# EXPLAINABLE AI FOR AUTONOMOUS NETWORK FUNCTIONS IN WIRELESS AND MOBILE NETWORKS

Premnath K Narayanan[1]and David K Harrison[2]

[1]LM Ericsson Ltd., SA OSS PDU OSS S&T Research & PCT, Athlone, Ireland
[2]School of Computing, Glasgow Caledonian University, Glasgow, United Kingdom

## *ABSTRACT*

*As the telecommunication network components and functions are getting commoditized, the complexity in configuration and optimization increases. Several automation techniques are evolving from traditional deterministic algorithms (pre-defined rulesets obtained from experience accumulated by humans) that were heuristic-based to more cognitive and stochastic-based algorithms. The aim of this paper is to introduce the seven layers in wireless telecommunication networks that uses stochastic or AI algorithms, explain the need for monitoring and possible potential biases in each layer of the stochastic algorithm stack and finally conclude with evaluation methods, techniques for detecting false positive and false negative proposals in autonomous network functions. The main subject of the paper is to provide a background on the need of explainable AI for autonomous network functions. The paper includes introduction of two models ANOBIA and INFEROBIA models that helps to achieve explainable AI for autonomous network functions in wireless and mobile networks.*

## *KEYWORDS*

*Explainable AI; Machine Learning; Artificial Intelligence; Precision; Recall; BIAS; Variance; Algorithm and Mitigation methods*

## 1. INTRODUCTION

Modern telecommunication networks are moving towards programmable network functions and adopting stochastic algorithms as part of the network autonomous functions [1]. Network operators could potentially buy several components of network equipment and assemble them as a full-stack network (e.g., baseband, radio unit, core network, transport network, and their sub-networks – including physical, virtual or containerized network functions). One of the key initiatives taken to fuel such programmable network functions is through the O-RAN alliance (Operator Defined Next Generation Radio Access Network Architecture and Interfaces) [2]. Near Realtime and Non-Realtime Radio Access Network (RAN) intelligent controllers are part of the O-RAN architecture.

Future network evolutions in 6G, are more focused on providing several deployment options in the core and radio by integrating virtual network functions as containerized network functions (CNFs). Such flexible deployment architecture helps network operators to provide services depending on the latency and throughput needs of the data or services.CNFs are more and more embracing stochastic algorithms as part of their autonomous functions. Until 4G and current 5G such autonomous functions use deterministic rules (policies) that are derived from simulations and early technology-specific network trials. As part of political, economic, social,

DOI:10.5121/ijwmn.2020.12303 

technological, legal and environmental (PESTLE) chart described in early 6G whitepapers [3] mentions that AI/ML stochastic algorithms is moving from medium to high impact in “To-be 6G” systems.

This paper describes possible biases in such AI/ML stochastic algorithms and proposes two models that can help in mitigating biases or malicious proposals made by autonomous network functions that use stochastic algorithms.

## 2. AI/ML/Deep Learning Software Stack and Trending Deployment Techniques

Several layers are involved in developing a cognitive use case for the autonomous network, as shown in Figure1. An autonomous network function is developed using frameworks, platforms, and software development kits (SDKs) specific to software and stochastic algorithms.

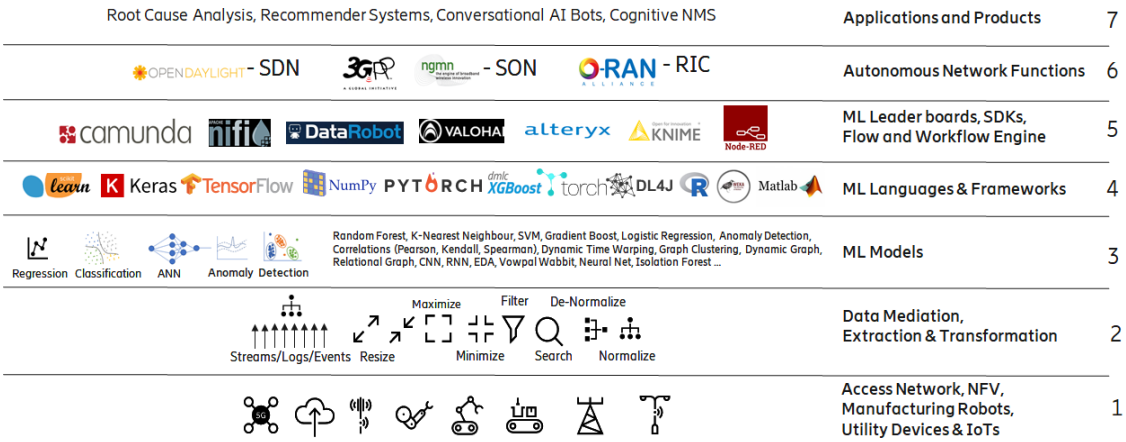


Figure 1. Stochastic algorithm software stack highlighting the layers

Layer 1) Network functions of core or radio network measures the performance of the network functions (e.g., downlink throughput, mobility).

