Explaining What a Neural Network has Learned: Toward Transparent Classification

Kasun Amarasinghe, Milos Manic
Department of Computer Science
Virginia Commonwealth University
Richmond, Virginia, USA
amarasinghek@vcu.edu, misko@ieee.org

***Abstract*—Deep Neural Networks (DNNs) have limited ability to explain their acquired knowledge or decision rationale. As a result, end-users perceive DNNs as black-boxes and are hesitant to fully adopt them in safety-critical applications. Therefore, developing explainable DNNs has become a prime interest in neural network research. This paper presents a methodology for linguistically explaining the knowledge a DNN classifier has acquired in training. The main objective is to help users understand what the DNN has learned about each class. The presented methodology is fuzzy logic based and involves end-users of the system in the explanation process, enabling users to customize the explanations to match their requirements. This paper presents the explanation methodology, metrics of explanation quality, validation steps, and a discussion of advantages and limitations. The explanation methodology was implemented on a benchmark classification problem. Experimental results demonstrated the method's capability to explain the DNN-knowledge and validated the quality metrics.**

***Index Terms*—Explainable Artificial Intelligence, Interpretable Neural Networks, Explainable Neural Networks, Deep Neural Networks, Linguistic Summarization, Fuzzy Logic**

I. Introduction

Deep Neural Networks (DNNs) have shown unprecedented performance in a multitude of domains [1], [2]. Despite the impressive performance, there is a lack of trust in DNN systems among end-users [3], [4]. As a result, there is a reluctance to fully adopt DNNs in safety-critical domains such as medical diagnoses [4].

It is recognized that the principal reason behind the lack of trust is the inability of DNNs to explain their decisions and thus being perceived as black-boxes [4], [3], [5]. Therefore, there is no insight whether the accuracy of the DNN is achieved with a concrete rationale or due to artifacts in data [6], [7]. Therefore, an essential step to build trust between humans and DNN systems is to break open the said black box. The users understanding how the DNN system comes to its conclusions is essential in achieving the goal of trustworthy Artificial Intelligence (AI). This desired quality of explaining the decision-making process of DNNs is named 'explainability' of DNNs [8].

In the last few years, there have been several attempts to address the issue of explainability in DNN/AI systems. The existing research approaches can be can be categorized into two groups: 1) altering the learning algorithms to learn explainable features, and 2) using additional methods with the standard learning algorithm to explain existing DNN algorithms. Furthermore, in the second approach, two types of explanations can be generated: 1) explanations of individual classification decisions, 2) explanations of the overall knowledge of the DNN.

In 2016, Defense Advanced Research Projects Academy (DARPA) spawned a program named Explainable Artificial Intelligence (XAI) [8]. XAI's goal is to develop a suite of algorithms that combine the learning performance of DNNs and the explainability of models such as decision trees. In studies that focused on methodologies for explaining existing DNNs, explaining individual prediction through visualization has been the focus [9], [10], [11], [12], [6], [4]. However, humans are more inclined to justify things verbally [13]. Hence, textual explanations would resonate more with humans than visualizations. Hendricks et al. presented a methodology for generating textual explanations for image classifications using a combination of DNN algorithms [14]. However, the generated explanation couldn't guarantee that the classification used the features described in the explanation [15]. Therefore, to the best of our knowledge, there is a gap in existing research for generating textual explanations of the overall behavior of a DNN.

The main contribution of this paper is a methodology for linguistically explaining the overall knowledge of a DNN classifier, without altering its learning algorithm. The presented framework can be used with any DNN classifier and the goal is to help a user to understand what the DNN has learned about each class it's trained to classify. In this paper, these classes are referred to as DNN concepts. Derived explanations answer the question "is it (DNN) doing the right thing for the right reasons?", whereas accuracy scores answer the question, "is it doing the right thing?". This paper elaborates the explanation methodology, implements the method on a benchmark classification problem and discusses methods for validating explanations. Furthermore, this paper presents a discussion of the advantages, limitations and possible extensions of the method.

