RISAN: Robust Instance Specific Deep Abstention Network

Bhavya Kalra[1] **Kulin Shah**[2] **Naresh Manwani**[1]

[1]Machine Learning Lab, International Institute of Technology, Hyderabad, India
[2]Microsoft Research, Bangalore, India,

Abstract

In this paper, we propose deep architectures for learning instance specific abstain (reject option) binary classifiers. The proposed approach uses double sigmoid loss function as described by Kulin Shah and Naresh Manwani in ("Online Active Learning of Reject Option Classifiers", AAAI, 2020), as a performance measure. We show that the double sigmoid loss is classification calibrated. We also show that the excess risk of 0-d-1 loss is upper bounded by the excess risk of double sigmoid loss. We derive the generalization error bounds for the proposed architecture for reject option classifiers. To show the effectiveness of the proposed approach, we experiment with several real world datasets. We observe that the proposed approach not only performs comparable to the state-of-the-art approaches, it is also robust against label noise. We also provide visualizations to observe the important features learned by the network corresponding to the abstaining decision.

1 INTRODUCTION

In classification problems, learning becomes difficult when the cost of misclassification is extremely high. It becomes more challenging when learning critical tasks such as stock markets, medical diagnosis, autonomous driving, biotech, cyber-security, identification technologies, and robot-assisted surgery. In such situations, it becomes advantageous to refrain from taking any decision when in a dilemma. Such classifiers are called abstain (reject) option classifiers. Abstain classifiers have been successfully used in medical diagnosis [da Rocha Neto et al., 2011], financial forecasting [Rosowsky and Smith, 2013], genomics [Hanczar and Dougherty, 2008], speech emotion recognition [Sridhar and Busso, 2019], crowdsourcing [Li et al., 2017] etc.

Let $\mathcal{X} \subset \mathbb{R}^D$ be the feature space and $\{+1, -1\}$ be the label space. An abstaining classifier can be defined using a function $f : \mathcal{X} \to \mathbb{R}$ and a rejection function $\rho : \mathcal{X} \to \mathbb{R}_+$ as follows.

$$g(f(\mathbf{x}), \rho(\mathbf{x})) = \begin{cases} 1, & \mathbb{I}[f(\mathbf{x}) > \rho(\mathbf{x})] \\ \text{reject}, & \mathbb{I}[|f(\mathbf{x})| \leq \rho(\mathbf{x})] \\ -1, & \mathbb{I}[f(\mathbf{x}) < -\rho(\mathbf{x})] \end{cases}$$

The goal here is to simultaneously learn the function $f(\cdot)$ and $\rho(.)$. The performance of a given abstain classifier is measured using loss L_d (0-d-1) as follows.

$$L_d(yf(\mathbf{x}), \rho(\mathbf{x})) = \mathbb{I}[yf(\mathbf{x}) < -\rho(\mathbf{x})] + d\,\mathbb{I}[|yf(\mathbf{x})| \leq \rho(\mathbf{x})] \quad (1)$$

where $d \in (0, 0.5)$ is the cost of rejection. Loss L_d is minimized by *generalized Bayes classifier* [Chow, 1970] described as follows.

$$f_d^*(\mathbf{x}) = \begin{cases} 1, & \eta(\mathbf{x}) > 1 - d \\ \text{reject}, & d \leq \eta(\mathbf{x}) \leq 1 - d \\ -1, & \eta(\mathbf{x}) < d \end{cases} \quad (2)$$

where $\eta(\mathbf{x}) = P(y = 1|\mathbf{x})$. Loss L_d is discontinuous. Thus, minimizing risk under L_d is difficult. In practice, various surrogate losses of L_d have been used for learning abstain classifiers.

Kernel Based Approaches: Different algorithms for learning abstaining classifiers are proposed based on different choices of surrogates of L_d. Generalized hinge [Bartlett and Wegkamp, 2008] and double hinge [Grandvalet et al., 2009] are convex surrogates of L_d. Risk minimization using these losses results in support vector machine (SVM) like algorithms. However, approaches proposed in [Bartlett and Wegkamp, 2008, Grandvalet et al., 2009] learn the rejection bandwidth as a post-processing step resulting in suboptimal solutions. Manwani et al. [2015], Shah and Manwani

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:1525–1534.

[2019] propose approaches based on nonconvex surrogate of L_d called double ramp loss. Cortes et al. [2016] propose max-hinge loss and plus-hinge loss for rejection option and propose a kernel-based approach that minimizes these losses. Online active learning of abstaining classifiers is discussed in [Shah and Manwani, 2020]. These approaches face three major challenges. (a) These approaches rely on kernel trick to learn nonlinear classifiers. Thus, the scalability of these methods with big data is an issue. (b) Function $\rho(.)$ is assumed to be a constant for all instances (i.e., $\rho(\mathbf{x}) = \rho,\ \forall \mathbf{x} \in \mathcal{X}$). Thus, these approaches do not learn instance-specific rejection functions. (c) Most of these approaches are not robust against the label noise. Though the approach proposed in Shah and Manwani [2019] is shown robust against label noise, it uses kernels to learn nonlinear classifiers and cannot produce instance-specific rejection bandwidth.