Layer 2) Data mediation components mediate with respective network devices and receive the data as per the use case needed for Layer 6 and Layer 7. The data collected from the network are generally structured (respective schema defines the structure). With older network equipment's the data received from the network are unstructured and schema-less. (e.g., response to shell commands).

Layer 3) Based on the use case need, statistical techniques and machine learning algorithms are chosen. Generally, these algorithms require the right hyperparameters tuned, based on the data and the use case.

Layer 4) To implement layer 3 algorithms frameworks with right application programmable interface (API) are chosen from this layer for implementing the selected algorithm, e.g., "scikit learn” provides the needed APIs to perform machine learning algorithms.

Layer 5) Data engineering flows and machine learning platforms are from this layer. Workflow engines, machine learning SDKs, ML pipeline configuration management, and Auto-ML are some of the examples of such platforms.

Layer 6) Autonomous network functions as defined in 3GPPforum such as Self-Organizing Networks functions, Open daylight forum's Software defined networks, and O-RAN'sreal-time and non-real-time Radio Access Network (RAN) intelligent controller. Autonomous network functions have their intents, goal, and implicit rules or algorithms to achieve a specific network function objective. Generally, autonomous functions take care of avoiding conflicts with other network functions. Alternatively, there are supervisory control algorithms or coordinators or orchestrators that filter the proposals between autonomous network functions.

Layer 7) Applications and products developed to ease network operations are available as part of this layer. Example of such applications include Cognitive NMS (with reinforcement learning support), Autonomous alarm resolution and root cause analysis products.

Further, in the paper, these 7 layers are mentioned as "7 Layers of Deep learning stack," and respective layer numbers refer to the description of the layers.

Note: In every layer, the respective trademarks and icons belong to respective companies. The authors of this paper do not make any recommendations for their accuracy and applicability. They are used for illustration purposes only (based on the vendor website details and practical experiences of the authors in using them as part of their day to day work).

## 3. REQUIREMENT FOR MONITORING AND MITIGATING BIAS IN AUTONOMOUS NETWORK FUNCTIONS

Stochastic algorithms are essential for any machine learning, deep learning, and artificial intelligence systems. All stochastic algorithms have hyperparameters to be correctly tuned, availability of input network data, underlying network function instantiation, and initial configuration based on "golden" parameter standards.

Randomness is well embraced in ML, DL, and AI systems for the right bias and variance trade-off. Elegant measures are taken to fit the network data generally. Such generalization drives to algorithm applicability for several data sets. In the case of classification algorithms, "Precision and Recall" measure the success of prediction for imbalanced classes. High Area Under Curve (AUC) denotes high recall and high precision. High precision score indicates low false-positive rates, and high recall indicates low false-negative rates. The area above the AUC curve indicates the potential proposals, which are either false positive or false negative. Such proposals could potentially bring down the network or generally go against the norm or do not go towards an intent (e.g., make the cell more congested, reduce coverage and reduce throughput/capacity when it is needed).

Randomness in algorithms and the possibility of non-reliability of data (due to technical and other external factors) could potentially lead to a malicious proposal that acts against the intended goal of the network.

