Interpretable Unsupervised Anomaly Detection For RAN Cell Trace Analysis

Ashima Chawla*, Paul Jacob*, Saman Feghhi†, Devashish Rughwani†,
Sven van der Meer†, Sheila Fallon*
*Software Research Institute, Athlone Institute of Technology, Athlone, Ireland
ashima.chawla@ericsson.com, pjacob@ait.ie, sheilafallon@ait.ie
†Ericsson, Network Management Lab, Athlone, Ireland
saman.feghhi@ericsson.com , devashish.rughwani@ericsson.com, sven.van.der.meer@ericsson.com

***Abstract*—The high complexity of modern communication networks requires an increasing degree of automation for performance and fault management tasks. A key task is the classification and identification of anomalous operation modes (and faults). This is important to separate them from normal operation conditions. In addition, these diagnoses should be interpretable by domain experts to (a) gain acceptance by these experts and (b) support effective root cause analysis and localisation. In this paper, we investigate the analysis of multivariant network data in order to identify anomalous data instances. Root cause analysis benefits from this by filtering features whose values lead (to some extent) to such anomalies. We are using Deep Neural Networks (DNNs), a powerful tool for anomaly detection in the telecommunication domain. We demonstrate the effectiveness of autoencoders (an unsupervised technique) to detect multivariant anomalies and anomalous features. To overcome the black box nature of neural networks (and thus increase their acceptance by domain experts), we apply SHapley Additive exPlanations (SHAP), which are used to explain the output model of a neural network.**

***Index Terms*—Autoencoders, SHAP, DNNs, Correlation**

I. Introduction

The rising volume of network data [1] due to the generation of approximately 1 million events every second in mobile Radio Access Network (RAN) data has contributed to increased momentum in network analysis and troubleshooting research. The importance of intelligent monitoring of network performance management is imperative for network operators. To ensure infrastructures provide a high level of robustness to customers, rule-based systems, based on domain knowledge, were implemented to analyze and detect the anomalies across multiple features. However, these traditional rule based network application approaches fail when exposed to new previously unseen complex patterns.

A lot of effort is spent doing analysis of cell trace files to understand why network fails and what is the reason for the bad network. In such scenarios, DNNs [2]-[5] have shown impressive results and state-of-the-art performances in the field of telecommunications. In this paper we demonstrate the effectiveness of our proposal whereby we provide human interpretable multi-anomaly detection. This multi-anomaly functionality allows for the identification of multiple anomalous cell traces. The contributions of this paper are as follows:

- To model the structure of normal trace data in order to subsequently identify anomalous cell traces.
- To highlight related cell trace event values and combinations of events that cause the cell traces to be anomalous. This is done using both a global approach depending on correlation values between features and local interpretations which examine effects on individual anomalous traces.

With the proposed unsupervised approach, we provide an algorithm to provide network insights which complement the existing autoencoder anomaly detection mechanism [16]. The proposed algorithm addresses the drawback of DNNs "black box" nature, which does not provide interpretability of an output. It focuses on the importance of interpretation of AI models and offers an explainable solution for stakeholder experts to better understand the reason behind decisions made by model. The solution not only detects signature based multivariate anomalies but also anomalies which could be missed by experts who rely on domain rules. This provides actionable insights in terms of developing and improving automated troubleshooting across anomalies. Additionally, this framework is flexible enough to integrate different sources of data (Evolved Packet Core: EPC) into the same anomaly detection algorithm. This would result in providing an automated cell trace troubleshooting across multivariate features using a deep learning approach.

The rest of the paper is organized as follows. Section II outlines the related work in neural networks and anomaly detection domain. In Section III, we review the methodology including stacked autoencoders and SHAP [7] algorithm to detect the most contributing features to anomalous features. Section IV outlines our data collection, experimental setup and results evaluation. Section V concludes the paper.

II. Related Work

Recently many researchers have applied anomaly detection mechanisms [8]-[10], [15] to network data, and these have been quite successful in capturing anomalous behaviours.

978-3-903176-31-7

In this section, we review the generic anomaly detection algorithms used by researchers in telecommunication domain. In [11], the authors empirically evaluated the analytic engine called as Autoregressive Integrated Moving Average (ARIMA) along with Java functions. The engine takes time series data stream and further calculates the anomaly score and anomaly probability on network traffic throughput dataset.