Deep Learning-based Approaches for Abstain Classifiers: A neural networks based classifiers with abstain option is proposed De Stefano et al. [2000]. In this model, rejections are done after the learning of the classifier. This results in a suboptimal abstain option classifier. A similar approach for deep neural networks(DNNs) is proposed in Geifman and El-Yaniv [2017], which finds the best abstaining threshold based on the softmax output corresponding to each class from already trained networks. The method proposed in El-Yaniv et al. [2010] optimizes a pair of functions, a classification function, and a selective function with a risk-coverage trade-off, where coverage is defined as the ratio of samples selected for classification amongst the complete dataset. Deep learning implementation of the same is proposed in Selectivenet [Geifman and El-Yaniv, 2019]. This approach learns the appropriate selection and classification function for a given coverage in a deep learning setting. However, this approach does not take rejection cost d into account in their objective function. The main issue with such an approach is that it does not allow the data to decide the rejection rate. For example, in instances where the classes are separable with sufficient margin, this approach rejects and learns the classifier using the remaining examples based on specified coverage parameters. Thulasidasan et al. [2019] consider abstaining option as another class. However, this changes the abstain option's interpretation as the purpose of abstaining option is to capture the overlapping regions of any two classes.

Proposed Approach: In this paper, we propose an instance-specific deep learning approach with abstain option. The proposed approach takes the cost of rejection also as an input. It simultaneously learns the decision surface ($f(\mathbf{x})$) and rejection function ($\rho(\mathbf{x})$) which depends on the cost of rejection d. We use double sigmoid loss function to compare the output of the network with the ground truth. Note that the double sigmoid loss is a smooth nonconvex surrogate of L_d (see Eq. (1)).

Key Contributions: Our key contributions in this paper are as follows.

1. We show that the double sigmoid loss function is classification calibrated. We provide the excess risk bounds of the double sigmoid loss.
2. We propose a novel instance-specific deep abstain network called RISAN. RISAN has two variants, with and without instance-specific rejection function.
3. We derive the generalization error bounds for the proposed approach RISAN.
4. We show the proposed approach's effectiveness by comparing it with various state-of-the-art algorithms on various benchmark datasets. We also show by experiments that RISAN is robust against label noise in the data.
5. We also show visualizations that focus on the areas in an image leading the network to choose to abstain option. These visualizations reflect that our network learns useful representations for the rejection as well as classification.

Paper Organization: The rest of the paper is organized as follows. We discuss the double sigmoid loss and its properties in Section 2. In Section 3, we discuss the proposed approach RISAN, its different variants, and generalization bounds. We show the experimental results in Section 4. Robustness results of RISAN are given in Section 5. We discuss the visualizations of the representations learned by RISAN in Section 6. We conclude the paper with some remarks and future directions in Section 7.

2 DOUBLE SIGMOID LOSS FOR ABSTENTION

As discussed earlier, $\mathcal{X} \subseteq \mathbb{R}^D$ is the feature space and $\mathcal{Y} \in \{\pm 1\}$ is the label space. Let $\mathcal{P}(\mathbf{x}, y)$ be the unknown joint distribution on $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ be the finite training set where each $(\mathbf{x}_i, y_i)$ is generated i.i.d. from the distribution $\mathcal{P}(\mathbf{x}, y)$. The goal here is to learn functions $f : \mathcal{X} \to \mathbb{R}$ and $\rho : \mathcal{X} \to \mathbb{R}_+$ using the training set $\mathcal{S}$.

Here, functions $f(.)$ and $\rho(.)$ are represented using deep neural network (to be discussed shortly). To evaluate the performance of the learnt functions $f(.)$ and $\rho(.)$, we use double sigmoid loss function Shah and Manwani [2020] as follows.

$$\begin{aligned} L_{ds}(yf(\mathbf{x}), \rho(\mathbf{x})) = 2d\sigma(yf(\mathbf{x}) - \rho(\mathbf{x})) \\ + 2(1-d)\sigma(yf(\mathbf{x}) + \rho(\mathbf{x})) \end{aligned} \quad (3)$$

where d is the cost of rejection and $\sigma(a) = (1+\exp(\gamma a))^{-1}$ is the sigmoid function with $(\gamma > 0)$. The risk under double sigmoid loss function is as follows.

$$R_{ds}(f, \rho) = \mathbb{E}_{\mathcal{X},\mathcal{Y}}\left[L_{ds}(yf(\mathbf{x}), \rho(\mathbf{x}))\right]$$

Here, we establish theoretical properties of the double sigmoid loss function.

Classification Calibration Double sigmoid loss is a linear combination of two sigmoid functions and hence is a non convex loss function. We first show classification calibration on double sigmoid loss by ensuring that the risk under L_{ds} is minimized by the generalized bayes classifier. To approximate the optimal classifier, classification calibration is the minimal requirement for any loss function.

Theorem 1. *For a fixed cost of rejection d, the risk under double sigmoid loss is minimized by the generalized Bayes classifier $f_d^*(.)$ (see Eq.(2)).*

Excess Risk Bound We now relate the excess risk of L_d, $(R_d(f, \rho) - R_d(f_d^*))$ with the excess risk of the double sigmoid loss $(R_{ds}(f, \rho) - R_{ds}(f_d^*))$. Note that here $R_d(f, \rho) = \mathbb{E}_{\mathcal{X},\mathcal{Y}}[L_d(yf(\mathbf{x}), \rho(\mathbf{x}))]$ and f_d^* (see Eq.(2)) is the generalized Bayes classifier which minimizes $R_d(f, \rho)$. $R_d(f_d^*)$ and $R_{ds}(f_d^*)$ represents risk of generalized Bayes classifies f_d^* under L_d and L_{ds} loss. We know that $L_d(yf(\mathbf{x}), \rho(\mathbf{x})) \leq L_{ds}(yf(\mathbf{x}), \rho(\mathbf{x}))$. Thus, taking expectations on both sides, we get, $R_d(f, \rho) \leq R_{ds}(f, \rho)$. We follow the approach of Bartlett et al. [2006] to establish an excess risk bound for the double sigmoid loss function L_{ds}.

