RESIST: Robust Transformer for Unsupervised Time Series Anomaly Detection

Naji Najari[1,2,3], Samuel Berlemont[1], Grégoire Lefebvre[1], Stefan Duffner[2,3], and Christophe Garcia[2,3]

[1] Orange Innovation, Meylan, France
{naji.najari, samuel.berlemont, gregoire.lefebvre}@orange.com
[2] LIRIS UMR 5205 CNRS, Villeurbanne, France
{naji.najari, stefan.duffner, christophe.garcia}@liris.cnrs.fr
[3] INSA Lyon, Villeurbanne, France

Abstract. In the last decades, Internet of Things objects have been increasingly integrated into smart environments. Nevertheless, new issues emerge due to numerous reasons such as fraudulent attacks, inconsistent sensor behaviours, and network congestion. These anomalies can have a drastic impact on the global Quality of Service in the Local Area Network. Consequently, contextual anomaly detection using network traffic metadata has received a growing interest among the scientific community. The detection of temporal anomalies helps network administrators anticipate and prevent such failures. In this paper, we propose RESIST, a Robust transformEr developed for unSupervised tIme Series anomaly deTection. We introduce a robust learning strategy that trains a Transformer to model the nominal behaviour of the network activity. Unlike competing methods, our approach does not require the availability of an anomaly-free training subset. Relying on a contrastive learning-based robust loss function, RESIST automatically downweights atypical corrupted training data, to reduce their impact on the training optimization. Experiments on the CICIDS17 public benchmark dataset show an improved accuracy of our proposal in comparison to recent state-of-the-art methods.

Keywords: Unsupervised anomaly detection · Robust Transformers · Self and Co-attention · Network traffic anomaly detection.

1 Introduction

With the substantial increase of network anomalies in modern communication networks, anomaly detection has gained considerable interest over the last few years. The classical detectors, i.e., signature-based detectors, identify anomalies based on a predefined set of rules that models known attack signatures. These signatures must repeatedly be updated to integrate new attacks. Despite their effectiveness in identifying known threats, these systems fail to detect new emerging anomalies, e.g., zero-day attacks and non-malicious faults. To address these limitations, all the more present with the development of the Internet

of Things (IoT), contextual anomaly detection becomes of big interest in the network analysis landscape.

Anomaly Detection (AD) in time series is a broad research field affecting numerous application domains such as network and object monitoring, medical data analysis, fraud detection, and network intrusion detection [8]. In such fields, detecting outliers mainly relies on the *temporal continuity* assumption, defined by Aggrawal [1] as "the fact that the patterns in the data are not expected to change abruptly unless there are abnormal processes at work." As such, a temporal outlier is an abrupt change in the data pattern, which results in a discontinuity of the data with its local context. This assumption makes temporal AD more challenging than the classical unsupervised punctual AD, since considering the ordinal causality between observations is of paramount importance.

Numerous extensive studies have been carried out in the field of temporal AD. Contributions have shifted their focus towards semi-supervision, a.k.a., One-Class Classification. Here, an algorithm is first trained to model the nominal patterns of the anomaly-free training data. Then, any deviation from the trained model is flagged as an outlier. Despite yielding encouraging results in some specific applications, these classical anomaly detectors generally assume the availability of anomaly-free training data, and their performance drastically declines in the presence of corrupted observations. Unfortunately, in real-world applications, the data collection process is prone to contamination, as the training data may be corrupted with an unknown fraction of outliers. For example, in network intrusion detection, diverse anomalies may occur during the collection of the training network trace, due to faulty sensors, traffic congestions, and security attacks. The manual filtering of training anomalies is laborious, because of increasing data volumes and the diversity of emergent anomalies. This motivates the development of robust unsupervised temporal anomaly detectors, insensitive to training contamination.

In this paper, we propose RESIST, a Robust transformEr designed for unSupervised tIme Series anomaly deTection. We introduce a novel training strategy that identifies and downweights the impact of contaminants. RESIST is trained to mine the common temporal correlations that link successive sliding windows. Only common patterns are modelled and instance-specific rare patterns are ignored, since they may be caused by training corrupted data. RESIST training optimizes the robust Geman-Mcclure loss function, to reduce the impact of training outliers.

This paper is organized as follows: Section II introduces related work in temporal AD, and focuses particularly on Transformers for robust AD. Section III presents our contribution: RESIST. Section IV depicts the datasets used in our experiments, the training protocols, and the experimental results. Finally, conclusions and perspectives are drawn.

2 Related Work

Time series AD is an active research field that has drawn increasing attention in the data mining and machine learning community [5, 8].