The rest of the paper is organized as follows. Section II elaborates the presented explanation methodology; Section III presents the experimentation process; Section IV presents a discussion of advantages and limitations; and finally, Section V concludes the paper.

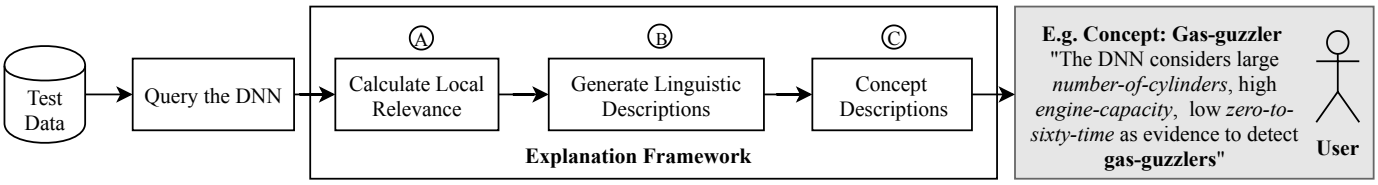

Fig. 1. Proposed methodology/framework for explaining the knowledge of a DNN

II. Explaining the Knowledge of a Neural Network

The goal of the derived explanations is to help users understand the knowledge the DNN has acquired from learning. A description is generated for each DNN concept (concept description). The concept descriptions contain input feature behavior that drives the DNN toward that concept (See Figure 1)

The presented methodology requires access to the trained DNN (weights and biases), and a test dataset. In this work, a feed-forward neural network is used for simplicity. However, any DNN classifier could be used. Further, we assume that the DNN is already trained to achieve sufficient accuracy. The presented method consists of three main steps: 1) calculating the local relevance of input feature, 2) generating linguistic descriptions, and 3) generating user feedback.

The method requires querying the DNN with a test dataset. The presented method uses typical DNN inference with ReLU activated hidden layers and a softmax-normalized output layer. Other components are discussed in detail below.

A. Calculating Input Feature Local Relevance

In this work, local relevance of an input feature is defined as a quantitative measure of its contribution to an individual classification decision. Explanations are based on the input features and their local relevance scores. Depending on the DNN architecture, different methodologies can be used in this step. Methods such as, sensitivity analysis [17], Deconvolution [10], [11] or Layer-wise relevance propagation [6], [19], [20] can be used to calculate the local relevance of input features. In this work, Layer-wise relevance propagation (LRP) is used.

Bach et al. introduced LRP as an approach for understanding pixel-wise contributions to image classification [6], [20]. LRP leverages the layered structure of a DNN. Each neuron of each layer has a relevance score $(R_d^{(l)})$ where d is the neuron (dimension) and the l is the layer.

LRP method calculates relevance scores for layer l when relevance scores for layer $(l+1)$ are available. The relevance scores are propagated backward through the layers from the output layer to the input layer in a single backward pass. Figure 2 depicts the process of propagating relevance scores through the layers. Relevance scores are back-propagated in messages, $R_{i\leftarrow j}^{(l,l+1)}$ (from neuron j in $l+1$ to neuron i in l). The LRP method propagates the relevance scores such that the total relevance is conserved through the layers.

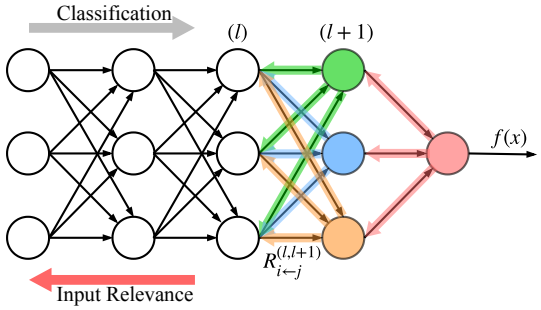

Fig. 2. Propagating relevance scores through the layers of the DNN in LRP

$$f(x) = \cdots = \sum_{d\in l+1} R_d^{(l+1)} = \sum_{d\in l} R_d^{(l)} = \cdots = \sum_{d} R_d^{(l)} \quad (1)$$

