UNIVERSITY OF HELSINKI

https://helda.helsinki.fi

Failure Prediction in 2D Document Information Extraction with Calibrated Confidence Scores

Kivimäki, Juhani Ilmari; Lebedev, Aleksey; Nurminen, Jukka K

2023

http://hdl.handle.net/10138/565446

Kivimäki, J I, Lebedev, A & Nurminen, J K 2023, Failure Prediction in 2D Document Information Extraction with Calibrated Confidence Scores. in 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, pp. 193-202, Annual Computers, Software, and Applications Conference, Torino, Italy, 26/06/2023. https://doi.org/10.1109/COMPSAC57700.2023.00033

Downloaded from Helda, University of Helsinki institutional repository. https://helda.helsinki.fi
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.
Please cite the original version.

Failure Prediction in 2D Document Information Extraction with Calibrated Confidence Scores

Juhani Kivimäki
University of Helsinki
Helsinki, Finland
juhani.kivimaki@helsinki.fi
0000-0002-9673-9760

Aleksey Lebedev
Basware Inc.
Espoo, Finland
aleksey.lebedev@basware.com

Jukka K. Nurminen
University of Helsinki
Helsinki, Finland
jukka.k.nurminen@helsinki.fi
0000-0001-5083-1927

***Abstract*—Modern machine learning models can achieve impressive results in many tasks, but often fail to express reliably how confident they are with their predictions. In an industrial setting, the end goal is usually not a prediction of a model, but a decision based on that prediction. It is often not sufficient to generate high-accuracy predictions on average. One also needs to estimate the uncertainty and risks involved when making related decisions. Thus, having reliable and calibrated uncertainty estimates is highly useful for any model used in automated decision-making.**

In this paper, we present a case study, where we propose a novel method to improve the uncertainty estimates of an in-production machine learning model operating in an industrial setting with real-life data. This model is used by Basware, a Finnish software company, to extract information from invoices in the form of machine-readable PDFs. The solution we propose is shown to produce calibrated confidence estimates, which outperform legacy estimates on several relevant metrics, increasing coverage of automated invoices from 65.6% to 73.2% with no increase in error rate.

***Index Terms*—machine learning, uncertainty estimation, confidence calibration, failure prediction, information extraction**

I. Introduction

In recent years, machine learning (ML) models have been shown to produce impressive results in a wide range of tasks. However, the same models which yield high-accuracy predictions are usually poor in expressing reliably the confidence in those predictions [1]. This is even more true if the predictions are made for data, which originates from a different distribution than the data used in training the model [2], [3]. In real life, these predictions are leveraged in automated decision-making. Thus, the inability to reliably quantify the ubiquitous uncertainty presents a serious risk. The ML systems of today are deployed in increasingly large-scale, autonomous, open-domain situations [4], which underlines the need to assess these uncertainties in a reliable and robust manner. Accurate uncertainty estimation provides models with resilience to retain their utility even under volatile circumstances.

There are many ways in which models can express uncertainty in their predictions. Bayesian models are trained to output distributions instead of point predictions [5]. Conformal predictions yield their predictions as sets [6]. In both cases, considerable effort is needed in implementing these uncertainty estimation techniques and further analysis is required in interpreting the results they yield. A more straightforward and easily interpretable way is to express uncertainty as a scalar value. A common choice is to use a *confidence score* [1], which is typically restricted on the interval $[0, 1]$, with 1 signaling maximal confidence. Confidence scores can be derived in many different ways depending on the model. For example, a probabilistic classifier with K classes is a mapping $f : \mathcal{X} \to \Delta_K$, where Δ_K is the $K-1$ dimensional probability simplex. The predicted class is $y = \text{ARGMAX}(f(\mathbf{x}))$ and $\text{MAX}(f(\mathbf{x}))$ is typically used as the confidence score for the prediction.

Confidence estimates have been utilized in information extraction (IE) as well, although deriving them might be less straightforward and depend on the specifics of the model being estimated. We present some previous strategies for this in Section V. In recent years, there has been a growing interest in developing IE solutions for *semi-structured documents*, where the 2D location of the information is highly relevant. Such documents include invoices, receipts, and forms with varying layouts. They are frequently used in financial, medical, and other domains. Currently, confidence estimation in these solutions is somewhat lacking, typically restricted to pixel or token level [7]–[10], while document level confidence scores might be preferred instead.

A naïve interpretation would equate the confidence score of a prediction with the probability that the prediction was correct. However, this is not usually the case: Empirical analysis shows that the probabilistic predictions of contemporary ML models are often either under- or overconfident [1], [11]. For example, if one looks at all the predictions a model made with say 80% confidence, it might be the case that the expected accuracy of those predictions was only 50%. A model, for which the interpretation of confidence scores as probabilities is justified, is called *calibrated*. If the confidence scores produced by a model are not calibrated, they can be *re-calibrated* by applying a post hoc *calibration mapping* on the confidence scores in order to have a better match between the confidence scores and empirical probabilities [1], [2], [11]–[17]. Also, some specific ways of training models induce an implicit

This work was partly funded by local authorities ("Business Finland") under grant agreements ITEA-2019-18022-IVVES of ITEA3 programme and 20219 IML4E of ITEA4 programme.

