Context-Aware Feature Selection and Classification

Juanyan Wang and **Mustafa Bilgic**
Illinois Institute of Technology, Chicago, IL, USA
jwang245@hawk.iit.edu, mbilgic@iit.edu

Abstract

We propose a joint model that performs instance-level feature selection and classification. For a given case, the joint model first skims the full feature vector, decides which features are relevant for that case, and makes a classification decision using only the selected features, resulting in compact, interpretable, and case-specific classification decisions. Because the selected features depend on the case at hand, we refer to this approach as *context-aware feature selection and classification*. The model can be trained on instances that are annotated by experts with both class labels and instance-level feature selections, so it can select instance-level features that humans would use. Experiments on several datasets demonstrate that the proposed model outperforms eight baselines on a combined classification and feature selection measure, and is able to better emulate the ground-truth instance-level feature selections. The supplementary materials are available at https://github.com/IIT-ML/IJCAI23-CFSC.

1 Introduction

A barrier to using highly accurate machine learning algorithms for decision support is their opacity [De Laat, 2018]. While opacity might be acceptable for certain tasks, such as handwritten digit recognition, when machine learning systems are used to assist humans in decision making, the algorithms must be able to explain their decision making processes and these explanations need to make sense to humans in order to be useful. For instance, when a machine learning system is employed for loan decisions, the reasons for approving or rejecting a loan must be explained in a human-understandable form to the loan officer (who will make the final decision based on the system's recommendations and explanations), to the applicant (who is directly impacted by the decision), to the developer (for debugging), and to the regulators (to ensure the system's decisions are not discriminatory) [Arya *et al.*, 2019].

While being interpretable and able to explain its decisions are important and necessary, they are not sufficient for a model to be effective in a decision support system. For example, a logistic regression model is assumed to be interpretable. However, when the model uses thousands of features, the explanations are rarely useful as inspecting the relative impact of all relevant features will be overwhelming for the stakeholders (decision makers, users, etc.). Likewise, a decision tree, while considered interpretable, is not necessarily useful for decision support. One can impose sparsity, such as using L_1 regularization for logistic regression, or a depth limit on the decision tree. While sparsity can make the models simpler, it often does so by prioritizing common features that have the greatest impact on the most number of objects (e.g., the word 'movie' tends to be a common and a statistically negative term in a movie review classification task), which may not be the most meaningful features for stakeholders [Lage *et al.*, 2019].

Additionally, when people make decisions, they tend to quickly scan all available information and then focus on few factors that are relevant for the case at hand [Shrestha *et al.*, 2019]. For instance, when loan officers review a loan application, they skim the entire application and then focus on what is relevant for the case. While all information is used, the income and the credit score might be the determining factors for one application while the number of missed payments might be the determining factor for another [Purohit *et al.*, 2012]. The human decision maker is essentially performing what we call *context-aware feature selection and classification*: skimming the full feature set, focusing on the features that are relevant for the current case, and making a classification decision using only the selected features.

Context-aware feature selection by machine learning models is limited. Traditional feature selection methods such as L_1 regularization and filter methods perform 'global' feature selection where the same set of features are used for all objects, as opposed to context-aware feature selection where the selected features depend on the object itself. While decision trees perform context-aware feature selection, the rules provided by decision trees are hierarchical, and hence some features, like the root and the features close to the root, will repeatedly be selected. They are also not necessarily accurate in learning the complex relationships between data points and features. Rule-based systems perform context-based feature selection, but they also tend to have low accuracy [Molnar, 2020]. Attention-based neural networks [Bahdanau *et*

al., 2014] can perform context-aware feature selection and be highly accurate. However, the selected features are not guaranteed to be meaningful to humans. We discuss related work in more detail in the next section.

When explanations that are meaningful to humans are desired, one approach is to ask for supervision from humans on which features they focused for each classification decision. We refer to these as instance-level feature labels, as opposed to global feature labeling [Melville and Sindhwani, 2009; Das *et al.*, 2013]. Eliciting instance-level features from humans requires extra time, effort, and cost; hence, instance-level feature labeling is practical for only domains where human-like explanations and decisions are desired. We experiment with both fully-automated baselines that do not need instance-level feature labels as well as with the ones that can use them if available.

