Towards Unsupervised Deep Learning Based Anomaly Detection

Trevor Landeen, *Student Member, IEEE,* and Jacob Gunther, *Member, IEEE*

***Abstract*—Novelty or anomaly detection is a challenging problem in many research disciplines without a general solution. In machine learning, inputs unlike the training data need to be identified. In areas where research involves taking measurements, identifying errant measurements is often necessary and occasionally vital. When monitoring the status of a system, some observations may indicate a potential system failure is occurring or may occur in the near future. The challenge is to identify the anomalous measurements that are usually sparse in comparison to the valid measurements. This paper presents a land-water classification problem as an anomaly detection problem to demonstrate the inability of a classifier to detect anomalies. A second problem requiring the identification of anomalous data uses a deep neural network (DNN) to perform a nonlinear regression as a method for the estimation of the probability that a given input is valid and not anomalous. A discussion of autoencoders is then proposed as an alternative to the supervised classification and regression approaches in an effort to remove the necessity of representing the anomalies in the training dataset.**

I. Introduction

ANOMALY detection, or synonymously novelty and outlier detection, is a problem of interest in many research disciplines. Whenever a dataset is involved, whatever the source may be, the possibility of novel samples being present exists. The method of handling the novel samples varies between applications. Where one application may treat novel data as an outlier to be removed from the dataset, another may view a novel input as significant and begin an alternate processing path. The purpose of anomaly detection is to robustly identify when input samples don't fit the standard, expected model of a normal behaving input.

A specific challenge with anomaly detection is first identifying and then quantifying what constitutes as the normal, expected behavior. What is normal, allowed, and expected in one research area is not likely to be considered normal or expected in an unrelated field. Furthermore, the quantification of the differences between normal and anomalous implies some comparison between the two sets, which requires access to enough samples from both valid and novel samples. When anomalous samples are present, they are likely sparse and occur irregularly and may not even be representative of the set of anomalous samples as whole. In some cases, access to the novel samples ahead of time may not be possible. For example, consider a system monitoring the status of a nuclear reactor. If the measurements returned by the system indicate possible reactor failure, they are anomalous as development of the processing system would have no safe access to a failing reactor in order to develop the baseline. The detection of the anomaly is important the handling of it should not simply throw it out.

The remainder of this paper will continue to build on the idea of detecting anomalous samples as follows. Section II presents both basic statistical based anomaly detection and a basic neural network based anomaly detection approaches. Two different supervised detection problems and their accompanying solutions are presented in Section III. The first is a binary classification problem posed as an anomaly detection problem, then solved using a deep neural network (DNN) to demonstrate some inherent weaknesses. The other problem is formulated as a regression problem solved using a DNN to learn the nonlinear regression. An overview of some unsupervised DNN approaches to be the subject of future research is given in section IV.

II. Review of Anomaly Detection Approaches

In a survey by Hodge and Austin [1], the authors give examples of outliers or anomalies and then discuss several approaches to detect them. They first discuss statistical models and identify an approach which allows a human auditor to visually identify any outlying data. They also present approaches which make discriminate between normal and novel using distance metrics such as the Mahalanobis distance. They review neural networks and other machine learning approaches before concluding that there is no universally applicable or generic outlier detection approach. The authors then emphasize the importance of the decision on how to handle any detected anomalies.

Augusteijn and Folkert argue in [2] that neural network classifiers trained using backpropagation cannot detect novel patterns. They then explore the probabilistic neural network and conclude that is is much more suited to the task than standard classifiers. Yadav and Devi pose the anomaly detection problem as a classification problem in [3]. The authors train a probabilistic neural network classifier to estimate the probability that a specific input is a member of every class. To provide the negative inputs necessary when training a classifier, they use a combination of hyperspheres and a k-Nearest Neighbor method to artificially generate new samples.

An in depth look at neural network based novelty detection was conducted by Markou adn Singh and presented in a summary survey[4]. The authors discuss the use of multilayer perceptrons, auto-associative networks, Hopfield netowrks, and others and concluded by citing the lack of direct comparisons between the different networks that currently exist in different fields of research as a problem hindering general progress.

