Deep Conformal Prediction for Robust Models

Soundouss Messoudi, Sylvain Rousseau, Sébastien Destercke

Deep conformal prediction for robust models

Soundouss Messoudi[1], Sylvain Rousseau[1][0000−0002−2212−5950], and Sébastien Destercke[1][0000−0003−2026−468X]

HEUDIASYC - UMR CNRS 7253, Université de Technologie de Compiègne - 57 avenue de Landshut, 60203 COMPIEGNE CEDEX - FRANCE
firstname.lastname@hds.utc.fr
https://www.hds.utc.fr/

Abstract. Deep networks, like some other learning models, can associate high trust to unreliable predictions. Making these models robust and reliable is therefore essential, especially for critical decisions. This experimental paper shows that the conformal prediction approach brings a convincing solution to this challenge. Conformal prediction consists in predicting a set of classes covering the real class with a user-defined frequency. In the case of atypical examples, the conformal prediction will predict the empty set. Experiments show the good behavior of the conformal approach, especially when the data is noisy.

Keywords: Deep learning · Conformal prediction · Robust and reliable models.

1 Introduction

Machine learning and deep models are everywhere today. It has been shown, however, that these models can sometimes provide scores with a high confidence in a clearly erroneous prediction. Thus, a dog image can almost certainly be recognized as a panda, due to an adversarial noise invisible to the naked eye [4]. In addition, since deep networks have little explanation and interpretability by their very nature, it becomes all the more important to make their decisions robust and reliable.

There are two popular approaches that estimate the confidence to be placed in the predictions of machine learning algorithms : Bayesian learning and Probably Approximately Correct (PAC) learning. However, both these methods provide major limitations. Indeed, the first one needs correct prior distributions to produce accurate confidence values, which is often not the case in real-world applications. Experiments conducted by [10] show that when assumptions are incorrect, Bayesian frameworks give misleading and invalid confidence values (i.e. the probability of error is higher than what is expected by the confidence level). The second method, i.e. PAC learning, does not rely on a strong underlying prior but generates error bounds that are not helpful in practice, as demonstrated in [13]. Another approach that offers hedged predictions and does not have these drawbacks is conformal prediction [14].

Conformal prediction is a framework that can be implemented on any machine learning algorithm in order to add a useful confidence measure to its predictions. It provides predictions that can come in the form of a set of classes whose statistical reliability (the average percentage of the true class recovery by the predicted set) is guaranteed under the traditional identically and independently distributed (i.i.d.) assumption. This general assumption can be relaxed into a slightly weaker one that is exchangeability, meaning that the joint probability distribution of a sequence of examples does not change if the order of the examples in this sequence is altered. The principle of conformal prediction and its extensions will be recalled in Section 2.

Our work uses an extension of this principle proposed by [6]. They propose to use the density $p(x|y)$ instead of $p(y|x)$ to produce the prediction. This makes it possible to differentiate two cases of different uncertainties: the first predicts more than one label compatible with x in case of ambiguity and the second predicts the empty set $\emptyset$ when the model does not know or did not see a similar example during training. This approach is recalled in Section 2.3. However, the tests in [6] only concern images and Convolutional Neural Networks.

Therefore, the validity and interest of this approach still largely remains to be empirically confirmed. This is what we do in Section 3, where we show experimentally that this approach is very generic, in the sense that it works for different neural network architectures (Convolutional Neural Networks, Gated Recurrent Unit and Multi Layer Perceptron) and various types of data (image, textual, cross sectional).

2 Conformal prediction methods

Conformal prediction was initially introduced in [14] as a transductive online learning method that directly uses the previous examples to provide an individual prediction for each new example. An inductive variant of conformal prediction is described in [11] that starts by deriving a general rule from which the predictions are based. This section presents both approaches as well as the density-based approach, which we used in this paper.

2.1 Transductive conformal prediction

Let $z_1 = (x_1, y_1), z_2 = (x_2, y_2), \ldots, z_n = (x_n, y_n)$ be successive pairs constituting the examples, with $x_i \in X$ an object and $y_i \in Y$ its label. For any sequence $z_1, z_2, \ldots, z_n \in Z^*$ and any new object $x_{n+1} \in X$, we can define a *simple predictor* D such as :

$$D : Z^* \times X \to Y. \tag{1}$$

This simple predictor D produces a point prediction $D(z_1, \ldots, z_n, x_{n+1}) \in Y$, which is the prediction for y_{n+1}, the true label of x_{n+1}.