Theorem 2. *Let $0 \leq d \leq 1/2$ and a measurable function z. Then we have the excess risk relation as*

$$\psi\left(R_d(f, \rho) - R_d(f_d^*)\right) \leq R_{ds}(f, \rho) - R_{ds}(f_d^*)$$

where

$$\psi(\theta) = \begin{cases} 0 & \theta = 0 \\ (2d-1)\zeta + \left(\frac{\theta+1-2d}{2}\right)\left(\frac{T+\zeta^2\theta}{\zeta\theta+T\zeta}\right) & \\ \qquad + \left(\frac{\theta+2d-1}{2}\right)\left(\frac{T-\zeta^2\theta}{\zeta\theta-T\zeta}\right) & \theta \in (0, 1-2d] \\ \theta + (2d-1)\zeta & \theta \in [1-2d, 1] \end{cases}$$

and $\theta = R_d(f, \rho) - R_d(f_d^*)$ and . Also, $\zeta = tanh(\frac{\rho}{2})$ and $T = (1-2d) - \sqrt{(1-2d)^2 - \theta^2}$.

The proof of the theorem is provided in Appendix A.

Since we have established statistical properties of double sigmoid loss, we can use this loss in deep networks to train classifiers with abstention option.

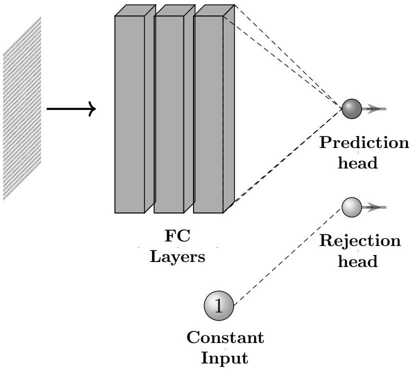


Figure 1: RISAN architecture with input independent ρ

3 PROPOSED APPROACH: RISAN

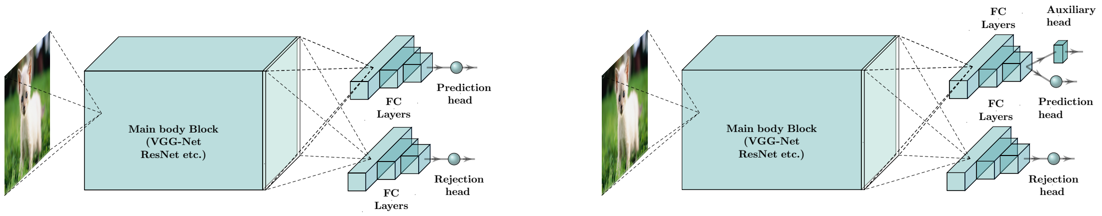
The proposed architecture models both decision surface $f(.)$ and rejection function $\rho(.)$ in a single DNN model. Schematic view of RISAN implementations is depicted in Figure-1 and Figure-2. The network's input is processed by the main body block and an associated (separate or same) network that would learn the rejection region parameter. The main body block consists of hidden layers or sub-blocks. The rejection function $\rho(.)$ can be modeled by a separate single neuron or a network similar to the main body block. The main body block can be assembled using any type of architecture relevant to the problem at hand (e.g., convolutional, fully connected, or recurrent architectures).

3.1 RISAN: INPUT INDEPENDENT REJECTION

RISAN architecture represented in Figure-1 describes the architecture when the rejection function takes the same value for all $\mathbf{x}$, that is, $\rho(\mathbf{x}) = \rho, \ \forall \mathbf{x} \in \mathcal{X}$. RISAN for input independent rejection has two output heads, prediction head $(f(\mathbf{x}))$ and rejection region parameter(ρ). The input data $\mathbf{x}$ is fed into the fully connected (FC) layers while a fixed constant is fed into the rejection head. The role of the prediction head is to learn the appropriate decision surface $f(\mathbf{x})$, and the rejection head learns the rejection region parameter (denoted as ρ). In this case, the main body block is a stack of fully connected layers that are used for processing the input data.

3.2 RISAN: INPUT-DEPENDENT REJECTION

RISAN architecture in Figure-2 describes the architectures when rejection function depends on the specific instance. The primary architecture is provided in Figure 2a for input dependent rejection. This architecture has two output heads similar to the input independent architecture. However, the rejection head is fed the input from the main body block.