The authors in [5] include a section on probabilistic novelty detection in their survey that other surveys overlooked. The

fundamental concept of these approaches is that the valid data is generated by probability density function. Estimating the generative probability density function, either with parametric or non-parametric approaches, allows the approach to decide if the inputs were generated under the same distribution. Parametric approaches include mixture models and state-space models. The authors also discuss reconstruction-based approaches, which reconstruct the inputs when presented to the system and base the decision on some distance between the input sample and the reconstruction sample.

More recently, Chandola, Banerjee, and Kumar provide another survey discussing the use of neural network based anomaly detection[6]. Results using recent progress in deep neural networks are still sparse.

Any approach using neural networks can be designated as supervised, semi-supervised, or unsupervised. The distinction is drawn from the neural network training process, specifically the availability of labeled output target values. The following sections discuss the use of both supervised and unsupervised approaches.

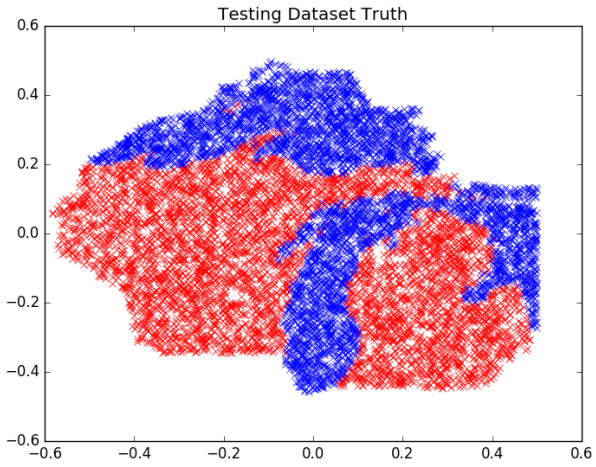

Fig. 1. Great lakes classification problem. The red points are land measurements and comprise the region of interest. The blue points are water measurements and are to be ignored.

III. Applications of Supervised Approaches

A. Novelty Detection through Classification

As background to this classification problem, the Global Ozone Monitoring Experiment-2 (GOME-2) is a flying optical spectrometer developed by the European Space Agency to measure atmospheric ozone, trace gases, and ultraviolet radiation[7]. GOME-2 has near daily global coverage and its measurements are freely available to researchers. In many cases, automated processing is implemented to parse the observations made over a region of interest. The region of interest may be non-standard and easily specified, thus requiring an increased-complexity processing algorithm.

Consider the case where the region of interest is the land area of Wisconsin and Michigan. Let the land measurements be considered valid while all others, but specifically the measurements over water, are considered invalid and therefore anomalous. In this application, The anomaly detection algorithm simply needs to identify when measurements are not made over land and then remove them from the dataset. In this example, the valid and invalid classifications allow the problem to be easily formed as a binary classification problem. Either a measurement is a land measurement or it isn't (it's a water measurement). More formally, define the classification to be:

$$c(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \notin A \\ 1 & \mathbf{x} \in A \end{cases} \tag{1}$$

where A is the set of all anomalous measurements. This application has the advantage over true anomaly detection problems because the anomalies are well known beforehand and represented proportionally in the dataset. While it is an abuse of terminology to consider the water anomalous, the lessons learned are still valid.

Prior to using a deep neural network (DNN) as a classifier, the DNN needs to first be trained using a labeled dataset. The natural feature to use in the classifier for this application is the latitude and longitude position. The dataset was generated using QGIS, an open source geographic information system, which provides the latitude and longitude of randomly sampled points for both the land and the water surrounding the region of interest. The normalized, labeled dataset is shown in Figure 1, with the classes represented by color, red is land and blue is the anomaly class (i.e. water).

The DNN was defined and trained using TFLearn running on Tensorflow. Upon completion of the DNN training, a separate validation dataset (generated using the same process as before) is used as a test for the DNN classifier. The positions in the validation dataset are kept separate from the training dataset to avoid biasing the classification. The DNN classifies every input as either land (keep it) or water (anomalous, so reject it). The validation dataset classifications made by the DNN are shown in Figure 2. The red points were classified as land and the blue points were classified as the anomalies with the black points indicating input sample points classified in error. Either they were valid and classified as anomalous or they were anomalous and classified as valid.

