Anomaly Detection for Wireless Communication Links via Data Integrity Modeling

Mahyar Nemati*, Jihong Park*, Moongu Jeon†, and Jinho Choi*
*School of Information Technology, Deakin University, Australia
E-mail: {nematim, jinho.choi}@deakin.edu.au, jihong.park@{deakin.edu.au, gist.ac.kr}
†School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Korea
E-mail: mgjeon@gist.ac.kr

***Abstract*—Wireless connectivity plays a crucial role in collecting data from a large number of devices and sensors for various Internet-of-Things (IoT) applications including supply chain management and personalized healthcare. In most IoT applications, for various reasons, a collected data set may include incorrect or corrupted data samples, which should be detected and removed. For example, malicious devices may send fake information or malfunctioned remote devices can respond improperly. In this paper, we study anomaly detection for wireless links, not data sets sent by devices, to see any anomalies in the physical and link layers associated with connected devices to a network. The resulting approach can be viewed as preemptive anomaly detection and be part of causal anomaly discovery that helps determine whether anomalies detected in a data set are caused by errors in wireless links or transceivers.**

I. Introduction

In the Internet-of-Things (IoT), a large number of devices including sensors and actuators are to be connected to the Internet for diverse applications including smart cities and factories [1] [2]. To allow devices to be connected, wireless connectivity plays a crucial role in the IoT and a number of solutions are studied [3]. Furthermore, non-terrestrial networks (NTN) will be to be part of IoT networks [4].

Anomaly detection is to detect samples that differ from most of the data or deviate from some form of normality, and has a wide range of applications such as fraud detection, intrusion detection, fault diagnosis, and so on [5] [6]. As with diverse applications, various approaches to anomaly detection have been studied, and some of classical approaches are well summarized in [6]. Deep learning is also applied to anomaly detection [7] [8].

As mentioned earlier, a number of IoT applications need to process data collected from a large number of devices [9] through wireless connectivity, which may cause various issues. Remote devices may malfunction and send partial or wrong information, which may feed to certain applications and result in undesirable outcomes. There can also be malicious devices that perform impersonation attacks by sending fake information. While any malicious behaviors of certain devices could be detected in application domains through data samples collected from them, it is also possible to detect them using radio frequency (RF) fingerprinting and mobility profiles [10]. As a result, in terms of a layered model for IoT systems [11], anomaly detection can take place on any layer.

In this paper, we study anomaly detection for wireless links using typical parameters in the physical and link layers using deep learning [12]. As an example, we consider satellite links. A set of parameters that are used to configure the physical and link layers becomes a data sample to train a model. The notion of data integrity modeling is adopted to train the model as a supervised learning. In particular, we consider a convolutional neural network (CNN) classifier [12] to detect any anomaly of a given wireless link. The resulting approach can be viewed as preemptive anomaly detection, and the outcomes can be used for causal anomaly discovery in the application domain. If a device is seen as a malfunctioned or malicious device, the data set uploaded by this device will be ignored and the device itself may be registered as a suspicious one as well.

II. Background

In this section, we present the background for the work in this paper.

A. Wireless Links

In this paper, we consider wireless links with the two bottom layers, namely physical and link layers so that anomalies in wireless channels, RF components, or transceivers can be detected.

In Fig. 1, we show the bottom two layers of communication systems [13]. The physical layer consists of a pair of modulator and demodulator, a pair of channel encoder and decoder, and a given channel. There are a number of parameters that can characterize the physical layer including the signal-to-noise ratio (SNR), modulation order, code rate, and so on [14]. Hybrid automatic request (HARQ) protocols are used in the link layer for reliable communications over unknown channels [15].

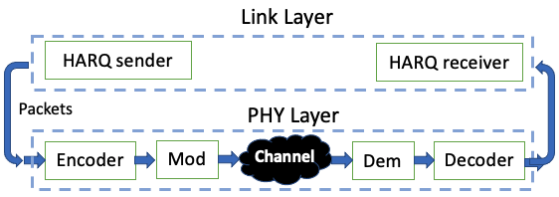

Fig. 1: An illustration of the bottom two layers, physical and link layers, of wireless communications.