By adding another parameter $\epsilon \in (0, 1)$ which is the probability of error called the *significance level*, this simple predictor becomes a *confidence predictor* Γ that

can predict a subset of Y with a *confidence level* $1-\epsilon$, which corresponds to a statistical guarantee of coverage of the true label y_{n+1}. Γ is defined as follows:

$$\Gamma : Z^* \times X \times (0,1) \to 2^Y, \tag{2}$$

where 2^Y denotes the power set of Y. This confidence predictor Γ^ϵ must be decreasing for the inclusion with respect to ϵ, i.e. we must have:

$$\forall n > 0, \quad \forall \epsilon_1 \geq \epsilon_2, \quad \Gamma^{\epsilon_1}(z_1, \ldots, z_n, x_{n+1}) \subseteq \Gamma^{\epsilon_2}(z_1, \ldots, z_n, x_{n+1}). \tag{3}$$

The two main properties desired in confidence predictors are (a) *validity*, meaning the error rate does not exceed ϵ for each chosen confidence level ϵ, and (b) *efficiency*, i.e. prediction sets are as small as possible. Therefore, a prediction set with fewer labels will be much more informative and useful than a bigger prediction set.

To build such a predictor, conformal prediction relies on a *non-conformity measure* A_n. This measure calculates a score that estimates how strange an example z_i is from a bag of other examples $\wr z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n \int$. We then note α_i the non-conformity score of z_i compared to the other examples, such as:

$$\alpha_i := A_n(\wr z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n \int, z_i). \tag{4}$$

Comparing α_i with other non-conformity scores α_j with $j \neq i$, we calculate a *p-value* of z_i expressing the proportion of less conforming examples than z_i, with:

$$\frac{|\{j = 1, \ldots, n : \alpha_j \geq \alpha_i\}|}{n}. \tag{5}$$

If the p-value approaches the lower bound $1/n$ then z_i is non-compliant to most other examples (an outlier). If, on the contrary, it approaches the upper bound 1 then z_i is very consistent.

We can then compute the p-value for the new example x_{n+1} being classified as each possible label $y \in Y$ by using (5). More precisely, we can consider for each $y \in Y$ the sequence $(\wr z_1, \ldots, z_n, z_{n+1} = (x_{n+1}, y))$ and derive from that scores $\alpha_1^y, \ldots, \alpha_{n+1}^y$. We thus get a conformal predictor by predicting the set :

$$\Gamma^\epsilon(x_{n+1}) = \left\{ y \in Y : \frac{|\{i = 1, \ldots, n, n+1 : \alpha_i^y \geq \alpha_{n+1}^y\}|}{n+1} > \epsilon \right\}. \tag{6}$$

Constructing a conformal predictor therefore amounts to defining a non-conformity measure that can be built based on any machine learning algorithm called the *underlying algorithm* of the conformal prediction. Popular underlying algorithms for conformal prediction include Support Vector Machines (SVMs) and k-Nearest Neighbours (k-NN).

2.2 Inductive conformal prediction

One important drawback of Transductive Conformal Prediction (TCP) is the fact that it is not computationally efficient. When dealing with a large amount of data,

it is inadequate to use all previous examples to predict an outcome for each new example. Hence, this approach is not suitable for any time consuming training tasks such as deep learning models. Inductive Conformal prediction (ICP) is a method that was outlined in [11] to solve the computational inefficiency problem by replacing the transductive inference with an inductive one. The paper shows that ICP preserves the validity of conformal prediction. However, it has a slight loss in efficiency.

ICP requires the same assumption as TCP (the i.i.d. assumption or the weaker assumption exchangeability), and can also be applied on any underlying machine learning algorithm. The difference between ICP and TCP consists of splitting the original training data set $\wr z_1, \ldots, z_n \int$ into two parts in the inductive approach. The first part $D^{tr} = \wr z_1, \ldots, z_l \int$ is called the *proper training set*, and the second smaller one $D^{cal} = \wr z_{l+1}, \ldots, z_n \int$ is called the *calibration set*. In this case, the non-conformity measure A_l based on the chosen underlying algorithm is trained only on the proper training set. For each example of the calibration set $i = l+1, \ldots, n$, a non-conformity score α_i is calculated by applying (4) to get the sequence $\alpha_{l+1}, \ldots, \alpha_n$. For a new example x_{n+1}, a non-conformity score α^y_{n+1} is computed for each possible $y \in Y$, so that the p-values are obtained and compared to the significance level ϵ to get the predictions such as:

$$\Gamma^{\epsilon}(x_{n+1}) = \{y \in Y : \frac{|\{i = l+1, \ldots, n, n+1 : \alpha_i \geq \alpha^y_{n+1}\}|}{n-l+1} > \epsilon\}. \qquad (7)$$

In other words, this inductive conformal predictor will output the set of all possible labels for each new example of the classification problem without the need of recomputing the non-conformity scores in each time by including the previous examples, i.e., only α_{n+1} is recomputed for each y in Equation (7).

