UNITED STATES PATENT APPLICATION

For

EXPLAINABLE AI FOR PNM

INVENTORS:

JASON W. RUPE

JINGJIE ZHU

SAYANDEV MUKHERJEE

Description

For AI and ML implementations for anomaly detection in PNM data, such as in the CableLabs Anomaly Detector, the nature of the training data can be used to quantify in various ways the impairment type and severity, describing the coverage of the input space for AD, and serving as a baseline for explainability. By quantifying an input to the AD in terms of the input space for the training data, we can determine whether the input case is within the space covered by the training data, and how much training was done around this space. As a result, one can create a quantification of the support for a given AD decision in terms of explainability that is not just a rationalization, but is based on the reasoning given to the AD during training.

For RxMER per subcarrier and spectrum data (or any frequency bin or time series data relating to maintenance which can reveal anomalies), an anomaly detection algorithm (which may or may not be AI or ML based) can be used to identify patterns in the data that indicate network faults or impairments. These impairments are classified by type as indicated by their pattern, and their severity can be quantified by a number of methods including power level, deviation from intended power level at each frequency, profile that must be assigned, loss of capacity in the RF signal, etc.

When training a solution for anomaly detection in these types of data, each training input can be quantified as described above. One can keep track of the quantification and additional statistics such as the anomaly type and where in the training data it exists, and more. Each training data input (set of data points in frequency or time bins for example) can then be accompanied by a set of statistics (meta-data) that describe where in the possibility space that data input resides. After a large number of training input sets, these could be plotted to envision the coverage and density of coverage in the training. For each statistics that makes up the full meta-data that makes up the possibility space, we can find a density function. We may even assign the data to bins for a histogram, or take a like approach to translate data points into a density function. Once the solution is trained, each input to the trained solution could then be assigned its meta-data in terms of the statistics provided by the training data, to provide information about how close to the training possibility space the input data may be, or how dense the training data were in areas near the input in terms of the meta-data. This can serve as a measure of confidence for the decision being made, and additional statistics in terms of the training data labels can accompany to provide explanatory evidence-information. A weighting model can be added to fairly assess inputs that are outside the training possibility space for comparison with those that are inside, per dimension if needed. This can be done simply by fitting a distribution model on each statistic within the meta-data for the training input, so that tail probabilities can provide reasonable approximate estimates of how much of an outlet an input may be, and thus the low confidence we may have in the model's ability to identify the anomaly properly.

Example for spectrum capture data

Spectrum capture data from a cable modem is magnitude within a frequency band only, not complex I, Q values. A CM captures data in frequency bands mostly in the downstream, though some CMs can capture upstream frequencies as well, which are lower frequencies. For this example, consider only the downstream frequency bands, and identify exclusion bands (frequencies not in use) as well as upstream bands. What frequencies remain are the frequencies over which downstream signals are carried to the CM. Consider the downstream frequencies as well as the exclusion bands between and around them, but only for the downstream frequencies, according to the frequency plan set by DOCSIS and the operator.

Using spectrum data to train AD for impairment detection involves adding meta data to the spectrum plot which includes the type of impairment and the frequencies over which the impairment is indicated. Each impairment has a unique signature that is a combination of the frequencies it is within, the shape of the spectrum magnitude as compared to the design target of a level signal, the magnitude deviation from the expected, and whether the level change is positive or negative compared to the neighbor frequencies that appear to not be impaired. Because the impairment is identified by a shape that is a deviation from an signal target that can be estimated, and the deviation in magnitude can be translated into an impact on each frequencies ability to carry a signal well, we can quantify the severity of each identified impairment.

Therefore the training input space can be characterized by the impairment types, and the signed values of deviation at each frequency. Because some impairment types don't occur in just any frequency, each impairment type has its own frequency space.

Let a given spectrum capture i with and impairment of type j (with J being the type and j being the sample's type) have a frequency deviation vector across frequencies of $df_{i,j} = (d1, d2, ,,, dn)$ where dk is the deviation from normal for frequency bin k due to the impairment. Now generalize this to allow for multiple impairments in a single spectrum capture so that we have $df_{i,j1,j2,\ldots,jn}$. Simplify the notation so that $df_{i,j}$ allows for multiple j with j being a vector of some small finite number.