Our contributions in this paper include:

- We introduce and formalize the *context-aware feature selection and classification* problem.
- We propose a context-aware feature selection and classification model that jointly utilizes class labels and instance-level feature selection annotations.
- We conduct experiments to empirically compare the proposed model to a number of baselines on several datasets, comparing them using both classification performance and feature selection performance.

The rest of the paper is organized as follows. We discuss related work in the next section. We present our model in Section 3. We discuss the experimental methodology in Section 4. We discuss our findings in Section 5 and followed by a discussion of limitations in Section 6, and then conclude.

2 Related Work

2.1 Feature Selection

Traditional feature selection methods can be broadly categorized into three: i) filter methods use a feature importance measure such as feature correlation [Hall, 2000; Yu and Liu, 2003] and mutual information [Gao *et al.*, 2016], to rank and select features; ii) wrapper methods that iteratively search for the best set of features for a given model [Kohavi and John, 1997; Arai *et al.*, 2016]; iii) and, methods that embed the feature selection into the learning process, such as decision trees, rule-based systems, and L_1-regularized models. We used decision tree and L_1 regularized logistic regression as two baselines in our experiments.

2.2 Rule-Based Systems

Rule-based systems have been extensively used for decision support [Adriaenssens *et al.*, 2004; Seerat and Qamar, 2015]. They were preferred for their interpretability. Approaches include OneR that created rules with one feature [Holte, 1993], IREP that used a combination of pre-pruning and post-pruning [Fürnkranz and Widmer, 1994], RIPPER that used rule pruning to optimize the rule set in a post-processing phase [Cohen, 1995], and Bayesian Rule Lists [Letham *et al.*, 2015]. We used RIPPER as a baseline in our experiments.

2.3 Learning with Rationales

A closely related area is *learning with rationales* which asks annotators to highlight segments of the text per document as 'rationales' for their labeling decisions. Zaidan et al. [2007] converted the rationales into constraints for training support vector machines. Sharma and Bilgic [2018] presented a method to manipulate feature weights in the training of off-the-shelf classifiers. Recent deep learning-based approaches on incorporating rationales either generated rationale-augmented representations of text [Zhang *et al.*, 2016] or utilized the rationales for richer supervision [Barrett *et al.*, 2018; Wang *et al.*, 2022]. Although the main purpose of these methods was to improve classification performance, instead of performing feature selection, some of them can be adapted to perform context-aware feature selection. We adapted Barrett et al. [2018]'s BiLSTM-based method as one of the baselines.

2.4 Model Interpretability

Several papers worked on generating explanations for complex or black-box models. For example, Ribeiro et al. [2016] replaced the underlying complex model with a surrogate model, Lundberg and Lee [2017] computed Shapley values as feature importance, Wachter et al. [2017] used examples for explanations, Li et al. [2015] computed input saliency for neural networks. Our approach differs from most of these post-processing methods as it selects case-specific features that are certainly used by the model for predicting a specific instance. While several papers used attention mechanism for interpretability [Wang *et al.*, 2016; Ghaeini *et al.*, 2018], other papers pointed out that attention weights often reflect how much the model attend to the hidden representation of each input, which might already have mixed in information from other inputs [Bastings and Filippova, 2020], and are not stable indicators for interpretability [Jain and Wallace, 2019; Serrano and Smith, 2019].

Several papers also incorporated interpretability into the approach itself. For example, rationalized neural network [Lei *et al.*, 2016] and causality-based approaches [Narendra *et al.*, 2018; Harradon *et al.*, 2018]. Closest to our task is rationalized neural network [Lei *et al.*, 2016]; they extracted short and continuous 'rationales' from each document for classifying and rationalizing the document. We adapted this approach as a baseline in our experiments.

3 Context-Aware Feature Selection and Classification (CFSC)

We are given a dataset $\mathcal{D}$ whose members are triplets $\langle \boldsymbol{x}_i, y_i, \boldsymbol{a}_i \rangle$ where $\boldsymbol{x}_i \in \mathcal{R}^m$ is an m dimensional input vector, $y_i \in \{c_1, c_2, ..., c_q\}$ is a discrete variable representing $\boldsymbol{x}_i$'s class label, and $\boldsymbol{a}_i \in \{0, 1\}^m$ is $\boldsymbol{x}_i$'s feature label indicating which features are used by a human in making the classification decision y_i for the instance $\boldsymbol{x}_i$.