Time series AD mainly include four main families: density-based, clustering-based, prediction-based, and reconstruction-based methods. Density-based methods rely on a *local density* criterion to identify outliers. Observations that have few adjacent neighbours are considered anomalous. Density-based methods, such as Local Outlier Factor (LOF) [6] and Deep Autoencoding Gaussian Mixture Model (DAGMM) [32], are extensively used in non-temporal anomaly detection. Many works extend these classical methods to time series anomaly detection, by restricting the local density criterion to local sliding windows [2]. Cluster-based methods firstly determine the optimal set of clusters that model the nominal data. Then, these clusters are used as a reference for normality: the anomaly score is defined as the distance to the closest cluster centre. The most common cluster-based anomaly detectors include Support Vector Data Description (SVDD) [22], and Deep-SVDD [16]. Similarly, numerous studies have been carried out to adapt such methods to temporal AD [2]. Prediction-based methods train a model to forecast a posterior observation using only past data. Anomalies are points that are different from their predictions. Various models were developed within this category, ranging from AutoRegressive Integrated Moving Average (ARIMA) [31], to Long Short-Term Memory recurrent neural networks [9]. Finally, reconstruction-based methods learn to compress the nominal data points into a low-dimensional representation and reconstruct the original data based on these compressed encodings. In other words, these methods learn to extract the most important information of the norm by mapping the data into a subspace of lower dimensionality, with the least reconstruction error. Since anomalies generally comprise non-representative features, it is harder to project them in this subspace without loss of information, which results in a larger reconstruction error. The most common reconstruction-based anomaly detectors are AutoEncoders. They are extensively used to identify non-temporal anomalies [13, 15, 14, 20]. To extend this approach to time series data, Su et al. [20] propose a hybrid method that combines a Variational AutoEncoder (VAE) and a Gated Recurrent Unit (GRU). While the GRU learns the temporal correlations of the input sequences, the VAE is trained to map the observations into a latent stochastic space. Similarly, Malhotra et al. [13] propose an LSTM-AE, tailored for time series AD.

Within the reconstruction-based category, a series of recent studies has shown the advantage of using Transformers over classical methods [7]. Benefiting from the self-attention mechanism and parallel computations, Transformer-based anomaly detectors show a higher detection performance and a more efficient training process [26]. Some recent studies, e.g., TranAD [23] and MT-RVAE [25], propose combining the Transformer-based architecture with common generative models, Generative Adversarial Networks (GANs) and VAEs, to further improve the model performance and robustness to training contaminations.

Alternatively, Xu et al. [27] renovate the self-attention mechanism by introducing a new *AnomalyAttention* module, specifically tailored for unsupervised time series anomaly detection. Their method, called AnomalyTransformer, is based on the intuition that, due to the rarity of anomalies, it is harder to find an association spread over the whole sequence. The authors remark that the self-attentions

of anomalous points generally tend to be located in their adjacent data points. Consequently, AnomalyTransformer leverages this *adjacent concentration bias* to make anomalous points more distinguishable. The authors formalize the *adjacent concentration bias* by defining the Association Discrepancy (AssDis) criterion. For each data point, the Association Discrepancy quantifies the disparity between the local attention relative to the adjacent points and the global attention with the whole series. As it is difficult to find a global mapping that links anomalous points with the whole sequence, both local and global self-attentions are mostly localized in the surrounding. As such, anomalies have smaller Association Discrepancy than nominal points. After training, AnomalyTransformer is used to assess the anomalousness of new samples. For a test data matrix $\mathbf{X} \in \mathbb{R}^{T \times d}$, containing T consecutive data points of dimension d, and its reconstruction $\widehat{\mathbf{X}} \in \mathbb{R}^{T \times d}$, the anomaly score is computed as follows:

$$\text{AnomalyScore}(\mathbf{X}) = \text{Softmax}(-\text{AssDis}) \left\| \mathbf{X} - \widehat{\mathbf{X}} \right\|_2^2. \tag{1}$$

The classical reconstruction error is amplified with a term inversely proportional to the AssDis. Since anomalies have smaller AssDis than inliers, their reconstruction error is amplified, which improves anomaly detection performance.

AnomalyTransformer shows that encouraging global attention spread over the entire sequence improves Transformer anomaly detection performance. Despite being more robust than vanilla Transformers, AnomalyTransformer attention is still restricted to the input sequence and lacks longer-term dependencies extracted from historical sequences. In fact, time-series data are usually split into fixed-length consecutive segments using a sliding window. The reference of normality in AnomalyTransformer is bounded to a single segment and ignores all previous windows. Even though anomalies are rare, the same anomaly may occur twice in the same window. In this case, the adjacent concentration bias becomes invalid, as anomalous observation self-attention is no longer limited to its surroundings. This is why we propose RESIST, which addresses this limitation, by extending the adjacent concentration prior to accounting for historical long-range properties.

3 Method

Unlike AnomalyTransformer, we propose to extend the Transformer attention to cover the historical data, in order to reject unusual observations. We hypothesize that rejecting training contaminants requires building pairwise associations not only between data points of the same sequence but also with instances of previous segments. The main intuition is that nominal instances present a regular behaviour shared across multiple segments. That is, reconstructing nominal sequences using either self-information extracted from the current input (i.e., self-reconstruction) or using relevant information extracted from the history (i.e., cross-reconstruction) would lead to similar results. In contrast, since anomalies are rare and different, building inter-sequence associations (or similarities) is more difficult and less informative. Building on this insight, we propose RESIST, a Robust transformEr

for unSupervISed Time-series anomaly detection. RESIST is trained to reconstruct input sequences using a hybrid representation that combines local intra-sequence information as well as global properties, shared between multiple segments. Firstly, we introduce a Siamese training strategy that ensures that the model pays equal attention to the input sequence as well as to the previous ones. Secondly, we train RESIST with a robust loss function to reduce the impact of large reconstruction errors caused by training outliers. In the following, we detail our contributions and the hypotheses that we will analyze in the experimental part. First, we depict a global architecture overview of RESIST to present its main building blocks. Then, we present each component separately. Finally, we present our hypotheses, the corresponding experimental protocols and results.

3.1 RESIST Architecture

RESIST presents an encoder-decoder architecture, comprised of four main components: a positional encoding and embedding layer, a siamese encoder, a fusion layer, and a decoder (cf. Figure 1). Similar to vanilla Transformers [24], the original data is firstly encoded using the linear embedding and the positional encoding units. Both encoder and decoder are composed of stacked identical blocks, where each block contains a multi-head attention unit followed by a Feed-Forward Network (FFN) layer.