2.3 Density-based conformal prediction

The paper [6] uses a density-based conformal prediction approach inspired from the inductive approach and considers a density estimate $\hat{p}(x|y)$ of $p(x|y)$ for the label $y \in Y$. Therefore, this method divides labeled data into two parts: the first one is the *proper training* data $D^{tr} = \{X^{tr}, Y^{tr}\}$ used to build $\hat{p}(x|y)$, the second is the *calibration* data $D^{cal} = \{X^{cal}, Y^{cal}\}$ to evaluate $\{\hat{p}(x_i|y)\}$ and set $\hat{t}_y$ to be the empirical quantile of order ϵ of the values $\{\hat{p}(x_i|y)\}$:

$$\hat{t}_y = \sup\left\{t : \frac{1}{n_y} \sum_{\{z_i \in D^{cal}_y\}} I(\hat{p}(x_i|y) \geq t) \geq 1 - \epsilon\right\}, \qquad (8)$$

where n_y is the number of elements belonging to the class y in D^{cal}, and $D^{cal}_y = \{z_i \in D^{cal} : y_i = y\}$ is the subset of calibration examples of class y. For a new observation x_{n+1}, we set the conformal predictor Γ^{ϵ}_d such that :

$$\Gamma^{\epsilon}_d(x_{n+1}) = \{y \in Y : \hat{p}(x_{n+1}|y) \geq \hat{t}_y\}. \qquad (9)$$

This ensures that the observations with low probability — that is, the poorly populated regions of the input space — are classified as $\emptyset$. This divisional procedure avoids the high cost of deep learning calculations in the case where the online approach is used. The paper [6] also shows that $|P(y \in \Gamma_d^\epsilon(x_{n+1})) - (1-\epsilon)| \to 0$ with $\min_y n_y \to \infty$, which ensures the validity of the model. The training and prediction algorithms are defined in the algorithms 1 and 2.

Algorithm 1 Training algorithm

Input: Training data $Z = (x_i, y_i)$, $i = 1 \dots n$, Class list $\mathcal{Y}$, Confidence level ϵ, Ratio p.
Initialize: $\hat{p}_{list} = list, \hat{t}_{list} = list$
for $y \in \mathcal{Y}$ **do**
  $X_y^{tr}, X_y^{cal} \leftarrow SubsetData(Z, \mathcal{Y}, p)$
  $\hat{p}_y \leftarrow LearnDensityEstimator(X_y^{tr})$
  $\hat{t}_y \leftarrow Quantile(\hat{p}_y(X_y^{cal}), \epsilon)$
  $\hat{p}_{list}.append(\hat{p}_y); \hat{t}_{list}.append(\hat{t}_y)$
end for
return $\hat{p}_{list}, \hat{t}_{list}$

Algorithm 2 Prediction algorithm

Input: Input to be predicted x, Trained $\hat{p}_{list}, \hat{t}_{list}$, Class list $\mathcal{Y}$.
Initialize: $C = list$
for $y \in \mathcal{Y}$ **do**
  if $\hat{p}_y(x) \geq \hat{t}_y$ **then**
    $C.append(y)$
  end if
end for
return C

We can rewrite (9) in such a way that it approaches (7) with a few differences, mainly the fact that Γ_d^ϵ uses a conformity measure based on the density estimation (calculating how much an example is compliant with the others) instead of a non-conformity measure as in Γ^ϵ, with $\alpha_i^y = -\hat{p}(x_i|y)$ [14], and the fact that the number of examples used to build the predicition set depends on y. Thus, Γ_d^ϵ can also be written as :

$$\Gamma^\epsilon(x_{n+1}) = \left\{ y \in Y : \frac{|\{z_i \in D_y^{cal} : \alpha_i^y \geq \alpha_{n+1}^y\}|}{n_y} > \epsilon \right\}. \tag{10}$$

The proof can be found in appendix A.

The final quality of the predictor (its efficiency, robustness) depends in part on the density estimator. The paper [7] suggests that the use of kernel estimators gives good results under weak conditions.