(a) RISAN without an auxiliary head (RISAN-NA)

(b) RISAN with an auxiliary head (RISAN)

Figure 2: Different implementations of RISAN with input dependent $\rho(.)$

An additional architecture for incorporating auxiliary loss has been provided in Figure 2b. This architecture has three output heads, prediction head ($f(\mathbf{x})$), rejection head ($\rho(\mathbf{x})$) and an auxiliary head. The auxiliary head, only used for training the networks, sometimes plays an important role in the initial process of acquiring complex features from convolutional blocks. We follow the notion of the auxiliary head for very deep neural networks as mentioned in Geifman and El-Yaniv [2019]. The auxiliary head's role is to learn a related prediction task that facilitates the consolidation of apropos features in the main body block. Thus, the prediction and rejection head are optimized with the auxiliary head helping build features that minimize L_c, the convex combination of categorical cross entropy loss L_{ce} and double sigmoid loss L_{ds}.

$$L_c = \alpha \times L_{ds} + (1-\alpha) \times L_{ce}$$

The number and size of fully connected layers preceding these two or three heads (depending on the architecture) are independent and can vary depending on the task type and complexity. The final neuron, however, for both the prediction head and rejection head are single neurons. The final layer of auxiliary head $h(\mathbf{x})$ depends on the application and could be a softmax layer. The relevance of the different architectures has been explored in the experiments section.

3.3 GENERALIZATION ERROR BOUNDS OF RISAN WITH INPUT INDEPENDENT REJECTION

We followed the approach of Neyshabur et al. [2015] to establish an upper bound on the Rademacher complexity of regularized DNN with double sigmoid loss function and an input independent ρ as shown in figure 1. We show in Theorem 3 that the Rademacher complexity for rectified linear unit based neural networks and consider two intuitive types of norm regularization (i) bounding the norm of the incoming weights of each unit (per-unit regularization) and (ii) bounding the overall norm of all the weights in the system jointly (overall regularization) Let ℓ_p, be the norm over all incoming weights to each unit and ℓ_q, the norm over all the units collectively. Now, considering the above definitions. Our neural network can be defined as a graph with group norm regularization as:

$$\xi_{p,q}(\mathbf{w}) = \left(\sum_{v\in V}\left(\sum_{(u\to v)\in E}|\mathbf{w}(u\to v)|^p\right)^{q/p}\right)^{1/q}$$

where u and v are nodes in adjacent layers belonging to set of vertices, V. And $\mathbf{w}(u \to v)$ represents the weight associated with the edge $u \to v$ belonging to set of edges, E.

Let us consider a deep abstain network with $n+1$ layers including input and output layers. Let us assume that all the hidden layer have the same number of nodes (H). Let W_j denotes the weight matrix corresponding to the connections from $(j-1)$ layer to j^{th} layer. Then, $W_1 \in \mathbb{R}^{H\times D}$, W_2,. . . ,$W_{n-1} \in \mathbb{R}^{H\times H}$, $W_n \in \mathbb{R}^{1\times H}$ and $\rho(\mathbf{x}) = \rho$. The output of the network can be defined written as,

$$f_W(x) = W_n\sigma\left(W_{n-1}\sigma\left(W_{n-2}\left(\ldots\sigma\left(W_1\mathbf{x}\right)\right)\right)\right) - \rho$$

where σ is the activation function and $W = (W_1,\ldots,W_n,\rho)$. Let $\mathcal{F}$ denotes the set all possible functions represented by such a neural network.

Theorem 3. *Let $\mathcal{D}$ be any distribution on $\mathcal{X} \times \{-1,+1\}$. Let $0 < \delta \le 1$. Then for any n, $q \ge 1$, $1 \le p < \infty$ and any set $S = \{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$; with probability at least $1-\delta$ (over $S \sim \mathcal{D}^m$), all functions $f \in \mathcal{F}$ satisfy*

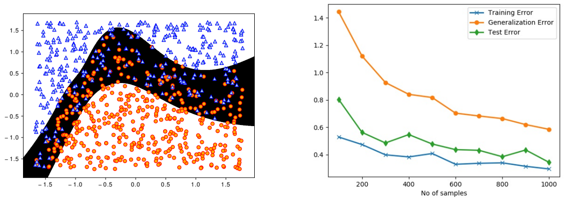
$$R_{ds}(f,\rho) \le \hat{R}_{ds}(f,\rho,+)\frac{\bar{\rho}}{\sqrt{m}} + \sqrt{\frac{8\ln\left(\frac{4}{\delta}\right)}{m}} + \sqrt{\frac{2\ln\left(\frac{2}{\delta}\right)}{m}} + \left(\frac{2\beta}{\sqrt{m}}\max_i \|\mathbf{x}_i\|_{p'}\right)\left(2H^{\left[\frac{1}{p'}-\frac{1}{q}\right]_+}\right)^{n-1}$$

where n is the number of layers in the network, H is the number of neurons in the hidden layers, rejection region parameter is bounded as $\rho \le \bar{\rho}$. Also $\frac{1}{p'}+\frac{1}{p} = 1$ and $[a]_+ =$ $\max(0,a)$. $\mathrm{er}_S^{\ell}[f]$ *is the empirical error and* $\beta_{p,q}(W) = \prod_{k=1}^{n}\|W_k\|_{p,q} \le \beta$.

The proof of the theorem is provided in appendix A. The key observations from the bound in Theorem 3 are as follows. The bounds depend on the number of neurons in each layer, H and the number of layers n. The bounds are also inversely proportional to the number of samples, $\sqrt{m}$. Thus, increasing m decreases the generalization error bound. Also, when $p' \geq q$, the dependence on the number of neurons in each layer vanishes. If we use overall ℓ_1 or ℓ_2 regularization, this dependence should disappear.