RESIST takes as input K non-overlapping sequences $\mathbf{X}_t^w = (\mathbf{x}_{t-K+1}^w, ..., \mathbf{x}_t^w)$: an input sequence $\mathbf{x}_t^w$ and its $K-1$ previous sequences. Here, each sequence is composed of w consecutive data points $\mathbf{x}_t^w = (\mathbf{x}_{t-w+1}, ..., \mathbf{x}_t)$, where $\mathbf{x_t} \in \mathbb{R}^d$ is an observation of dimension d, recorded at the timestamp t. In Figure 1, we illustrate our method for $K = 2$. Firstly, the linear embedding and the positional encoding units encode the input sequences $(\mathbf{x}_{t-K+1}^w, ..., \mathbf{x}_t^w)$ and output the K embedded sequences $(\mathbf{e}_{t-K+1}^w, ..., \mathbf{e}_t^w)$. Secondly, the encoder extracts from each embedded sequence $\mathbf{e}_t^w$ a low-dimensional latent encoding $\mathbf{z}_t^w$. Then, the fusion layer aggregates these encodings into a single representation. The decoder maps the fusion encoding to the input space in order to reconstruct the original sequence $\mathbf{x}_t$. Finally, RESIST minimizes the Geman-McClure robust function between the reconstructed sequence $\widehat{\mathbf{x}_t^w}$ and the original one $\mathbf{x}_t^w$.

After presenting the global architecture of our method, we will thoroughly review each component in the following Sections.

Siamese Encoder RESIST encoder, illustrated in Figure 1, learns to project K consecutive sequences into K low-dimensional embeddings. The encoder receives a sequence $\mathbf{x}_t^w$ and its associated history, which contains the $K-1$ sequences preceding $\mathbf{x}_t^w$. It models the point-wise correlations between $\mathbf{x}_t^w$ and the history. Then, it learns to project these data into a common reduced space of dimension $d_{enc} \in \mathbb{N}^*$, where common data points share similar representations. This task is notoriously hard for anomalies, since they present non-representative uncommon patterns. For this reason, we propose an encoder with a Siamese architecture, with K identical sub-networks that share the same parameters. Input sequences

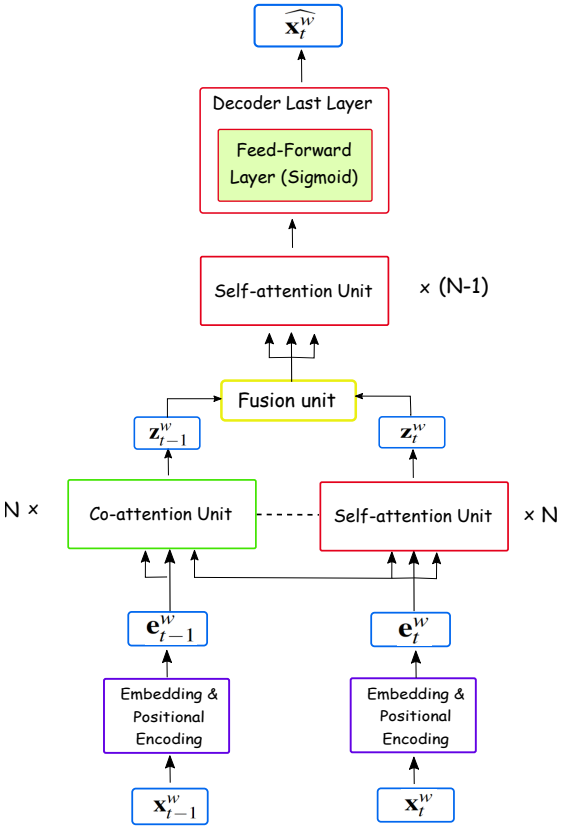

Fig. 1. RESIST architecture.

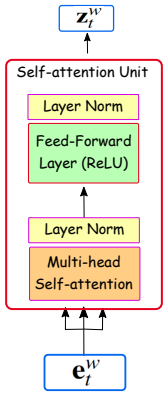

Fig. 2. Self-attention unit

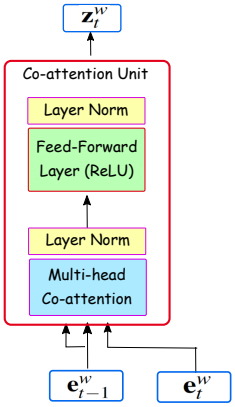

Fig. 3. Co-attention unit

are simultaneously processed using these networks. The sequences that share common proprieties have close encodings.

Unlike classical Siamese Neural Networks, our encoder is not trained to learn a similarity metric between input sequences. Its objective is to reduce the data dimensionality to only keep the most important information. Each siamese encoder sub-network is composed of a stack of $N = 2$ identical blocks. Each block comprises two sub-modules: a multi-head attention unit followed by a FFN layer (cf. Figures 2 and 3). While the attention mines the temporal correlations in the data, the FFN layers are used for dimensionality reduction.

The Siamese encoder is a hybrid composition of both Self-Attention (SA) and Co-Attention (CA) units. While the SA units are used to extract the contextual properties of the current sequence, the CA unit is destined to extract inter-segment properties and only keep common relationships.

Self-attention and Co-attention Module Attention modules are intended to mine pairwise interactions between data points. We propose to leverage the SA

and CA layers, initially introduced in multimodal Visual Question Answering (VQA) [28], to our task of unsupervised AD.