(Note one novel idea is to take combinations of RF impairments to define a new type of impairment.)

In a simple approach (with more complex methods to be explored later), each j then defines an impairment type or combination of impairments, which we will refer to as a meta-impairment.

For each meta-impairment type J, we can define for a set of spectrum captures the input space covered by the set of training data Id. In a simple approach, this can just be the $df_{i,j}$ values for all i with j in J. The density function or histogram of df values in each frequency then characterizes the range and density of the samples covered by the training

data set. In a more complex approach, each impairment type may be characterized by a model. For example, a standing wave can be characterized by its periodicity as it is a periodic wave, and its magnitude, making for a way to simplify df.,j for convenience and potentially better modeling of the behavior. Refer to the resulting densify function set for each J meta-impairment type as DFJ.

Now an input i' to a trained AD can be classified as an impairment type j and then characterized by its dfi',j. Compare this to DFJ for j=J to determine how much overlap the input i' has with the training set. This result provides a measure of trust in the model because it provides meta-information about the training of the AD model in terms relevant to the specific input.

Now extend this idea into an approach where we do the same for input i' but with each J not equal to j, and let the dfi',j' values for each j' possible in J represent a factor for any competing answers, say if an AD were to provide a most likely result and potentially a top few most likely results. Or this can be used to examine each meta-impairment type containing a specific impairment type j of interest that may be the result of the AD for i'.

Note that this approach can be extended to other measures of validity or trustworthiness or confidence, say.

If the training data were to be held in a referenceable database attached to the AD, then training data that most resemble the input i' could be attached to the result as support for the decision, thus adding validation and reason for trust in the result. A human can then easily validate visually or statistically or both, and a meta-process encoded to collect statistics on the training to input comparison to determine if additional training is needed to cover the applied use case population, or to watch for changes in the input process that suggests the state of the network or system is deviating from the believed use case conditions.

In addition, those AD decisions acted upon can provide feedback to better understand if low validity in the AD result aligns with incorrect decisions, or not, to further determine if specific types of impairments, and their specific qualities in terms of df, are needed. Combining this automated approach with "human in the loop", a positive feedback loop that is manual can be created in the pipeline to continuously improve the explainability and trustworthiness and other measures of performance of the AD models in iterations.

[0001] In another embodiment of this general idea, the meta-category types J are the "classes" of the AD (assumed to be implemented via deep learning), while the types j in each J are "labels" and the AD is a multi-class classifier that can assign more than one label to an input i'. The outputs of the softmax layer of the AD corresponding to the different classes correspond to the confidence of the AD's classification. If the confidence assigned by the AD to its highest-confidence class is not high enough then a human expert may be employed to manually assign the class and label corresponding to

this input and thereby add to the curated training data for the AD. Moreover, even the existing inputs in the training data should be slightly dithered by adding noise in an attempt to create adversarial examples and thereby strengthen the resilience of the AD to such inputs when it is deployed.

In general terms, the idea is to characterize the input training space for a given AD or other AI or ML solution in specific terms for each input to the trained model as a way to explain the model as well as provide information that helps to validate the result.

Another idea for improving the trustworthiness of the AD model is to develop multiple machine learning-based models or statistical algorithms-based models and let them cross-validate. This could introduce additional development work, however, when deploying ML models in the network, an essential goal is to make the model trustworthy and explainable, which can be achieved using this approach.

Consider 3 different ML models are trained based on different feature sets or are implemented with different AD methods. For example, the first model is a deep 1-D CNN that's trained based using the raw one-dimensional spectrum data; the second model is a Fully-Connected Network (FCN) based regression model trained using statistical features of the impairments; and the third AD is purely a statistical algorithms based model. The labeling workload will not increase since the labeling is done on the ground truth of the raw spectrum data, where the summarized features can seamlessly map to the labels of each impairment. Using these 3 models, if there are significant prediction result deviations, the input sample should be given attention by human in the loop. If all of the AD models generate consistent predictions on a same input data, the trustworthiness of the prediction result could be considered high.