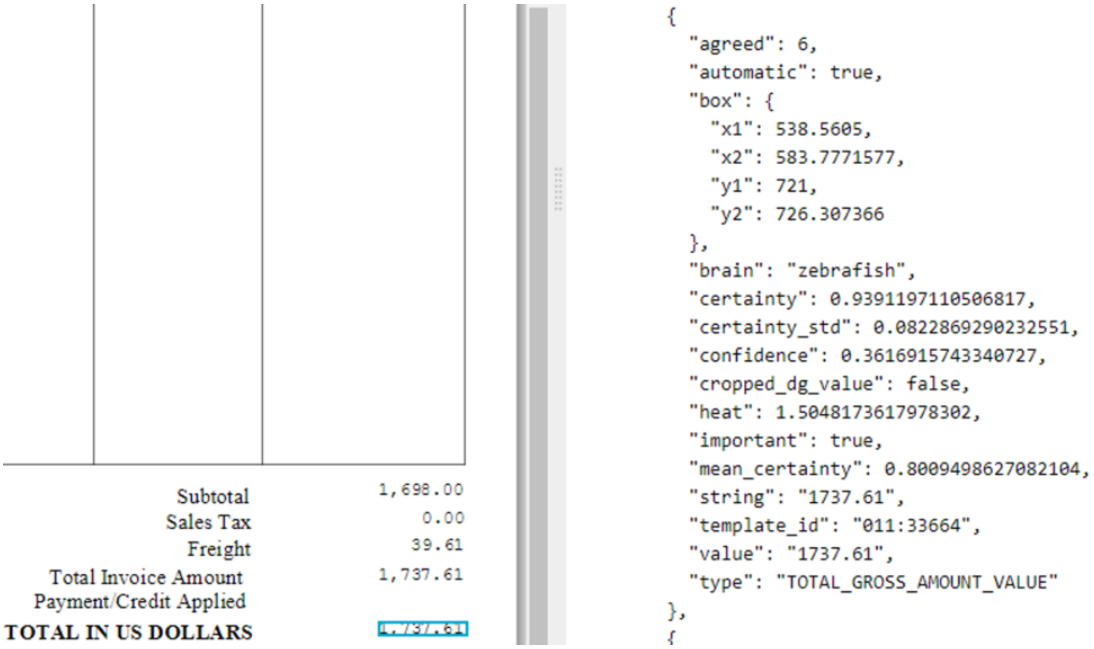


Fig. 1. An example of an invoice, along with the corresponding prediction

calibration effect on the confidence scores [3], [18]–[22].

In this work, we present a case study, where we suggest a novel hybrid ML method to improve upon existing confidence estimates of a deployed IE model by the Finnish software company Basware used to process machine-readable PDF invoices on an industrial scale. We propose to augment this model with an unintrusive auxiliary part, which uses latent representations and statistics of the base model gathered during inference as inputs and learns to assign calibrated confidence scores for each prediction of the base model. By unintrusive we mean that no changes in the architecture or training process of the base model are needed. These confidence scores can then be used in automated decision-making as well as downstream tasks such as model monitoring. We strive for calibrated confidence scores, since they enable a more accurate assessment of the risks involved, yield a principled way of monitoring the predictive accuracy, and allow rejection of low confidence predictions [3]. We show that the proposed solution improves on the legacy confidence estimates by a significant margin on several relevant metrics.

In summary, the key contributions of this paper are:

1) We propose a novel hybrid ML method for estimating the confidence of the predictions of an IE model ensemble by inspecting latent representations of the models together with statistics gathered during inference (in Section III).
2) We demonstrate the effectiveness of the proposed method with a case study in an industrial information extraction setting with real-life data (in Section IV) and compare it to related research (in Section V).
3) We provide insight and analysis for the generalizability of the method to other similar cases (in Section VI).

II. Case description

Basware is a publicly listed software company selling enterprise software for financial processes, purchase-to-pay, and financial management operating in over 170 countries. Their AI product consists of a complex distillation pipeline (DPL), which processes PDF invoices. At the core of the DPL is an ensemble of six image segmentation models, called a committee. Each model in the committee independently produces a heatmap representation for each field of targeted information, such as total amount, tax, invoice number, freight, and so on. The heatmaps are then interpreted to locate the desired information on the invoice. Once the location has been established, information in that location can be directly read from the PDF file.

For each invoice processed, a decision of whether the inference can be trusted needs to be made. Low confidence predictions are rejected and sent to manual inspection, which cuts profits. Thus, maximizing the fraction of automated invoices (with acceptable precision) is a key performance indicator (KPI). We refer to this fraction in this work as *coverage*. The confidence estimate is based on both *hard* and *soft predictions* [23]. The hard confidence is simply the number of models within the committee in agreement with the majority prediction. The soft confidence is derived from statistics on the heat expressed in the heatmaps. A sample invoice along with the gathered statistics is shown in Fig. 1. In this work, we refer to this soft confidence as *legacy confidence*. Complex heuristics are applied to arrive at both the final value for the extracted information as well as the final confidence estimates. The final decision of whether an invoice can be processed automatically is made on the basis of whether the predictions for five important fields, namely

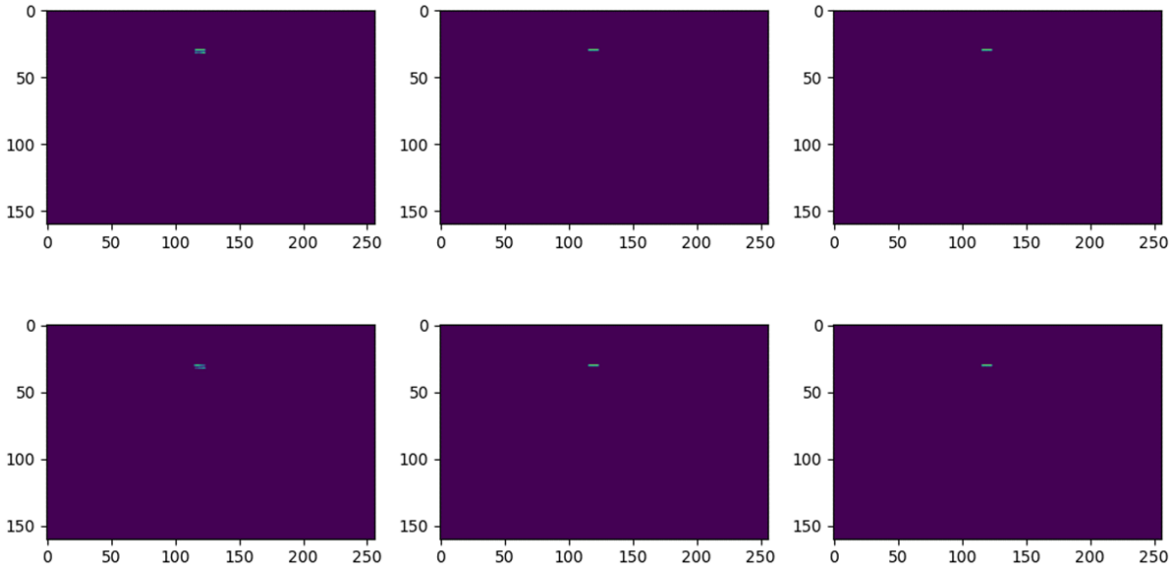


Fig. 2. A heatmap stack of six heatmaps from the *Heatmap extractor* for field `total` for one randomly picked invoice. Typically, most pixels express very little heat. Although these representations are very similar, small differences can be observed. Our proposed solution seeks to leverage these differences in predicting whether the inference of the DPL led to a correct prediction.