The authors of [12] proposed a novel self-adaptive deep learning framework for detecting anomalies in 5G RAN and EPC. The Autoencoder based Anomaly detector present in RAN quickly captures the anomaly symptoms, which is further evaluated by Long Short Term Memory (LSTM) recurrent network model in EPC. [14] proposed an approach to detect probability of radio anomalies using LSTM based RNNS. The proposed framework calculates the error distribution using a parametric multivariate Gaussian distribution function to evaluate the model performance.

In [13], the authors intended to capture the malicious network attacks present in a network through parallel clustering technique. They implemented MCOD (Micro-Cluster based Outlier Detection) classifier to identify outliers over streaming data. Recently the idea of providing interpretation to neural network model was introduced lately in 2019 by [7]. The paper emphasized on the importance of how hard is to explain the anomalies extracted by outlier detection algorithms. The illustration in the paper provided a framework to explain the instances with high reconstruction error.

To the best of our knowledge none of this previous work has focused on doing multivariate anomaly detection using network cell trace data. We note that part of this paper's contribution enables the use of interpretability of data-driven solutions.

III. Methodology

A. Autoencoders

Autoencoders [6] are a type of neural networks trained to learn a compressed representation of the input data and to reconstruct the input from this representation. Anomaly detection is then carried out by identifying instances which can not be accurately reconstructed, and these are identified as anomalous instances.

In effect, autoencoders build a model to identify hidden structures and useful features from unlabelled input data. They work on the concept of learning interesting hidden layer representations by limiting the number of neurons in the hidden layer. An Autoencoder consist of an encoder, a latent space representation and a decoder. The encoder learns a compressed vector representation of the input data, and the decoder uses this information to reconstruct the input from its hidden representation. The Encoder-Decoder learns to reconstruct normal behavior, and thereafter uses the reconstruction error to detect anomalies. The model jointly trains the encoder and decoder and applies backpropagation [17] to minimize the reconstruction error. The Mean Squared Error (MSE) is used as the loss function for training the model and depends on the the difference between the actual (x) and reconstructed (x') attribute values. The Root Mean Squared Error (RMSE) [18] is used as an evaluation metric to calculate the squared error difference between the observed (x) and predicted (x') values of a set of n.

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(x_i - x_i^{'})^2} \tag{1}$$

B. Sparse Autoencoders

Sparse Autoencoders [19] impose a sparsity constraint where they learn feature representations in the data even for a large number of hidden units. We add an L1 norm regularization [21] which learns sparse and enlarged representation of the input data. In Keras [20], this can be done by adding an activity regularizer to the Dense layer. In our proposed algorithm, we have implemented sparse autoencoder with L1 regularization of 3e-7 with MSE loss which learnt better representation than vanilla autoencoders.

C. SHAP

Normally an Artificial Neural Network (ANN) is considered a "black box" that cannot provide easily interpretable insights into the relationship between input and output. Particularly, when there are high dimensional data and layers, it becomes much harder to understand the reason behind an anomaly without a proper explanation. To overcome this, a relatively new technique in machine learning, known as a 'SHAP mechanism', supports interpretation of the neural network, or any complex machine learning model, by determining how input features contribute to the value of output features.

The SHAP framework [22] unifies methods such as LIME [23] and DeepLIFT [24] under the class of additive feature attribution methods. [7] demonstrate how the game theory based SHAP framework could be used for explaining anomalies detected by an autoencoder. Kernel SHAP is a method that uses a special weighted linear regression mechanism. It uses SHapley values from game theory to explain a specific prediction by assigning an importance value (SHAP value) to each contributing feature.

The KernelExplainer function takes as input the model, an instance of anomalous input x and a set of background instances. For a particular feature x_i' it calculates a set of SHAP values which measures the importance of each of the features $x_1, x_2....x_n$ in predicting x_i'. This local interpretability can be represented graphically by using force plots which enables us to pinpoint the SHAP value of features with respect to each other.