VQA is a visual reasoning task where we train a model to answer a question concerning an image. Identifying joint visual-linguistic representations is crucial in VQA. In [28], Yu et al. propose a Transformer-based VQA model where they introduce a co-attention layer, a.k.a., guided attention layer (see Figure 3). This layer is mainly designed to model multimodal interactions between a sentence and an image. The architecture of co-attention is the same as the self-attention layer. The main difference is that co-attention receives two different input sequences, a sentence and an image. It extracts the Query from the image and the pair (Key, Value) from the sentence. Recent studies [21] show the potential of co-attention to learn contextual representations and to improve model generalization performance.

We propose to extend CA to our task of unsupervised anomaly detection. CA can be seen as a module that filters similar data points between a sequence and the history. Then, it weights the current sequence observations with the relative normalized similarities. The aim is to guide the reconstruction with inter-sequence common information and to filter out sequence-specific rare patterns. This encourages the model to ignore unusual patterns that are only relevant for a single sequence. Different compositions of CA and SA may result in different configurations of RESIST. In Section 4.3, we will present these configurations and we will experimentally evaluate their impact on the AD performance.

Fusion Layer We propose to leverage multiple data views for robust reconstruction. The fusion layer combines the multiple encodings extracted by RESIST encoder into a single vector representation. In this work, we propose an addition-based fusion. This module comprises a fusion layer, followed by a FFN layer. The RESIST additive fusion strategy is inspired from the well-known manifold *mixup* method [30]. The original *mixup* method was initially proposed for data augmentation in supervised learning. For two training inputs x_i and x_j, having two labels y_i and y_j, respectively, *mixup* generates a new training instance, $\hat{x}$, using a linear interpolation:

$$\hat{x} = \beta x_i + (1-\beta)x_j \quad \text{and} \quad \hat{y} = \beta y_i + (1-\beta)y_j. \tag{2}$$

$\hat{y}$ is the corresponding label of $\hat{x}$. The interpolation term $\beta \in [0,1]$ is an hyper-parameter. In other words, *mixup* trains supervised classifiers to adapt a linear behaviour in the boundaries between training classes. *Mixup* reduces classifier regularization error and makes classifiers more robust to corrupted labels [30].

We extend the *mixup* method to robust unsupervised anomaly detection. Similar to the original *mixup* strategy, *mixup* fusion merges K instances into a single vector through linear interpolation. The merged representation of K encodings $(\mathbf{z}^w_{t-K+1}, ..., \mathbf{z}^w_t)$ is defined as follow:

$$\widehat{\mathbf{z}^w_t} = \frac{1}{K} \sum_{i=t-K+1}^{t} \mathbf{z}^w_i \tag{3}$$

We propose a uniform contribution of all encodings. For $K = 2$, we have $\widehat{\mathbf{z}_t^w} = 0.5\mathbf{z}_{t-1}^w + 0.5\mathbf{z}_t^w$. When the input sequence $\mathbf{x}_t^w$ presents common properties relative to its history, represented by $\mathbf{x}_{t-1}^w$, we expect that the siamese encoder extracts close latent representations $\mathbf{z}_t^w$ and $\mathbf{z}_{t-1}^w$. In this case, the fusion representation would be similar to the encoding of a vanilla Transformer, i.e., $\widehat{\mathbf{z}_t^w} \approx \mathbf{z}_{t-1}^w$. In contrast, when the current sequence comprises an uncommon pattern, the encoder self-attention and co-attention modules potentially extract different encodings. Therefore, the linear interpolation may generate an inconsistent sample and the reconstruction task become more difficult.

Finally, this compact representation $\widehat{\mathbf{z}_t^w}$ is forwarded to the FFN of the fusion module and the final output is:

$$\mathbf{F}_t^w(\mathbf{z}_{t-K+1}^w, ..., \mathbf{z}_t^w) = ReLU(\widehat{\mathbf{z}_t^w}\mathbf{W}_f + \mathbf{b}_f), \tag{4}$$

where $\mathbf{W}_f \in \mathbb{R}^{d_{enc} \times d_f}$ refers to the linear layer weights and $\mathbf{b}_f \in \mathbb{R}^{d_f}$ to the bias vector. d_{enc} is the dimension of the fusion module inputs and d_f is the dimension of the outputs. In all experiments, we use $d_f = d_{enc} = 16$.

RESIST Decoder Finally, the RESIST decoder learns to reconstruct the last sequence of the input using the compact representation that is the output of the fusion module. It is composed of a stack of $N = 2$ identical blocks. Each block comprises two sub-modules: a multi-head self-attention unit followed by a FFN layer. While Rectified Linear Unit (ReLU) activation function is used in the first block, the last block is followed by a Sigmoid function to ensure that the output has the same range as the input $[0, 1]$.

3.2 Robust Training Loss

To hedge against training contaminants, we train RESIST using a robust loss function. Indeed, the commonly used Mean Squared Error (MSE) is sensitive to outliers, since squaring large deviations results in the dominance of anomalies during the training. In contrast, a robust loss can resist noise and anomalies by reducing the influence of their large reconstruction errors. There have been numerous studies to explore robust leaning in the presence of outliers. The robust function list includes Charbonnier loss, Cauchy loss, Geman-McClure loss, and Welsch loss. Recently, Barron [4] generalizes these common losses in a single parametric function, $\rho(x, \alpha, c)$, parameterized by the scale c and the robustness parameter α.

$$\rho(x, \alpha, c) = \begin{cases} \frac{1}{2}(\frac{x}{c})^2 & if\ \alpha = 2 \\ \log(\frac{1}{2}(\frac{x}{c})^2 + 1) & if\ \alpha = 0 \\ 1 - \exp(-\frac{1}{2}(\frac{x}{c})^2) & if\ \alpha = -\infty \\ \frac{|\alpha-2|}{\alpha}((\frac{(\frac{x}{c})^2}{|\alpha-2|} + 1)^{\frac{\alpha}{2}} - 1) & otherwise \end{cases} \tag{5}$$

Particular values of α define common robust losses: L2 loss ($\alpha = 2$), Charbonnier loss ($\alpha = 1$), Cauchy loss ($\alpha = 0$), Geman-McClure loss ($\alpha = -2$), and

Welsch loss ($\alpha = -\infty$). These cases are visualized in Figure 4, extracted from [4]. We refer the reader to [4] for a detailed description of these losses.