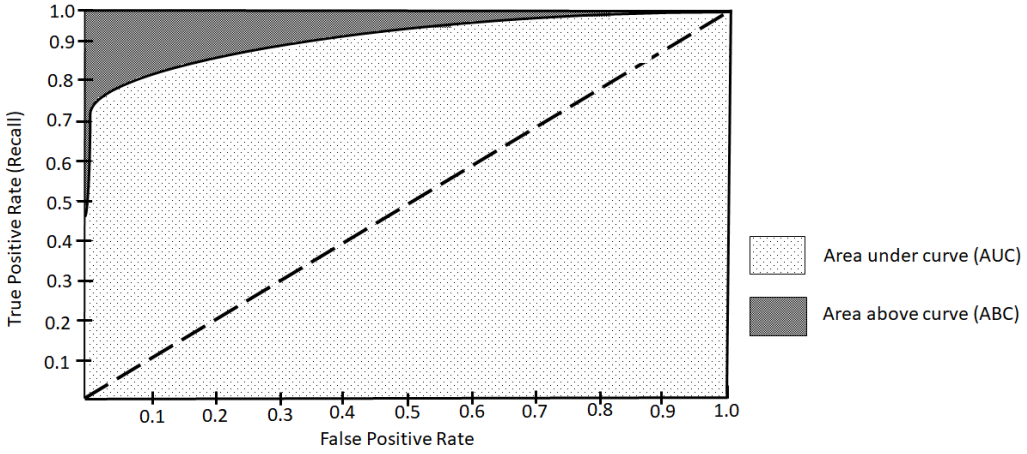


Figure 2. Indication of Area above curve

The proposal that falls in "ABC-Area," as shown in Figure 2. are generally malicious (“false positives or false negatives”). Such proposals degrade the performance of the network. Similarly, Mean Squared Error (MSE) / Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Adjusted R Squared, Mean Absolute Percent Error (MAPE)/Mean Squared Percentage Error (MSPE) are few examples in Regression algorithms. Accuracy and Log-loss is another example in case of “Classification” algorithms. Within cluster sum of squares (WCSS)/ Between Cluster sum of squares (BCSS), Mutual Information, Silhouette Co-efficient are examples of unsupervised learning. Bilingual Evaluation Understudy (BLEU) Score is an example of unsupervised learning. Generally Cross validation (CV) error is applicable almost for all the ML algorithms that uses CV as validation step.

Table 1. Possible biases to be mitigated in each layer

| | **Possible biases to be mitigated** | |
|---|---|---|
| **Layer ID** | **Description** | **Possible biases** |
| Layer 1 | The devices support different formats. Data is sent from the node as streams or can be retrieved as files. In any case, the node sends only the requested data. | i. Sample bias<br>ii. Latent or prejudice bias<br>iii. Survivorship bias<br>iv. Interaction bias<br>v. Cobra effect (unwanted data)<br>vi. Jevons paradox |
| Layer 2 | Operations such as extraction, transformation, and imputing of missing data are performed based on the use case and the algorithm need. | i. Sample bias<br>ii. Calculation accuracy<br>iii. Cognitive reflection for calculation in decision making.<br>iv. Statistical bias<br>v. Delayed latency due to IT infrastructure issues |
| Layer 3 | The critical activity done by the majority of the stochastic algorithm is with bias, variance trade-off. Extreme caution and attention are needed in tuning the hyperparameters. | i. Sample bias<br>ii. Cognitive bias or algorithm bias<br>iii. Concept drift<br>iv. Inductive bias or |

| | | |
|---|---|---|
| | | learning bias |
| Layer 4 | Specific machine learning languages, such as Python, are dynamically typed. When enough attention is not paid, the type (e.g., string, integer) could change during the program execution. | i. Calculation accuracy<br>ii. Statistical bias<br>iii. Bias/Variance trade-off |
| Layer 5 | Frameworks use several configurations to execute a flow. | i. Calculation accuracy<br>ii. Statistical bias |
| Layer 6 | Policies drive autonomous network functions (e.g., trade-off configurations). A wrong configuration of such parameters could lead to biases. | Cognitive bias or algorithm bias |
| Layer 7 | Application is use case driven, and different configurations are used for specific use cases. A wrong configuration of such parameters could lead to biases. | Cognitive bias or algorithm bias |

### 3.1. Sample Bias

Also referred to as selection bias. The problem with training data that does not accurately represent the environment is referred to as sample bias. An algorithm cannot be trained virtually on the entire universe of data it could interact with. The sampling techniques and methods involved in selecting the subset of that universe, both large enough and representative enough to mitigate sample bias is crucial and many times ignored for data feasibility reasons. Example: Data collected only during busy hours and performing optimization. [11]

### 3.2. Latent (or) prejudice Bias

Training the model based on cultural prejudice in previous data will lead to high false positives or false negatives. Example: KPI driven network optimization. Always believing in throughput and physical resource block utilization than focusing on customer experience and quality of service. [11]

### 3.3. Survivorship Bias

Survivorship bias is the logical error made during the data selection process where critical data is overlooked. Wald [12] during the second world war took survivorship bias into calculations when considering how to minimize bomber losses to enemy fire. Based on the returning aircraft from mission, research concluded that they add extra armor for the areas where that showed the most damage. Wald noted that the study only considered the aircraft that had survived their missions and the holes in returning aircraft, then represented areas where a bomber could take damage and still return home safely. Wald proposed that the Navy reinforce areas where the returning aircraft were undamaged. (e.g., Optimizing coverage areas based on signal measurements. Low-quality signals increase coverage, and interfering signals decrease coverage. Spots with coverage hole will not have any measurements. That is the critical area needed for focus to ensure 100% coverage and improve quality).[12]

### 3.4. Interaction Bias

When a model is generated based on a specific set of data and without a diverse set of human interaction could lead to wrong models in the algorithm. Example: Collecting network performance data only from macrocells (large coverage area cells) and applying the algorithm for all the cell types (e.g., small, micro, indoor, and macrocells).