3.4 EXAMPLE: CLASSIFIER LEARNT USING RISAN

We generated 1000 examples in the square $[-1.5, 1.5]^2$ uniformly randomly. We used $x_2 - x_1 - 2sin(x_1) = 0$ as separation boundary. We ensured equal representation of each class. We then randomly flipped labels of the samples present within the ±0.75 margin of the decision boundary. We also used RISAN with input independent ρ (see Figure 1) and $d = 0.25$. The resulting classification boundary and rejection region of the synthetic dataset are shown in Figure 3a where the dark region signifies the rejection region. We also plot the generalization bounds for the input independent rejection on a 2D dataset (see Figure 3b). For $d = 0.25$, we ran the experiments for 30 epochs, increasing the no. of samples from 100 to 1000 with a step size of 100. We observed that an increase in the number of samples leads to decreased training error, test error, and generalization error simultaneously. Also, we observed that the generalization error upper bounds the test error for each experiment.



(a)

(b)

Figure 3: (a) Resulting classifier on a synthetically generated dataset with a nonlinear rejection region (black) (b) Generalization error upper bounds the Test error on the synthetic dataset

4 EXPERIMENTS

This section describes the experimental details: datasets used, baseline algorithms used for comparison purposes, and our choice of architectures and hyper-parameters.

4.1 DATASETS USED

Note that our proposed approach works for binary classification problems. Thus, to show the effectiveness of the proposed approach, we performed experiments on the following datasets.

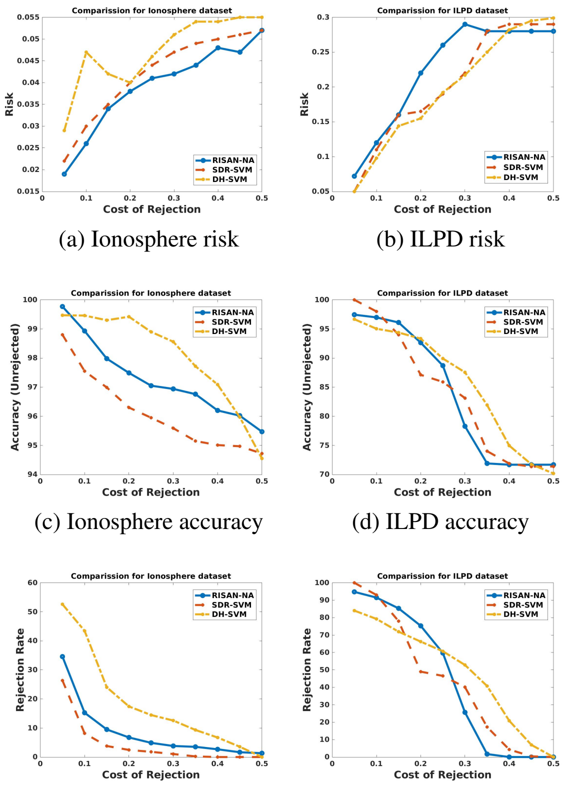
1. Small Datasets: Ionosphere and ILPD [Dua and Graff, 2017].
2. Phishing dataset [Dua and Graff, 2017].
3. Cats vs. Dogs [Elson et al., 2007]: Each image rescaled to 64x64 from original images of size 360x400.
4. CIFAR-10 [Krizhevsky et al., 2009]: We selected classes *automotive* and *truck* from CIFAR-10 for our task. We have selected these classes as they have many similarities, contain overlapping features, and are tough to classify even for humans sometimes.
5. MNIST [LeCun et al., 2010]: We selected classes *1* and *7* from MNIST dataset for our task.
6. CBIS-DDSM [Lee et al., 2017]: This is a medical image dataset with positive referring to the presence of some form of calcification or mass, and the absence refers to negative examples. The dataset has 14% positives, and 86% negative labeled pre-processed images with ROI extracted. We further sampled the images to create a subset dataset with a similar number (4500) of positives and negatives examples each, all re-scaled to 64x64 from the original size of 299x299.

We divide our experiments into two categories, namely, small dataset and large dataset experiments because some baseline methods are optimized for the smaller datasets and fail to converge for larger datasets. Hence, we use different baseline methods for small and large datasets.

4.2 BASELINES

Baselines for Small Datasets Experiments: We compare our network with two state of the art methods, (a) DH-SVM: reject option classifier introduced in Grandvalet et al. [2009] which minimizes the double hinge loss and (b) SDR-SVM: sparse reject option classifier proposed in Shah and Manwani [2019] which minimizes ℓ_1 regularized risk under double ramp loss function.

Baselines for Large Datasets Experiments: We compare the proposed approach with the following baselines for Cats vs. Dogs, CIFAR-10, CBIS-DDSM, MNIST, and Phishing website datasets. (a) SelectiveNet(SNN) [Geifman and El-Yaniv, 2019]: a deep neural architecture with an integrated reject option that simultaneously optimizes a prediction and a selection function . We also compare results on a variant of SNN without the auxiliary loss, the SNN-NA. (b) DAC: deep abstaining classifier, a deep neural network trained with a modified cross entropy loss function introduced in Thulasidasan et al. [2019] to accommodate an abstain (reject) class.