$$\sum_{i} R_{i\leftarrow j}^{(l,l+1)} = R_j^{(l,l+1)} \quad (2)$$

$f(x)$ is the output (unnormalized probabilities in the DNN) and $R_d^{(1)}$ is the relevance score of the d$^{\text{th}}$ dimension of the input layer. The relevance score of the i$^{\text{th}}$ neuron in layer l can be expressed as:

$$R_i^{(l,l+1)} = \sum_{j} R_{i\leftarrow j}^{(l,l+1)} \quad (3)$$

This relevance scores are distributed based on the ratio of pre-activations as follows:

$$R_{i\leftarrow j}^{(l,l+1)} = \left(\frac{a_i^{(l)} w_{ij}^{(l,l+1)}}{\Sigma_i a_i^{(l)} w_{ij}^{(l,l+1)} + b_j^{(l+1)}} \right) \cdot R_j^{(l+1)} \quad (4)$$

where, a^l is the activation of the l$^{\text{th}}$ layer, $w_{ij}^{(l,l+1)}$ is the weight of the connection between i$^{\text{th}}$ neuron in layer l and j$^{\text{th}}$ neuron in layer $l+1$ and $b_j^{(l+1)}$ is the bias of the j$^{\text{th}}$ neuron in layer $l+1$.

With the above propagation rule, the relevance score of each input feature $d; R_d^{(l)}$ for each test prediction is obtained. A positive $R_d^{(l)}$ indicates that dimension d supports detection of the concept and vice versa.

B. Generating Linguistic Descriptions

Linguistic description (LD) generation takes place after the calculation of local relevance scores.

Depending on user requirements, different types of LDs can be derived in this module. In this work, the focus is on

generating LDs that explain the relationship between input feature values and their relevance to classification. Only the input feature behaviors that support the detection of each concept are considered.

Since LRP propagates the unnormalized probabilities of the NN-ADS, the relevance scores are in different scales. Therefore, in order to ensure consistency, relevance scores are scaled to the same range prior to generating LDs. Further, relevance scores can be positive, zero or negative. This work only considers feature behavior that supports classification decisions (positive relevance). The relevance scores are normalized as follows. The normalized score is referred to as the feature influence.

$$I_d = \begin{cases} \frac{R_d}{\sum_i R_i^{(+)}}, \text{if } R_d > 0 \\ 0 \qquad\quad , otherwise \end{cases} \tag{5}$$

Where, I_d is the local influence score of the d^{th} input feature, R_d is the local relevance, and $R_i^{(+)}$ are the features with positive relevance scores. Influence score can be viewed as the relative positive relevance of the input feature, and it ensures that the influence scores are scaled between 0 and 1.

In this work, the derivation of LDs makes use of Linguistic Summarization (LS) techniques. LS techniques are used in data mining to obtain succinct descriptions of databases [21], [22], [23]. All the LDs derived in this work are based on type-I fuzzy sets introduced by Zadeh [24]. The LDs take the form of an IF-THEN type linguistic summary [25]. Given below, is an example LD derived for a DNN classifying cars into two classes: eco-friendly and gas-guzzler.

$$\text{IF } engine\_capacity \text{ IS } high \text{ THEN } influence \text{ IS } high \tag{6}$$

where $engine\_capacity\ IS\ high$ portion represents the input feature and its behavior, $influence\ IS\ high$ represents the level of influence, and low and $high$ are type-I fuzzy sets for the input feature behavior and influence respectively.