When we consider a satellite link, the transmitter can be a ground station (GS) and the receiver can be a satellite. In this case, the channel can be characterized by the distance between the GS and the satellite, weather condition, and so on.

B. Anomaly Detection

Denote by $f_0(\mathbf{x})$ the distribution that generates the training vectors, i.e., $\mathbf{x}_{(i)} \sim f_0(\mathbf{x})$. In other words, $f_0(\mathbf{x})$ is the ground truth law of normal behavior. Then, the following two hypotheses can be considered:

$$\begin{aligned} H_0 : &\quad \mathbf{y} \sim f_0(\mathbf{x}) \\ H_1 : &\quad \mathbf{y} \sim f_1(\mathbf{x}), \end{aligned} \tag{1}$$

where $f_1(\mathbf{x})(\neq f_0(\mathbf{x}))$ is an anomaly distribution. As a default uninformative prior, a uniform distribution can be used for $f_1(\mathbf{x})$ [16]. Then, with known $f_0(\mathbf{x})$, a set of anomalies can be defined as $\mathcal{A}(\tau) = \{\mathbf{x} \in \mathcal{X} \,|\, f_0(\mathbf{x}) \leq \tau\}$, where $\tau \geq 0$ is a threshold. If a test vector $\mathbf{y}$ belongs to $\mathcal{A}(\tau)$, it can be seen as an anomaly. From (1), there are two types of decision errors: Type 1 (or false-alarm) error that results from choosing H_1 when a test vector follows $f_0(\mathbf{x})$; and Type 2 (or miss) error that results from choosing H_0 when a test vector follows $f_1(\mathbf{x})$.

If $f_0(\mathbf{x})$ is not available, but a dataset, machine learning approaches can be used for anomaly detection [17]. Provided that a large number of data samples, $\{\mathbf{x}_k\}$, are available, various deep learning approaches can be considered for anomaly detection [17]. For example, the unsupervised anomaly detection setting can be considered by taking $\{\mathbf{x}_k\}$ as unlabeled data samples.

III. Anomaly Detection for Wireless Links with CNN Classifiers

In this section, we propose an approach to anomaly detection for wireless links. As mentioned earlier, we consider anomalies in wireless channels, RF components, or transceivers, which may not be detectable in higher layers, e.g., network layer, through the notion of data integrity.

A. A Modeling of Data Integrity

Suppose that a data sample consists of M elements. Thus, the kth data sample can be expressed by the following vector:

$$\mathbf{x}_k = [x_{1,k} \ \ldots \ x_{M,k}]^{\mathrm{T}} \in \mathbb{R}^M. \tag{2}$$

We assume that $\mathbf{x}_k$ is a pattern/configuration obtained from an unknown distribution, $f_0(\mathbf{x})$, i.e., $\mathbf{x}_k \sim f_0(\mathbf{x})$. In addition, the features of $\mathbf{x}_k$ are correlated.

We consider a case where one of the elements of $\mathbf{x}_k$ has a finite support. To be specific, assume that $x_{M,k} \in \mathcal{L} = \{1, \ldots, L\}$, where L is finite. For the anomaly detection of wireless links, the number of (re-)transmissions of a data packet in HARQ can be $\ell_k = x_{M,k} \in \mathcal{L}$, which can be regarded as a label. Thus, $\mathbf{x}_k$ can be divided into $(\tilde{\mathbf{x}}_k, \ell_k)$, where $\tilde{\mathbf{x}}_k = [x_{1,k} \ \ldots \ x_{M-1,k}]^{\mathrm{T}}$ is the unlabeled data sample and ℓ_k is its corresponding label. The notion of data integrity towards anomaly detection is as follows: Taking $\tilde{\mathbf{x}}_k$ as an input and ℓ_k as an output, there should be a consistent relationship across all the pairs of input and output through a predictor or classifier, denoted by $\varphi(\cdot)$, if $\mathbf{x}_k$ is drawn from $f_0(\mathbf{x})$. That is, we can expect that $\ell_k = \varphi(\tilde{\mathbf{x}}_k) \in \mathcal{L}$, where $\varphi(\cdot)$ represents a classifier, with a high probability. In general, if a data sample drawn from a certain distribution is divided into a pair of input and output, the output should be predicted (or explained) by the input unless the input and output are independent. Therefore, for successful modeling of data integrity, elements in a data sample must be highly correlated.