Similarly, in addition to employing different feature engineering approaches to create different models that consume different input data to improve the trustworthiness of the prediction results, different data augmentation techniques can be applied to multiple AD models to create "model memory" deviations.

Background

AI and ML solutions experience friction with adoption in part due to their lack of explainability. While there is much research and development to add explainability to these solutions, they mostly rationalize a black box algorithm, or suggest monitoring of the inner model components, neglecting the value of the input space as a form of explainability. Further, for AI and ML solutions applied to PNM, including our CableLabs anomaly detector (AD), a black box explainer would be redundant and with questionable value, while monitoring the inner model components gets complicated and does not explain well to the user, which does not provide confidence and reduce friction in acceptance of the decision provided by the AI or ML.

XAI for Communication Networks

***Abstract*—Explainable AI (XAI) is a topic of intense activity in the research community today. However, for AI models deployed in the critical infrastructure of communications networks, explainability alone is not enough to earn the trust of network operations teams comprising human experts with many decades of collective experience. In the present work we discuss some use cases in communications networks and state some of the additional properties, including accountability, that XAI models would have to satisfy before they can be widely deployed. In particular, we advocate for a human-in-the-loop approach to train and validate XAI models. Additionally, we discuss the use cases of XAI models around improving data preprocessing and data augmentation techniques, and refining data labeling rules for producing consistently labeled network datasets.**

Keywords*—XAI, trust, accountability, communications networks

INTRODUCTION

As machine learning models in general and deep learning models in particular are applied to more use cases across different industries, the need for Explainable AI (XAI) models has increased. Nowhere is this more true than in critical infrastructure deployments, where there is scope for great harm from poor decisions, whether made by humans or AI models. In the present work, we focus on communication networks, which are an important component of the critical infrastructure of every country.

We begin with the definition of XAI and the distinction between explainability and similar but distinct terms like interpretability and comprehensibility. Next, we provide an overview of the general literature on XAI before focusing on the literature addressing the needs of critical infrastructure. We illustrate the problem with a discussion of some use cases from a hybrid fiber coax (HFC) network. Our discussion highlights the importance of vetting and validating the explanations of the trained models by introducing humans in the loop at all stages, even starting from the initial data labeling stage.

REVIEW OF LITERATURE

Definition of XAI

We adopt the DARPA definition of XAI as "AI systems that can explain their rationale to a human user, characterize their strengths and weakness, and convey an understanding of how they will behave in the future" [1]. As explained in [2], this definition is distinct from related terms like "interpretability", "comprehensibility" and "transparency". Moreover, in a critical infrastructure deployment, the XAI system should also be *accountable*, meaning that there should be sanctions or redress to those harmed by its wrong or poor decisions. We will discuss how accountability can be designed into an XAI system later, but first let us review the literature on XAI systems in general.

Surveys of the XAI literature

In [3], the authors provide a survey of explainable AI techniques. Pointing out challenges, they list several of what they call "ethical principles" of XAI: accountability, responsibility, transparency, fidelity, bias, causality, fairness, safety. They differentiate responsible AI as having accountability, responsibility, and trust. They further explain that for XAI there is a distinction between explanation and interpretation, and define important features such as trustworthiness, interactivity, stability, robustness, reproducibility, and confidence. Then they classify methods into intrinsically interpretable methods, model agnostic visualization approaches, model agnostic feature interaction, model agnostic global surrogate, model agnostic local surrogate, propagation-based methods, instance-based explanations, and knowledge based techniques.

In [4], the authors provide a review of expert and recommender systems, plus explainable artificial intelligence. The architectural differences may assist in defining methods for verification, validation, and instilling trust. For explainable AI, the authors point out that XAI was built to overcome issues with interpretability, trust, and transparency (which could be interpreted as verification and validation). They further breakdown the issue into definitions including understandability, comprehensibility, explainability, transparency, interpretability, and data security.

In [5], the authors present an approach for explaining image classifiers using a counterfactual approach, finding a minimum set of features that if removed would change the classification. They also explain an important distinction in explanations: local to explain decisions, and global to explain models.