B. Issues with Classification

Many of the popular DNNs in literature (including this paper) are trained to classify inputs as belonging to one or more predefined classes. The classification problem is well documented over a broad range of application areas. In nearly every area though, the possible classes are well defined. This is of necessity rather than convenience and becomes apparent when considering the operations involved when a DNN classifier makes an estimate on an input. For fully connected DNNs the value of every neuron is computed by applying a nonlinear function to a linear combination of the previous layer's neurons. The sequence of layers performs a nonlinear mapping of the input space, where no linear separation exists between the classes, to the output space where the classes are sufficiently, linearly separable.

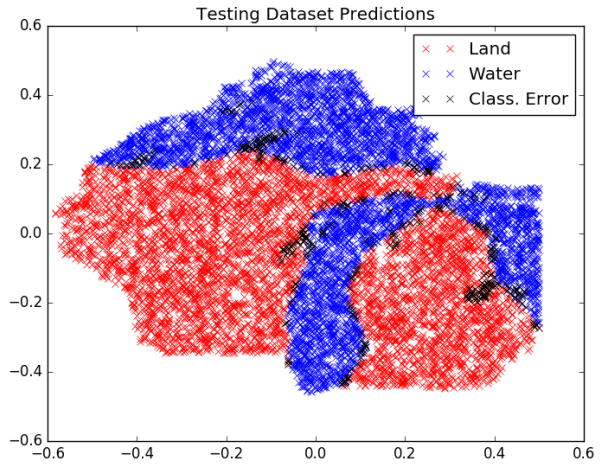

Fig. 2. Classifications of positions in the great lakes, Wisconsin, and Michigan region. Points classified by the DNN as water are blue and points classified by the DNN as land are red. Black points are the points classified incorrectly.

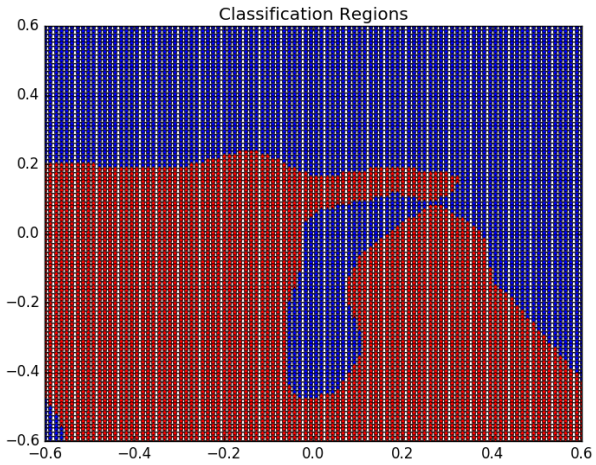

Fig. 3. Using a classifier to classify inputs not represented in the training dataset can result in erroneous classification. There is land north of Lake Superior and there isn't a body of water in the bottom left corner.

The nonlinear mapping is learned by an iterative process which adjusts the weights between each layer by attempting to minimize an objective function applied to the output of the DNN for a given input. Classification is nearly always a supervised learning approach which requires the class to be provided for every input during training. If a certain class is not adequately represented in the training data, there is no way to encourage or force a DNN to learn the desired behavior. This inability is significant because the class exists in the input space and is mapped somewhere to the output space where it will be classified according the region it is mapped to. The output classification will be made according to the 'rules' the model learned, but may seem egregiously incorrect if the true class is known. In some applications, any expectations of identifying the mistaken classification should be discarded.

Overall, the great lakes classifier works well and has high classification accuracy. However, the concern of classifying previously unseen data is demonstrated in Figure 3. The classified areas not included in the training datset include the land to the north of Lake Superior and the land below Michigan and Wisconsin. Any person who has seen a complete map of the area can easily identify these areas as land whereas the classifier incorrectly classifies the land north of Lake Superior and a small portion of the land in the bottom left corner as water. If the classification problem is still interpreted as anomaly detection, these areas would be classified incorrectly as anomalous and would then be removed from the dataset.