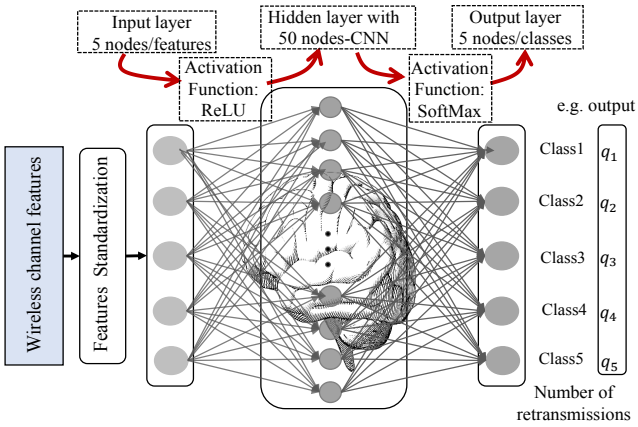

Fig. 2: Illustration of the multi-class classification using CNN. ReLU activation function used for output of input layer given 5 input nodes; and SoftMax activation function used for output of hidden layer. Each class represents number of (re-)transmissions while class5 indicates fail transmission.

In terms of wireless links, the input parameters such as modulation order, channel conditions, and code rate should predict the numbers of (re-)transmissions of data packets in HARQ. Thus, a classifier, $\varphi(\cdot)$, can be trained to predict the number of (re-)transmissions with a training data set. Anomalies in input parameters or outputs can cause differences between the predicted output and actual output, which can be used for anomaly detection.

B. Application of the CNN Classifier

Since there are a large number of parameters/features affecting wireless channels[1], classic machine learning classifiers are not applicable to classify different channel conditions. Therefore, deep learning [12] needed here. One of the typical deep learning algorithms based on neural network structures is CNN. A CNN classifier is a multilayered neural network with a special architecture to detect complex features in data and label them accordingly. For instance, CNN is used in image recognition, self-driving vehicles, powering vision in robots, and so on.

In this paper, we aim to build a CNN classifier capable of classifying satellite wireless channel conditions/features according to the number of required (re-)transmissions of data

[1]For example, carrier frequency, distance, shadowing, weather, Doppler frequency, modulation order, code rate, etc.

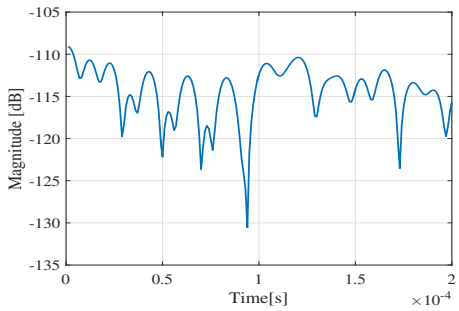

Fig. 3: Time-domain channel characteristics setup model. Modeled by Jakes model, Max Doppler: 50 kHz for LEO orbital speed of 10 km/s, sampling time: $1e-6$s, LEO-GS distance 225 km.

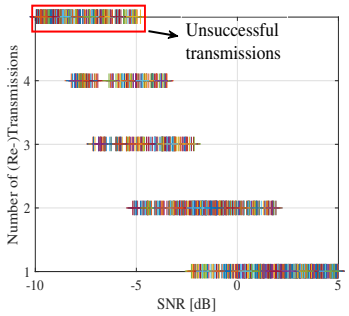

Fig. 4: Different configurations result in various number of (re-)transmissions, i.e., 300 configurations each with 200 realizations. Different realizations for a unique configuration, i.e., same SNR, may result in different number of (re-)transmissions.

packets in HARQ. A high-level overview of this classifier is shown in Fig. 2 that will be discussed in Section IV. In training phase, all we have to do is to feed this model with a set of ground truth data set to supervise the learning process. Once it is built, it can be used to predict the number of (re-)transmissions, i.e., $\ell_k = \varphi(\tilde{\mathbf{x}}_k) \in \mathcal{L}$, with the input features of different wireless channel configurations. Then, the difference between the predicted number of (re-)transmissions and the actual number of (re-)transmissions can be used for anomaly detection.