### 3.5. Cobra effect

The "Cobra effect" occurs when an attempted solution to a problem makes the problem worse.Economist Siebert, Horst coined the term "Cobra effect" based on the following:

During the British rule in India, bureaucrats in Delhi grew concerned about the proliferation of cobras in the city. To get the problem under control, authorities offered a bounty on cobra skins. This economic incentive did not work well since some of the population in Delhi responded by farming cobras. By noticing this unethical practice, bureaucrats decided to stop the bounty program, and the framing entrepreneurs left the cobra in the fields. This led to an increase in the population of cobras. e.g., Offering incentives or bonuses for keeping specific KPIs in the network above certain levels. This leads to not exploring better optimization possibilities and increases unwanted network expansion or at time de-commissioning of networks.

### 3.6. Jevons Paradox

During the industrial revolution, there was a general belief that coal consumption can be reduced by improving technology. Jevon's study pointed out that this view as incorrect since an increase in efficiency leads to higher productivity and market reach, in turn, an increase in usage of coal. e.g., Energy-saving devices could reduce energy consumption for the telecom network. When customer consumption pattern increases due to cheap network running cost, more network could be rolled out.

## 4. EVALUATION METHODS AND TECHNIQUES TO MITIGATE BIAS IN AUTONOMOUS NETWORK FUNCTIONS

Two methods ANOBIA and INFEROBIA, are proposed as part of this paper to mitigate malicious proposals from autonomous network functions.

### 4.1. ANOBIA Model

The ANOBIA model is used for detecting anomalies or outliers that are close to false positive and false negative in an autonomous network function based on online (current network data) and historical network data.

The ANOBIA model uses several anomaly detection algorithms from the literature (e.g., statistical-based models – online learning, labelled data-based models – supervised learning). The model picks the right combination of algorithm (ensemble model) from the ANOBIA algorithm leader board for detecting outliers in an autonomous network function proposal that is closer to false positive and false negative proposals (or the proposals that are above the AUC curve). It also detects malicious proposals by identifying anomalies in the proposals. Generally, anomaly models can identify anomalies in the series of data. In this case, a series of data is the proposals made over time by the autonomous network functions.

The ANOBIA method learns, over proposals. This learning helps to mitigate malicious proposals over time.

Advantages:

- The model does not need any attributes of autonomous network functions. (such as input parameters, hyperparameter settings).

Disadvantages:

- Cannotdetermine the reasons why the autonomous network function behaves maliciously.

### 4.1.1.ANOBIA Model – Procedure 1

An autonomous function proposal (v) is marked as a malicious proposal when the value of v is outside the Min or Max range (anomaly).

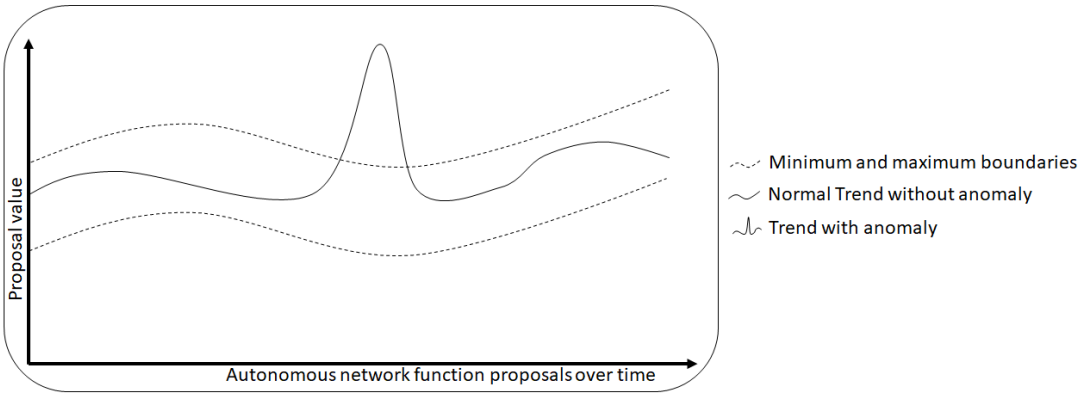


Figure 3.Anomaly detection based on boundaries (autonomous function proposal)

In Figure 3, minimum and maximum boundaries are detected using the following standard statistical measures (referred to as Min(v) and Max(v)):

1. Standard Deviation (SD)

$\sigma = \sqrt{\sum \frac{(x-\bar{x})^2}{N}}$ ; Where $x$ is the current proposal of autonomous function and $\bar{x}$is the mean of all the proposals.