`invoice_number`, `issue_date`, `subtotal`, `tax` and `total` have been made with high enough confidence.

Currently, confidence is mainly based on hard predictions, since the quality of the soft predictions is assessed to be lacking. However, the hard predictions act as a rather coarse metric, since it can only take six distinct values (1-6). In this work, we seek to develop a new solution to replace the legacy confidence estimates with calibrated confidence scores for better-quality soft predictions, which would enable a more fine-grained scale for the confidence scores of the predictions of the DPL.

III. Methodology

Our proposed solution can be labeled as a mechanism for failure prediction [24] or selective classification [23]. Either way, the idea is to identify data points (in our case invoices), which a given model is less likely to predict correctly, due to the data point being ambiguous, out-of-distribution (OoD), or otherwise anomalous. In general, when a failure prediction mechanism is based on confidence scores, it is essentially solving a ranking task [25]. Ideally, correctly predicted samples should be assigned with higher confidence scores than misclassified samples. In the following subsections, we present our solution designed to solve this task. Our proposed hybrid method consists of convolutional networks, an XGBoost model, and a Beta calibration mapping.

A. Convolutional Networks

In order to test our proposed method, we sampled a set of roughly 240,000 real-world invoices with known ground truth values. At the beginning of the DPL is an in-house clustering algorithm, which tries to assign a template id for each invoice processed by the system, though this is not possible for some outliers. Our dataset contains only invoices with a known template id. Invoices were selected so that all known template ids are represented and the imbalances in their distribution are reflected. We restricted our dataset to contain only single-page invoices for simplicity.

We made use of the 160×256 pixel heatmap representations produced by the DPL. For each of the five important fields, we stacked the heatmaps produced by the six models (shown in Fig. 2) in the committee as three-dimensional representations and trained a convolutional network to predict, whether the information for that field was extracted correctly by the DPL. Since a lightweight solution was sought after, we used a modernized version of the LeNet-5 architecture [26] as a starting point. We experimented with kernel and filter sizes as well as the number of layers, until settling on the solution presented in this work. We used spatial dropout and dropout as a form of regularization, with the dropout rates tuned to achieve the best possible generalization performance.

On the first layer, the model performs 3-dimensional convolutions directed to compare differences between the heatmaps of the different models. This layer also flattens the 3D representations into two dimensions. Then, four more convolutional layers, each followed by a max pooling layer, were added. These were followed by four fully connected layers, where the final layer has only one node and a sigmoid activation. Padding is performed before convolutions to retain input dimension with the exception of the first convolutional layer. ReLU activations are used after each convolutional and dense layer, with the exception of two final layers. A summarization of the architecture is given in Table I.

These five convolutional network models were trained to predict, whether the DPL resulted in a correct prediction for a given field, given the heatmap representations for all

TABLE I
LAYER DESCRIPTIONS FOR THE PROPOSED ARCHITECTURE

Layer	input dimension	description	output dimension
1	$160 \times 256 \times 6 \times 1$	Conv3D (kernel: $1 \times 5 \times 6$, filters: 8, flatten to 2D)	$160 \times 256 \times 8$
2	$160 \times 256 \times 8$	Conv2D (kernel: 5×7, filters: 16)	$160 \times 256 \times 16$
3	$160 \times 256 \times 16$	MaxPool2D (kernel: 3×3, stride: 2×2, spatial dropout rate: 0.1)	$80 \times 128 \times 16$
4	$80 \times 128 \times 16$	Conv2D (kernel: 3×5, filters: 32)	$80 \times 128 \times 32$
5	$80 \times 128 \times 32$	MaxPool2D (kernel: 3×3, stride: 2×2, spatial dropout rate: 0.1)	$40 \times 64 \times 32$
6	$40 \times 64 \times 32$	Conv2D (kernel: 3×5, filters: 48)	$40 \times 64 \times 48$
7	$40 \times 64 \times 48$	MaxPool2D (kernel: 3×3, stride: 2×2, spatial dropout rate: 0.1)	$20 \times 32 \times 48$
8	$20 \times 32 \times 48$	Conv2D (kernel: 1×1, filters: 12)	$20 \times 32 \times 12$
9	7680	Dense (dropout rate: 0.33, activation: ReLu)	300
10	300	Dense (dropout rate: 0.33, activation: ReLu)	150
11	150	Dense (dropout rate: 0.25, activation: Tanh)	24
12	24	Dense (activation: sigmoid)	1

members of the committee. Each model was trained with Adam optimizer (learning rate: 0.0005) for 8 epochs with a batch size of 128. We applied simple data augmentation by shuffling the order of the heatmaps in the heatmap stacks between training epochs. The initial results for this approach, although promising, were deemed not good enough. For fields `invoice_number` and `total`, the results were quite good ($\sim 95\%$ accuracy on an independent test set), but for other fields, they were significantly worse ($80 - 90\%$ test set accuracy). One shortcoming of this approach is that the convolutional networks can only express confidence on the per-field level, whereas confidence on the document level was sought after. Furthermore, the convolutional networks are trained to predict the per-field confidences independently of each other. In reality, there are dependencies between these predictions, as the predictions for each field are derived from the same data (a particular invoice), which acts as a confounder. Thus, we decided to augment our solution to account for these dependencies.

B. Augmented feature vectors

After training the convolutional networks, instead of using them to derive final confidence estimates, we used them as feature extractors by dropping the final layer from the networks and using the 24-dimensional output vector from the penultimate layer as an embedded latent representation of the heatmap stack for a given field on a given invoice. In order to get more expressive latent representations, we replaced the ReLu activation on the penultimate layer with Tanh, with no decrease in predictive performance. We then sought to increase the expressiveness of these *Heatmap features* even further.

Since the convolutional networks pay attention only to local details in the heatmaps, we also decided to complement the Heatmap features with statistics from the heatmaps, which would be able to capture more global characteristics of the heatmaps. First, we created a mean map by averaging the per-pixel heat of all pixels over all 6 models. Then, the maximum, average, and variance of the mean map values were calculated. In addition, an entropy map was also created, containing the per-pixel entropy between the models. After this, three different heat thresholds, namely 0.001, 0.01, and 0.1 were employed. If the maximum heat value for a pixel over all heatmaps in the stack was below a given heat threshold, that pixel would be discarded from the calculations for that threshold level. For all heat threshold levels, 6 statistics were calculated: The number of pixels included, the average and variance of the mean map for all included pixels, and finally, the minimum, average, and variance of the entropy map for all included pixels. Thus, a total of $21 (= 3 + 6 \cdot 3)$ values for each of the five important fields were gathered, referred to here as *Heatmap statistics*.