(a) Ionosphere risk

(b) ILPD risk

(c) Ionosphere accuracy

(d) ILPD accuracy

(e) Ionosphere rejection rate

(f) ILPD rejection rate

Figure 4: Small Dataset Results

4.3 EXPERIMENTAL SETTINGS

We execute experiments on ILPD and Ionosphere datasets in a 10-fold cross-validation fashion for 10 repetitions. We do these for the cost of rejection (d) varying from $[0.05, 0.5]$ with a step size of 0.05. We monitor the accuracy (on unrejected samples), rejection rate, and the cross-validation risk $(0 - d - 1)$ for each value of d. The experiments on large datasets compare five algorithms where each one takes a different parameter to introduce rejection. While DAC takes an abstention rate as input parameter, Selective Net (SNN) and SNN-NA take as input a coverage parameter. Here, coverage denotes fraction of points without abstention as the output label by the final trained classifier. For our networks RISAN and RISAN-NA, we have a cost of rejection which depends on the dataset. To get wide range of rejection rate, we choose cost of rejection d parameter for our RISAN and RISAN-NA methods from set $\{0.0001, 0.005, 0.001, 0.05, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.5\}$. Both the abstention rate parameter for DAC and coverage parameter from SNN and SNN-NA are varied from [0.1,1.0] with a step size of 0.1. We plot the rejection rate vs accuracy plots to compare the five methods. The details of architectures and hyperparameters used in the experiments is given in Appendix A.4.

4.4 REPRODUCIBILITY

The code for the implementation would be available at `https://github.com/kalra20/RISAN-Robust-Instance-Specific-Abstain-Network`

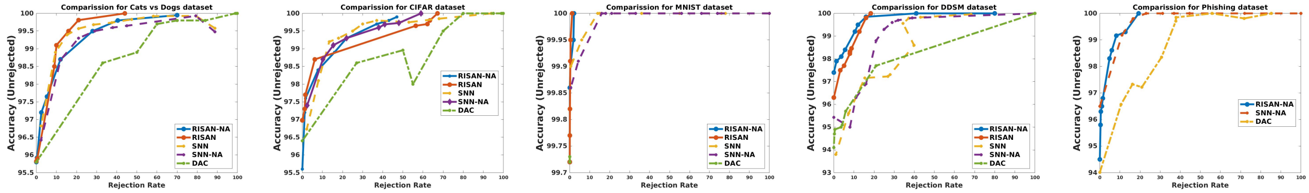
4.5 EMPIRICAL OBSERVATIONS

In Figure 4, we give results on smaller tabular datasets. We observed that proposed method achieves lower risk on the Ionosphere dataset (Figure 4a) and performs comparably on the ILPD dataset except at a couple of points (Figure 4b). Note that baseline methods on smaller datasets are optimized for small-sized datasets and fail to converge for large dataset. We perform better or comparable to such baseline methods. The proposed algorithm RISAN and RISAN-NA don't suffer from failing-to-converge issue on large datasets and perform comfortably to other neural network based algorithms (Figure 5). We also make some interesting observations from results on the larger datasets. Both RISAN and RISAN-NA perform comparably on Cats vs. Dogs and CIFAR dataset with other datasets. However, RISAN performs slightly better than SNN, while RISAN-NA performs better than SNN-NA. This trend is consistent across all the datasets. We do acknowledge a consistent improvement of (1-2%) in accuracy with the addition of an auxiliary loss. We also observed that DAC fails to reject any examples for the MNIST dataset where the accuracy is too high (99.7%) for VGG architecture despite complete coverage. However, the RISAN and RISAN-NA perform better than SNN and SNN-NA while all four maintain a non-zero rejection rate. The fact that RISAN and RISAN-NA opting not to reject more samples even for an extremely small value of d verifies that a cost-based abstain classifier is a more natural choice to learn the classifier than a coverage-based classifier. Since it chooses not to reject samples when the data is well separated, i.e., high accuracy without any rejection. This observation prompted the inspection of results on datasets with label noise.

5 ROBUSTNESS OF RISAN AGAINST LABEL NOISE

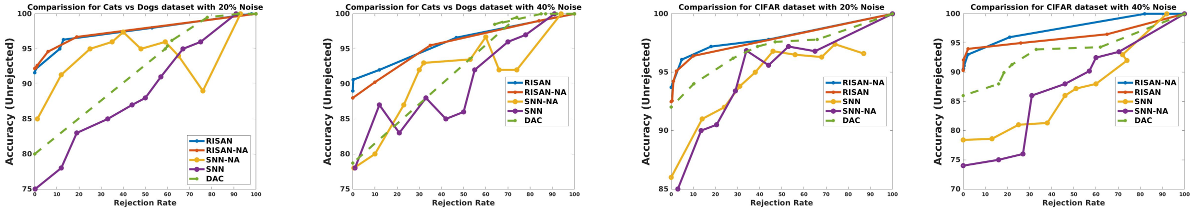
In this section, we show the robustness results of RISAN against uniform label noise.

Experimental Setup: We use Cats vs. Dogs and CIFAR 10 datasets for showing the robustness of RISAN against label noise. We introduce uniform label noise with a noise



(a) Cats vs Dogs Dataset (b) CIFAR Dataset (c) MNIST Dataset (d) CBIS-DDSM Dataset (e) Phishing Dataset

Figure 5: Large Dataset Results



(a) Cats vs Dogs with 20% label noise (b) Cats vs Dogs with 40% label noise (c) CIFAR with 20% label noise (d) CIFAR with 40% label noise

Figure 6: Comparison Results With Label Noise

rate of 20% and 40%. We ran the experiments with identical coverage values for SNN, SNN-NA, and DAC used in large dataset experiments. We used values of d from set $\{0.05, 0.1, 0.15, \dots, 0.4, 0.45, 0.5\}$.