In this great lake classification problem, the areas where the classifier fails is not critical as it isn't ever considered to be in the region of interest. In other applications this issue may need to be addressed.

C. Regression

A DNN regression model is proposed as an alternative to the classification models. Rather than classify inputs as valid or invalid, the DNN learns a highly nonlinear regression model and is used to provide non-binary outputs for new inputs from the same domain. The regression approach is still supervised learning and doesn't break away from the dependence on representing anomalous data within the dataset. The regression DNNs require target values instead of target classes, and in this problem the target values are interpreted to be the probability that the input vector is valid. The decision of normal or anomalous is made using the estimated probability in a hypothesis test.

Using a regression based network to identify anomalous data still has challenges. The training dataset requires a target value and computing the target value is not necessarily trivial. In the end, the training data will include

$$\begin{aligned} \text{Inputs} &= \mathbf{x} \\ \text{Target Values} &= P(\mathbf{x}) \end{aligned} \tag{2}$$

Once the dataset is created and the training process is completed using common techniques [8]. After the DNN is trained it can then be used to estimate the probability that an input vector $\mathbf{x_i}$ is normal or novel. After the probability is provided, it is necessary to make the decision.

1) Hypothesis Testing: Define the hypotheses to be

$$\begin{aligned} H_0 &: \mathbf{x} \notin A \\ H_1 &: \mathbf{x} \in A \end{aligned} \tag{3}$$

where A is the set of anomalies. That is, A is the set of vectors $\mathbf{x}$ such that $\mathbf{x}$ isn't like other inputs previously seen. The decision to either reject the null hypothesis or fail to reject the null hypothesis can be made using a ratio of probabilities:

$$\ell(x) = \frac{P(\mathbf{x} \notin A)}{P(\mathbf{x} \in A)} \tag{4}$$

and define the decision rule

$$\phi(\mathbf{x}) = \begin{cases} \mathbf{x} \notin A & \ell(x) \geq \tau \\ \mathbf{x} \in A & \ell(x) < \tau \end{cases}. \tag{5}$$

To determine the decision threshold τ, express the ratio test as

$$\frac{P(\mathbf{x} \notin A)}{P(\mathbf{x} \in A)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau. \tag{6}$$

Which means decide the input isn't anomalous if the ratio is greater than or equal to τ and decide it is anomalous if ratio is less than τ. The set of inputs $\mathbf{x_i} \in A$ and the set of inputs $\mathbf{x_i} \notin A$ are disjoint, therefore

$$P(\mathbf{x} \notin A) + P(\mathbf{x} \in A) = 1 \tag{7}$$

and substituting into (6) to obtain

$$\frac{P(\mathbf{x} \notin A)}{1 - P(\mathbf{x} \notin A)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau \tag{8}$$

which can be solved for $P(\mathbf{x} \notin A)$ to get the simplified ratio test,

$$P(\mathbf{x} \notin A) \underset{H_1}{\overset{H_0}{\gtrless}} \frac{\tau}{1+\tau} = \tau'. \tag{9}$$

The decision rule (5) can now be rewritten in terms of the simplified ratio test as

$$\phi(\mathbf{x}) = \begin{cases} \mathbf{x} \notin A & P(\mathbf{x}) \geq \tau' \\ \mathbf{x} \in A & P(\mathbf{x}) < \tau' \end{cases} \tag{10}$$

The result in equation (10) is the decision rule based entirely on the probability estimated by the regression DNN. If the estimated probability for an input is greater than τ', the decision should be H_0, the input vector $\mathbf{x}$ is not an anomaly. If the estimated probability is less than τ', the decision should be H_1, the input vector $\mathbf{x}$ is an anomaly. The value of τ' needs to carefully selected to balance the probability of correctly detecting anomalies and probability of incorrectly deciding a normal sample is an anomaly.