The intuition behind this way of gathering statistics arises mainly from a finding that the heatmaps typically consist mostly of pixels with low heat. This is illustrated in Fig. 2. If the heat value of a certain pixel is very close to zero in all heatmaps, comparing the maps over that pixel is uninformative, which justifies the use of thresholding. If the models place heat in highly different locations, the number of pixels exceeding a threshold grows larger, and the per-pixel entropies and overall variance tend to grow larger as well. On the other hand, if the models place the heat in similar locations but the absolute heat values are small, the average values tend to become smaller.

Finally, we gathered 6 more *Committee statistics* for each field, produced by the DPL during inference as shown in Fig. 1. We combined the Committee statistics together with the Heatmap features and Heatmap statistics to form 51-dimensional *augmented feature vectors* per important field. Finally, we concatenated the 5 per-field augmented feature vectors as a single 255-dimensional augmented feature vector representation for each invoice.

C. XGBoost and Calibration

Using the 255-dimensional augmented feature vectors as inputs, we trained a binary XGBoost classifier [27] to predict, whether the DPL could predict all important fields for the corresponding invoices correctly or not. XGBoost was chosen since it is generally recognized as a state-of-the-art model in classification tasks. Because the XGBoost model has access to features related to all important fields, it can leverage the dependencies between the different fields when making its predictions. The hyperparameters of the XGBoost model were

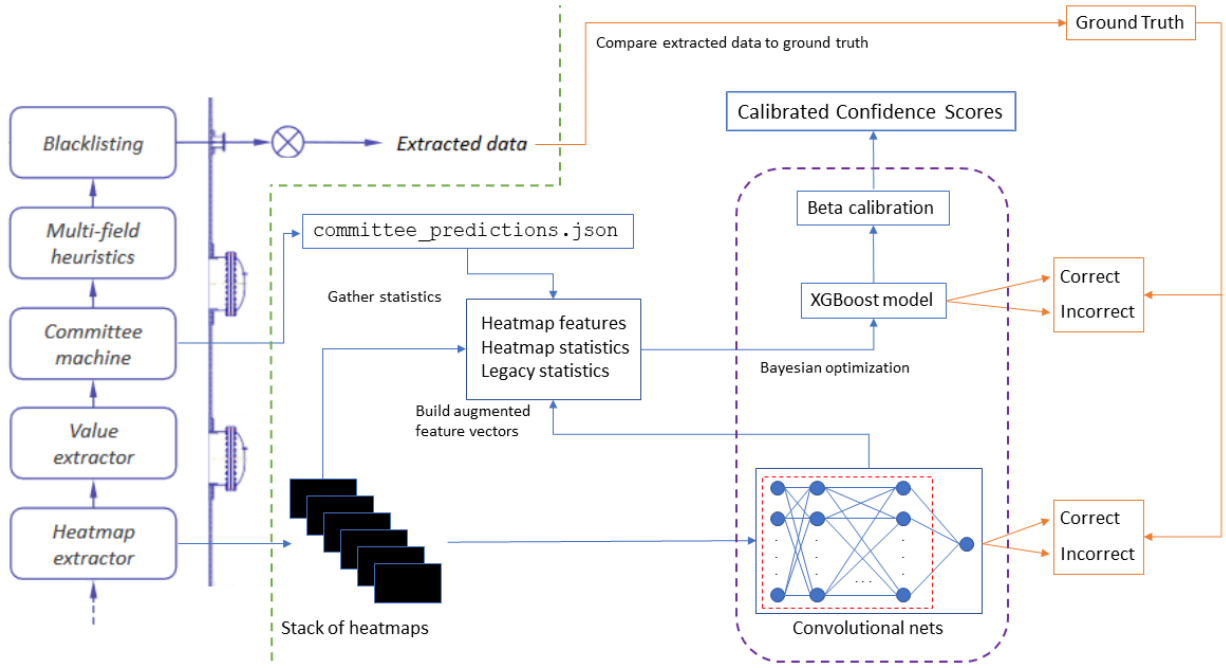


Fig. 3. An overview of the proposed confidence estimator hybrid model trained to produce calibrated confidence scores for the predictions of the Basware DPL on the document level. The final parts of the DPL, where the *Heatmap extractor* is responsible for producing the heatmaps, are shown on the left side of the green dashed line. Parts with orange color are only present during training. The trainable models in our proposed method are marked with the purple dashed line.

tuned with Bayesian hyperparameter optimization [28], using the `scikit-optimize`[1] library.

After training the XGBoost model, its soft outputs were interpreted as confidence scores for the whole invoice. However, it was to be expected that these confidence scores would be uncalibrated. Thus, as a final part of our confidence estimation pipeline, we trained a Beta calibration [12] mapping to map the soft outputs of the XGBoost model to better match empirical probabilities. We chose Beta calibration for our post hoc calibration method out of several tested methods, since it gave the best performance and was deemed more stable than some equally good options.

The flow chart of the entire proposed hybrid solution is depicted in Fig. 3. It shows how our proposed hybrid confidence estimator attaches to the final stages of the DPL, gathers the extracted heatmap stacks and Committee statistics, creates the augmented feature vector representations, and predicts a calibrated confidence score for each invoice. Naturally, ground truth is only available during training. We used a stratified split for the data, where 40% of the invoices was used in training the convolutional networks, 40% in training the XGBoost model, 10% in training the Beta Calibration mapping and the remaining 10% as an independent test set.

IV. Results

The ultimate goal of our proposed solution is to produce calibrated confidence scores to be used in dividing predictions of the DPL into two categories (trust or reject). Thus, the quality of the split depends on the chosen confidence threshold or *operating point* [29] and it is justified to assess the predictive quality over all possible operating points. A commonly used metric in this context is the Receiver Operating Characteristic (ROC) curve [29], where the true positive rate is plotted against the false positive rate over all confidence thresholds. Area Under the ROC Curve (AUC) can be used as a scalar overall performance metric when needed. It can be interpreted as a ranking quality metric, measuring the probability that the confidence scores given by a classifier will rank a randomly chosen correctly predicted instance higher than a randomly chosen incorrect one [30].