Algorithm 1 explains the process how SHapley values are being interpreted to explain the cell trace anomalies using SHAP output. The Kernel Explainer SHapley function outputs allFeatures, shapValue and inputValue for anomalous cell trace. The shapValue corresponds to the impact of each feature

Algorithm 1 Kernel Explainer Interpretation

Input: trainData - Background instances
f - Autoencoder model
x - Anomalous cell trace to be explained
Output: SHAP values saved in CSV file and Top Contributors Feature list for stakeholders.

```
procedure ANOMALIESINTERPRETATION
    allFeatures, shapValue, inputValue ←
        KernelExplainer.getshapValues(f, trainData, x)
    sortFeatures ← sortDecrease[allFeatures, shapValue]
    for i in sortFeatures.length: do
        feature ← sortFeatures[i]
        featureEffect ← feature.shapValue
        featureValue ← feature.inputValue
        if featureEffect > 0 then
            print2CSV(featureValue, featureEffect)
        end if
        topContributor ← sortFeatures[0]
    end for
    print Unique topContributor
end procedure
```

on all other features in the cell trace. The features are sorted based on the shapValue in descending order. Then, the positive shapValue features and their corresponding inputValue are stored in the CSV output as Effect (*featureEffect*) and Value (*featureValue*) respectively. Lastly, it displays the top unique contributing feature.

IV. Evaluation and Results

A. Data Stream Processing

The cell trace data for this study is collected from multiple Evolved NodeB (eNodeBs) as files. There are a mix of periodic and procedural events in each file. Periodic events are produced at fixed intervals capturing key metrics. Procedural events are produced every time the node or UE performs an operation e.g. UE Context Setup, Handover Preparation, etc. Fig 1 illustrates the data stream solution architecture which collects events from eNodeB. These event streams are parsed by the events parser and sent upstream to the correlation engine to produce a coherent dataset. This dataset is collection of session records which includes UE, eNodeB and EPC. The output from this process produced cell trace files which provided the input data for anomaly detection using an autoencoder and subsequent interpretation using correlation and SHAP values.

B. Experiment Settings

All experiments have been conducted on the computational machine which includes Intel® Core™ i7-8650U CPU @ 1.90GHz × 8, 25.8 GiB memory, Ubuntu 18.04.4 LTS operating system. The Keras python library 2.3.1 was used for running on top of a source build of Tensorflow 2.1.0.

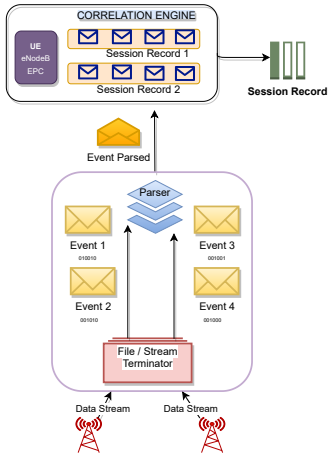

Fig. 1. Data Stream Processing

The sparse autoencoder model architecture comprises of an input layer, 3 hidden layers and finally an output layer. (1) 3 hidden layers with 100-50-100 neuron units, (2) 300-150-300 neuron units, (3) 400-20-400 neuron units and (4) 500-250-500 neuron units. We matched the cardinality of the input and output layer. We used the above 4 deep stacked perceptron models to train approximately 2 million rows by tweaking the hyperparameters. The model is trained in minibatches gradient descent with the set of normal cell traces correlated dataset. The model optimizes the Mean Squared Error (MSE) loss using Adam optimizer [26] with a learning rate of 0.0008, followed by non linear activation relu function, defined as:

$$f(x) = max(0, x) \tag{2}$$

Callback with an early stopping mechanism [25] of validation loss has been used as one of the regularization technique to avoid model overfitting. This technique stops training at the point when performance on a validation dataset stops improving. The model trained after tweaking the hyperparameters is then used to reconstruct the input data. The reconstruction errors calculated are then used to detect the anomalies in the test set correlated cell trace file.

C. Anomaly Detection

We had access to the test data set where anomalous instances were labeled. As the RMSE was to be used as the key metric error indicator we first obtained the average RMSE for normal and anomalous instances. RMSE for normal instances and anomalous instances are 0.0197 and 0.2215 respectively. This seems to suggest that RMSE could be used to identify anomalous instances. Then, True Positive and False Positive rates are calculated for different threshold values to find anomalous cell trace instances. The AUROC [27] (Area Under Receiver Operating Characteristic) and AUPR (Area Under Precision-Recall) are holistic metrics that summarize the performance of a detection method across

multiple thresholds. Table I shows performance values for several model configurations. The best-fitted model with 500-250-500 neuron units produces 0.96 AUC with 0.70 AUPR.