2. Mean Absolute Deviations

Absolute value of $(x - y)^{\text{nth Percentile}}$; Where "x" is the current proposal of autonomous function, and y is the mean of all the proposals (v).

3. Average Absolute Deviation (AAD)

Average of proposals with mean proposal value.
For the actual proposal (v) made by the autonomous network function we derive Min(v) and Max(v) as follows:

Min(v) = -1 x (n x SD) (or) -1 x (n x MAD) (or) -1 x (n x AAD);
Max(v) = (n x SD) (or) (n x MAD) (or) (n x AAD);
where "n" is a natural number (non-zero positive whole number).

On the series of autonomous function proposals, the Min and Max calculations can be applied to the mean or median or percentile of the proposals. Choosing mean, median or percentile and applying SD or MAD or AAD to derive Min and Max purely depends on the type of data. For, e.g., Load balancing algorithms that proposes changes for cell offsets attributes are in dB's, remote electrical tilt parameters are in degrees.

### 4.1.2.ANOBIA Model – Procedure 2

Alternative procedure for detecting malicious proposal using Local Outlier Factor (LOF) algorithm LOF [13] contends that for many scenarios, assigning a degree of being an outlier is more meaningful.

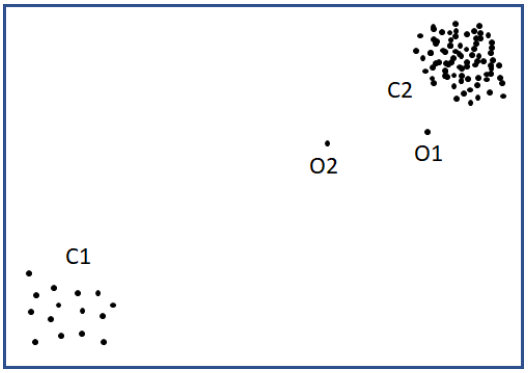


*Figure 4.* Illustration of clusters and outliers

For the objects deep inside a cluster, C1, or C2, as shown in Fig. 4., their LOF is close to 1 (LOF $\approx$ 1). When LOF is close to 1, the object is not labeled as an outlier. For the objects far from dense areas, these are outliers, and their LOF value is close to 0 (e.g., O1 and O2). In this case potentially they are malicious proposals.

The paper [13] introduces the following method to calculate LOF:

1 k-distance – The distance of an object "p" to its kth neighbour. If, for example, k was 5, then the k-distance would be the distance of an object "p" to the fifth closest point.

2 Reachability distance – is calculated based on k-distance.
reach-distk(p, o) = max { k-distance(o), d(p, o) }, if object p is away from o then actual distance between p & o becomes reachability distance or else replaced by k-distance.

3 Local reachability distance (lrd) –

$$lrd(p) = \frac{1}{(\Sigma\, reach-distk(p,o)/N)'}$$

Where "p" and "o" are two objects, and N is the total number of nearby objects, lrd indicates how far one must travel to reach the next point or cluster of points. Higher lrd indicates the cluster is denser and shorter to travel.

4 LOF is calculated based on the average ratio of lrds of the neighbours of object p to the lrd of p. LOF of the object indicates the density of the point compared to the density of its neighbours. When the density of an object is much smaller than the density of its neighbours

(LOF close to 0), the object is far from dense areas, and hence we can detect the outlier (malicious proposal from autonomous functions).

## 4.2. INFEROBIA Model

The model is used for detecting potential bias in an autonomous network function based on input parameters, hyperparameters, and historical results of network function proposals.

The INFEROBIA model adds labels to every proposal made by an autonomous network function as "good" or "bad" in the data set. Labelling is done based on the historical effects on the network (e.g., through KPI measurements). The effects on the network are evaluated based on the network KPI values. When the KPI values are moving towards the intended goal, the effect of the proposal is marked as "good."

The INFEROBIA model considers the following entities along with the good/bad label:

- Autonomous network function input parameters (e.g., Network KPIs, events).
- Algorithm hyperparameters (e.g., Algorithms specific thresholds, trade-off configurations).