In addition to ROC, we measure our model's precision/coverage trade-off by an Accuracy-Reject Curve (ARC) [31], which shows the accuracy of the non-rejected predictions (precision) as a function of the rejection rate. In this work, we replace the rejection rate with coverage, since the KPI of the Basware DPL is expressed in terms of coverage. Also, rejection rate and coverage are additive inverses, so they convey essentially the same information. Similar to AUC, Area Under the ARC Curve (AURC) can be taken to indicate model overall performance [31].

The ROC and ARC curves are shown in Fig. 4. They show that the confidence scores produced by our proposed solution dominate the legacy confidence scores across all confidence thresholds. Our proposed solution improves AUC from 0.795 to 0.911 and AURC from 0.760 to 0.842 when compared to the legacy confidence estimates. The improved separation

[1]https://scikit-optimize.github.io/stable/

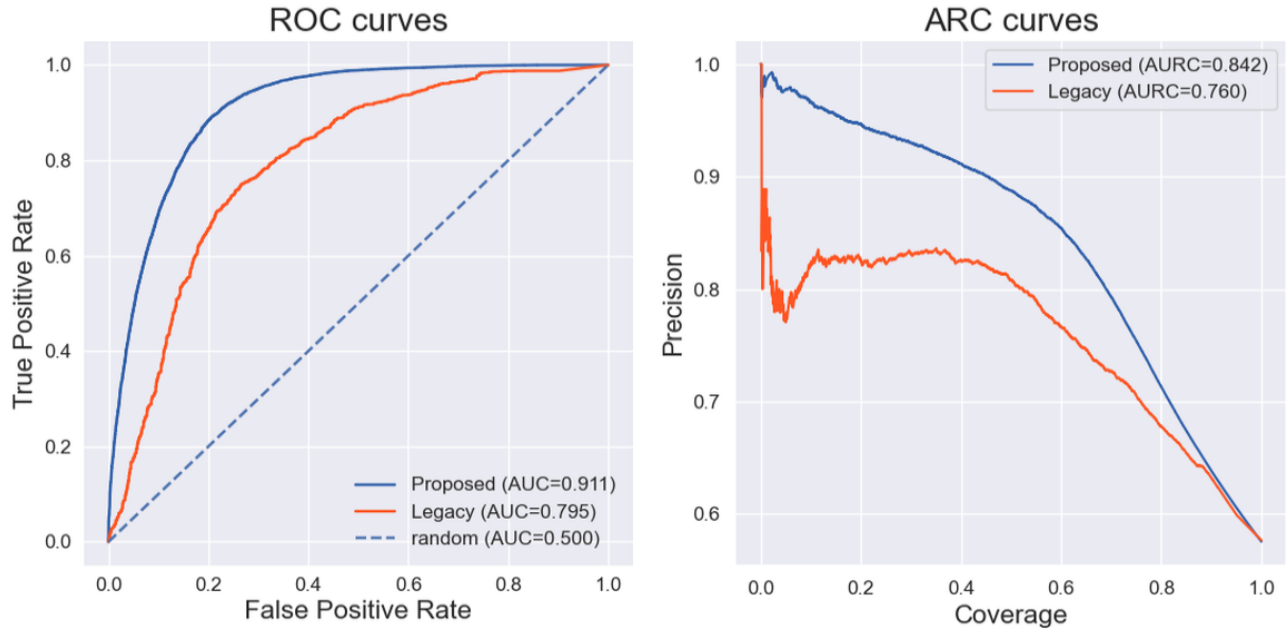


Fig. 4. The ROC and ARC curves showing the performance of our proposed solution in comparison with the legacy confidence. The confidence scores produced by our proposed method dominate the legacy confidence scores over all operating points by a clear margin.

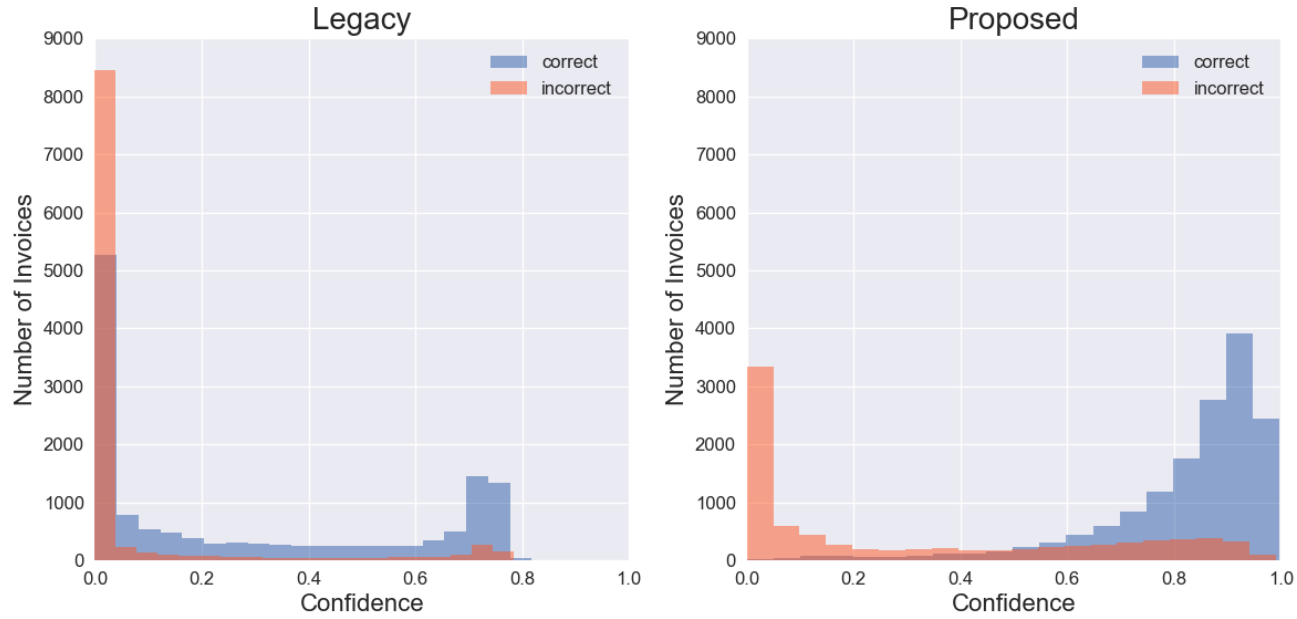


Fig. 5. The separation between correct and incorrect predictions for 24,000 invoices from the independent test set, using either legacy confidence scores or scores from our proposed method. The legacy method assigns very low confidence scores for most invoices, including over 5,000 invoices that were correctly predicted, and no invoices are assigned with a high confidence score. Our proposed method produces more refined separation.

between correct and incorrect predictions is seen in Fig. 5. By fixing the confidence threshold to correspond with the target precision currently used in production, the confidence estimates produced by our proposed method would increase coverage of automated invoices from 65.6% to 73.2%.