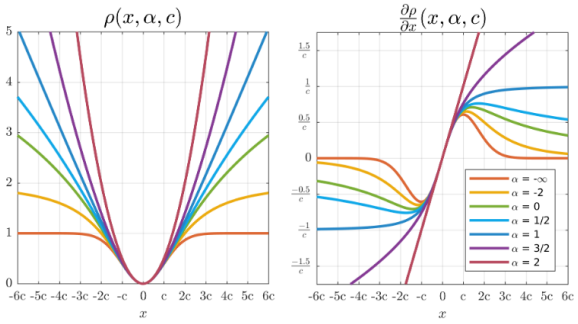

Fig. 4. The general robust loss function proposed in [4].

In particular, we propose to train RESIST by minimizing the Geman-McClure robust function, which reduces the influence of high reconstruction errors in gradient computations during training. The Geman-McClure function is:

$$L(x) = \rho(x, \alpha = -2, c) = 2\frac{(\frac{x}{c})^2}{4 + (\frac{x}{c})^2} \tag{6}$$

where c is a scale parameter that modulates the loss robustness range. In all our experiments, we set $x = \lambda \text{IQR}$, where IQR is the interquartile range and $\lambda = 0.1$.

3.3 Hypotheses

We synthesize our contributions into the following hypotheses:

- **Hypothesis 1 (H1)** *We conjecture that guiding the Transformer reconstruction with both intra-sequence properties, extracted using SA units, and inter-sequence pairwise interactions with the history, extracted with CA units, results in a more robust anomaly detector.*
- **Hypothesis 2 (H2)** *We hypothesize that training RESIST with a robust loss function, and particularly the Geman-McClure loss, reduces the impact of training noise and anomalies;*

4 Experiments and Results

In this section, we explore the validity of the assumed hypotheses on the benchmark dataset: the Canadian Institute of Cybersecurity Intrusion Detection System (CICIDS17) evaluation dataset [18]. In addition, we extensively compare our contribution against common unsupervised anomaly detectors. First, we provide an overview of the dataset. Then, we develop the training and testing protocols. Finally, we present and analyze the empirical results.

4.1 Dataset Description

CICIDS17 [18] is a recent public dataset developed by the Canadian Institute of Cybersecurity (CIC) for IDS evaluation. Overall, this dataset comprises about 3 million labelled network flows collected over 5 days, starting from July 3, 2017, and ending on Friday, July 7, 2017. 83% of this traffic is benign and the remaining 17% is anomalous. To collect the traffic, Sharafaldin et al. developed a testbed containing two networks: an Attack-Network and a Victim-Network. The Victim-Network comprises three servers, one firewall, two switches and ten interconnected PCs. One switch was configured to mirror all the traffic passing through the network. The Attack-Network is a separate network that runs network attacks on the Victim-Network.

CICIDS17 provides full packet capture of the collected data in *pcap* files. In addition, the raw data are processed using CICFlowMeter, a flow-based feature extractor, to extract metadata from the packet traces. Each flow record is represented by 85 features: a flow ID, 83 flow metadata features, and a class label. A detailed description of the 83 flow-based features is presented in [10]. CICIDS2017 comprises 15 classes: a nominal class and 14 attack types, including DoS, Distributed DoS (DDoS), Web attacks, and Infiltration attacks. This dataset was extensively used in many recent publications [10], since it covers various recent attacks and it comprises both punctual and collective anomalies.

4.2 Data Preprocessing

We follow the same preprocessing steps proposed in [11]. Since the original dataset is voluminous, we focus on the data subset that is collected during one day: Thursday, July 6 2017. This subset contains 170231 network flow and represents around 6% of the whole dataset. 98.7% of this traffic is benign and the remaining 1.3% is anomalous. We rescale numerical features to be in the range $[0, 1]$, using the min-max normalization method. Then, we randomly split the benign data into 40% for the training and 60% for testing.

4.3 Training and Testing Protocols

Protocol 1 (P1): Modular Composition of Co-attention and Self-attention Modules RESIST encoder is composed of two attention-based components: the self-attention and the co-attention modules. Different combinations of these modules result in different variants. In this section, we study the performance of RESIST with three modular compositions of these units.

For ease of illustration, we only visualize the RESIST encoder part for the three configuration. The first variant, RESIST-SS (cf. Figure 5), is the baseline. This first configuration does not consider the history for data reconstruction. In this case, only the input sequence flows through self-attention units to gradually extract the intra-properties of each sequence Then, RESIST-SS decoder is trained to reconstruct the sequence based only on this self-encoded representation. The

second configuration, RESIST-SC (cf. Figure 6), considers inter-sequence similarities between the input and the history. Indeed, both sequences are processed using a first self-attention unit to model intra-sequence relationships. Then, the encoded representation of the current sequence is processed using a self-attention unit, while the historical representations are fed into a co-attention unit to introduce pairwise similarities between consecutive sequences. Finally, the third variant is RESIST-CC (cf. Figure 7). Here, the input sequence is encoded through cascaded self-attention units and adjacent sequences are encoded using co-attention units. The main difference between RESIST-SC and RESIST-CC is that the former encodes the history with a hybrid encoder that alternates CA and SA, while in the latter, only CA units are used to encode the previous segments.