The objective is to train a model $f : \boldsymbol{x}_i \rightarrow \langle y_i, \boldsymbol{a}_i \rangle$ that can generalize to unseen data points $\boldsymbol{x}_j$ and correctly predict both the label y_j and the feature selections $\boldsymbol{a}_j$ for $\boldsymbol{x}_j$. Predicting $\boldsymbol{a}_j$ is equivalent to solving a multi-label classification problem as each entry indicates if feature k is selected,

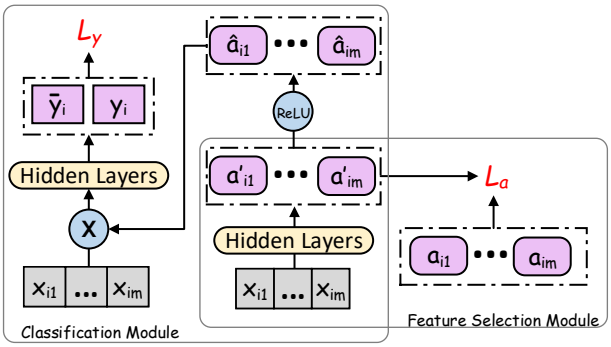

Figure 1: The architecture for the CFSC (**C**ontext-Aware **F**eature **S**election and **C**lassification) approach.

$a_{jk} \in \{0, 1\}$. Note that while f has access to the full triplet $\langle \boldsymbol{x}_i, y_i, \boldsymbol{a}_i \rangle$ during training time, f has access to only $\boldsymbol{x}_j$ at test time and not $\boldsymbol{a}_j$. Hence, at test time f must predict both $\boldsymbol{a}_j$ and y_j. We refer to this task as a *context-aware feature selection and classification* problem. The step of predicting $\boldsymbol{a}_j$ is the *context-aware feature selection* where the feature selection decisions, $\boldsymbol{a}_j$, depend on the particular instance $\boldsymbol{x}_j$, as opposed to global feature selection methods. Moreover, to ensure sparsity and interpretability, we require the predicted $\boldsymbol{a}_j$ to have exactly 0 for features that are not selected for predicting y_j of $\boldsymbol{x}_j$, and the step of predicting y_j should use only the selected features.

We propose the neural network model depicted in Figure 1 and refer to it as *Context-aware Feature Selection and Classification*, or CFSC in short. The model contains a context-aware feature selection module and a classification module. It is first trained to predict $\boldsymbol{a}_i$ using $\boldsymbol{x}_i$, without initially worrying about y_i, and then fine-tuned to predict $\boldsymbol{a}_i$ and y_i jointly.

The feature selection module processes the input $\boldsymbol{x}_i$ through an optional number of hidden layers. The hidden layer part of this module can be as simple or complicated as needed, such as a simple fully-connected dense layer, or a deep neural network consisting of self-attention and dense layers. The output of this process is $\boldsymbol{a}'_i$. $\boldsymbol{a}'_i$ are not constrained to be between 0 or 1 and they are not constrained to be sparse just yet, as sparsity will be imposed in the next step. Hence, we use the identity activation at these nodes at this stage.

To train the feature selection module using human-provided triplets $\mathcal{D} = \{\langle \boldsymbol{x}_i, y_i, \boldsymbol{a}_i \rangle\}$, we formulate the objective as a multi-label classification task. First, we pass each a'_{ik} through a sigmoid of the form $1/(1 + e^{-a'_{ik}})$. The weights of the feature selection module, $\mathbf{W}_a$, are then trained using binary cross entropy loss. Let this loss be L_a. This training process imposes that for a feature that should be selected, i.e., for $a_{ik} = 1$, the a'_{ik} needs to be positive, and a'_{ik} needs to be negative for features where the ground truth is $a_{ik} = 0$.

For predicting y_i, we need to use only the features that are selected by the feature selection module. Because the output of the feature module, a'_{ik}, are real-valued, where $a'_{ik} > 0$ indicates if a feature should be selected, we pass the $\boldsymbol{a}'_i$ vector through a ReLU function. That is, $\hat{\boldsymbol{a}}_i = \text{ReLU}(\boldsymbol{a}'_i)$, guaranteeing that features for which $a'_{ik} < 0$ will have exactly zero values and $\hat{\boldsymbol{a}}_i$ acts as a mask function, performing feature selection.