Based on these labels and input parameters, decision trees are generated to indicate specific input hyperparameters and input network values that influence the decision to be "bad" or "good" for an autonomous network function. Based on these decision tree rules, the INFEROBIA model evaluates the input parameter values and predicts whether, or not, the autonomous network function proposal is "genuine" or "malicious."

Advantages:

Highly suitable for abstraction and reasoning of malicious proposals:

- It can determine the reason why the autonomous network function is behaving maliciously either due to hyperparameter (e.g., range) or input network data (e.g., missing data).

- The rules derived out of decision trees (e.g., the combination of hyperparameter and network data) helps to derive new meanings.

Disadvantages:

- It is mandatory to know the effect on the network for the action taken by an autonomous network function. Without the effect on the network, it is not possible to label "good" and "bad."

- Model fits only for autonomous network functions where input network data, hyperparameters of the algorithms, are known.

Table 2. Sample feature categories for the inference engine

| Classification feature category | Description of parameters |
|---|---|
| Algorithm Name | Name of the stochastic algorithm. |
| Algorithm Hyperparameters | Stochastic algorithms have specific hyperparameters that helps the algorithm to balance between bias and variance.<br>Example: "k" is a hyperparameter for k-nearest neighbor in clustering algorithms like K-Nearest Neighbours (KNN). |
| Policy configuration | Since coverage and capacity are mutually exclusive, autonomous network functions carry policy configurations such as mobility, coverage, and capacity. |
| Algorithm input parameters | General network configurations, current network performance metrics, alarms, events, and logs are some of the examples of autonomous network function algorithm inputs. |
| Algorithm runtime infrastructure configurations | Number of CPUs, memory, network speed, container, or other physical or virtual configurations to run the algorithms. |
| Algorithm Framework, Programming language-related configurations | Framework settings (e.g., port number, name), programming language runtime parameters (e.g., classpath), dependency jars used, are some of the examples of Layer 4 configurations. |
| Layer 1 data selection configurations in the network | Configurations specific to data collection decide what data to be collected and the duration of data. |
| Layer 2 data wrangling, ETL techniques used and their settings or configurations | As part of data wrangling, several techniques could be used to impute missing data (e.g., the average value of the feature in the place of missing data). |
| Layer 5 tools used and their settings or configurations | As part of the complete implementation of autonomous network functions, SDKs, workflow engines, and other data pipeline configuration management systems are used. All their names, configuration, or settings used can vary from execution to execution. |

### 4.2.1. INFEROBIA Model – Procedure 1

Classification is one of the existing techniques in data mining, and the tree classification algorithms such as C4.5 decision tree classification method [14] can provide the best insights into the algorithm.

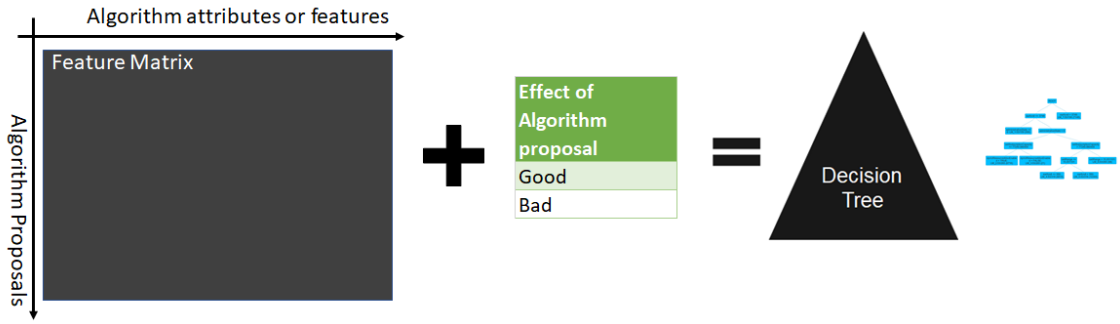


Figure 5.Decision tree to infer reasons for malicious algorithm proposals

A feature matrix is created as discussed in Table II. that includes the internal details of the algorithm, such as input network data, hyperparameters of the algorithms, Layer 1 to Layer 7 settings, and configurations of the stochastic algorithm software stack as indicated in Figure 1.

The algorithm proposals and their impact on the network can be monitored based on network assurance or analytical reports. Analytics reports indicate whether the proposal made by autonomous network function is good or bad. By adding respective impacts (good or bad) of the autonomous function to algorithm features as indicated in Fig. 5. (includes all 7 layers of the stochastic algorithm software stack), a detailed feature matrix is created for all the autonomous function proposals.

Generate classification tree based on "Effect of algorithm proposal" as target. The split conditions from the root of the tree until the leaf node indicates the apparent reason indicating what algorithm features made the autonomous network function to fire malicious proposals.