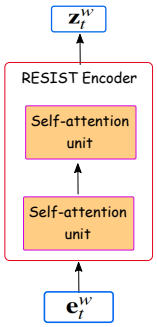

Fig. 5. RESIST-SS

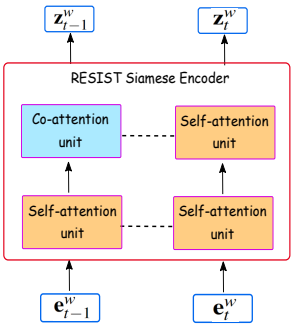

Fig. 6. RESIST-SC

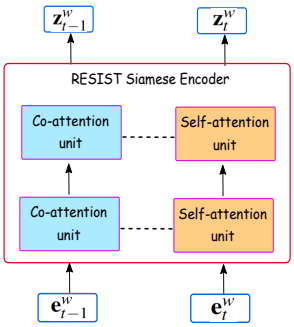

Fig. 7. RESIST-CC

Protocol 2 (P2): Robust Loss Function In this section, we explore the importance of the robust loss function to reduce model sensitivity with respect to anomalies. As previously mentioned in Section 3.2, various robust losses are developed in the literature, such as Charbonnier loss, Cauchy loss, Geman-McClure loss, and Welsch loss. In particular, we compare three different training losses. The first function is the classical L2 loss. Here, we train this first variant of RESIST with the common L2 loss to study its sensitivity to training outliers, in the absence of a robust training loss. Then, we compare three common robust functions: Charbonnier loss, Cauchy loss, and Geman-McClure loss. [4].

Protocol 3 (P3): Comparison with Competing Methods Finally, we globally compare our contribution against common unsupervised time series anomaly detectors. In this experiment, we select the best performing configuration of RESIST: a siamese encoder that comprises a hybrid composition of self and co-attention units, and trained with the Geman-McClure loss. The baselines selected in our experiments belong to the different categories of unsupervised anomaly detection presented in Section 2. These baselines include one-class classifiers: IF [12], OSVM [22]; density-based methods: LOF [6]; reconstruction-based algorithms: OmniAnomaly [20], LSTM-AE [13], MSCRED [29], USAD [3],

and vanilla Transformer [24]. In addition, we assess the performance of robust Transformers including TranAD [23] and AnomalyTransformer [27].

4.4 Training Parameter Settings and Evaluation Criteria

We follow the well-established protocol used by many recent papers [19]. We transform the input time series into consecutive sub-sequences using non-overlapped sliding windows of length $w = 100$. After preliminary tests, we use the same architecture for all autoencoder-based models. The autoencoders are a 5-layer MLP with 78-32-16-32-78 units. All latent layers are followed by ReLU activation function. The last layer is followed by a sigmoid function. We use the Adam optimizer to train all the neural networks, with an initial learning rate of 0.001, and a step-scheduler with a step of 0.5. All models are trained for 100 epochs, with a batch size of 64 in all experiments, and random parameter initialization. To limit the impact of random parameter initialization, we repeat each experiment five times and average the results over these five runs. Regarding Transformer-based anomaly detectors, we set the dimension of the embedding to 128 and we use 2-head attention units. In all our experiment, RESIST hyperparameter c is set as $c = 0.1\text{IQR}$ (cf. Equation 6). Similar to the validation protocol adapted by Ruff et al. [17], the competing methods hyperparameters are tuned on the predefined validation subset. To minimize hyperparameter selection problems, we select the optimal hyperparameters that maximize their validation Area Under the Curve of the Receiver Operating Characteristics (AUROC). This deliberately grants competing methods an advantage over RESIST. Lastly, all the experiments were run on a laptop equipped with a 12-core Intel i7-9850H CPU clocked at 2.6GHz and with NVIDIA Quadro P2000 GPU.

4.5 Results

Protocol 1 (P1) : Modular Composition of Co-attention and Self-attention Modules For a fair comparison, the three variants have the same architecture and configuration. The three variants use the *mixup* fusion strategy and are trained with the same robust loss: Geman-McClure loss. The only difference between the three variants, is the modular composition of co-attention and self-attention units. The experimental results of these 3 variants are shown in Figure 8. These first results highlight that the structure of RESIST encoder has a significant impact on the global performance, since varying the encoder composition of self and co-attention units is clearly reflected in the results.

Firstly, RESIST-SS, whose encoder is purely composed of a cascade of self-attention units, performs poorly compared to the other variants. Indeed, RESIST-SS is similar to a vanilla Transformer trained to reconstruct the input, using the robust Geman-McClure loss, and without considering the historical data. This variant shows the lowest AUROCs in this first set of experiments, with a mean equal to 76.6%, and with a large standard variation of 5%. The other two variants, which integrate intra and inter-sequence properties with co-attention units, globally show better results with reduced standard variations. This confirms

our first hypothesis (H1), in the sense that guiding the Transformer reconstruction with both intra-sequence properties and inter-sequence pairwise interactions with the history results in a more robust anomaly detector.