$\boldsymbol{x}_i$ is first multiplied with the predicted feature mask $\hat{\boldsymbol{a}}_i$, $\boldsymbol{x}_i \otimes \hat{\boldsymbol{a}}_i$, which is then passed through a number of optional hidden layers and then used for predicting y_i. This overall process is meant to mimic the human decision-making process where the full feature vector $\boldsymbol{x}_i$ is first skimmed to decide which features are relevant for the case at hand ($\hat{\boldsymbol{a}}_i$), and only those feature values ($\boldsymbol{x}_i \otimes \hat{\boldsymbol{a}}_i$) are used for predicting y_i. Cross entropy is used as the classification loss using the predicted $\bar{y}_i$ and the ground truth y_i. Let this loss be L_y. The overall loss L is a weighted combination of L_y and L_a:

$$L = \lambda_a L_a + (1 - \lambda_a) L_y \tag{1}$$

where λ_a is the weight of the feature loss. The weights of feature selection module, $\mathbf{W}_a$, are trained to optimize L_a first, and then the weights of the full model, $\mathbf{W} = \mathbf{W}_a \cup \mathbf{W}_y$, are jointly trained to optimize the combined loss L.

4 Experimental Methodology

We conduct experiments to compare the proposed CFSC method to several baselines on both classification and feature selection performance. In this section, we describe the baselines, the datasets, the two simulated experts, the combined classification and feature selection measure, the density measure, and the parameter settings.

4.1 Baselines

To the best of our knowledge, no existing paper directly addresses the context-aware feature selection and classification problem for tabular data, except decision trees and rule-based systems. However, several approaches can be adapted to perform this task. We modified and experimented with an attention-based BiLSTM model [Barrett *et al.*, 2018], a rationalized neural network [Lei *et al.*, 2016], a pipeline model, and a global feature selection model.

Attention-based Bi-directional LSTM (ATT-FL). This is the main baseline that is closest to our approach and utilizes both $\boldsymbol{a}_i$ and y_i during training. This baseline is based on the method by Barrett et al. [2018] for text classification. They regularized the attention layer of a Bi-directional LSTM model using 'estimated' human attention from an eye tracking corpora. We adapted their method to be used for vector-based data as follows: each instance $\boldsymbol{x}_i$ is passed through a hidden layer, followed by a Bi-LSTM layer, an attention layer, and finally the classification layer. The softmax in the attention layer is replaced with sparsemax [Martins and Astudillo, 2016] to ensure sparsity and enable feature selection. We refer to this method as ATT-FL. For comparison and to evaluate how much the human-provided feature labels help, we also present results with the fully-automated version of this method that still performs context-aware feature selection but does not need feature labels; we refer to this as ATT.

Rationalizing Neural Predictions (RNP). This method is based on the work of Lei et al. [2016]. It used a generator to extract short and continuous 'rationales' for text classification, where rationales are pieces of text from the document for classifying the document. We adapted this method to a

vector-based domain as follows: each instance $\boldsymbol{x}_i$ is passed through a generator consisting of two hidden layers and one output layer for rationale selection first. The selected features pass through an encoder consisting of one hidden layer and one output layer for classification. Following the implementation of Lei et al. [2016], we used gumbel-softmax activation function [Jang *et al.*, 2016] coupled with L_1 penalty for rationales in the loss function to impose sparsity.

Logistic Regression Pipeline (`LR-PL`). In contrast to our joint learning approach, one simple baseline is to build a pipeline where one logistic regression model is trained for feature selection and another logistic regression model is trained for classification separately. The first model is trained to perform $f_1 : \boldsymbol{x}_i \rightarrow \boldsymbol{a}_i$. The second model is trained to perform $f_2 : \boldsymbol{x}_i \otimes \bar{\boldsymbol{a}}_i \rightarrow y_i$ where $\bar{\boldsymbol{a}}_i$ is the predicted binarized feature labels. At test time, f_1 is used to predict $\boldsymbol{x}_j \rightarrow \boldsymbol{a}_j$ and then f_2 is used to predict $\boldsymbol{x}_j \otimes \bar{\boldsymbol{a}}_j \rightarrow y_j$.

We also experimented with three classification algorithms that simultaneously perform feature selection. A decision tree classifier (`DT`), a rule-based learner (`RL`) based on RIPPER [Cohen, 1995](we trained two rule-based learners for binary classification: one aimed at predicting the positive class as `RL-P` and the other for the negative class as `RL-N`), and a L_1-regularized logistic regression model (`LR`). Decision tree classifier and rule-based learner perform context-aware feature selection, whereas the L_1-regularized logistic regression model performs global feature selection. Finally, we include a feed-forward neural network (`FF`) that acts as a baseline for only the classification performance.