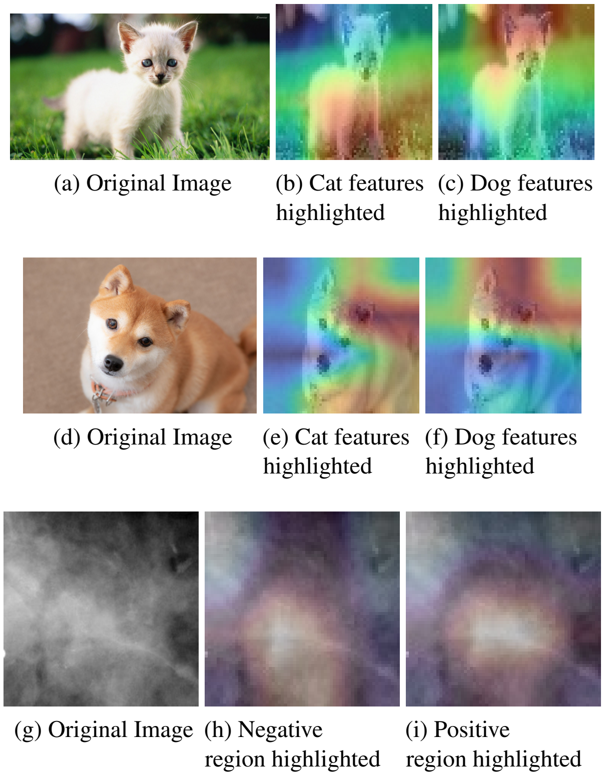
Results: Results with label noise are shown in Figure 6. We observe that with 20% and 40% label noise rates, RISAN and RISAN-NA performances do not drop much. On the other hand, the other approaches' performances drop significantly with label noise on both datasets. For 20% label noise and low rejection rate, RISAN and RISAN-NA achieve at least $6 - 7\%$ higher accuracy than other methods on both datasets. For 40% label noise and low rejection rate, RISAN and RISAN-NA achieve around 10% higher accuracy on the Cats vs. Dogs dataset and around 5% higher accuracy on the CIFAR-10 dataset. As Thulasidasan et al. [2019] claim that their approach (DAC) is robust to noisy labels, proposed algorithm RISAN improves around 5-10% accuracy on unrejected samples from previously proposed robust learning algorithms. For large rejection rates, models are expected to get good accuracy on unrejected samples because the model is allowed to abstain large fraction of the data.

6 EXPLAINING THE REJECTION DECISIONS

In this section we introduced a visualization technique into the abstain network as a post processing step and examined the rejected examples.

6.1 REPRESENTATIONS LEARNT BY RISAN

In this section, we explored the following hypotheses about the trained abstain network: (i) Our network would reject images that contain pertinent features amongst both the classes (ii) Prediction network will learn features that are more prominent and easily distinguishable for each class (iii) The prediction network will give lesser precedence to features that are common to both classes. The implementation details of GradCAM in RISAN have been shifted to the Appendix A.5. The GradCAM Selvaraju et al. [2017] technique was used on the sigmoid outputs of the auxiliary head associated with the prediction network. Thus visualizing features learned by the prediction network to produce highlighted regions corresponding to the image's different classes. We executed GradCAM on some selected examples that were ambiguous and tough to classify. Our network, as expected, choose to reject these samples. The images used in this task were re-scaled to 64x64 for the network to process the image. In Fig. 7a, we examined a cat image that could be mistaken for a dog. We observed that the cat's body, especially the legs, were majorly highlighted with reference to the *cat* class in Fig. 7b. It's contrasted by the head region of the subject being highlighted in Fig. 7c with respect to *dog* class. The legs and body region are important features for *cat* class, as will be established in our later conducted experiments. In comparison, the head region of a *dog* is equally important. We examined another example, a dog in Fig. 7d that can be mistaken for a cat. We observed that subject's ear and the body is being majorly highlighted with reference to the *cat* class in Fig. 7e. It's contrasted by the


(a) Original Image (b) Cat features highlighted (c) Dog features highlighted

(d) Original Image (e) Cat features highlighted (f) Dog features highlighted

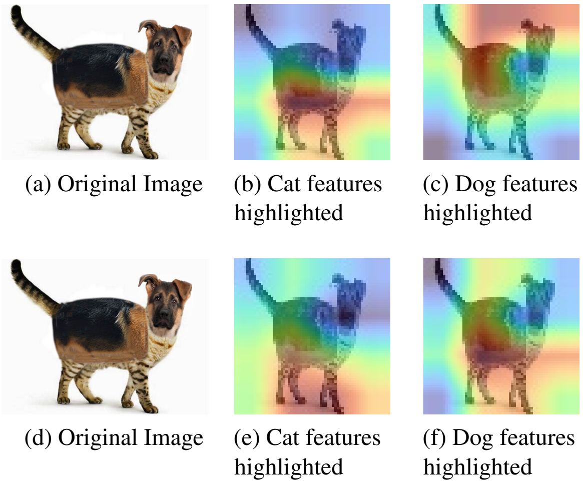
(g) Original Image (h) Negative region highlighted (i) Positive region highlighted

Figure 7: GradCAM on image of a cat rejected by RISAN highlights what the network perceives as cat and dog regions in the image(a,b,c). GradCAM on image of a dog rejected by RISAN highlights what the network perceives as cat and dog regions in the image(d,e,f). GradCAM on image of a cancerous development rejected by RISAN highlights what the network perceives as negative and positive regions in the image(g,h,i)

head region of the subject being highlighted in Fig. 7f with respect to *dog* class.