Since fuzzy inference is used in generating the LDs, the calculated influence scores and input features are fuzzified into a preset number of fuzzy sets. The number of fuzzy sets, labels, and shapes can be changed for each dimension and application/domain to make the generated LDs descriptive and customized. These choices are entirely at users' discretion and the application requirements. Alternatively, data-driven techniques can be used to determine the optimal configuration of the fuzzy sets [26].Details on fuzzification of crisp influence and feature values are omitted in this paper.

The quality of an LD is assessed using two measurements: 1) degree of truth (DT) and 2) degree of coverage (DC). These metrics were proposed by Wu et. al for IF-THEN linguistic summaries [27]. If an LD is of high quality, both these figures should be high. Similarly to the fuzzy set configuration, the acceptable levels of DT and DC are entirely at the users' discretion.

DT is a measurement of accuracy. DT is calculated using the fuzzy membership degrees of each data record to the antecedent and consequent fuzzy sets. A high DT would imply that the LD is accurately describing the DNN behavior. DT is calculated as follows:

$$T = \frac{\sum_{m=1}^{M_c} min(\mu_{S_a}(v_{m,a}), \mu_{S_c}(v_{m,c}))}{\sum_{m=1}^{M_c} \mu_{S_a}(v_{m,a})} \tag{7}$$

where, $\mu_{S_a}(v_{m,a})$, is the degree of membership of the m$^{\text{th}}$ test prediction to the antecedent fuzzy set (behavior of the input feature) and $\mu_{S_c}(v_{m,c})$ is the degree of membership of the m$^{\text{th}}$ test prediction to the consequent fuzzy set (level of input feature influence).

DC is a measure that indicates how good the LD is in terms of generalization. DC is a non-linear mapping of the portion of data which satisfies the LD. A high DC would imply good generalization. For instance, a high DT and a low DC can be an indication of LDs describing outliers. Therefore, both quality measures are equally important. DC is calculated as follows:

$$C = f_c\left(\frac{\sum_{m=1}^{M_c} t_m}{M_c}\right) \tag{8}$$

where:

$$t_m = \begin{cases} 1, & \text{if } min(\mu_{S_a}(v_{m,a}), \mu_{S_c}(v_{m,c})) > 0 \\ 0, & otherwise \end{cases} \tag{9}$$

The function f_c is a sigmoid function, This maps the ratio of data points that satisfy the LD to a value between 0 and 1. The shape of the function can be fine-tuned to match the user requirements.

C. Generating user feedback

The final explanation of the DNN is presented to the user in the form of concept descriptions. Once the high-quality LDs are derived, a concept description is generated for each DNN concept.

Concept descriptions can take different semantic and syntactic arrangements depending on the user requirements. For example, in a classifier that classifies cars into two classes, eco-friendly and gas-guzzlers, the concept description of a "gas-guzzler" can be expressed as follows:

> —The DNN considers *large number-of-cylinders, high engine-capacity, high seating-capacity, low zero-to-sixty-time* as evidence for detecting a gas-guzzler

where, *number-of-cylinders, engine-capacity, seating-capacity* and *zero-to-sixty-time* are input features. and the specific behavior are extracted from the LDs with high DT and DC.

These concept descriptions help to uncover the input feature behavior the DNN uses to identify a specific class/concept. These input feature behaviors act as the rationale, which drives the decisions of the DNN. Therefore, by inspecting all the concept descriptions, users can infer the process of decision-making in the DNN.