4.2 Datasets

We experimented with five real-world and three synthetic datasets. The **Credit** [Goyal, 2020] dataset contains 3,254 bank credit card customers with 37 features and binary labels indicating if the customer is an 'Attrited Customer.' The **Company** [Zieba *et al.*, 2016] dataset has 4,182 companies with 64 features and binary labels indicating whether the company bankrupted within the forecasting period. The **Mobile** [Sharma, 2017] dataset contains 2,000 mobile phone data with 20 features and binary labels indicating if the price of a phone is in the high cost range. The **NHIS** [CDC, 2017] dataset has 2,306 adult survey data with 144 features and binary labels indicating if the person is suffering from chronic obstructive pulmonary disease. The **Ride** [City of Chicago, 2019] dataset has 4,800 ride trip records with 46 features and binary labels indicating if the trip is shared with other persons. We chose these real-world datasets because the domains are relatively easy for the laypeople, as opposed to more specialized domains. These datasets did not contain instance-level feature labels. Hence, similar to work on generating synthetic explanations [Ribeiro *et al.*, 2018; Guidotti, 2021], we created simulated experts, which we discuss in detail in the next subsection.

We created three synthetic datasets containing instance-level feature labels where the first two are for binary classification and the third one is a multi-class classification task. We first generated the input data by allocating each class a normally-distributed cluster of points. We trained a shallow decision tree based on the original data and then reassigned class labels based on the predictions of the decision tree. **Synthetic1** contains 1,000 instances with 5 features whereas **Synthetic2** contains 1,500 instances with 10 features. **Synthetic3** contains 3,024 instances with 20 features and four classes. For each $\boldsymbol{x}_i$, the a_{ik} is 1 for the features used in the decision path, and 0 otherwise. To experiment with the datasets with different settings, we kept the root node in the decision path and hence one feature was always 'on' for Synthetic1 and Synthetic3 datasets, whereas for Synthetic2 dataset, we removed the root node from the decision path and hence no feature was always 'on.'

4.3 Simulated Expert

We used two strategies to simulate experts that can provide instance-level feature labels. The first strategy is based on the evidence counterfactual method [Moeyersoms *et al.*, 2016]. A logistic regression model is trained on a given dataset. For object $\boldsymbol{x}_i$, let the predicted label be y and w_{yk} be the coefficient of feature k for class y. Starting with the least important feature for $\boldsymbol{x}_i$ (i.e., the feature that has the lowest $|w_{yk} \times x_{ik}|$), features are removed one by one until the predicted label changes. The top features, including the last one that caused a label flip, are retained as the justification of the classification decision. Because features are ranked by $|w_{yk} \times x_{ik}|$ and not simply by $|w_{yk}|$, this process performs context-aware feature selection, rather than global feature selection.

The second strategy uses a decision tree to generate the instance-level feature labels. First, a decision tree on a given dataset is trained. Then, at prediction time, the features that are used in the decision path for classifying $\boldsymbol{x}_j$ are marked as 'on' and the rest are marked as 'off.'

4.4 Evaluation Measures

One can use traditional evaluation measures, such as accuracy and F_1, to evaluate the classification performance. To evaluate the performance of the context-aware feature selection, which is a multi-label classification task, measures such as hamming loss and subset accuracy can be used. However, these measures are crude and inadequate for transparency purposes; while some features are frequently used, others are never used, and some are used rarely. Summarizing everything up in a single measure would not portray the whole picture. Therefore, we introduce a more granular evaluation approach for evaluating context-aware feature selection.

Let $\mathcal{T}$ be the evaluation set, $\mathcal{A}_\delta^{\mathbf{on}}$ be the features that are used for all instances in $\mathcal{T}$: $\mathcal{A}_\delta^{\mathbf{on}} = \{k \mid a_{ik} = 1 \text{ for } \forall \boldsymbol{x}_i \in \mathcal{T}\}$, and $\mathcal{A}_\delta^{\mathbf{off}}$ be the set of features that are never used for any of the objects: $\mathcal{A}_\delta^{\mathbf{off}} = \{k \mid a_{ik} = 0 \text{ for } \forall \boldsymbol{x}_i \in \mathcal{T}\}$. Let $\mathcal{A}_\nu$ represent the rest of the features, i.e., the features that are used for some objects but not for others. We use accuracy separately for the features in $\mathcal{A}_\delta^{\mathbf{on}}$ and for features in $\mathcal{A}_\delta^{\mathbf{off}}$. For features in $\mathcal{A}_\nu$, however, because the ground-truth $\boldsymbol{a}_j$ tend to be sparse for human-interpretable decisions, a measure for imbalanced classes, such as F_1, is more appropriate. For feature $k \in \mathcal{A}_\nu$ we first calculate F_{1k}, and take a weighted average as ${F_1}^{\mathbf{w}}$, where each F_{1k} is weighted based on the frequency the feature k is 'on.'