For critical infrastructure deployments, we emphasize here that in addition to global and local explanations, we also need validation and verification, not to mention additional meta information to provide a foundation for trust.

Literature on XAI and trust

In [6], the authors conduct an evaluation of XAI tools. They use Local Interpretable Model-agnostic Explanations (LIME)[7] and Shapley Additive Explanations (SHAP) to evaluate Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) using some empirical data sets to evaluate the explaining methods. As the data sets were limited to the medical realm, we cannot apply their findings with confidence to other domains.

In **Error! Reference source not found.**, the authors report on a project to improve trust in AI decision support tools in air traffic management, in light of regulations, using an AI trust assurance survey. While focused on the flight industries in the UK, some of the findings may apply to other industries such as communications, or at least a similar approach may reveal like findings in other industries. The framework they created as a result of the work is a good starting point for other critical infrastructure deployments of XAI models.

Discussion

The above works treat AI as a black box and attempt to explain the decisions it provides using an external validation method. Because the AI is treated as a black box, there is no way to truly explain the AI in any provable way; at best, you can create a rationalization for the decisions provided. But

rationalizing these decisions supports artificial trust in the artificial intelligence. Therefore, provable explainability, and true explainability, requires at least a grey box approach, and further will be very specific to the AI method chosen.

COMMUNICATION NETWORK USE CASES

Communications network use cases for AI, ML, and related techniques for decision-making are vast and can apply to many areas of the network service provider business:

- Technology Strategy - we envision AI could assist in assessing requirements needs, and suggesting strategy involving technology choices.
- Technology Evaluation - we envision AI could assist in assessing technology choices against requirements, and potentially develop options that result.
- Architecture Assessment – AI and ML could potentially assist in assessing architectures for various measures of performance and against requirements.
- Planning and Engineering – these network functions rely on mathematical models to assist in assessments that guide decisions, optimize for resources and requirements, and more. Some of these models may be augmented by AI and ML based solutions. But even if not, traditional models benefit from explainability equally.
- Installation – AI and ML could be used to suggest the best course of installation action, be that by technician (to support training and decisions) or customer executing a self-installation.
- Fault Management – this work is about identification, localization, and mitigation of faults (be that through repair, isolation, or other means). As we will show below in an example use case, ML solutions have already been applied here, and explainability and more bring benefit to this use case.
- Configuration Management – we envision that AI and-or ML solutions can help automate the configuration of network equipment when needed, and there are examples where these functions make adjustments to the network in real time, some including a human in the loop and others fully automating.
- Accounting Management – AI and ML could be used to find problems with accounts, and help sales representatives to manage their accounts as well.
- Intelligent Resource Allocation – ML solutions could be used to intelligently and efficiently allocate network bandwidth resources while maintaining the network latency at the target levels, such as the work described in [9].
- Performance Management – AI and ML are applied to network performance decisions today.
- Security Management – AI and ML solutions today assist in managing the security and privacy of network data and services today.
- Reliability Management – We envision that AI and ML can augment decisions for network configuration, repair prioritization, and more in support of reliable services and networks.
- Network Maintenance – To assist with network maintenance decisions, AI and ML solutions can aid the human in the process. Our use case example below serves as an example.
- Customer Support - AI and ML and other complex automation methods are used to aid in customer support today.
- Process Improvement - We conceive that AI and ML solutions can aid in process improvement too.

AI and ML solutions may contribute to each of these areas of the communication network service provider function. An overview of how AI and ML may be introduced into the operations of an HFC network is given in the White Paper written by a team of experts from the cable industry and published by CableLabs in **Error! Reference source not found.**.

PROACTIVE NETWORK MAINTENANCE USE CASE

Next, we describe a machine learning solution applied to anomaly detection in support of fault management. We refer to our solution as the anomaly detector (AD).
In the cable industry, which supports the access network as part of the service portfolio, network maintenance is a significant cost of operations, and yet a critical one in that it has significant impact on the customer: high availability and rapid repair time are important to all customers, and critical to some. Due to the resiliency mechanisms designed into DOCSIS® technology, operators can run tests and query telemetry to determine the health of the network. By looking at the performance of the network at various frequencies that carry the RF signal that carries the data, operators can perform proactive network maintenance (PNM), repairing impairments in the network before they impact service or are visible to the customer.