In another example, we considered mammography of a malignant mass that's tough to spot and classify in Fig. 7g. The network was trained to classify the presence of any irregularities(calcification or mass) in the image as *positive*. We observed that in Fig. 7i, the mass (irregular lighter region running through the image diagonally) is being highlighted with respect to the *positive* class. It's contrasted by the larger region highlighted in Fig. 7h, containing more surrounding *negative* region, with respect to *negative* class. Though there appears to be an overlap of highlighted regions, *negative* class region focuses more on the surroundings of the mass while *positive* class focuses more on the mass itself. But since features from both classes are present in the image, it's a good candidate for rejection. Hence, we observed our prediction network highlighted pertinent features corresponding to each class found in the images and chose to reject these examples.

To verify our second and third assumptions, we then chose an interesting example of an animal, where a dog's body and head with a cat's legs and tail were infused. To compare and analyze our network's learned features, we also trained a separate network (CCEN) with categorical cross-entropy loss. We executed GradCAM on both the networks to compare the resulting highlighted regions in the images. We observed that while our network rejected the image, CCEN predicted the *dog* class. When we examined the highlighted regions corresponding to different classes, we saw that in Fig. 8b and Fig. 8e both networks chose to highlight the cat's legs fairly well in reference to *cat* class. This is also coherent with our previous analysis of features highlighted for the *cat* class. However, when we analyzed Fig. 8c, our network gave attention to the dog's body and head and less attention to the animal's legs. Whereas, as seen in Fig. 8f, CCEN pays attention to the animal's body and legs for *dog* class. This holds with our belief that when rejecting, features corresponding to non-similar regions would get more attention and help make decisions only when the classifier is extremely certain.


(a) Original Image (b) Cat features highlighted (c) Dog features highlighted

(d) Original Image (e) Cat features highlighted (f) Dog features highlighted

Figure 8: The GradCAM on image of a cat's lower body and tail infused with a dog's upper body and head showcase the contrasting features learnt by RISAN (a,b,c) and CCEN (d,e,f)

7 CONCLUSION AND FUTURE WORK

We introduced a novel implementation of double sigmoid loss in a deep neural network setting, RISAN for binary classification. We established the statistical properties of double sigmoid loss function such as classification calibration and excess risk bound. We also derived the generalization error bounds for input independent RISAN. We then demonstrated the various architectures and how each can be utilized for varied sized datasets. We also show that RISAN performs competitively to other state of the art shallow and deep neural network methods, SelectiveNet and Deep Abstaining Classifier in absence of noise but gains significant advantage when the data becomes noisy. We were also able to visualize the highlighted regions for corresponding classes in images and make inferences about rejected images. The results motivates the use of RISAN in applications where cost of misclassification is extremely high.

We leave a number of issues for future research such as extending the proposed method from binary classification to multiclass classification. Also, the study of representations learnt by other abstain neural networks and how they compare to RISAN is also an open future direction.

References

Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(Aug):1823–1840, 2008.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory (ALT)*, pages 67–82, 2016.

Ajalmar R. da Rocha Neto, Ricardo Sousa, Guilherme de A. Barreto, and Jaime S. Cardoso. Diagnostic of pathology on the vertebral column with embedded reject option. In *Pattern Recognition and Image Analysis*, pages 588–595, 2011.

Claudio De Stefano, Carlo Sansone, and Mario Vento. To reject or not to reject: that is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(1):84–94, 2000.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Ran El-Yaniv et al. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5), 2010.

Jeremy Elson, John (JD) Douceur, Jon Howell, and Jared Saul. Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*, October 2007.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in neural information processing systems*, pages 4878–4887, 2017.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. *arXiv preprint arXiv:1901.09192*, 2019.

Yves Grandvalet, Alain Rakotomamonjy, Joseph Keshet, and Stéphane Canu. Support vector machines with a reject option. In *Advances in neural information processing systems*, pages 537–544, 2009.

Blaise Hanczar and Edward R Dougherty. Classification with reject option in gene expression data. *Bioinformatics (Oxford, England)*, 24(17):1889—1895, September 2008.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy, and Daniel L Rubin. A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific data*, 4: 170177, 2017.

Q. Li, A. Vempaty, L. R. Varshney, and P. K. Varshney. Multi-object classification via crowdsourcing with a reject option. *IEEE Transactions on Signal Processing*, 65 (4):1068–1081, 2017.

Naresh Manwani, Kalpit Desai, Sanand Sasidharan, and Ramasubramanian Sundararajan. Double ramp loss based reject option classifier. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 151–163. Springer, 2015.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

Yasin I. Rosowsky and Robert E. Smith. Rejection based support vector machines for financial time series forecasting. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2013.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

Kulin Shah and Naresh Manwani. Sparse reject option classifier using successive linear programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4870–4877, 2019.

Kulin Shah and Naresh Manwani. Online active learning of reject option classifiers. In *AAAI*, pages 5652–5659, 2020.

Kusha Sridhar and Carlos Busso. Speech Emotion Recognition with a Reject Option. In *Proc. Interspeech 2019*, pages 3272–3276, 2019.

Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention. *arXiv preprint arXiv:1905.10964*, 2019.