IV. Experiments

In this section, we present experimental setups and discuss how anomaly detection can be carried out with a CNN classifier for satellite links.

A. Setup

Suppose we have a large and well-chosen sample set of observations from which the normal behavior is to be learned. For this purpose, we first propose a satellite channel generator capable of producing $\mathbf{x}_k$, $k = 1, \cdots, K = 3000$ various wireless link configurations to mimic realistic behaviors of wireless links for each configuration. In particular, $\mathbf{x}_k$ consists of five wireless channel features as the inputs of our classifier, $\tilde{\mathbf{x}}_k = [x_{1,k}, \cdots, x_{M-1,k}]$, and the number of (re-)transmissions as its output, $\varphi(\tilde{\mathbf{x}}_k) = x_{M,k} (= \ell_k)$. Our simulator is designed according to a downlink scenario where a low-earth-orbit (LEO) satellite communicates with a GS. The height of LEO is set at 225 km and the coordinates of the GS follows a uniform distribution in the LEO's foot print area. The rest of parameters are given in Table I, unless otherwise specified. The five inputs which are the wireless channel features are set as (i) shadowing variance, (ii) distance between GS and LEO, (iii) pathloss exponent, (iv) modulation order, and (v) Rician K-factor[2].

[2]The satellite is the sender and the GS is the receiver. Wireless link is LoS and modeled as Rician fading and the receiver might be surrounded by various obstacles like in suburban or residential areas. Inputs are not limited to these and other parameters can be considered as well; however, in this study, we only focus on these most variable parameters.

This experimental setup allows up to four (re-)transmissions and further (re-)transmission request is considered *unsuccessful* transmission. As a result, there are 5 potential output classes, i.e., (re-)transmissions, for each configuration, i.e., $x_{M,k} = \ell_k \in \mathcal{L} = \{1, \cdots, 5\}$. That is, ℓ_k is the number of (re-)transmissions, while $\ell_k = 5$ means that the transmission is unsuccessful.

Fig. 3 shows a channel configuration generated for the kth configuration, i.e., $\mathbf{x}_k$. Besides, Fig. 4 shows the number of (re-)transmissions for each SNR corresponding to each configuration. Note that the actual SNR may not be available and it is just to show the outcome of each configuration setup. There are a total of $R = 200$ realizations for each configuration in Fig. 4 and each might have different numbers of (re-)transmissions due to the randomness behavior of wireless channels. As a result, for instance, in SNR interval $[-5, 0]$ dB, there are multiple possible (re-)transmissions/classes for a single configuration. Then, using the maximum likelihood (ML) estimation the label that has the largest likelihood can be found, given the data were observed in R realizations for the kth configuration. Thus, the probability of $x_{M,k} = s$ (re-)transmissions at the kth configuration is $\Pr(x_{M,k} = s \,|\, \{\mathbf{x}_k\})$, $s \in \{1, ..., 5\}$. Our CNN can be trained to predict the probability of the number of (re-)transmissions from the data set without anomalies.

In order to train our CNN classifier, we use 1000, 500 and 1500 wireless channel configurations/patterns as train-

TABLE I: Wireless channel generator parameters

System parameter	Corresponding value
Carrier frequency [GHz]	1.5
Satellite reference distance [m]	100
LEO-GS distance in LEO's footprint area [km]	$\sim U(150, 300)$
Pathloss exponent-Shadowed urban area	$\sim U(2.7, 4.5)$
Shadowing variance [dB]	$\sim U(0.5, 2.5)$
Max. Doppler frequency [KHz]	50
Sampling time	10^{-6}
Number of samples	200
Modulation order	QPSK, 16QAM, 64QAM
Noise power [dBm]	-195
Rician K-factor	$\sim U(-20, -16)$