CableLabs developed an anomaly detector (AD) for the industry, one of the first applications of ML to PNM, and demonstrated its use for finding impairments in Receive Modulation Error Ratio (RxMER) per subcarrier data, and helped others test this concept using spectrum data. This work is reported in **Error! Reference source not found.**. An example output from AD of RxMER data with impairments is shown in Figure 1. Since reporting initial success in that paper, we have learned a lot more about this particular use case.

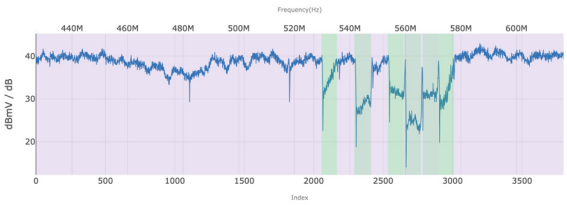

Fig. 1. Example output from AD of RxMER data with identified anomalies; the standing wave is highlighted across almost the full range of data, and multiple wireless ingresses are highlighted in green color on the right side of the figure.

Next, we explain some of what we learned in the application of our AD solution, and how we have augmented that with additional tools for operators, all of which impacts the explainability, verification and validation, trust, and other aspects of effective ML.

The AD we created for network operations takes PNM test telemetry and locates possible impairments, then classifies them according to type which aligns to failure modes and components in the network that are most likely. Further localization is done through correlating test points with the network layout, assessing additional test and query telemetry, and eventually through technicians manually localizing the faults in the network further, as needed, when sending a repair technician to the field is warranted.

Implementations for AD

In a fully automated implementation, the AD would generate a repair ticket if supported to do so, utilizing a framework such as the Proactive Operations Platform (ProOps) **Error! Reference source not found.**, Depicted in Figure 2. In this approach, the telemetry would be captured and assessed, further information collected, automated localization conducted, and criticality assessed. If the repair criteria are met, then the technician is sent to fix the problem. In this scenario, the technician may trust the result and go right to where indicated. They may then conduct the repair or find that the problem is elsewhere and then re-localize the fault. They may then find and fix, or not find the problem, or find it is inaccessible such as in a customer's house. In this approach, the cost of an incorrect assessment is at worst some amount of wasted technician time, at best a very efficient proactive fix, or in between it could be a fix that was not urgent or took longer than it should.

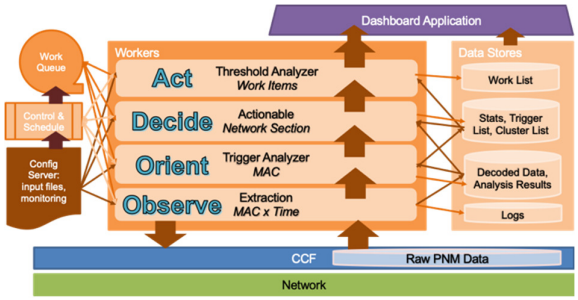

Fig. 2. A depiction of ProOps. In the case of our example, using AD, the AD module is inserted just after the raw PNM data so that the first layer receives the PNM data after AD labeling of anomalies.

Note that in an implementation that involves a human in the process earlier, there is a chance to further correct incorrect decisions in this use case. The result is a potential for better overall decisions in the process, though not guaranteed. The tradeoff is where to optimize cost in this case, as long as the repair is truly proactive.

In cases where customers are impacted, and may have called in for repair, there may be more urgency and cost of an incorrect identification and localization. While customers may provide additional useful information for repair, the impact on service favors proactive repair when possible.

To further train the AD, and help other operators to do the same, CableLabs constructed an *impairment classification tool*. This tool is not a machine learning model; rather, it is a software tool for humans, which allows users to take RxMER per subcarrier or spectrum capture data and label various types of anomalies and the spectrum they occupy. See Figure 3 for a screen shot of the data labeling tool using spectrum data.