To evaluate the performance of the DNN probability estimator, the quality of the estimates needs to be determined. To do this consider two types of errors made using the decision test. A **Type I** error, or false alarm, is deciding that $\mathbf{x}$ is not in A when it actually is. A **Type II** error, or missed detection, is deciding that $\mathbf{x}$ is in A when it actually is not. The decision test is evaluated by defining the probability that the type of error will occur. The probability of missed detection is given by first defining the probability of detection, P_D.

$$\begin{aligned} P_{FA} &= P(\text{decide } \mathbf{x} \in A \,|\, \mathbf{x} \notin A \text{ is true}) \\ P_D &= P(\text{decide } \mathbf{x} \in A \,|\, \mathbf{x} \in A \text{ is true}) \\ P_{MD} &= 1 - P_D \end{aligned} \tag{11}$$

A receiver operating characteristic (ROC) curve is generated by plotting the probability of detection vs. the probability of false alarm over the range of τ'. ROC curves for the DNN regression model are shown in Figure 4. The four curves are separate ROC curves where each represents the performance of the decision testing for varying levels of anomalies. Every anomaly is a small perturbation of a valid sample with the higher levels meaning increased similarities, thus more difficult decisions. The performance on the level 4 dataset (the smallest difference between anomalies and valid inputs) is mediocre. In order to detect the anomalies 90% of the time, nearly 40% of the detections will be incorrect. In more favorable conditions the detector is able to detect the anomalies 90% of the time while only incorrectly identifying the valid inputs as anomalies 5% of time.

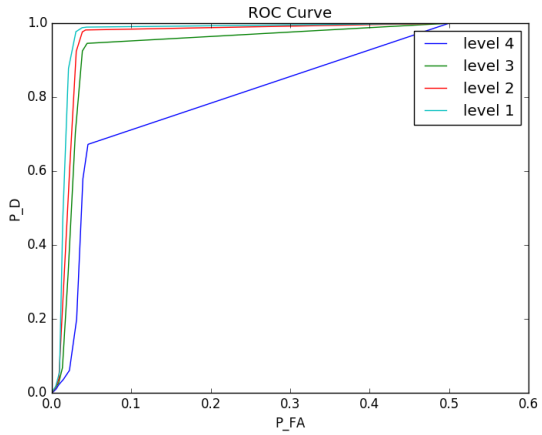

Fig. 4. ROC curves of the decision rule when the anomalies and normal inputs increasing levels of similarity. The higher the level, the higher the similarity between normal and novel.

IV. Allure of Unsupervised Approaches

The drawback of the regression model, even though it is more informative than a classification model, is its reliance on a labeled dataset. Furthermore, it requires the knowledge or intuition of how to assign the probabilities as the target values. Unsupervised models are attractive because they do not have the same reliance on labeled training data. One of the most popular unsupervised deep learning approaches is the autoencoder.

A. Autoencoders

An autoencoder, like other neural networks, has at least one hidden layer but it uses that hidden layer differently. The distinction between the autoencoder and other neural networks discussed thus far is because the autoencoder is considered unsupervised, i.e. it does not require labeled training data. Instead it learns to recreate the inputs on the output layer as shown in Figure 5. This is accomplished in two separate ways depending on the application. In one method the number of neurons in the hidden layer is less than the number of neurons in the input and output layers. The other method allows for the number of neurons in the hidden layer to be greater than the number of neurons in the input or output layers. A sparsity constraint must be enforced or the autoencoder will learn the uninteresting identity function. Both methods force the network to learn an alternate representation of the data to be used when solving a problem.