The results of the paper show that the training and prediction of each label are independent of the other classes. This makes conformal prediction an adaptive method, which means that adding or removing a class does not require retraining the model from scratch. However, it does not provide any information on the relationship between the classes. In addition, the results depend on ϵ: when ϵ is small, the model has high precision and a large number of classes predicted for each observation. On the contrary, when ϵ is large, there are no more cases classified as $\emptyset$ and fewer cases predicted by label.

3 Experiments

In order to examine the effectiveness of the conformal method on different types of data, three data sets for binary classification were used. They are :

1. **CelebA** [8] : face attributes dataset with over 200,000 celebrity images used to determine if a person is a man (1) or a woman (0).
2. **IMDb** [9] : contains more than 50,000 different texts describing film reviews for sentiment analysis (with 1 representing a positive opinion and 0 indicating a negative opinion).
3. **EGSS** [1] : contains 10000 examples for the study of the electrical networks' stability (1 representing a stable network), with 12 numerical characteristics.

3.1 Approach

The overall approach followed the same steps as in density-based conformal prediction [6] and meets the conditions listed above (the i.i.d. or exchangeability assumptions). Each data set is divided into proper training, calibration and test sets. A deep learning model dedicated to each type of data is trained on the proper training and calibration sets. The before last dense layer serves as a feature extractor which produces a fixed size vector for each dataset and representing the object (image, text or vector). These feature vectors are then used for the conformal part to estimate the density. Here we used a gaussian kernel density estimator of bandwidth 1 available in Python's scikit-learn [12]. The architecture of deep learning models is shown in Figure 1. It is built following the steps below:

1. Use a basic deep learning model depending on the type of data. In the case of CelebA, it is a CNN with a ResNet50 [5] pre-trained on ImageNet [2] and adjusted to CelebA. For IMDb, this model is a bidirectional GRU that takes processed data with a tokenizer and padding. For EGSS, this model is a multilayer perceptron (MLP).
2. Apply an intermediate dense layer and use it as a feature extractor with a vector of size 50 representing the object, and which will be used later for conformal prediction.
3. Add a dense layer to obtain the class predicted by the model (0 or 1).

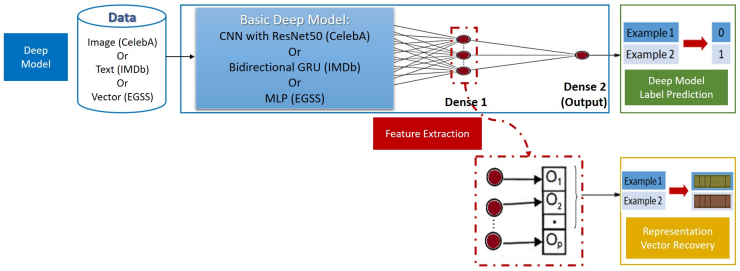

Fig. 1: Architecture of deep learning models.

Based on the recovered vectors, a Gaussian kernel density estimate is made on the proper training set of each class to obtain the values $P(x|y)$. Then, the calibration set is used to compute the density scores and sort them to determine the given ϵ threshold of all the values, thus delimiting the density region of each class. Finally, the test set is used to calculate the performance of the model. The code used for this article is available in Github [1].

The visualization of the density regions (figure 2) is done via the first two dimensions of a Principal Component Analysis. The results show the distinct regions of the classes 0 (in red) and 1 (in blue) with a non-empty intersection (in green) representing a region of random uncertainty. The points outside these three regions belong to the region of epistemic uncertainty, meaning that the classifier "does not know".

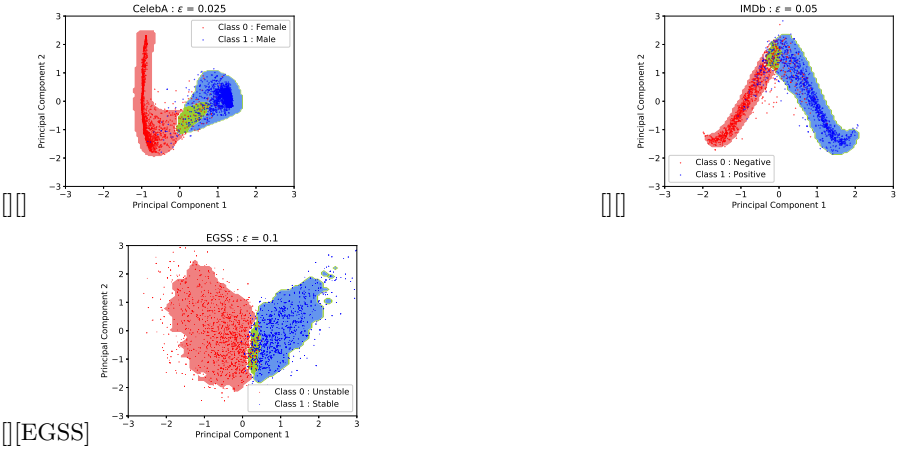

Fig. 2: Conformal prediction density regions for all datasets.