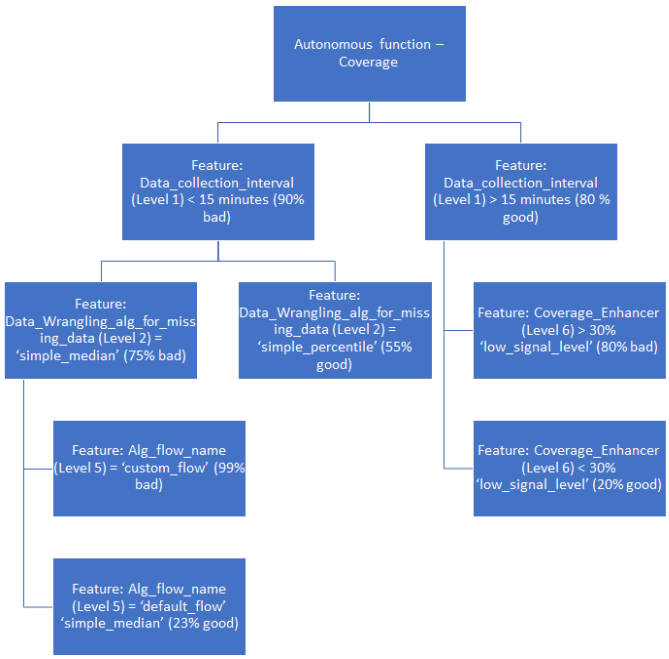


Figure 6.Decision tree to infer reasons for malicious algorithm proposal

### 4.2.2. INFEROBIA Model – Procedure 2

Distance-based metric to infer the significant difference between genuine and malicious proposals made by autonomous network functions.

Table 3.Data set with autonomous network function proposal and algorithm attributes (from every layer)

| **Autonomous network function name** | **Time of proposal** | **Effect of Algorithm Proposal** | **Layer1 Data Sampling Reliability** | **Layer3 Algorithm hyper parameter 1** | ··· |
|---|---|---|---|---|---|
| Coverage and Capacity Optimization (CCO) | 13:00 | good | Above Average (AA) | k=7 | |
| CCO | 13:20 | good | AA | k=12 | |
| CCO | 13:40 | bad | Below Average (BA) | k=7 | |
| CCO | 14:00 | good | AA | k=12 | |
| CCO | 14:20 | good | AA | k=7 | |
| CCO | 14:40 | bad | BA | k=12 | |
| CCO | 15:00 | bad | BA | k=7 | |
| CCO | 15:20 | good | AA | k=12 | |
| CCO | 15:40 | good | Average | k=6 | |
| CCO | 16:00 | bad | Average | k=3 | |

Distance between the elements, as shown in TABLE III in the set, can be achieved with the help of a distance function.

Euclidean Distance: One of the commonly used distance metrics to find distance between two data points in a plane.

$$Distance(a, b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

Where “a” and “b” are features, “i” is the index of features from “1” to “n."

Using this distance formula, the feature difference between a bad proposal and its nearest good proposal can be calculated. Nearest good proposals are the ones where most of the features match with the bad proposal’s feature.

Based on the feature difference following attributes need attention:

- “Layer1 Data Sampling Reliability” attribute with value ‘Below Average’ is responsible for bad proposals. On the contrary, the value ‘Above Average’ is responsible for good proposals. Hence it is essential to ensure data sampling is above average for the model to work effectively.
- “Layer3 Algorithm hyper parameter 1” attribute with value ‘k=3’ is responsible for bad proposals. On the contrary, the value ‘k=6’ is responsible for good proposals. Hence it is important to ensure hyperparameter value should be 6 for the model to work effectively.

Similarly, KNN and K-means clustering techniques can be adopted to find whether the proposal is close to AUC centroids.

### 4.3. Further Research

ANOBIA and INFEROBIA can be extended with feature analysis using SHapley Additive explanation (SHAP) models [15]. Unified framework for interpreting predictions [16] can be extended with ANOBIA and INFEROBIA model for detailed insights not only restricted to feature analysis, but also to all the attributes and parameters of seven layers as described and introduced in this paper.

## 5. CONCLUSIONS

ANOBIA and INFEROBIA are machine learning models that can protect a telecommunication network from wrong decisions made by autonomous network functions. Such models will help to validate the effectiveness of increasing autonomous network functions in the telecommunication network. ANOBIA and INFEROBIA can potentially act as a red button before unintended effects are created in the network. The phobia in adopting autonomous network functions can be minimized by adopting such models that perceive, learn, abstract, and reason the malicious behaviour of autonomous network functions. Reasoning will help the area of "explainable AI" in the area of telecommunication.