A fundamental idea exploited by autoencoders is known as the manifold hypothesis. A manifold is a neighborhood

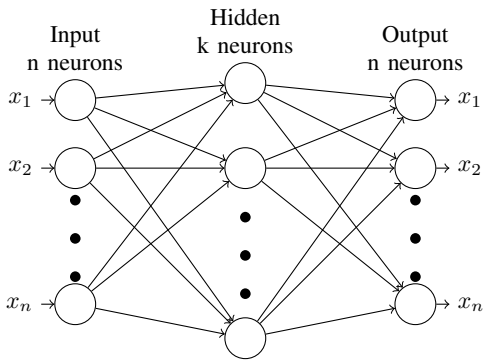

Fig. 5. This is an example of an autoencoder with a single hidden layer. The number of layers may be more than 1. Both the input and output layers are composed of *n* neurons. There number of neurons, *k*, in the hidden layer may be more or less than the number of neurons in the other layers if care is taken to prevent the model from learing the identity. The autoencoder's purpose is to reconstruct the inputs from the hidden layer representation learned during training.

of connected points and the manifold hypothesis is the idea that the probability distribution over the set of inputs lies in a highly concentrated region[9]. For example, consider writing a document and picking the letters at random. The probability that the letters form correctly spelled words is very small. The more likely scenario is that the document conveys no meaning and contains no real information. Thus, in the region of all possible character sequences, actual words only make up a very small, connected portion. This is the manifold hypothesis. Essentially the autoencoder operates under the assumption that in some given input space of dimension *n*, the inputs of interest are grouped together in a subspace of dimension *k* with $k < n$.

Work that has been done on anomaly detection using autoencoders is limited. The authors in [10] use a convolutional sparse autoencoder to learn features that a fault detection or anomaly detection framework can be built on. This is still posed as a classification problem and may still struggle with novel inputs. Another autoencoder based novelty assessment approach is posed in [11]. The authors believe that the after the autoencoder recreates the input, the error between the recreated vector and the input vector will be small. In comparison, when novel inputs are given to the autoencoder, it will recreate the input as best as it can to be near the manifold of acceptable inputs. The error between the novel input vector and the reconstructed output vector will be much larger. Analysis of the error vector and its size can then be used to determine if an input was novel.

Another use of autoencoders proposed by Smaragdis and Venkataramani in [12] isn't focused on novelty detection but can likely be applied. As an alternative to non-negative matrix factorization, autoencoders are used to decompose an audio spectrogram to obtain a set of basis signals as well as their activation times. This has potential to be used in anomaly detection in applications involving spectrograms. In systems monitoring radio frequency (RF) transmissions, anomalous signals (whether interference or something else) may be identified by computing the error between the projection of the RF transmission onto the learned signal space.

V. Conclusion

The two unique applications this paper discussed also identified some of the limitations in their respective solutions. Though each proposed solution adequately solved the problem, it did so over a limited scope. It is generally accepted that there is no general anomaly detection approach applicable in all areas. However, unsupervised approaches are possibly better suited than other supervised approaches to a wider range of applications because it decreases the dependence on human skill and expertise. Future research will focus on developing unsupervised approaches by exploring various types of autoencoders and exploiting their unique properties.

References

[1] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[2] M. F. Augusteijn and B. A. Folkert, "Neural network classification and novelty detection," *International Journal of Remote Sensing*, vol. 23, no. 14, pp. 2891–2902, 2002. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01431160110055804

[3] B. Yadav and V. S. Devi, "Novelty detection applied to the classification problem using probabilistic neural network," in *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, Dec 2014, pp. 265–272.

[4] M. Markou and S. Singh, "Novelty detection: A review—part 2: Neural network based approaches," *Signal Process.*, vol. 83, no. 12, pp. 2499–2521, Dec. 2003. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2003.07.019

[5] M. A. F. Pimentel, D. A. Clifton, L. A. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[7] "Gome-2," http://www.ospo.noaa.gov/Products/atmosphere/gome/gome-A.html.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[10] K. Chen, J. Hu, and J. He, "Detection and classification of transmission line faults based on unsupervised feature learning and convolutional sparse autoencoder," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2016.

[11] B. B. Thompson, R. J. Marks, J. J. Choi, M. A. El-Sharkawi, M.-Y. Huang, and C. Bunje, "Implicit learning in autoencoder novelty assessment," in *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, vol. 3, 2002, pp. 2878–2883.

[12] P. Smaragdis and S. Venkataramani, "A neural network alternative to non-negative audio models," *CoRR*, vol. abs/1609.03296, 2016. [Online]. Available: http://arxiv.org/abs/1609.03296