[1] https://github.com/M-Soundouss/density_based_conformal_prediction

3.2 Results on the test examples

To obtain more information on the results of this experiment, the accuracy of the models was calculated with different values ϵ between 0.01 and 0.5 when determining the threshold of conformal prediction density as follows:

- DL accuracy: the accuracy of the basic deep model (CNN for CelebA, GRU for IMDb or MLP for EGSS) on all the test examples.
- Valid conformal accuracy: the accuracy of the conformal model when one considers only the singleton predictions 0 or 1 (without taking into account the $\{0, 1\}$ and the empty sets).
- Valid DL accuracy: The accuracy of the basic deep model on the test examples that have been predicted as 0 or 1 by the conformal model.

The percentage of empty sets $\emptyset$ and $\{0, 1\}$ sets was also calculated from all the predictions of the test examples made by the conformal prediction model. The results are shown in the figure 3.

The results show that the accuracy of the valid conformal model and the accuracy of the valid basic deep learning model are almost equal and are better than the accuracy of the base model for all ϵ values. In our tests, the addition of conformal prediction to a deep model does not degrade its performance, and sometimes even improves it (EGSS). This is due to the fact that the conformal prediction model allows to abstain from predicting (empty set $\emptyset$) or to predict both classes for ambiguous examples, thus making it possible to have a more reliable prediction of the label. It is also noticed that as ϵ grows, the percentage of predicted $\{0, 1\}$ sets decreases until it is no longer predicted (at $\epsilon = 0.15$ for CelebA for example). Conversely, the opposite is observed with the percentage of empty sets $\emptyset$ which escalates as ϵ increases.

3.3 Results on noisy and foreign examples

CelebA : Two types of noise were introduced: a noise masking parts of the face and another Gaussian on all the pixels. These perturbations and their predictions are illustrated in the figure 4 with "CNN" the prediction of the CNN and "CNN + CP" that of the conformal model. This example shows that the CNN and the conformal prediction model correctly identify the woman in the image (a). However, by masking the image (b), the CNN predicts it as a man with a score of 0.6 whereas the model of conformal prediction is more cautious by indicating that it does not know ($\emptyset$). When applying a Gaussian noise over the whole image (c), the CNN predicts that it is a man with a larger score of 0.91, whereas the conformal model predicts both classes. For outliers, examples (d), (e), and (f) illustrate the ability of the conformal model to identify different outliers as such ($\emptyset$) in contrast to the deep model that predicts them as men with a high score.

IMDb : The figure 5 displays a comparison of two texts before and after the random change of a few words (in bold) by other words in the model's vocabulary. The actual text predicted as negative opinion by both models becomes positive for

the GRU after disturbance. Nevertheless, the conformal model is more cautious by indicating that it can be both cases ($\{0, 1\}$). For the outlier example formed completely of vocabulary words, the GRU model predicts positive with a score of 0.99, while the conformal model says that it does not know ($\emptyset$).

EGSS : The figure 6 displays a comparison of the positions of the test examples on the density regions before (a) and after (b) the addition of a Gaussian noise. This shows that several examples are positioned outside the density regions after the introduction of the disturbances. The outlier examples (c) created by modifying some characteristics of these test examples with extreme values (to simulate a sensor failure, for example) are even further away from the density regions, and recognized as such by the conformal model ($\emptyset$).

4 Conclusions and perspectives

We used the conformal prediction and the technique presented in [6] to have a more reliable and cautious deep learning model. The results show the interest of this method on different data types (image, text, tabular) used with different deep learning architectures (CNN, GRU and MLP). Indeed, in these three cases, the conformal model not only adds reliability and robustness to the deep model by detecting ambiguous examples but also keeps or even improves the performance of the basic deep model when it predicts only one class. We also illustrated the ability of conformal prediction to handle noisy and outlier examples for all three types of data. These experiments show that the conformal method can give more robustness and reliability to predictions on several types of data and basic deep architectures.

To improve the experiments and results, the perspectives include the optimization of density estimation based on neural networks. For instance, at a fixed ϵ the problem of finding the most efficient model arises that could be done by modifying the density estimation technique, but also by proposing an end-to-end, integrated estimation method. Also, it would be useful to compare the conformal prediction with calibration methods, for example, evidential ones that are also adopted for cautious predictions [3].