Finally, we show calibration performance in the form of a reliability diagram [32] in Fig. 6. We divided all confidence scores into 20 bins, where bin boundaries are selected adaptively so that all bins contain a roughly equal amount of predictions. In each bin, the average confidence score is compared to the fraction of positive samples (average accuracy) in the bin and shown in the diagram as deviation from zero (perfect calibration). We use bootstrap sampling to estimate confidence intervals as suggested in [32]. We measure calibration performance with *Adaptive Expected Calibration Error* (AdaECE) [19], since adaptive binning has been shown

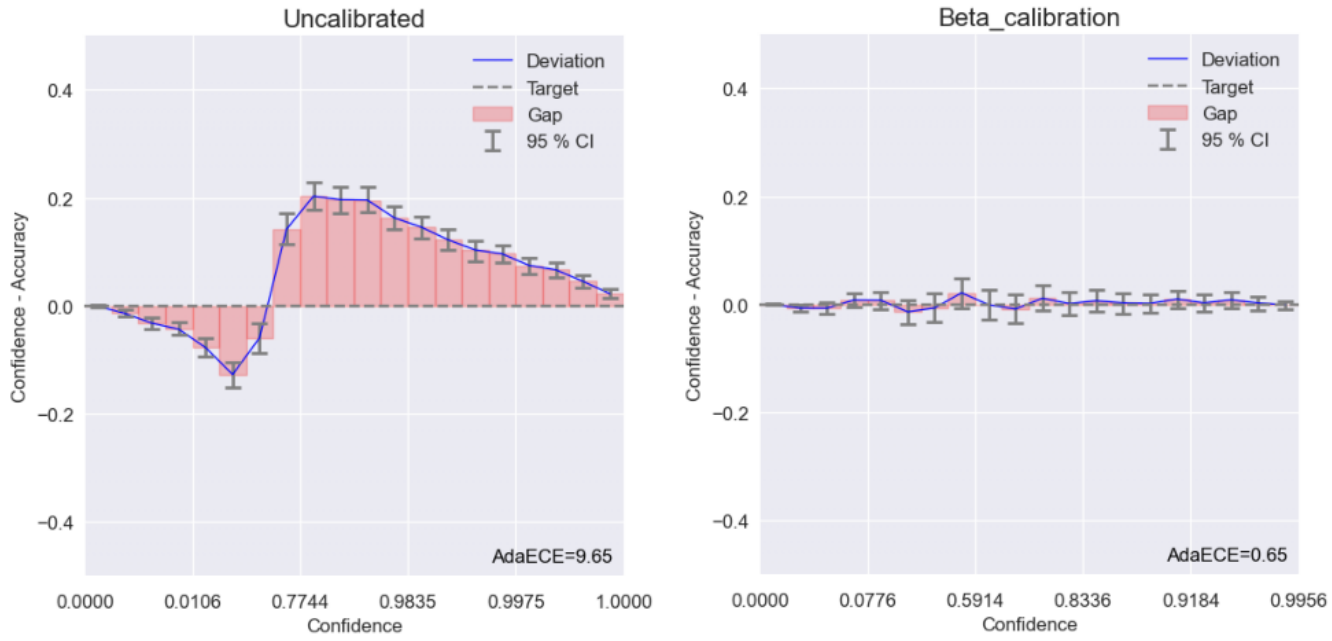


Fig. 6. The reliability diagrams and AdaECE (%) for the confidence scores before and after calibration. with a dynamic binning scheme using 20 equisize bins. The blue line plots in the reliability diagram mark the per-bin average accuracies and the thin gray lines mark the 95 % empirical confidence intervals (calculated using 1,000 bootstrap samples).

to be more robust [33] than equiwidth binning when measuring calibration error. It is defined as

$$AdaECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |A_m - C_m| , \tag{1}$$

where n is the total number of samples, $M = 20$ the number of bins, B_m the set of samples in bin m, and A_m and C_m the average accuracy and confidence of samples in bin m respectively.

This metric is best understood as a weighted average of binwise deviations (or *gaps*) $|A_m - C_m|$. It is minimized when all deviations equal zero, which signals perfect calibration. As is seen in Fig. 6, the Beta calibration mapping improves model calibration dramatically, resulting in near-perfect calibration.

Interpreting the significance of measured calibration error is rather tricky [32]–[34] since each binning strategy induces a distinctive bias in the results. One way around this problem is to approach calibration performance from a hypothesis-testing viewpoint. One suggested approximate test calculates empirical confidence intervals and rejects the hypothesis of perfect calibration if zero deviation is not included in all bins [32]. A visual inspection of Fig. 6 shows that our proposed method passes this test.

V. Comparison to related work

A. Failure Prediction and OoD Detection

A simple baseline method to assess whether an instance given to a softmax classifier should be labeled as abnormal (being either misclassified or OoD) is to rank the predictions on the basis of (uncalibrated) confidence scores and label an instance as abnormal if the related confidence score does not exceed a user set confidence threshold [24]. This approach is motivated by a finding that even though modern neural networks make high-confidence incorrect predictions, the distributions of confidences for correctly and incorrectly classified samples might still be different enough to make the separation between the two somewhat possible, at least in some cases [24]. The user is responsible for setting a suitable confidence threshold.

Several methods have improved upon this baseline since. In a method labeled ODIN [35], small perturbations are added to the training samples and *temperature scaling* [1] is applied on the softmax outputs. This method is used mainly as an OoD detector and is shown to gain significant improvements in many cases. A potentially complementary method is presented in [36], where the authors devise a new loss function to be used in training the base model. The proposed loss function requires OoD samples to be available at training time and the authors suggest using a generative adversarial network (GAN) to generate 'boundary' samples from the low-density areas of the training samples. Although both of these methods yield promising results in image classification, it is not obvious how one would apply them in the case of semi-structured documents.

Any method, which uses confidence scores as a criterion for whether to trust or reject a prediction, requires the separation of predictions into correct and incorrect classes to be *sharp* [37] enough, meaning that there is minimal overlap between the two classes. In this sense, our proposed method is shown to produce good quality confidence scores, although the overlap between the correct and incorrect predictions is still not completely eliminated.