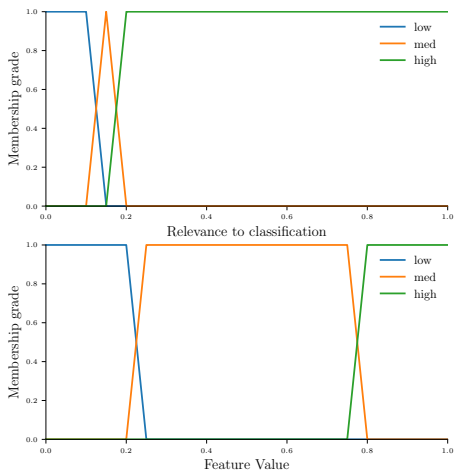

Fig. 3. The fuzzy membership functions used to fuzzify feature influence (top) and feature value (bottom)

III. Experiments

In order to demonstrate the explanation methodology, a benchmark intrusion detection dataset, NSL-KDD was used. NSL-KDD is a popular dataset for testing network intrusion detection systems [28]. The NSL-KDD contains network traffic data of 41 features for five classes: 1) normal communication, 2) Denial-of-Service (DoS) attacks, 3) Probe attacks, 4) Root-to-local attacks (R2L), and 5) User-to-local (U2R). For brevity and simplicity, this study only considered classification between normal communication and DoS attacks (128, 722 records).

As mentioned, all the input features and influence scores are fuzzified. In a real world application, these fuzzy set configurations would be adapted to each dataset and each input feature. In this work, the same fuzzy sets were used to fuzzify all input features for simplicity. Similarly, the same fuzzy set configuration was used fuzzify the influence levels of all input features. The presented method is independent of the fuzzy set configuration. Therefore, the fuzzy sets can be changed without affecting the explanation methodology.

In this work, the antecedent and the consequent were fuzzified into three fuzzy sets—low, medium and high—using the fuzzy sets shown in Figure 3.

A. Experimental Results

The DNN classifier was able to classify the NSL-KDD data with a test accuracy of 99.35%. It has to be noted that the explanation methodology is only applied to DNNs which are capable of achieving desired classification accuracy levels and classification accuracies are reported for completeness.

Once the model with the highest classification accuracy was selected, the explanation methodology was applied. For the NSL-KDD dataset, the DNN was trained to learn two concepts, Normal and DoS. Table I shows the LDs for the DoS concept and Table II shows the LDs for 'normal' concept.

TABLE I
LINGUISTIC DESCRIPTIONS OF 'DOS' CONCEPT IN DNN TRAINED FOR NSL-KDD DATASET

No.	Linguistic Description	DT	DC
LD-01	IF dst_host_rerror_rate IS high THEN influence IS high	0.99	1.0
LD-02	IF dst_host_serror_rate IS high THEN influence IS high	0.80	1.0
LD-03	IF srv_error_rate IS high THEN influence IS med	0.71	1.0
LD-04	IF logged_in IS low THEN influence IS low	0.99	1.0
LD-05	IF dst_host_srv_serror_rate IS high THEN influence IS low	0.99	1.0
LD-06	IF dst_host_srv_serror_rate IS high THEN influence IS low	0.99	1.0
LD-07	IF dst_host_same_srv_rate IS low THEN influence IS low	0.99	1.0
LD-08	IF dst_host_same_src_port_rate IS low THEN influence IS low	0.99	1.0
LD-09	IF dst_host_srv_serror_rate IS high THEN influence IS low	0.99	1.0
LD-10	IF dst_host_count IS high THEN influence IS low	0.98	1.0

TABLE II
LINGUISTIC DESCRIPTIONS OF 'NORMAL' CONCEPT IN DNN TRAINED FOR NSL-KDD DATASET

No.	Linguistic Description	DT	DC
LD-01	IF wrong_fragment IS low THEN influence IS low	0.98	1.0
LD-02	IF srv_rerror_rate IS low THEN influence IS low	0.98	1.0
LD-03	IF dst_host_srv_rerror_rate IS low THEN influence IS low	0.98	1.0
LD-04	IF rerror_rate IS low THEN influence IS low	0.97	1.0
LD-05	IF dst_host_rerror_rate IS low THEN influence IS low	0.97	1.0
LD-06	IF logged_in IS high THEN influence IS low	0.97	1.0
LD-07	IF hot IS low THEN influence IS low, truth	0.97	1.0
LD-08	IF srv_error_rate IS low THEN influence IS low	0.91	1.0
LD-09	IF src_bytes IS low THEN influence IS low	0.91	1.0
LD-10	IF dst_host_count IS low THEN influence IS low	0.90	1.0

Only the LDs with a DT > 0.8 and DC > 0.9 were chosen in this work. These thresholds can be adapted to different user requirements.