A Appendix

This appendix is to prove that equations (9) and (10) in Section 2.3 are equivalent. We recall that equation (10) is

$$\Gamma^{\epsilon}(x_{n+1}) = \left\{ y \in Y : \frac{|\{z_i \in D_y^{cal} : \alpha_i^y \geq \alpha_{n+1}^y\}|}{n_y} > \epsilon \right\}. \tag{11}$$

We recall that equation (9) uses the "greater or equal" sign. Here we need to use the "greater" signs in equations (12) and (13) to have an equivalence, which is

$$\Gamma_d^{\epsilon}(x_{n+1}) = \{y \in Y : \hat{p}(x_{n+1}|y) > \hat{t}_y\}, \tag{12}$$

such that

$$\hat{t}_y = \sup\left\{ t : \frac{1}{n_y} \sum_{\{z_i \in D_y^{cal}\}} I(\hat{p}(x_i|y) > t) \geq 1 - \epsilon \right\}. \tag{13}$$

Let $f(t)$ be the decreasing function $f(t) = \frac{1}{n_y} \sum_{\{z_i \in D_y^{cal}\}} I(\hat{p}(x_i|y) > t)$.

Let us prove that (12) $\implies$ (11).

Since $\hat{t}_y$ is the upper bound such that $f(\hat{t}_y) \geq 1 - \epsilon$, then $\hat{p}(x_{n+1}|y)$ does not satisfy this inequality, thus

$$\begin{aligned} f(\hat{p}(x_{n+1}|y)) &= \frac{1}{n_y} \sum_{\{z_i \in D_y^{cal}\}} I(\hat{p}(x_i|y) > \hat{p}(x_{n+1}|y)) < 1 - \epsilon \\ &= \frac{1}{n_y} \sum_{\{z_i \in D_y^{cal}\}} 1 - I(\hat{p}(x_i|y) \leq \hat{p}(x_{n+1}|y)) < 1 - \epsilon \\ &= 1 - \frac{1}{n_y} \sum_{\{z_i \in D_y^{cal}\}} I(\hat{p}(x_i|y) \leq \hat{p}(x_{n+1}|y)) < 1 - \epsilon \\ &= \frac{1}{n_y} \sum_{\{z_i \in D_y^{cal}\}} I(\hat{p}(x_i|y) \leq \hat{p}(x_{n+1}|y)) > \epsilon \end{aligned} \tag{14}$$

Since $\hat{p}(x_{n+1}|y))$ is a conformity score, whereas α_i^y is a non-conformity score, we can write $\hat{p}(x_{n+1}|y)) = -\alpha_i^y$ [14]. So (14) becomes

$$\frac{1}{n_y} \sum_{\{z_i \in D_y^{cal}\}} I(\alpha_i^y \geq \alpha_{n+1}^y) > \epsilon \implies \frac{|\{z_i \in D_y^{cal} : \alpha_i^y \geq \alpha_{n+1}^y\}|}{n_y} > \epsilon$$

This shows that (12) $\implies$ (11).

Let us now prove that (11) $\implies$ (12). Using the indicator function of the complement, and changing the non-conformity score into a conformity score as shown before, we can simply find that

$$\frac{|\{z_i \in D_y^{cal} : \alpha_i^y \geq \alpha_{n+1}^y\}|}{n_y} > \epsilon \implies \frac{1}{n_y} \sum_{\{z_i \in D_y^{cal}\}} I(\hat{p}(x_i|y) > \hat{p}(x_{n+1}|y)) < 1 - \epsilon$$

Using the same function f, we then have

$$f(\hat{p}(x_{n+1}|y)) < 1 - \epsilon. \tag{15}$$

Let us show by contradiction that $\hat{p}(x_{n+1}|y) > \hat{t}_y$. Suppose that $\hat{p}(x_{n+1}|y) \leq \hat{t}_y$. Since f is a decreasing function, we have $f(\hat{p}(x_{n+1}|y)) \geq f(\hat{t}_y)$. By the definition of $\hat{t}_y$, we have $f(\hat{t}_y) \geq 1 - \epsilon$. Thus $f(\hat{p}(x_{n+1}|y)) \geq f(\hat{t}_y) \geq 1 - \epsilon$. However, this contradicts (15). So we proved that (11) $\implies$ (12), which concludes the proof.

