mixture Model (DAGMM) is an unsupervised DL-based anomaly detection algorithm [6], where it utilizes a deep autoencoder to generate a low-dimensional representation and reconstruction error for each input data point and is further fed into a Gaussian Mixture Model (GMM). Deviation Network(DevNet) [7] is a novel method that harnesses anomaly scoring networks, Z-score based deviation loss and Gaussian prior together to increase efficiency for anomaly detection.

VI. Discussion and Conclusion

In this paper, we have proposed multiple versions of the SoTA anomaly detection algorithms along with a forecasting-based LSTM method. We have demonstrated that combining the quantile technique with LSTM can be successfully implemented to detect anomalies in industrial and non-industrial datasets without label availability for training. We have also exploited the parameterized Elliot activation function and shown anomaly distribution against quantile values, which helps decide the quantile anomaly threshold. The design of a flexible form activation, i.e., PEF, also helps accommodate variance in the unseen data as the shape of the activation is learned from data. PEF, as seen in Table 8, captures anomalies better than other activation functions. The quantile thresholds are generic and will not differ for different datasets. The proposed techniques have addressed the data imbalance issue and expensive training dataset labeling in anomaly detection. These methods are useful where data is abundant. Traditional deep learning-based methods use classical conditional means and assume random normal distributions as the underlying structure of data. These assumptions make the methods vulnerable to capturing the uncertainty in prediction and make them incapable of modeling tail behaviors. Quantile in LSTM (for time series data) is a robust alternative that we leveraged in isolating anomalies successfully. This is fortified by the characteristics of quantiles making very few distributional assumptions. The distribution-agnostic behavior of Quantiles turned out to be a useful tool in modeling tail behavior and detecting anomalies. Anomalous instances, by definition, are rare and could be as rare as just one anomaly in the entire data set. Our method detects such instances (singleton anomaly), while some recent state of art algorithms, such as DAGMM require at least two anomalies to be effective. Extensive experiments on multiple industrial time-series datasets (Yahoo, AWS, GE, machine sensors, Numenta and VLDB Benchmark data) and non-time series data show evidence of effectiveness and superior performance of LSTM-based quantile techniques in identifying anomalies. The proposed methods have a few drawbacks 1) quantile based LSTM techniques are applicable only on univariate datasets. 2) A few of the methods, such as quantile-LSTM, iqr-LSTM have a dependency on multiple thresholds. We intend to extend our quantile-based approaches to detect anomalies in multivariate time series data in the future.

ACKNOWLEDGMENT

Snehanshu Saha would like to thank the Anuradha and Prashanth Palakurthi Center for Artificial Intelligence Research (APPCAIR), SERB SURE-DST(SUR/2022/001965), SERB CRG- DST (CRG/2023/003210) the DBT-Builder project (BT/INF/22/SP42543/2021), Govt. of India and CDRF-BITS Pilani for supporting the work.

REFERENCES

[1] A. Morales-Forero and S. Bassetto, "Case study: A semi-supervised methodology for anomaly detection and diagnosis," in *In 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2019, pp. 1031–1037.

[2] P. Sperl, J. P. Schulze, and K. B. ottinger, "A3: Activation anomaly analysis," *CoRR*, 2020. [Online]. Available: https://arxiv.org/pdf/2003.01801.pdf

[3] A. Jinwon and S. Ch, "Variational autoencoder based anomaly detection using reconstruction probability," in *Special Lecture on IE*, 2015, pp. 1–18.

[4] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis - MLSDA'14*, 2014, pp. 1031–1037.

[5] C. Yin, J. W. S. Zhang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for iot time series," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2020, doi: 10.1109/tsmc.2020.2968516.

[6] 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. [Online]. Available: https://openreview.net/forum?id=BJJLHbb0-

[7] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 353–362.

[8] A. I. Tambuwal and D. Neagu, "Deep quantile regression for unsupervised anomaly detection in time-series," *SN COMPUT. SCI*, vol. 6, no. 2, 2021.

[9] T. F. Liu, M. K. Ting, and Z. H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, Dec. 2008, p. 273–280, doi: 10.1109/ICDMW.2016.0046.