In addition to individual classification and feature selection

measures, we also present a linear combination of the two:

$$M(y, \bar{y}, \boldsymbol{a}, \bar{\boldsymbol{a}}) = \gamma_a \times M_a(\boldsymbol{a}, \bar{\boldsymbol{a}}) + (1 - \gamma_a) \times M_y(y, \bar{y}) \quad (2)$$

where $\bar{\boldsymbol{a}}$ is the binarized $\hat{\boldsymbol{a}}$, $M_a(\boldsymbol{a}, \bar{\boldsymbol{a}})$ refers to the evaluation of the context-aware feature selection, and $M_y(y, \bar{y})$ refers to the evaluation of the classification decisions, and γ_a controls the relative importance of the feature selection measure.

Finally, we introduce a new measure aimed at understanding how 'dense' each context-aware classification decision is. We calculate the number of features used per instance, on average, in a given dataset:

$$\textbf{Density} = \frac{1}{|\mathcal{T}|} \sum_{\boldsymbol{x}_j \in \mathcal{T}} \sum_{k=1}^{m} a_{jk} \quad (3)$$

As an example, assume a domain with 20 features, that there are 100 instances in the evaluation set, and the model uses 2 features for classifying 50 instances, 3 features for 25 instances, 4 features for 15 instances, and 5 features for the remaining 10 instances. The density for such a model would be $(50 \times 2 + 25 \times 3 + 15 \times 4 + 10 \times 5)/100 = 2.85$.

4.5 Model Structures and Parameter Settings

CFSC has one hidden layer with 16 units for the classification module and two hidden layers with 64 and 256 units respectively for the feature selection module. The ATT-FL model has one hidden layer with 64, one BiLSTM layer with 32, and one attention layer with 256 units. The RNP model has one hidden layer with 16 units for the classification module and two hidden layers with 64 and 256 units respectively for the feature selection module. The FF model has one hidden layer with 16 units. We used the same structures for all datasets and did not perform structure search.

Hyper-Parameter Tuning. For each dataset, we use $1/3$ of the data as the test set and perform 5-fold validation on the rest of the data where one fold is used for validation and four folds are used for training. We set γ_a to 0.5 (Equation 2) for all models[1]. We performed grid search with cross validation to optimize all the other tunable hyper-parameters of each method using the combined measure on the validation set. We provide the range of all tunable parameters for each method in the supplementary materials for reproducibility.

5 Results

We first compare CFSC to the baselines on a combined classification and feature selection measure[2]. Then, we conduct a deep dive analysis of the instance-level feature selection of three methods. Finally, we conduct an ablation study to investigate different parameter settings for γ_a and λ_a.

5.1 Combined Performance Measures

Tables 1 and 2 present the combined classification F_1 and feature selection F_1 on eight datasets under the counterfactual and decision-tree expert strategies respectively. All experimental results are reported by taking an average over five different runs, computed over the five-fold validation splits. We compare CFSC with all baselines using the combined measures computed by Equation 2, with $\gamma_a = 0.5$, which balances equally between the F_1 for classification and the weighted $F_1^{\mathbf{w}}$ for feature selection. Note that CFSC and all the baselines except FF tuned their hyper-parameters to maximize the equally-balanced and combined measures.

	FF	LR	DT	RL-P	RL-N	ATT	RNP	LR-PL	ATT-FL	CFSC
Credit	.707	.680	.701	.576	.538	.549	.675	.622	.745	**.785**
Company	.589	.480	.456	.174	.336	.506	.567	.328	.608	**.701**
Mobile	.853	.852	.852	.716	.715	.779	.842	.785	.903	**.907**
NHIS	.606	.598	.500	.448	.477	.497	.550	.685	.596	**.781**
Ride	.679	.675	.671	.492	.540	.591	.659	.573	.706	**.765**

Table 1: Comparison between CFSC and baselines using the combined measure. Feature labels are generated via the evidence counterfactual strategy. CFSC significantly outperformed all baselines, except being comparable to ATT-FL on the Mobile dataset.