References

1. Arzamasov, V.: UCI electrical grid stability simulated data data set (2018), https://archive.ics.uci.edu/ml/datasets/Electrical+Grid+Stability+Simulated+Data+
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)
3. Denoeux, T.: Logistic regression, neural networks and dempster–shafer theory: A new perspective. Knowledge-Based Systems **176**, 54–67 (2019)
4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6572
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
6. Hechtlinger, Y., Póczos, B., Wasserman, L.: Cautious deep learning. arXiv preprint arXiv:1805.09460 (2018)
7. Lei, J., Robins, J., Wasserman, L.: Distribution-free prediction sets. Journal of the American Statistical Association **108**(501), 278–287 (2013)
8. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
9. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-vol. 1. pp. 142–150 (2011)
10. Melluish, T., Saunders, C., Nouretdinov, I., Vovk, V.: Comparing the bayes and typicalness frameworks. In: European Conference on Machine Learning. pp. 360–371. Springer (2001)
11. Papadopoulos, H.: Inductive conformal prediction: Theory and application to neural networks. In: Tools in artificial intelligence. IntechOpen (2008)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python . Journal of Machine Learning Research **12**, 2825–2830 (2011)
13. Proedrou, K., Nouretdinov, I., Vovk, V., Gammerman, A.: Transductive confidence machines for pattern recognition. In: European Conference on Machine Learning. pp. 381–390. Springer (2002)
14. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world. Springer Science & Business Media (2005)

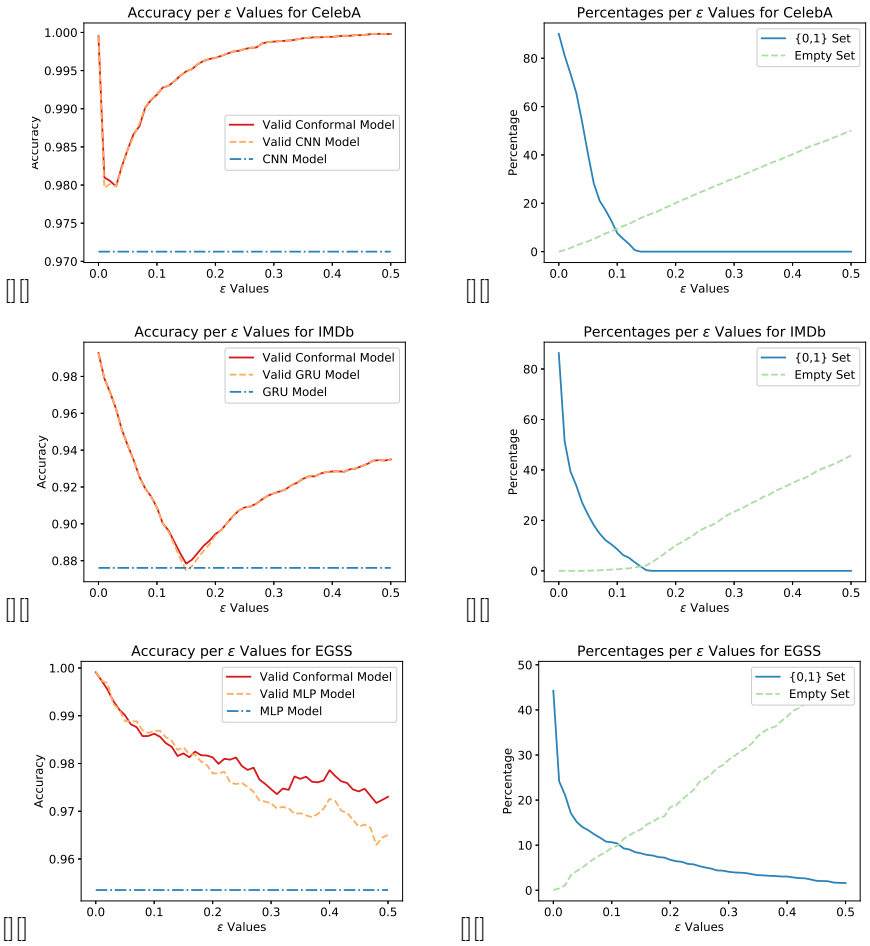

Fig. 3: The accuracy and the percentages according to ϵ for CelebA (top), IMDb (middle) and EGSS (bottom).