Furthermore, we note that the hybrid RESIST-SC reports higher AUROC, $80.5\% \pm 0.9$, compared to RESIST-SS, $78.8\% \pm 0.3$. This advantage is statistically significant, according to Welch's test with p-value = 0.05. This observation reveals that encoding the history with both self-attention and co-attention units is better than using only co-attention units. In RESIST-SC, the self-attention unit firstly extracts the intra-dependencies of the history. Then this first representation, which considers the history local context, is combined with the intermediate representation of the input. In contrast, RESIST-CC neglects history intra-sequence context and focuses only on inter-sequence properties. This result is consistent with other works in VQA [28]. In the following, we will use RESIST-SC encoder architecture as the basis for all next RESIST variants.

Protocol 2 (P2) : Robust Loss Function Similar to the protocol followed previously, all the variants share the same configuration, except the training loss function. The three variants have a hybrid siamese encoder, similar to RESIST-SC encoder. The results are reported in Figure 9. From this figure, we can see that the training loss function has a significant influence on the performance. We note that the results steadily improve when decreasing the robustness parameter α of the loss function $\rho(x, \alpha, c)$, defined in Section 3.2. Firstly, RESIST-MSE, trained with the common Euclidean distance, i.e., $\alpha = 2$, show the worst performance, with an AUROC around 74%. This result is in line with previous studies, which state that the mean-squared error is considerably influenced by outliers. Secondly, the Charbonnier loss, a.k.a, the pseudo-Huber loss, with $\alpha = 1$, does not improve the performance (cf. Figure 9). As shown in Figure 4 (left), even though the gradients of large error are reduced compared to the L2 loss, these gradients saturate to a non-zero value. That is, even though their contribution is slightly reduced, training contaminants still contribute to parameter optimization during the training. Nevertheless, when $\alpha \leq 0$, the gradient magnitude decreases and converges to 0, when the error is higher than the scale parameter c. As such, large errors are completely ignored and do not impact the training. The speed of converging to 0 clearly depends on the parameter α. The lower α, the higher the decreasing speed of large error gradients. Our results confirm this interpretation, in the sense that RESIST-GM, trained with Geman-McClure loss ($\alpha = -2$), exceeds RESIST-Cauchy, trained with Cauchy loss ($\alpha = 0$), by 2.6% on average. We can conclude that the second hypothesis (H2) is validated. Training RESIST with the Geman-McClure loss significantly reduces the impact of anomalies.

Protocol 3 (P3) : Comparison with Competing Methods In this section, we compare RESIST performance against common unsupervised anomaly detectors, presented in Section 4.3. We aim to demonstrate that RESIST outperforms these competing methods. The RESIST configuration used in this part is composed of the default architectures: a hybrid siamese encoder, i.e., the encoder

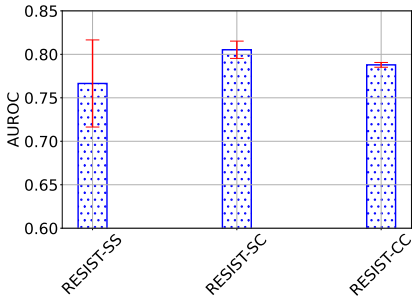

Fig. 8. Comparison between RESIST three variants: RESIST-SS, RESIST-SC, and RESIST-CC, on the CICIDS17.

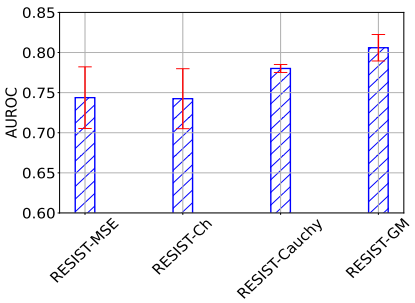

Fig. 9. Experimental results for RESIST trained with different loss functions, on the CICIDS17 dataset.

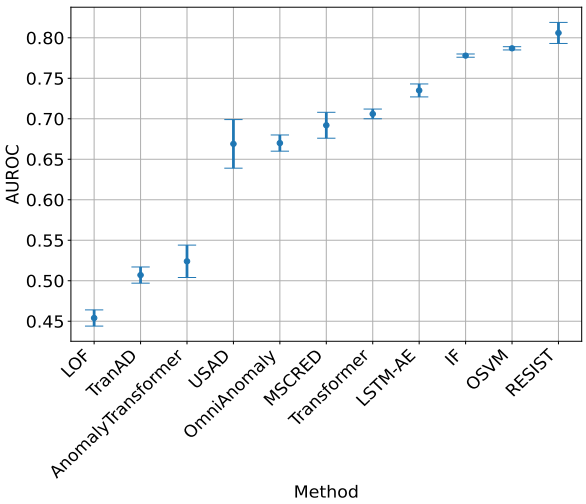

Fig. 10. Comparison between RESIST and the baselines on CICIDS17 dataset.

of RESIST-SC, the *mixup* fusion layer, and the robust Geman-McClure loss, with $c = 0.1\text{IQR}$. The experiment results are reported in Figure 10. Globally, RESIST achieves superior results compared to all the baselines, on the CICIDS17 dataset, with an average AUROC of $80.6\% \pm 1.3$. First, we note that RESIST is substantially more robust than vanilla Transformers. RESIST improves vanilla Transformer average AUROC by 10%. Second, the lowest results are reported with a density-based anomaly detector: LOF. Indeed, detecting contextual and collective outliers based on the local density of high-dimensional data is challenging. Surprisingly, Transformer-based anomaly detectors show poor performance on this dataset, even with a careful tuning of these architectures. TranAD and AnomalyTranformer report AUROCs of $50.7\% \pm 1.0$ and $52.4\% \pm 2.1$. This implies that these methods are significantly sensitive to training outliers, on this network traffic dataset. It is however difficult to explain such poor results, despite the careful fine-tuning of the hyperparameter on the dedicated validation subset. Third, classical anomaly detectors, i.e., IF and OSVM, give better results than