	FF	LR	RL-P	RL-N	ATT	RNP	LR-PL	ATT-FL	CFSC
Credit	.796	.772	.609	.626	.631	.802	.868	.829	**.945**
Company	.860	.751	.170	.335	.769	.808	.806	.821	**.895**
Mobile	.625	.624	.654	.673	.524	.620	.454	.601	**.903**
NHIS	.758	.754	.421	.538	.517	.753	.905	.602	**.909**
Ride	.787	.786	.504	.625	.696	.781	.871	.844	**.881**
Synthetic1	.902	.832	.633	.636	.769	.876	.921	.943	**.980**
Synthetic2	.754	.697	.526	.600	.634	.853	.811	.890	**.946**
Synthetic3	.814	.835	-	-	.526	.845	.908	.627	**.964**

Table 2: Comparison between CFSC and baselines using the combined measure. Feature labels are generated using decision trees. CFSC significantly outperformed all baselines, except being comparable to ATT-FL on the Synthetic2 dataset.

The results show that CSFC performs better than all baselines on all datasets for both expert simulation settings. For the evidence counterfactual simulation setting (Table 1), CSFC versus the best runner-up baseline's performances are: 0.78 versus 0.74 for Credit, 0.70 versus 0.61 for Company, 0.91 versus 0.90 for Mobile, 0.78 versus 0.68 for NHIS, and 0.76 versus 0.71 for the Mobile dataset. The t-test results[3] confirm that differences are statistically significant for all datasets except for the Mobile dataset. The results are similar for the decision tree simulation setting (Table 2); CSFC outperforms all baselines, and the differences are significant for all datasets except on the Synthetic2 dataset.

Among the baselines, only LR-PL and ATT-FL are also supervised by feature labels during training time. The other methods serve as baselines for full automation when human supervision for instance-level feature selection is not available. ATT-FL performed better than most baselines, as expected. LR-PL was competitive but sometimes failed badly (e.g., 0.33 on the Company dataset in Table 1 and 0.45 on the Mobile dataset in Table 2). Among the full automation baselines, FF usually performed the best, which was somewhat surprising. This is contributed by its great classification F_1 score and reasonable-but-not-great feature selection $F_1^{\mathbf{w}}$ score (but note that FF always used all features so it did not provide any interpretability). DT and LR were comparable to FF in most cases. RNP generally performed worse than FF on the evidence counterfactual simulation setting mostly due

[1] We provide an ablation study of varying γ_a in Section 5.3.

[2] The separate results for classification and feature selection are included in the supplementary materials.

[3] The p values are included in the supplementary material.

to its worse classification F_1 score. For the decision tree simulation setting, however, RNP can be better or comparable to FF, as it often had much better feature selection $F_1^{\mathbf{w}}$.

5.2 Feature Selection Analysis

We next conduct a deep dive analysis of the instance-level feature selection of the methods. We compare CFSC with the other two baselines that also used feature supervision (LR−PL and ATT−FL) to the ground truth feature selections at the instance level.

Instance-Level Feature Selection Analysis

Using ground-truth feature selection vectors $\boldsymbol{a}_j$ for each $\boldsymbol{x}_j$, we group each instance $\boldsymbol{x}_j$ in the test data based on which features are used to classify them. For example, on the Synthetic1 dataset, 206 instances in the test set had exactly the following three features 'on' based on ground truth: Feature0, Feature3, and Feature4. We also create groupings based on the predicted feature selection vectors $\bar{\boldsymbol{a}}_j$. We then compare the ground truth and the predicted groups.