'DoS': It was observed that the DNN was influenced by factors such as high error rates, not logged in connections, high speed of communication, and connections being spread across different services. Two feature behaviors were noted to be highly influential, high *dst_host_rerror_rate* and *dst_host_serror_rate*. High *srv_error_rate* showed a medium level of influence and several others with low influence. Therefore, using these LDs, a concept description was generated for DoS. Users have the flexibility to choose what levels of influence to use and what DT and DC threshold to use for these descriptions. For instance, the DoS description can be expressed as follows:

> —The DNN considers, **high values** for *dst_host_rerror_rate, dst_host_serror_rate, srv_error_rate, dst_host_srv_serror_rate, dst_host_srv_serror_rate, dst_host_srv_serror_rate,* and *dst_host_count* and **low values** for *logged_in, dst_host_same_srv_rate,* and *dst_host_same_src_port_rate*, as evidence for detecting a DoS attack.

'Normal': It was observed that there were no input feature behaviors that stood out. Instead, a range of input features influenced at a low level. This observation is agreeable with the real world scenario as normal communications should be detected by looking at the system as a whole. It was observed that the DNN was influenced to detect normal communications by low error rates, correctly logged in status, and low communication speeds. As with the DoS attacks, a concept description can be generated for normal communication using the LDs.

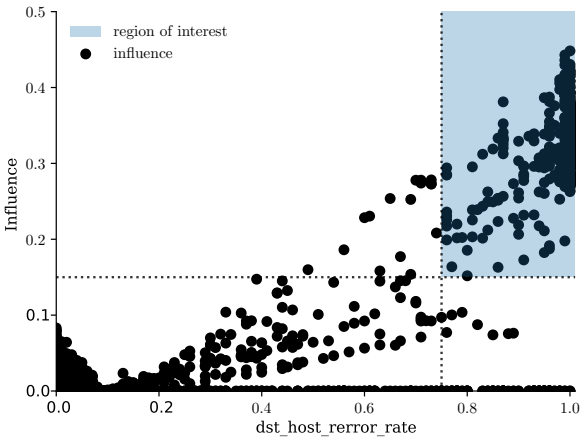

Fig. 4. Influence with Feature value: IF *dst_host_rerror_rate* IS high THEN influence IS high (LD-01, DoS)

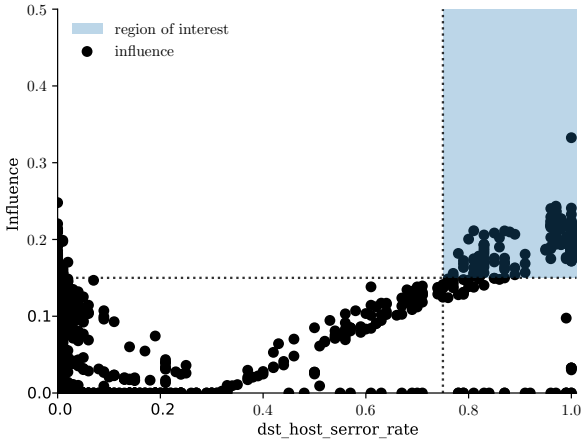

Fig. 5. Influence with Feature value: IF *dst_host_serror_rate* IS high THEN influence IS high (LD-02, DoS)

B. Validating explanations

In addition to the quality measures that were proposed, it is important to devise methods to validate the generated high quality LDs.

As a simple method of validation, a visual inspection of the influence (consequent) score against the feature value (antecedent) was carried out. This was possible since the generated LDs were single-antecedent-single-consequent. Figures 4 and 5 show the influence scores against the feature value for the 'DoS' concept. The shaded region indicates the region of interest for the LD being validated. It can be observed that the influence scores followed the pattern indicated by the LD.