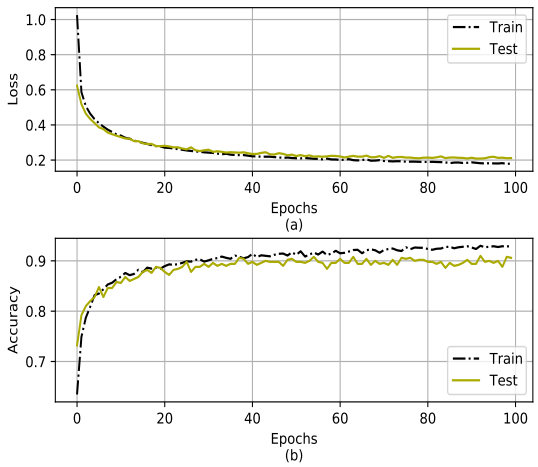

Fig. 5: (a) Multi-class *Categorical_Cross_Entropy* loss and (b) classification accuracy of the proposed trained CNN classifier. The classification loss and accuracy of our CNN classifier for both training and test data sets are below 0.25 and above 90%, respectively.

ing, validation, and test data sets, respectively. As shown in Fig. 2, we use one layer multi-class CNN model with 50 nodes to train and model our classifier. For training the classifier, we use $Keras$ model and $kernel\_initializer$ is set $he\_uniform$. It is also noteworthy that since inputs of data sets, $x_{1,k}, \cdots, x_{M-1,k}$, are of different types, they need to be scaled to be standardized. For this purpose, we use $StandardScaler().fit\_transform(\mathbf{x}_k)$ function in Python.

Figs. 5 (a) and (b) show multi-class *Categorical_Cross_Entropy* loss and classification accuracy of our CNN classifier for both training and test data sets, respectively, which are below 0.25 and above 90%, respectively.

B. Anomaly Detection Evaluation

The CNN classifier in Fig. 2 with the softmax activation function is used to predict the probability of the number of (re-)transmissions, which is called a proposed distribution. Suppose that the softmax output of a classifier is given by vector $\mathbf{q}_k = [q_{1,k}, \cdots, q_{5,k}]$. For convenience, we omit index k. Then, the output of the CNN classifier as a soft-decision is given by

$$q_s = \Pr(Y = s \,|\, \{\mathbf{x}_k\}), \qquad s \in \{1, ..., 5\}, \tag{3}$$

where Y represents the predicted output.

Fig. 6 compares the hard-decisions of the classifier predictions with the ground truth labels of 20 configurations for different scenarios where an anomaly exists in one of the features in each subplot. For example, in Fig. 6 (a), there are no anomalies in the data set and the predictions agree with the actual numbers of (re-)transmissions. However, as shown in Fig. 6 (b)–(f), with anomalies in different inputs, the predictions have differences from the actual numbers of (re-)transmissions. In particular, we can see that anomalies

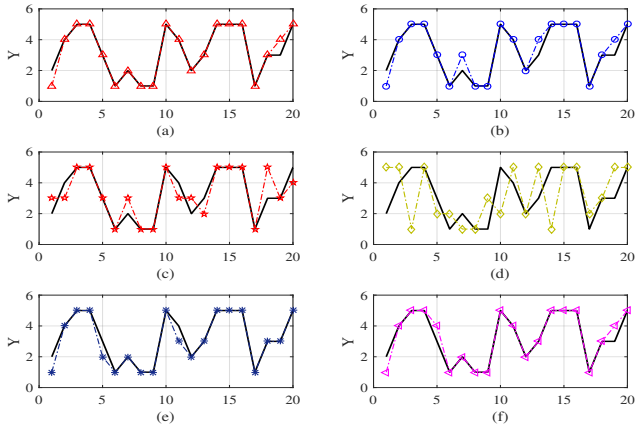

Fig. 6: Representation of anomaly detection. Curves with markers indicate predictions. Solid curve is the ground truth labels. (a) normal data set without anomaly, (b) anomalous shadowing variance, (c) anomalous distance, (d) anomalous pathloss exponent, (e) anomalous modulation, and (f) anomalous Rician K-factor. Standard deviation for all anomalies is set as $\sigma = 1$.