B. Confidence estimation in information extraction

In IE, early approaches utilized hidden Markov Models and Conditional Random Fields in estimating confidence in the extracted tokens, fields, or even entire records [38]. More recently, techniques from natural language processing (NLP), such as recurrent neural networks have shown competitive results [8]. These approaches view documents as non-structured 1D sequences of serialized tokens. Important structural information, useful for extracting information from semi-structured documents, is thus discarded [10].

There has also been development to use techniques from computer vision (CV) to account for the spatial information in semi-structured documents [7], [10]. These techniques view 2D document understanding as an image segmentation task. However, in CV-based IE methods, predictive confidence is expressed on a pixel-by-pixel level. Although these pixel-wise confidence scores can be used in estimating segmentation quality [39], [40], this approach is less relevant in IE, where confidence should be expressed in relation to the extracted fields or entire documents instead of segmentation masks. The most recent approach, with current state-of-the-art results in semi-structured document understanding, is to combine techniques from both NLP and CV to train a transformer-based model with multi-modal representations of the semi-structured documents [9]. In this approach, confidence is expressed on a word token level.

Thus, all current methods leveraging techniques from CV require additional aggregation, if document-level confidence scores are needed. No prior published method for how such aggregation should be carried out exists to the best of our knowledge. Our proposed method performs this aggregation as part of its internal function.

C. Auxiliary models

There have been some studies, where an existing ML model has been augmented with an auxiliary part, which tries to assign confidence scores to predictions of the original model. This approach is partly motivated by several findings that a model might not be the best judge of its own trustworthiness [41]. A neural network performing multi-class classification is augmented with a confidence estimation branch in [42]. The confidence branch receives logits from the penultimate layer of the original model as inputs. It then processes these logits through one or more fully connected layers and is trained to output a scalar confidence estimate. This approach is shown to beat the baseline method presented in [24], when used in OoD detection. A similar approach is taken in [43], where an image recognition system consisting of convolutional and linear layers is augmented with a confidence network, which takes the feature maps produced by the convolutional layers as input and outputs a scalar confidence score. The confidence network is trained to output the probability of the true class. The method is shown to produce competitive results in failure prediction.

An auxiliary class used to represent misclassified samples is added to labels in a multi-class setting in [44]. The base model (a neural network) is first trained in the normal fashion to produce standard K-dimensional prediction vectors through a softmax function. Then, a new $(K+1)$-dimensional set of labels is generated, where each correctly predicted instance retains its original label and each incorrectly predicted instance is assigned the label $K+1$. Another lightweight model (a shallow neural network with a small number of neurons) is then trained using the logits of the base model as inputs and the augmented labels as targets. The authors show that this approach yields state-of-the-art calibration performance in several multi-class classification tasks, while also providing better separation between the correctly and incorrectly predicted samples.

Perhaps the most similar approach to current work is taken in [45], where the authors inject *linear classifier probes* between the hidden layers of a ResNet model used for image classification. These probes are essentially logistic regression models trained in the same task as the base model, as if the layer preceding them, was the final layer. Thus, each probe outputs a prediction vector with the number of elements corresponding to the number of classes included in the task. The prediction vectors, along with the prediction vector produced by the base model are concatenated as a single feature vector. This feature vector is then given as an input to a *meta-model*, which is trained to produce a confidence score, reflecting the probability that the base model made the right prediction.

Although all these methods overlap with the method presented in this study, they also differ in several ways. In all of the above cases, the base models perform multi-class classification and output probability distributions through softmax activations. In [42], [43] and [44] the auxiliary part is then used to modify these softmax outputs in one way or another, and thus they are not applicable since the base model in this case study does not perform multi-class classification. Furthermore, the method used in [42] requires changes in the training process of the base model, which does not satisfy our desiderata of an unintrusive solution. Also, due to computational restraints, an approach similar to [45] would have been hard to implement, since the outputs of the internal layers of the image segmentation models in the DPL are huge in size. Finally, the solution presented in this paper differs from all of the above by instead of looking at the internal workings of a single base model, it compares latent representations and summary statistics from an ensemble of models.

VI. Discussion

Although our proposed method is described in the context of a specific case, the method can easily be generalized to any ensemble, which generates latent representations for the input data during inference, as long as it is reasonable to assume that the resemblance of these representations to one another can be taken as an indicator for the successfulness of the inference. For ensembles used in classification, averaging out results of the individual models has been shown to yield calibrated confidence scores, which are robust even under data shift [3]. However, uncertainty estimation for other types of ensembles

is less straightforward, and novel methods for this purpose are needed. We seek to bridge this gap with our proposed auxiliary hybrid method.

Since our proposed method is unintrusive, it is straightforward to apply it to models that have already been deployed as well. As such, it offers a simple and low-cost solution to increase the utility and resilience of the base model since no decommissioning of the base model is needed for including the calibrated uncertainty estimates. On the most general level, we promote the idea of auxiliary confidence estimators probing the internal workings of a base model. Decoupling the confidence estimation from the base model adds modularity to the design process with all of its benefits [46]. The designers of the base model can use all their efforts in building the best possible model, and questions related to confidence estimation can be dealt with separately.

Our proposed model, although consisting of multiple parts, can be considered lightweight. Each of the five convolutional nets consists of 2.4 million parameters and the pickled XGBoost model takes roughly 30 Mb of hard disk space. The Beta calibration mapping consists of only three stored parameters. The most time-consuming operation for the confidence estimator pipeline is building the augmented feature vectors. The added overhead from inference is negligible when compared to this. Thus, an obvious improvement to speed up training would be to parallelize the operations needed in building the augmented feature vectors. On the other hand, during inference, invoices arrive one at a time and thus no speed-up is generated via parallelizing. Currently, training the entire confidence estimation pipeline (including Bayesian optimization) takes less than a day with a single RTX 3060 GPU with 12 GB RAM. Processing a single invoice through the confidence estimation pipeline during inference takes roughly 0.1 seconds. Thus, the added overhead from confidence estimation by our proposed model can be considered minimal.

Since the ground truth (GT) label values are a result of human labor, they contain some errors with varying amounts for different fields. In this work, we chose to ignore these GT errors. However, we make the point that our proposed method could be used to locate potential GT errors by inspecting invoices not aligning with GT, but still receiving a high confidence score. These high confidence predictions labeled as 'incorrect' are present in Figure 5 with both the legacy confidences and our proposed method and are the cause of fluctuation in the left ends of the ARC curves in Figure 4. The results presented in Chapter IV would likely improve significantly if GT errors were cleaned.