In addition to visually analyzing the LDs, the generality of the LDs was tested by using different subsets of the test dataset to generate the LDs and the distribution of DT values was observed. Figure 6 shows a box-and-whisker plot of the DT values distribution of LDs (top 5 LDs) for the DoS concept. It was observed that the DT values remained consistent across different test datasets, which indicated good generalization of the LDs.

These analyses help users to filter the LDs generated from the system even further and improve the explanation process. Therefore, using these tests, the concept descriptions can be more refined by using only the LDs that pass the validation checks.

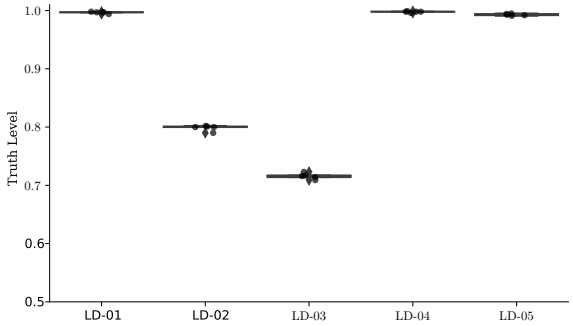

Fig. 6. Variability of DT for some LDs derived with different test datasets.

IV. Discussion

This section discusses the advantages, limitations and how the method can be adapted to complex real-world scenarios.

The **primary advantage** of the presented methodology is the ability to understand the knowledge DNN has gained in training. This is a substantial improvement of transparency for black-box DNNs. The **second advantage** is the ability to evaluate the DNN based on its decision rationale. This is an additional layer of evaluation. **Thirdly**, the presented methodology is customizable. Users have the ability to adapt the explanations to the application/user requirements and can cater to users of different levels (E.g. Technicians vs Managers). Additionally, the presented method involves the user in the explanation process. Therefore, the method is more transparent to the user.

The main limitation of the presented methodology is that it cannot be applied if the position of the object of interest is not consistent in the feature space. For instance, this methodology cannot be applied directly to explain image classifiers at pixel-level. In order to use this method in such a case, the features will need to be in a higher order, so that the position of the object of interest is consistent in the feature space. Therefore, the presented method is not suitable for generating linguistic explanations in domains such as computer vision.

In complex real-world scenarios, systems in question will typically consist of very high dimensional data (E.g. sensor readings from a thermo-chemical plant). In such cases, explanations based on individual features can be incomprehensible. As a solution, input features can be grouped into categories (or a hierarchy of categories) as a preprocessing step. Then, the explanations can be derived using the input feature categories instead of using individual features. Further, the complexity of the LDs can be increased for improved precision. For instance, multi-antecedent explanations (E.g. IF feature1 IS high AND

feature2 IS low THEN influence IS high) can be derived.

V. Conclusions

This paper presented a fuzzy logic based methodology for explaining the overall knowledge of a trained DNN classifier. The presented methodology can be used to derive linguistic explanations of what a DNN classifier has learned about each class during training. This paper elaborated the explanation methodology, explanation quality metrics and validation methods. The presented method was implemented on a benchmark classification problem (intrusion detection using NSL-KDD). Experimental results showed that the validated explanations improved transparency of the DNN classifier while providing an additional evaluation layer. Further, results validated the quality metrics. As future work, complex explanations and additional validation methods will be explored.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[3] Jason Bloomberg, "Don't Trust Artificial Intelligence? Time To Open The AI 'Black Box'," 2018. [Online]. Available: https://www.forbes.com/sites/jasonbloomberg/2018/09/16/dont-trust-artificial-intelligence-time-to-open-the-ai-black-box/.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *22nd International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: ACM, 2016, pp. 1135–1144.