deep neural network-based anomaly detectors, including OmniAnomaly, MSCRED, Vanilla Transformer, and LSTM-AE. This observation ties well with the previous study conducted by Lai et al. [11]. We speculate that this might be due to the fact that the latter are developed for semi-supervised AD. Indeed, they assume that the training data are anomaly free. In the case of data pollution with anomalies, this assumption is not respected and consequently, these methods fail to distinguish both classes. Fourth, RESIST exceeds IF AUROC by 4% and OSVM AUROC by 3%, on average. These results demonstrate that RESIST is more robust than these competing anomaly detectors on the CICIDS17 dataset.

5 Conclusion and Perspectives

In this paper, we introduced RESIST, a Robust transformEr designed for unSupervised tIme Series anomaly detection. Thanks to the modular composition of self and co-attention units, RESIST learns to reconstruct each input sequence using a hybrid representation that aggregates both the local information that is specific to the current input and the global information shared with the history. Moreover, we proposed a robust training strategy that minimizes the Geman-McClure function, to reduce the impact of training contaminants. We extensively studied the contributions of RESIST components in the global performance, and the experimental evaluation on the CICIDS17 benchmark dataset confirmed that RESIST outperforms existing unsupervised anomaly detection.

References

1. Aggarwal, C.C.: Outlier Analysis. Springer International Publishing (2017)
2. Angiulli, F., Fassetti, F.: Distance-based outlier queries in data streams: the novel task and algorithms. Data Mining and Knowledge Discovery **20**, 290–324 (2009)
3. Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A.: USAD: UnSupervised Anomaly Detection on Multivariate Time Series. In: SIGKDD (2020)
4. Barron, J.T.: A General and Adaptive Robust Loss Function. In: CVPR (2019)
5. Blázquez-García, A., Conde, A., Mori, U., Lozano, J.A.: A review on outlier/anomaly detection in time series data. ACM Comput. Surv. (2021)
6. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying Density-Based Local Outliers p. 12 (2000)
7. Chaudhari, S., Mithal, V., Polatkan, G., Ramanath, R.: An Attentive Survey of Attention Models. Transactions on Intelligent Systems and Technology **12** (2021)
8. Choi, K., Yi, J., Park, C., Yoon, S.: Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines **9**, 120043–120065 (2021)
9. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T.: Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In: ACM SIGKDD. pp. 387–395 (2018)
10. Kurniabudi, Stiawan, D., Darmawijoyo, Bin Idris, M.Y., Bamhdi, A.M., Budiarto, R.: CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection **8**, 132911–132921 (2020)
11. Lai, K.H., Zha, D., Zhao, Y., Wang, G., Xu, J., Hu, X.: Revisiting Time Series Outlier Detection: Definitions and Benchmarks. NeurIPS (2021)

12. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: ICDM (2008)
13. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. CoRR (2016)
14. Najari, N., Berlemont, S., Lefebvre, G., Duffner, S., Garcia, C.: Robust Variational Autoencoders and Normalizing Flows for Unsupervised Network Anomaly Detection. In: AINA (2022)
15. Najari, N., Berlemont, S., Lefebvre, G., Duffner, S., Garcia, C.: RADON: Robust Autoencoder for Unsupervised Anomaly Detection. SIN pp. 1–8 (2021)
16. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification **80**, 4393–4402 (2018)
17. Ruff, L., Vandermeulen, R.A., Görnitz, N., Binder, A., Müller, E., Müller, K.R., Kloft, M.: Deep semi-supervised anomaly detection. ICLR (2020)
18. Sharafaldin, I., Lashkar, A.H., Ghorbani, A.: Intrusion Detection Evaluation Dataset (CICIDS2017), Canadian Institute for Cybersecurity (2017)
19. Shen, L., Li, Z., Kwok, J.T.: Timeseries Anomaly Detection using Temporal Hierarchical One-Class Network. NeurIPS (2020)
20. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In: SIGKDD (2019)
21. Tan, H., Bansal, M.: LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In: EMNLP-IJCNLP. pp. 5099–5110 (2019)
22. Tax, D.M., Duin, R.P.: Support Vector Data Description. Machine Learning (2004)
23. Tuli, S., Casale, G., Jennings, N.R.: TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. CoRR (2022)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: NeurIPS. vol. 30 (2017)
25. Wang, X., Pi, D., Zhang, X., Liu, H., Guo, C.: Variational transformer-based anomaly detection approach for multivariate time series. Measurement **191** (2022)
26. Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L.: Transformers in Time Series: A Survey. CoRR (2022)
27. Xu, J., Wu, H., Wang, J., Long, M.: Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. ICLR p. 20 (2022)
28. Yu, Z., Yu, J., Cui, Y., Tao, D., Tian, Q.: Deep Modular Co-Attention Networks for Visual Question Answering. In: CVPR. pp. 6274–6283 (2019)
29. Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., Chawla, N.V.: A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. AAAI (2019)
30. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: Beyond Empirical Risk Minimization. ICLR p. 13 (2018)
31. Zhang, Y., Hamm, N.A.S., Meratnia, N., Stein, A., van de Voort, M., Havinga, P.J.M.: Statistics-based outlier detection for wireless sensor networks. Int. J. Geogr. Inf. Sci. **26**, 1373–1392 (2012)
32. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. ICLR (2018)