[10] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41(3), p. 212–223, 1999, doi: 10.1080/00401706.1999.10485670.

[11] R. Koenker, *Quantile Regression*. Cambridge University Press, 2005.

[12] A. Tambwekar, A. Maiya, S. Dhavala, and S. Saha, "Estimation and applications of quantiles in deep binary classification," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 275–286, 2022.

[13] J. L. Horowitz, "A smoothed maximum score estimator for the binary response model," *Econometrica*, vol. 60, no. 3, pp. 505–531, 1992. [Online]. Available: http://www.jstor.org/stable/2951582

[14] T. Dunning, "The t-digest: Efficient estimates of distributions," *Software Impacts*, vol. 7, 2021.

[15] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4027–4035.

[16] H. Zenati, M. Romain, C. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 727–736.

[17] 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.

[18] S. Voronov, E. Frisk, and M. Krysander, "Data-driven battery lifetime prediction and confidence estimation for heavy-duty trucks," *IEEE Transactions on Reliability*, vol. 67(2), pp. 623–639, 2018, doi: 10.1109/TR.2018.2803798.

[19] K. N. Verma, R. K. Sevakula, and R. Thirukovalluru, "Pattern analysis framework with graphical indices for condition-based monitoring," *IEEE Transactions on Reliability*, vol. 66(4), pp. 1085–1100, 2017, doi: 10.1109/TR.2017.2729465.

[20] D. D. S. Anton, S. Sinha, and H. D. Schotten, "Anomaly-based intrusion detection in industrial data with svm and random forests," in *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2019, doi: 10.23919/softcom.2019.8903672.

[21] J. Sternby, E. Thormarker, and M. Liljenstam, "Anomaly detection forest," in *ECAI*, 2020.

[22] P. Karczmarek, A. Kiersztyn, W. Pedrycz, and E. Al, "K-means-based isolation forest," *Knowledge-Based Systems*, vol. 195, 2020.

[23] A. Russo, S. Pecchia, and S. Sarkar, "Assessing invariant mining techniques for cloud-based utility computing systems," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 44–58, 2020.

[24] S. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognit*, vol. 58, p. 121–134, 2016.

[25] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD*, 2017, doi: 10.1145/3097983.3098052.

[26] M. Munir, A. S. Siddiqui, A. Dengel, and S. Ahmed, "Deepant: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 1(1), pp. 1085–1100, 2018, doi: 10.1109/access.2018.2886457.

[27] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015, pp. 89–94.

[28] E. N. Osegi, "Using the hierarchical temporal memory spatial pooler for short-term forecasting of electrical load time series," *Applied Computing and Informatics*, vol. 7, no. 2, pp. 264–278, 2021.

[29] S. Saurav, P. Malhotra, V. T. N. Gugulothu, L. Vig, P. Agarwal, and G. Shroff, "Online anomaly detection with concept drift adaptation using recurrent neural networks," in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, ser. CoDS-COMAD*, 2018, p. 78–87.

Snehanshu Saha is a Professor in the Department of Computer Science and Information Systems and heads the AI research center-APPCAIR, BITS Pilani Goa Campus, India. His current and future research interests lie in theoretical understanding and development of methods in Optimization and Machine Learning.

Jyotirmoy Sarkar is a Staff Software Engg.at GE Healthcare, Bangalore. Jyotirmoy's fields of interest include cloud computing, mathematical modeling, machine learning, reliability, and dependable systems. Jyotirmoy is an expert in robustness studies.

Soma S. Dhavala is a Principal Researcher with the Wadhwani Institute for Artificial Intelligence, Mumbai, India, working on AI for social good in public health space. Soma is an expert Statistician.

Santonu Sarkar is currently serving as Professor and Chair in the Department of Computer Science and Information Systems, BITS Pilani Goa Campus, India. His current research interest lies in building machine-learned applications for the cloud and cyber-physical systems.

Preyank Mota is a computer science and mathematics student pursuing M.Sc. (Hons.) Mathematics and B.E. (Hons.) Computer Science from BITS Pilani K. K. Birla Campus, Goa, India, set to graduate in 2024. He takes interest in Time Series Signal generative model, hyperparameter tuning, and anomaly detection.