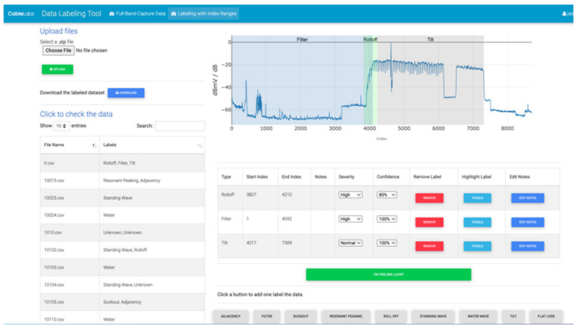

Fig. 3. A screen shot of the impairment classification tool after the user labeled impairments.

In sharing this tool, we have had several interesting observations:

- Experts do not all agree on how to classify and label an impairment, even if some of the impairments' signatures are obvious.
- The experts' feedback drives positive adjustments in the chosen data preprocessing and data augmentation techniques. (This is an example of the positive effect of the "human in the loop" building trust in the AI model trained with the labeled data, in this case the AD.)
- As the industry experts practice data labeling more, their labeling consistency improves, which positively affects the AD's accuracy.
- Having labeled impairment data is useful for training technicians and other operations personnel as well as the AD. As such, the process overall has a chance of improve performance by sharing the training data with the humans in the loop.

We note that the description of the impairment classification tool above implicitly puts the human expert "in the loop" of training in such a way that as the ML model is trained, it also gains the trust of the human expert. A similar human-in-the-loop process was proposed in **Error! Reference source not found.** as a means of ensuring accountability of the XAI system, accountability being essential for the legal and societal frameworks underlying critical infrastructure.

An important consequence for XAI is that the nature of the training data can be used to quantify in various ways the impairment type and severity, describe the coverage of the input space for AD, and serve as a baseline for explainability. By quantifying an input to the AD in terms of the input space for the training data, we can determine whether the input case is within the space covered by the training data, and how much training was done around this space. As a result, one can create a quantification of the support for a given AD decision in terms of explainability that is not just a rationalization but is based on the reasoning given to the AD during training. We provide more details of this approach below.

A HUMAN-IN-THE-LOOP APPROACH TO TRAIN AN EXPLAINABLE AD MODEL WITH SPECTRUM DATA

Creating a training dataset from spectrum data

Spectrum capture data from a cable modem (CM) is magnitude within a frequency band only, and not the full information in a pair of complex in-phase (I) and quadrature (Q) values. See the plot in Figure 3. A CM captures data in frequency bands mostly in the downstream (higher frequencies), though some CMs can capture upstream (lower) frequencies as well. For simplicity, let us consider only the downstream frequency bands, and identify exclusion bands (frequencies not in use) as well as upstream bands. What frequencies remain are the frequencies over which downstream signals are carried to the CM. Consider the downstream frequencies as well as the exclusion bands between and around them (but only for the downstream frequencies) according to the frequency plan set by DOCSIS® and the operator.

Using spectrum data as a labeled training set to train an AD for impairment detection requires adding meta data to the spectrum plot which includes the *type* of impairment and the *frequencies* over which the impairment is indicated. The following shows how to add this meta data.

Each impairment has a unique signature that is a combination of the frequencies it is within, the shape and deviation of the spectrum magnitude as compared to the design target of a level signal, and whether the level change is positive or negative compared to the neighbor frequencies that appear not to be impaired. Because the impairment is identified by a shape that is a deviation from a signal target that can be estimated, and the deviation in magnitude can be translated into an impact on each frequency's ability to carry a signal well, we can quantify the severity of each identified impairment.

Therefore, the training input space can be characterized by the impairment types, and the signed values of deviation at each frequency. Not all impairment types occur in all frequencies, so each impairment type has its own frequency space.