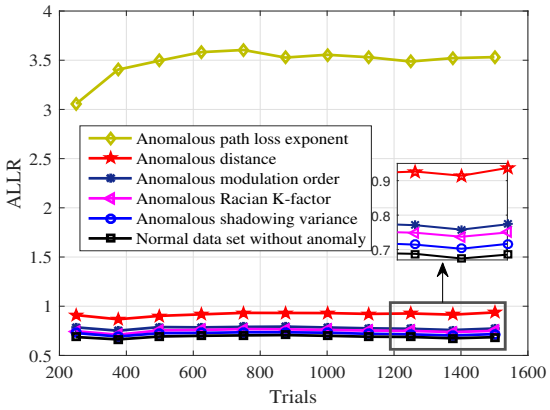

Fig. 7: ALLR between realizations and proposed distribution when there are anomaly in inputs. Standard deviation for all anomalies is set as $\sigma = 1$.

in path loss exponent lead to large differences between the prediction and actual number of (re-)transmissions.

For anomaly detection, it is necessary to compare the soft-decisions in (3) with actual number of (re-)transmissions. To this end, we can use the augmented log-likelihood ration (ALLR) derived in Appendix, which allows to compare a proposed distribution (i.e., the soft-decisions of the classifier) and the realizations (i.e., actual number of (re-)transmissions). As discussed in Appendix, ALLR is based on the Kullback-Leibler (KL) distance [18] and can be used to measure the distance between a proposed distribution and realizations or samples drawn from an unknown distribution. If the samples are drawn from the proposed distribution, the values of ALLR are expected to be small. Otherwise, the value of ALLR will be large. Thus, ALLR can be used as test statistics for anomaly detection.

Fig. 7 shows the ALLR between different proposed distributions and the actual realizations when $R = 200$. As

shown in Fig. 7, the black curve with square markers shows the difference between the data-driven distribution, $\mathbf{p}_k$, from realizations and its proposed distribution $\mathbf{q}_k$ when there is no anomaly in the data set is about 0.7. Note that this difference is different from the *Categorical_Cross_Entropy* loss shown in Fig. 5 (a). The blue curve with circle markers depicts the difference when there is an anomaly in shadowing variance. It is illustrated that anomaly in the shadowing variance slightly increases the difference metric. Similarly, we see the impact of anomaly in different input features: Rician K-factor, modulation order, distance and path-loss exponent. Note that in this experiment, anomalies are random with the same distribution as the original data set, i.e., standard deviation is set $\sigma = 1$. As a result, we conclude that anomaly in the path loss exponent has the highest impact on the ALLR. In addition, larger deviations, i.e., $\sigma > 1$, result in larger ALLRs.

At this stage, we aim to evaluate the false-alarm and miss probabilities with respect to the threshold τ. Therefore, the objective is to illustrate the diagnostic ability of false alarm and miss events (error type 2) as the discrimination threshold of the system varies. Fig. 8 illustrates the receiver operating characteristic (ROC) curves to evaluate the behavior of the empirical distribution at all thresholds. For this experiment the anomalous path loss exponent with three differed deviations, $\sigma = 8, 12, 16$, is considered for error type 2. Note that the y-axis is correct detection probability ($= 1-$ miss detection probability). Each point on the ROC curve represents a different trade-off between false alarm and miss detection. Therefore, the ROC curve provides a convenient gestalt of the trade-off between miss detection and false alarm performance for different anomalous data sets. Fig. 8 shows that the area under the Curve (AuC) increases when deviation in the anomalous data set increases. As a result, the likelihood of anomaly detection with the proposed ALLR test statistic becomes larger.