TABLE I
PERFORMANCE VALUES FOR MODELS

Parameters	Neurons	Learning Rate	AUROC	AUPR
46,631	100-50-100	0.0008	0.88	0.50
199,531	300-150-300	0.0008	0.95	0.68
305,981	400-200-400	0.0008	0.94	0.60
432,431	500-250-500	0.0008	**0.96**	**0.70**

D. Global Interpretation - Correlation Coefficients

Table II shows global interpretation results i.e. strongly correlated features in the top anomalous rows detected by the autoencoder model. For the purposes of evaluation, anomalous rows were sorted based on the root mean squared error obtained and top 100 anomalous cell traces were obtained for further analysis. To get a representation of the relationships between features for these top anomalous cell traces we calculated the pairwise correlation coefficients between features.

TABLE II
TOP FEATURE CORRELATIONS

Feature 109	Feature 173	1.0
Feature 108	Feature 173	1.0
Feature 109	Feature 111	1.0
	Feature 110	1.0
Feature 108	Feature 110	1.0
	Feature 111	1,0
Feature 111	Feature 173	1.0
Feature 79	Feature 80	1.0
Feature 75	Feature 80	1.0
	Feature 79	1.0

Fig. 2 highlights the collection of feature attributes in the top 100 anomalous cell traces. These features are the top correlations obtained from the strongly correlated features list.

E. Local Interpretation - SHAP values

In this section we review local interpretation results generated by SHAP. The KernelSHAP Explainer as described in section III (c) outputs features, SHAP values and input value for anomalous rows. Table III illustrates the contribution of input features to a sample anomalous row where features with higher Effect value indicate higher impact on the entry being anomalous. This particular entry indicates that a bad cell trace happening due to handover failure could be caused by no available neighboring RSRP (**Feature 109**), ERAB fail and data lost (**Feature 108**), high latency time (**Feature 75**) and low aggregated time for downlink delay in MAC layer (**Feature 173**). Our proposed solution validated that ERAB fail and data lost was the most important feature for failed handover, which contributed for bad RSRP, increased latency and low aggregated downlink delay.

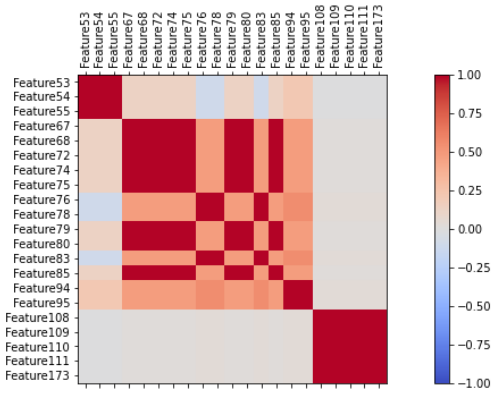

Fig. 2. Collection of Anomalous Features

Similar to this example, the SHAP values can help explain the impact of features to each anomalous entry detected by the proposed autoencoder model.

TABLE III
KERNEL EXPLANATION- SHAPLEY VALUE

Feature 108	Feature 109	Feature 75	Feature 173
Value = 0	Value = -1	Value = 2147483648	Value = 806
Effect=**0.050**	Effect=0.014	Effect=0.0054	Effect=0.016

V. CONCLUSION AND FUTURE WORK

This paper outlined the design of a neural network model and associated interpretation techniques with a view to providing insights and support for the troubleshooting of network cell trace data without prior domain knowledge. This automated solution can save a lot of manual troubleshooting, with a resulting reduction in costs and resources.We have been able to show that sparse autoencoders can be trained to reconstruct the input data and reconstruction error can be used to identify previously unseen anomalies. The results obtained were impressive and we achieved an AUROC score of 0.96. One possible threshold for anomaly detection gives us a true positive rate of 0.9 and a false positive rate of 0.1. The AUPR curve was 0.70.

Global interpretability of a set of anomalous data is supported using pairwise correlations of features. The local explanations provide a deeper insight for every anomalous trace. SHAP values identify the features and the extent to which they have caused the anomaly.

Additionally, this work includes a more systematic scalable evaluation of the our system's utility in network performance management. Future work includes applying this on multiple incident anomaly detection to work with larger datasets on GPU machines. The interpretation of data-driven solutions concept and its application to network performance management will be an important future research area for Ericsson.