## ACKNOWLEDGEMENTS

The authors thank the University of Bolton and Amity [IN] London university for providing an opportunity to work on the thesis that measures the effectiveness of autonomous network functions. The authors additionally thank their respective university, organization (LM Ericsson Ltd, and Glasgow Caledonian University) for supporting the research that would benefit the telecommunication industry. Further, the methods proposed in this thesis will act as a catalyst for introducing more autonomous network functions that could potentially reduce the increasing operational expenditure of a telecommunication network.

## REFERENCES

[1] Paulo Valente Klaine, Muhammad Ali Imran, OluwakayodeOnireti, Richard Demo Souza, "A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks," IEEE Communications Surveys and Tutorials, IEEE, Volume 19, issue 4, pp. 2392-2431, DOI: 10.1109/COMST.2017.2727878, 2017

[2] O-RAN alliance, "Operator Defined Next Generation RAN Architecture and Interfaces," https://www.o-ran.org/, 2019 (last accessed August 2019)

[3] Matti Latva, Kari Leppanen, "Key drivers and research challenges for 6G ubiquitous wireless intelligence," http://jultika.oulu.fi/files/isbn9789526223544.pdf, 2019 (last accessed January 2020)

[4] Willmott Cort J, Matsuura Kenji, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," Climate Research, pp. 79-82, DOI:10.3354/cr030079, 2005

[5] Magee L, "R2 measures based on Wald and likelihood ratio joint significance tests". The American Statistician. 44. pp. 250–3. DOI:10.1080/00031305.1990.10475731, 1990

[6] de Myttenaere, B Golden, B Le Grand, F Rossi, "Mean absolute percentage error for regression models," Neurocomputing 2016 archived preprint, 2016

[7] Powers, David M W, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness& Correlation," Journal of Machine Learning Technologies, pp. 37–63, 2011

[8] Shen Yi, "Loss Functions For Binary Classification and Class Probability Estimation," University of Pennsylvania, 2005

[9] Kriegel Hans Peter, Schubert Erich, Zimek Arthur, "The (black) art of runtime evaluation: Are we comparing algorithms or implementations?", Knowledge and Information Systems, pp. 341–378, DOI:10.1007/s10115-016-1004-2, 2016
[10] Peter J Rousseeuw, "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis." Computational and Applied Mathematics, pp. 53–65, DOI:10.1016/0377-0427(87)90125-7, 1987
[11] Glen Ford, "4 human-caused biases we need to fix for machine learning", https://thenextweb.com/contributors/2018/10/27/4-human-caused-biases-machine-learning/, 2017 (last accessed October 2018)
[12] Wald Abraham, "A Method of Estimating Plane Vulnerability Based on Damage of Survivors," Statistical Research Group, Columbia University, CRC 432 — reprint from July 1980, 1943
[13] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander, "LOF: identifying density-based local outliers," Vol. 29, No 2, pp. 93–104, ACM, May 2000.
[14] Quinlan, J. R, "Programs for Machine Learning," Morgan Kaufmann Publishers, 1993.
[15] Lundberg, S., Erion, G. and Lee, S.,"Consistent Individualized Feature Attribution For Tree Ensembles," https://arxiv.org/pdf/1802.03888.pdf, 2019(last accessed June 2020)
[16] Lundberg, S. and Lee, S., "A Unified Approach to Interpreting Model Predictions," https://arxiv.org/pdf/1705.07874v2.pdf, 2017 (last accessed June 2020)

## AUTHORS

**Eur Ing Professor David K Harrison BSc (Hons) MSc PhD CEng FIET FIMechE FIES CITP MBCS.**

David is currently Professor of Design & Manufacturing at Glasgow CaledonianUniversity where he has held a range of managerial roles. He has spent his working career in manufacturing industry or industry facing academia. A graduate of UMIST, he has edited several books and conference proceedings and has published his work widely. He has supervised 81 PhD students through to graduation. Around half of these students have been based outside the United Kingdom.

**Mr. Premnath K Narayan BSc., (CS), MCA., MBA.**

Premnath is a seasoned Software Engineer (System Engineering, software architecture and development) with 22 years of practical experience in realizing Commercially of the shelf (COTS)/cloud products for ICT (Information and Communications Technology) industry. He has designed & developed products, trained users and mentored employees. Working as a master engineer at Ericsson primarily focused on researching and developing autonomous network functions for telecommunication network products.