Although confidence calibration is not strictly required in selective classification, the statistical analysis needed in setting a suitable rejection threshold is simplified. Furthermore, if the confidence scores are not calibrated, a new threshold value needs to be decided each time there is a significant shift in the data distribution, or if the model is retrained. Some downstream tasks, such as model monitoring, can also be done in a more principled manner if the scores are calibrated. We will delve deeper into these topics in future work.

VII. Conclusion

In this work, we presented a case study, where the uncertainty estimates of an information extraction model deployed in an industrial setting were improved. Our proposed solution was to train an auxiliary model, which uses latent representations and statistics from the base model as inputs. The auxiliary model then learns to associate each prediction of the base model with a calibrated confidence estimate. The proposed solution was shown to produce confidence estimates, which outperform the legacy estimates over all possible confidence thresholds. This was shown to increase coverage of automated invoices from 65.6% to 73.2% without increasing the error rate.

References

[1] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, pp. 1321–1330.

[2] A. M. Alexandari, A. Kundaje, and A. Shrikumar, "Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[3] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[4] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *arXiv preprint arXiv:1606.06565*, 2016.

[5] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[6] H. Linusson, "Nonconformity measures and ensemble strategies : An analysis of conformal predictor efficiency and validity," Ph.D. dissertation, Stockholm University, Department of Computer and Systems Sciences and Department of Information Technology, University of Borås, 2021.

[7] O. Bensch, M. Popa, and C. Spille, "Key information extraction from documents: Evaluation and generator," *arXiv preprint arXiv:2106.14624*, 2021.

[8] X. Holt and A. Chisholm, "Extracting structured data from invoices," in *Proceedings of the Australasian Language Technology Association Workshop 2018*, Dunedin, New Zealand, Dec. 2018, pp. 53–59.

[9] Y. Huang, T. Lv, L. Cui, Y. Lu, and F. Wei, "Layoutlmv3: Pre-training for document ai with unified text and image masking," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4083–4091.

[10] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, and J. B. Faddoul, "Chargrid: Towards understanding 2d documents," *arXiv preprint arXiv:1809.08799*, 2018.

[11] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2002.

[12] M. Kull, T. S. Filho, and P. Flach, "Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 623–631.

[13] M. Kull, M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[14] A. Kumar, P. S. Liang, and T. Ma, "Verified uncertainty calibration," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[15] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press, 2015, pp. 2901–2907.

[16] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, pp. 61–74, 03 1999.

[17] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 609–616.

[18] A. Kumar, S. Sarawagi, and U. Jain, "Trainable calibration measures for neural networks from kernel mean embeddings," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 2805–2814.

[19] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. Torr, and P. Dokania, "Calibrating deep neural networks using focal loss," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 15 288–15 299.

[20] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[21] S. Seo, P. H. Seo, and B. Han, "Learning for single-shot confidence calibration in deep neural networks through stochastic inferences," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9022–9030.

[22] J. Zhang, B. Kailkhura, and T. Y.-J. Han, "Mix-n-match : Ensemble and compositional methods for uncertainty calibration in deep learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 11 117–11 128.

[23] K. Hendrickx, L. Perini, D. Van der Plas, W. Meert, and J. Davis, "Machine learning with a reject option: A survey," *arXiv preprint arXiv:2107.11277*, 2021.

[24] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations*, 2017.

[25] J. Moon, J. Kim, Y. Shin, and S. Hwang, "Confidence-aware learning for deep neural networks," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 7034–7044.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[27] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[28] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[29] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, rOC Analysis in Pattern Recognition.

[30] A. Fernández, S. García, M. Galar, R. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Springer International Publishing, 2018.

[31] M. S. A. Nadeem, J.-D. Zucker, and B. Hanczar, "Accuracy-rejection curves (arcs) for comparing classification methods with a reject option," in *Machine Learning in Systems Biology*. PMLR, 2009, pp. 65–81.

[32] J. Vaicenavicius, D. Widmann, C. Andersson, F. Lindsten, J. Roll, and T. Schön, "Evaluating model calibration in classification," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 3459–3467.

[33] J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran, "Measuring calibration in deep learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[34] N. Posocco and A. Bonnefoy, "Estimating expected calibration errors," in *International Conference on Artificial Neural Networks*. Springer, 2021, pp. 139–150.

[35] S. Liang, Y. Li, and R. Srikant, "Principled detection of out-of-distribution examples in neural networks," *arXiv preprint arXiv:1706.02690*, 2017.

[36] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[37] T. Gneiting, F. Balabdaoui, and A. E. Raftery, "Probabilistic forecasts, calibration and sharpness," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 69, no. 2, pp. 243–268, 2007.

[38] A. Culotta and A. McCallum, "Confidence estimation for information extraction," in *Proceedings of HLT-NAACL 2004: Short Papers*, ser. HLT-NAACL-Short '04. USA: Association for Computational Linguistics, 2004, p. 109–112.

[39] K. Hoebel, V. Andrearczyk, A. Beers, J. Patel, K. Chang, A. Depeursinge, H. Müller, and J. Kalpathy-Cramer, "An exploration of uncertainty information for segmentation quality assessment," in *Medical Imaging 2020: Image Processing*, vol. 11313. SPIE, 2020, pp. 381–390.

[40] A. Mehrtash, W. M. Wells, C. M. Tempany, P. Abolmaesumi, and T. Kapur, "Confidence calibration and predictive uncertainty estimation for deep medical image segmentation," *IEEE transactions on medical imaging*, vol. 39, no. 12, pp. 3868–3878, 2020.

[41] H. Jiang, B. Kim, M. Y. Guan, and M. Gupta, "To trust or not to trust a classifier," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 5546–5557.

[42] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[43] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[44] Z. Shao, J. Yang, and S. Ren, "Calibrating deep neural network classifiers on out-of-distribution datasets," *arXiv preprint arXiv:2006.08914*, 2020.

[45] T. Chen, J. Navrátil, V. Iyengar, and K. Shanmugam, "Confidence scoring using whitebox meta-models with linear classifier probes," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1467–1475.

[46] C. Y. Baldwin and K. B. Clark, *Design rules: The power of modularity*. MIT press, 2000, vol. 1.