ACKNOWLEDGMENT

This work is funded by Irish Research Council Enterprise Partnership Scheme Postgraduate Scholarship 2020 under Project EBPPG/2019/76.

REFERENCES

[1] Al Mtawa Y, Haque A, Bitar B. The mammoth Internet: Are we ready?. IEEE Access. 2019 Sep 12;7:132894-908.

[2] Castanedo, F., Valverde, G., Zaratiegui, J. and Vazquez, A., 2014. Using deep learning to predict customer churn in a mobile telecommunication network.

[3] Komar, M., Sachenko, A., Golovko, V. and Dorosh, V., 2018, May. Compression of network traffic parameters for detecting cyber attacks based on deep learning. In 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT) (pp. 43-47). IEEE.

[4] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. and Kavukcuoglu, K., 2016, June. Asynchronous methods for deep reinforcement learning. In International conference on machine learning (pp. 1928-1937).

[5] Arjovsky, M., Chintala, S. and Bottou, L., 2017. Wasserstein GAN. stat, 1050, p.26.

[6] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," 2018,arXiv:1802.09089. [Online]. Available: http://arxiv.org/abs/1802.09089.

[7] L. Antwarg, B. SHAPira, and L. Rokach. Explaining anomalies detected by autoencoders using SHAP. arXiv preprint arXiv:1903.02407, 2019.

[8] Li Y, Ma R, Jiao R. A hybrid malicious code detection method based on deep learning. International Journal of Security and Its Applications. 2015 May;9(5):205-16.

[9] Fadlullah ZM, Tang F, Mao B, Kato N, Akashi O, Inoue T, Mizutani K. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Communications Surveys & Tutorials. 2017 May 23;19(4):2432-55.

[10] Li H, Ota K, Dong M. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. IEEE network. 2018 Jan 26;32(1):96-101.

[11] M. Wang and S. Handurukande, "Anomaly Detection for Mobie Network Management," International Journal of Next-Generative Computing, vol. 9, no. 2, pp. 80-97, 2018.

[12] L. F. Maino, A. L. P. Gomez, F. J. G. Clemente, M. G. Perez and G. M. Perez, "A Self-Adaptive Deep Learning-Based System for Anomaly Detection in 5G Networks," in IEEE, 2018.

[13] F. Kieran, F. Enda, C. Paul and A. Abir, NetFlow Anomaly Detection Though Parallel Cluster Density Analysis in Continuous Time-Series, 2019.

[14] T. O'Shea, T. C. Clancy and R. W. McGwier, "Recurrent Neural Radio Anomaly Detection," in arXiv:1611.00301v1, 2016.

[15] Al Mamun, S.A. and Valimaki, J., 2018. Anomaly detection and classification in Cellular Networks using automatic labeling technique for applying supervised learning. Procedia Computer Science, 140, pp.186-195.

[16] Bengio, Y., Courville, A. and Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), pp.1798-1828.

[17] Hecht-Nielsen, R., 1992. Theory of the backpropagation neural network. In Neural networks for perception (pp. 65-93). Academic Press.

[18] Chai T, Draxler RR. Root mean square error (RMSE) or mean absolute error (MAE)?—arguments against avoiding RMSE in the literature. Geosci Model Dev. 2014;7(3):1247–50.

[19] A. Ng, Sparse autoencoder, 2010, [online] Available:https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf

[20] Keras Home Page, https://keras.io/, Last accessed on June 17, 2020

[21] A. Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In ICML, 2004.

[22] Lundberg, S.M. and Lee, S.I., 2017. A unified approach to interpreting model predictions. In Advances in neural information processing systems (pp. 4765-4774).

[23] Ribeiro, M.T., Singh, S. and Guestrin, C., 2016, August. " Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

[24] Shrikumar, A., Greenside, P. and Kundaje, A., 2017. Learning important features through propagating activation differences. arXiv preprint arXiv:1704.02685.

[25] Caruana R, Lawrence S, Giles L. Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. Adv Neural Inf Processing Syst. 2001:402-408.

[26] D. P. Kingma et al., "Adam: A Method for Stochastic Optimization," CoRR,abs/1412.6980 (2014).

[27] Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PloS one. 2015 Mar 4;10(3):e0118432.