V. Conclusions

In this paper, based on the modeling of data integrity, we studied anomaly detection for wireless links. Key parameters and performance measures in the physical and link layers of wireless networks have formed data samples that can be used to see data integrity through the input and output relationship of a CNN classifier. With a trained CNN classifier, anomaly detection of wireless links has been performed through the differences between the predicted output and actual output. To measure the differences, we also proposed the ALLR based on KL distance. In the presence of anomalies in test data samples, the ALLR tends to have large values, while its value is close to 0 for data samples without anomalies.

Acknowledgment: This research was supported by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (DP200100391) and by the Korean government (MSIT) through Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.2014-3-00077, AI National Strategy Project).

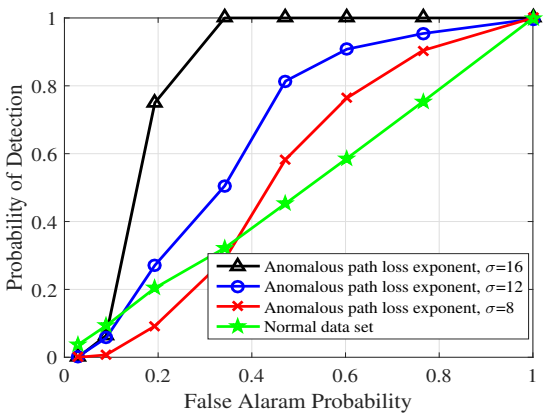

Fig. 8: Receiver operating characteristic curves showing the performance of the proposed model at all thresholds.

Appendix
Augmented Log-Likelihood Ratio

In this appendix, we discuss a distance measure that allows us to see the difference between a set of outcomes from experiments and a proposed distribution.

The information divergence or KL distance between two distributions, $p(x)$ and $q(x)$, [18] is given by

$$D(\mathbf{p}||\mathbf{q}) = \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}. \tag{4}$$

The divergence can be used to see the difference between two distributions.

The likelihood function of a parameter vector of interest, denoted by $\mathbf{p}$, for a given distribution is given by

$$f(X|\mathbf{p}). \tag{5}$$

The likelihood measures the goodness of fit of a statistical model to a sample of data, X, for given values of the unknown parameters $\mathbf{p}$.

Example 1. *Consider a categorical distribution of* $X \in \{1, \ldots, K\}$ *[19] as follows:*

$$f(X|\mathbf{p}) = \prod_{k=1}^{K} p_k^{\mathbb{1}[X=k]}, \tag{6}$$

where $\mathbb{1}[\cdot]$ *is the indicator function that is defined as*

$$\mathbb{1}[X = k] = \begin{cases} 1, & \text{if } X = k \\ 0, & \text{o.w.} \end{cases} \tag{7}$$

Let X_m *be the* m*th observation of* X*. Then, the ML estimate of* $\mathbf{p}$ *is given by*

$$\begin{aligned} \hat{\mathbf{p}} &= \underset{\mathbf{p}\in\mathcal{A}_K}{\operatorname{argmax}} \prod_m f(X_m|\mathbf{p}) \\ &= \underset{\mathbf{p}\in\mathcal{A}_K}{\operatorname{argmax}} \prod_k p_k^{c_k}, \end{aligned} \tag{8}$$

where $\mathcal{A}_K$ represents the K-simplex and $c_k = \sum_m \mathbb{1}[X_m = k]$. After some manipulations, we can show that

$$\hat{p}_k = \frac{c_k}{\sum_k c_k}. \tag{9}$$

By combining the KL divergence and ML estimate, we can find a way to see the difference between a dataset (obtained from experiments) and a proposed distribution.