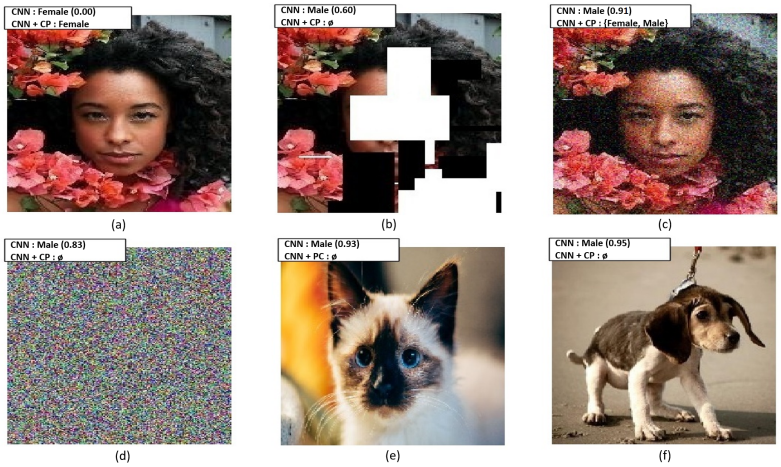

Fig. 4: Examples of outlier and noisy images compared to the actual image for CelebA.

Original Example	« **Every great** romantic comedy **needs** conflict between the romantic leads [...] **This story** falls completely flat in this **area** [...] suspense is flat, **there** is no anticipation, **and** there **really is no** allure [...] I was quite surprised. During the movie, I **expected them** more **to** play **a** game of checkers and chat **about** the weather than see **any** moving passion [...] While I'm a fan **of** both actors [...] The writing **was** very **weak, which** also **might** have **impacted** the **performances** [...] »	Label : Negative sentiment GRU : Negative sentiment (0.09) GRU + CP : Positive sentiment
Noisy Example	« **ambidexterous trentini** romantic comedy **dispassionately** conflict between the romantic leads [...] **fanout phoolan** falls completely flat in this **centerpiece** [...] suspense is flat, **binding** is no anticipation, **scroller** there **wiedzmin laudable thunderball** allure [...] I was quite surprised. During the movie, I **lives' 'are** more **subtextual** play **commemorations** game of checkers and chat **stratovarius** the weather than see **linking** moving passion [...] While I'm a fan **unti** both actors [...] The writing **ripoff's** very **nare releases** also **sharikov** have **maes** the **'sketching'** [...] »	Label : Negative sentiment , GRU : Positive sentiment (0.74) GRU + CP : {Positive sentiment , Negative sentiment }
Outlier Example	« wolverines 'sandwich' controversial posit homme subfunctions snowmobile symbiotic malamud challenge needle's personl witch's nonce wills' swooshes cobbled brash mcq wanky 'bought regenerated southstreet amazed ravenna 'mainly belyt hijixn shrugs deodorant mesquida anodynesprech romishness malice seldomely settling dispicable vocation [...] reduce macfarlane's disclosing officers' wiretapping balbao seagals mi3 dibnah romulan controls dolled maguire' [...] »	GRU : Positive sentiment (0.99) GRU + CP : ∅

Fig. 5: Examples of outlier and noisy texts compared to the original one for IMDb.

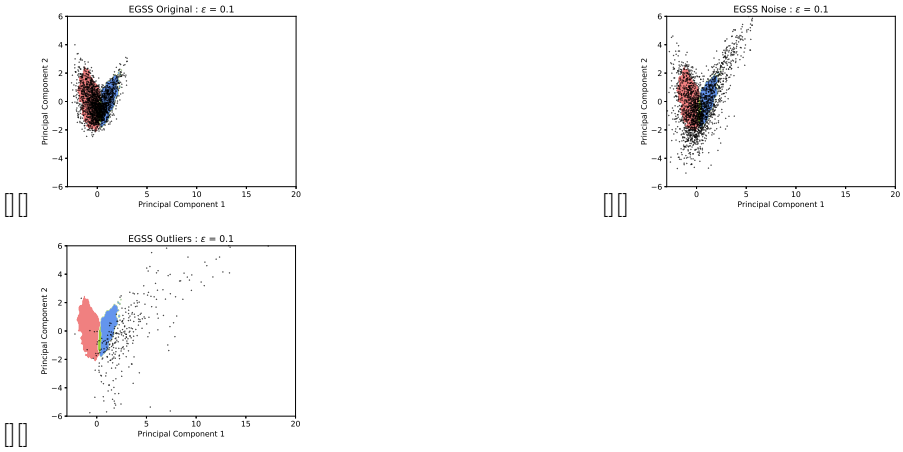

Fig. 6: Density visualization of (a) real, (b) noisy and (c) outlier examples for EGSS.