Characterization of training data for impairments

Let a given spectrum capture measurement i with an impairment of type j have a deviation vector across n frequency bins of measurement given by $\boldsymbol{d}_{i,j} = \left(d_{j,1}^{(i)}, d_{j,2}^{(i)}, \dots, d_{j,n}^{(i)}\right)$, where $d_{j,k}^{(i)}$ is the deviation from the expected target for frequency bin k due to the impairment j. In general, we need to allow for multiple (say m, with m small) impairments $\boldsymbol{j} = (j_1, \dots, j_m)$ in a single spectrum capture so that we can write the vector $\boldsymbol{d}_{i,\boldsymbol{j}}$ to define the concatenation of the vectors $\boldsymbol{d}_{i,j_l}$, $l = 1, \dots, m$.

In a simple approach (with more complex methods to be touched upon below), each $\boldsymbol{j}$ then defines an impairment type or combination of impairments, which we will refer to as a *meta-impairment* type. Multiple meta-impairment types may belong to a single meta-impairment *category* $\mathcal{J}$.

For each meta-impairment category $\mathcal{J}$, we can define for a set of spectrum captures $\boldsymbol{d}_{i,\boldsymbol{j}}, i \in I$, the (fraction of the) input space for this meta-impairment category $\mathcal{J}$ covered by the training data. In a simple approach, this can just be the measurements themselves: $\{\boldsymbol{d}_{i,\boldsymbol{j}} \,|i \in I, \boldsymbol{j} \in \mathcal{J}\}$. The density function or histogram of $d_{j,\cdot}^{(i)}$ values in each frequency then characterizes the range and density of the samples covered by the training data set.

In a more complex approach, feature engineering can be employed in the data preprocessing steps. For instance, each impairment type may be identified by a set of features that summarize its characteristics. For example, a standing wave can be characterized by its period and its magnitude, making for a way to simplify $\boldsymbol{d}_{i,\boldsymbol{j}}$ for convenience and potentially better modeling of the behavior.

For either a simple or a complex approach, let the set of density functions or histograms for meta-impairment category $\mathcal{J}$ be denoted $\mathcal{D}_{\mathcal{J}}$.

Introducing the human in the loop

An input i' to a trained AD can be classified as a meta-impairment type $\boldsymbol{j}$ and then characterized by its $\boldsymbol{d}_{i',\boldsymbol{j}}$. Compare this to $\mathcal{D}_{\mathcal{J}}$ for $\boldsymbol{j} \in \mathcal{J}$ to determine how much overlap the input i' has with the training set. This result provides a measure of trust in the model because it provides meta-information about the training of the AD model in terms relevant to the specific input.

Now extend this approach into a new approach where we do the same for input i' but for each $\mathcal{J} \not\ni \boldsymbol{j}$, and let the $\boldsymbol{d}_{i',\boldsymbol{j}'}$ values for each $\boldsymbol{j}' \in \mathcal{J}$ represent a factor for any competing answers, say if an AD were to provide a most likely result and potentially a top few most likely results. Alternatively, this can be used to examine each meta-impairment type containing a specific impairment type of interest that may be the result of the AD for i'.

Note also that this approach can be extended to other measures of validity or trustworthiness or confidence.

If the training data were to be held in a referenceable database attached to the AD, then training data that most resemble the input i' could be attached to the result as support for the decision, thus adding validation and reason for trust in the result. Confidence in each training data input can also be indicated (such as what level of experts and how many agree on the anomaly as indicated in the input). A human can then easily validate visually or statistically or both, and a meta-process encoded to collect statistics on the training to input comparison to determine if additional training is needed to cover the applied use case population, or to watch for changes in the input process that suggests the state of the network or system is deviating from the believed use case conditions. In addition, those AD decisions acted upon can provide feedback to better understand if low validity in the AD result aligns with incorrect decisions, or not, to further determine if specific types of impairments, and their specific qualities in terms of $\boldsymbol{d}_{i,\boldsymbol{j}}$ are needed.

For an example of how this could be done in practice, think of the meta-category categories $\mathcal{J}$ as the "classes" of the AD (assumed to be implemented via deep learning), while the types $\boldsymbol{j} \in \mathcal{J}$ are "labels" and the AD is a multi-class classifier that can assign more than one label to an input i'. The outputs of the softmax layer of the AD corresponding to the different classes correspond to the confidence of the AD's classification. If the confidence assigned by the AD to its highest-confidence class is not high enough, then a human expert may be employed to manually assign the class and label corresponding to this input and thereby add to the

curated training data for the AD. Moreover, even the existing inputs in the training data should be slightly dithered by adding noise to create adversarial examples and thereby strengthen the resilience of the AD to such inputs when it is deployed.