First, let consider the following hypothesis testing:

$$\mathbf{x}_m \sim \mathbf{p} \text{ versus } \mathbf{x}_m \sim \mathbf{q}. \tag{10}$$

As a test statistics for hypothesis testing, the log-likelihood ratio (LLR) can be considered, which is given by

$$\mathsf{LLR}(\mathbf{p};\mathbf{q}) = \log \frac{\prod_m f(\mathbf{x}_m|\mathbf{p})}{\prod_m f(\mathbf{x}_m|\mathbf{q})}. \tag{11}$$

For the categorical distribution as an example, the LLR can be found as

$$\begin{aligned}\mathsf{LLR}(\mathbf{p};\mathbf{q}) &= \sum_k \sum_m \mathbb{1}[X_m = k] \log \frac{p_k}{q_k} \\ &= \sum_k c_k \log \frac{p_k}{q_k}.\end{aligned} \tag{12}$$

Then, the following normalized LLR can be considered:

$$\frac{\mathsf{LLR}(\mathbf{p};\mathbf{q})}{\sum_k c_k} = \sum_k \hat{p}_k \log \frac{p_k}{q_k}. \tag{13}$$

If $\mathbf{p}$ is replaced with its ML estimate, the normalized LLR becomes

$$\frac{\mathsf{LLR}(\hat{\mathbf{p}};\mathbf{q})}{\sum_k c_k} = \sum_k \hat{p}_k \log \frac{\hat{p}_k}{q_k} = D(\hat{\mathbf{p}}||\mathbf{q}), \tag{14}$$

which shows the relationship between the KL divergence and the normalized LLR.

Based on (14), define the following quantity, namely augmented LLR (ALLR):

$$\begin{aligned}\mathsf{ALLR}(\{\mathbf{x}_m\};\mathbf{q}) &= \log \frac{\max_{\mathbf{p}} \prod_m f(\mathbf{x}_m|\mathbf{p})}{\prod_m f(\mathbf{x}_m|\mathbf{q})} \\ &= \log \frac{\prod_m f(\mathbf{x}_m|\hat{\mathbf{p}})}{\prod_m f(\mathbf{x}_m|\mathbf{q})},\end{aligned} \tag{15}$$

which allows us to measure the difference between the proposed distribution and a given dataset. Clearly, thanks to the maximization in the numerator, the ALLR is non-negative, i.e.,

$$\mathsf{ALLR}(\{\mathbf{x}_m\};\mathbf{q}) \geq 0. \tag{16}$$

As shown above, for a categorical distribution, the ALLR is proportional to the KL divergence.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, pp. 1645–1660, Sept. 2013.

[2] J. Kim, J. Yun, S. Choi, D. N. Seed, G. Lu, M. Bauer, A. Al-Hezmi, K. Campowsky, and J. Song, "Standard-based IoT platforms interworking: implementation, experiences, and lessons learned," *IEEE Communications Magazine*, vol. 54, pp. 48–54, July 2016.

[3] J. Ding, M. Nemati, C. Ranaweera, and J. Choi, "IoT connectivity technologies and applications: A survey," *IEEE Access*, vol. 8, pp. 67646–67673, 2020.

[4] M. Conti, A. Guidotti, C. Amatetti, and A. Vanelli-Coralli, "NB-IoT over non-terrestrial networks: Link budget analysis," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, 2020.

[5] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, pp. 85–126, Oct 2004.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, July 2009.

[7] A. Goel and P. Moulin, "Locally optimal detection of stochastic targeted universal adversarial perturbations," 2020.

[8] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, Mar. 2021.

[9] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big IoT data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[10] M. Barbeau, J. Hall, and E. Kranakis, "Detecting impersonation attacks in future wireless and mobile networks," in *Secure Mobile Ad-hoc Networks and Sensors* (M. Burmester and A. Yasinsac, eds.), (Berlin, Heidelberg), pp. 80–95, Springer Berlin Heidelberg, 2006.

[11] M. Weyrich and C. Ebert, "Reference architectures for the Internet of Things," *IEEE Software*, vol. 33, no. 1, pp. 112–116, 2016.

[12] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. http://www.deeplearningbook.org.

[13] J. Proakis, *Digital Communications*. McGraw-Hill, fourth ed., 2000.

[14] E. Biglieri, *Coding for Wireless Channels*. New York: Springer, 2005.

[15] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1995.

[16] I. Steinwart, D. Hush, and C. Scovel, "A classification framework for anomaly detection," *Journal of Machine Learning Research*, vol. 6, no. 8, pp. 211–232, 2005.

[17] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, 2021.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. NJ: John Wiley, second ed., 2006.

[19] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.