[5] IBM, *What's next for AI–Building trust*. [Online]. Available: https://www.ibm.com/watson/advantage-reports/future-of-artificial-intelligence/building-trust-in-ai.html (visited on 11/28/2018).

[6] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation," *PLoS One*, vol. 10, no. 7, O. D. Suarez, Ed., e0130140, 2015.

[7] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Muller, "Evaluating the Visualization of What a Deep Neural Network Has Learned," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 11, pp. 2660–2673, 2017.

[8] D. Gunning, *Explainable Artificial Intelligence*, 2016. [Online]. Available: https://www.darpa.mil/program/explainable-artificial-intelligence (visited on 04/26/2018).

[9] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, no. 3, p. 1, 2009.

[10] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[11] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *2011 Int. Conf. Comput. Vis.*, IEEE, 2011, pp. 2018–2025.

[12] H. J. Escalante, I. Guyon, S. Escalera, J. Jacques, M. Madadi, X. Baro, S. Ayache, E. Viegas, Y. Gucluturk, U. Guclu, M. A. J. van Gerven, and R. van Lier, "Design of an explainable machine learning challenge for video interviews," in *2017 Int. Jt. Conf. Neural Networks*, IEEE, 2017, pp. 3688–3695.

[13] Z. C. Lipton, "The Mythos of Model Interpretability," *arXiv preprint arXiv:1606.03490*, 2016.

[14] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, "Generating Visual Explanations," *arXiv preprint arXiv:1603.08507*, 2016.

[15] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, M. Srivastava, A. Preece, S. Julier, R. M. Rao, T. D. Kelley, D. Braines, M. Sensoy, C. J. Willis, and P. Gurram, "Interpretability of Deep Learning Models: A Survey of Results," in *IEEE Smart World Congr. DAIS - Work. Distrib. Anal. Infrastruct. Algorithms Multi-Organization Fed.*, 2017.

[16] D. Baehrens, T. Schroeter, S. Harmeling, K. Hansen Khansen, and C.-b. Klaus-Robert Mueller, "How to Explain Individual Classification Decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, 2010.

[17] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *arXiv preprint arXiv:1311.2901*, 2014.

[18] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," *arXiv preprint arXiv:1411.4389*, 2014.

[19] G. Montavon, W. Samek, and K.-R. Müller, "Methods for Interpreting and Understanding Deep Neural Networks," *arXiv preprint arXiv:1706.07979*, 2017.

[20] A. Binder, G. Montavon, S. Bach, K.-R. Müller, and W. Samek, "Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers," *arXiv preprint arXiv:1604.00825*, 2016.

[21] R. R. Yager, K. M. Ford, and A. J. Cañas, "An approach to the linguistic summarization of data," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 521 LNCS, no. July 1990, pp. 456–468, 1991.

[22] J Kacprzyk, "Fuzzy logic for linguistic summarization of databases," in *Fuzzy Syst. Conf. Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE Int.*, vol. 2, 1999, 813–818 vol.2.

[23] J. Kacprzyk and S. Zadrożny, "Linguistic Database Summaries Using Fuzzy Logic: Towards a Human-Consistent Data Mining Tool," Tech. Rep., 2009.

[24] L. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.

[25] D. Wu, J. M. Mendel, and J. Joo, "Linguistic summarization using IF-THEN rules," in *Int. Conf. Fuzzy Syst.*, IEEE, 2010, pp. 1–8.

[26] D. Wijayasekara and M. Manic, "Data driven fuzzy membership function generation for increased understandability," in *2014 IEEE Int. Conf. Fuzzy Syst.*, IEEE, 2014, pp. 133–140.

[27] D. Wu and J. M. Mendel, "Linguistic summarization using IFTHEN rules and interval Type-2 fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 136–151, 2011.

[28] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, no. Cisda, pp. 1–6, 2009.

[29] W. H. Wolberg, W. N. Street, and O. Mangasarian, *UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set*.