In general terms, the procedure described is designed to characterize the input training space for a given AD or other AI or ML solution in specific terms for each input to the trained model to explain the model as well as provide information that helps to validate the result.

CONCLUSIONS

For ML models applied to network operations use cases in modern communication networks (both wired and wireless), we believe a human-in-the-loop supervised learning process may be crucial for XAI systems to win the confidence of human experts that these XAI systems may be allowed to operate autonomously with only occasional oversight by humans. But we suggest some new options to help reduce the adoption friction by providing explainability and evidence of trustworthiness along with ML solutions by formulating effectiveness measures and providing support for trust and validity to the human user.

ACKNOWLEDGMENT

It is a pleasure to acknowledge discussions with our colleagues in the cable industry, both within and outside CableLabs.

REFERENCES

[1] D. Gunning and D. Aha, "DARPA's explainable artificial intelligence (XAI) program," *AI Magazine*, vol. 40, no. 2, pp. 44–58, 2019.

[2] A. Rawal, J. Mccoy, D. B. Rawat, B. Sadler and R. Amant, "Recent Advances in Trustworthy Explainable Artificial Intelligence: Status, Challenges and Perspectives," in *IEEE Transactions on Artificial Intelligence*, 2022 (to appear).

[3] A. Hanif, X. Zhang and S. Wood, "A Survey on Explainable Artificial Intelligence Techniques and Challenges," *2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW)*, Gold Coast, Australia, 2021, pp. 81-89.

[4] M. Ravi, A. Negi and S. Chitnis, "A Comparative Review of Expert Systems, Recommender Systems, and Explainable AI," *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, Mumbai, India, 2022, pp. 1-8.

[5] J. Chandrasekaran, Y. Lei, R. Kacker and D. R. Kuhn, "A Combinatorial Approach to Explaining Image Classifiers," *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Porto de Galinhas, Brazil, 2021, pp. 35-43.

[6] Y. Hailemariam, A. Yazdinejad, R. M. Parizi, G. Srivastava and A. Dehghantanha, "An Empirical Evaluation of AI Deep Explainable Tools," *2020 IEEE Globecom Workshops (GC Wkshps)*, Taipei, Taiwan, 2020, pp. 1-6.

[7] M.T. Ribeiro, S. Singh and C. Guestrin. "*"Why should I trust you?"* Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135-1144.

[8] C. S. Hernandez, S. Ayo and D. Panagiotakopoulos, "An Explainable Artificial Intelligence (xAI) Framework for Improving Trust in Automated ATM Tools," *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, San Antonio, TX, USA, 2021, pp. 1-10.

[9] Q. Zhou, J. Zhu, J. Zhang, Z. Jia, B. Huberman and G.-K. Chang, "Intelligent Bandwidth Allocation for Latency Management in NG-EPON using Reinforcement Learning Method," *2020 Conference on Lasers and Electro-Optics (CLEO)*, 2020, pp. 1-2.

[10] M. Teflian, J. Ryan, R. Pettus, C. Welin, J. Keller, J. Paul, M. Eagles, B. Huberman, S. Mukherjee et al., "AI-Enabled Operating Model—The Communications Industry of the Near Future," CableLabs, 2022.

[11] J. Zhu, K. Sundaresan and J. Rupe, "Proactive Network Maintenance using Fast, Accurate Anomaly Localization and Classification on 1-D Data Series," *2020 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2020, pp. 1-11.

[12] J. Rupe and J. Zhu, "Kickstarting Proactive Network Maintenance with the Proactive Operations Platform and Example Application," *SCTE Expo* 2019.

[13] T. N. Nguyen and R. Choo, "Human-in-the-Loop XAI-enabled Vulnerability Detection, Investigation, and Mitigation," *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Melbourne, Australia, 2021, pp. 1210-1212.