For a given ground-truth group $\mathcal{G}_t$ (i.e., instances that have exactly the same features 'on'), let the features that are on be $\mathcal{A}_j = \{a_{jk} = 1\}$. For a given model, f, find all objects x_l where exactly the same features $\mathcal{A}_j$ are predicted to be 'on;' let this group be $\mathcal{G}_f$. We compute the true positive ($\mathcal{G}_t \cap \mathcal{G}_f$), false positive ($\mathcal{G}_f \setminus \mathcal{G}_f$), precision, recall, and F_1 for $\mathcal{G}_t$. The results are presented in Tables 3 and 4 for the counterfactual and decision tree simulated experts respectively. As an illustration, take the Credit data in Table 3 as an example: 691 objects in the test data had only one feature (Total_Trans_Ct) 'on' based on ground truth. The LR−PL strategy predicted that only Total_Trans_Ct was 'on' for 929 objects, of which 636 were true positives and 293 were false positives. Hence, the precision of LR−PL for this group is $636/929 = .685$ and recall is $636/691 = .920$.

We first observe that the features that are 'on' for the top group in the evidence counterfactual strategy and the decision tree strategy are quite different: for the former, the top groups have only one feature 'on' whereas multiple features are 'on' for the latter strategy. A possible reason is as follows: when a counterfactual strategy is for a logistic regression model, the top $|w_{yk} \times x_{ik}|$ might dominate the classification decision and removing smaller values would not flip the label. For the decision tree approach, however, the root feature is often followed by other features before a classification decision is made, because the splits at the top are often not pure enough.

Comparing CFSC, LR−PL, and ATT−FL, we see that in Table 3, CFSC had better or comparable F_1 measures on these groups for most datasets, except for the Company dataset, where recall was low (.492). LR−PL had fluctuating performance, sometimes with low precision (Company), sometimes with low recall (NHIS), and sometimes both low precision and low recall (Ride). Though ATT−FL performed well in general (best F_1 on two datasets, and within .05 F_1 on another two datasets), it predicted 0 cases for the NHIS group.

For the decision tree expert simulation strategy (Table 4), CFSC had better or comparable F_1 results to LR−PL, whereas the ATT−FL method again struggled, with many cases with 0 instances. Further analysis show that ATT−FL used totally different features with most instances. For example, the top $\mathcal{G}_f$ of ATT−FL used 16 features on NHIS dataset for the evidence counterfactual simulation setting.

Density

We next present the density statistics (as defined in Equation 3) as a measure of how many features are used per instance on average by each method. A model with low density is often preferred to the one with higher density because of its easier interpretability. Table 5 shows the density values for all methods under the evidence counterfactual simulation setting (results for the decision tree strategy are similar and included in the supplementary materials).

The ground truth density values for these datasets are presented as the last column in the table. They are all low values, ranging from 1.2 (Mobile) to 4.8 (NHIS), as most instances were classified with only a handful of features. The rule-based learners (RL−P and RL−N) often have the lowest density among all models as they use none of the features if no rules apply for an instance. The two feature-selection supervised baselines, ATT−FL and LR−PL, have lower density than the true values in most cases, whereas CFSC often has the closest density to the ground truth density.

5.3 Ablation Study for CFSC

In the results that we presented so far, the classification F_1 and the feature selection $F_1^{\mathbf{w}}$ were given equal weights through $\gamma_a = 0.5$ (Equation 2) and the λ_a parameter for CFSC (used to combine the classification loss L_y and feature selection loss L_a in training of the network) was tuned using a validation set. Here, we study what would happen if we manually set the λ_a and γ_a to fixed values and force CFSC to focus on the feature selection and classification tasks with varying degrees. Table 6 shows the results for three cases:

- *Case 1*: $\lambda_a = \lambda_a^*, \gamma_a = 0.5$. This is the same setting used in earlier results.
- *Case 2*: $\lambda_a = 0.5, \gamma_a = 0.5$. Both classification and feature selection are given equal weights. Different from Case 1, λ_a is not tuned on the validation set and instead is fixed to 0.5. Other hyper-parameters such as learning rate are tuned on the validation set.
- *Case 3*: $\lambda_a = 1, \gamma_a = 1$. The model focuses exclusively on feature selection and ignores classification loss during training. It also ignores classification performance when tuning other hyper-parameters on the validation set. The purpose of this setting is to put the feature selection performances of Case 1 and Case 2 into perspective.

CFSC performs the best on the combined measure when $\lambda_a = \lambda_a^*, \gamma_a = 0.5$ as expected. The fully-balanced and fixed setting, $\lambda_a = 0.5, \gamma_a = 0.5$, has comparable[4] combined performance to the tuned λ_a case in general. When $\lambda_a = 1, \gamma_a = 1$, feature F_1 is the best but the classification F_1 is junk as expected, which also resulted in a poor combined performance. Case 3 results show that context-aware feature selection is a difficult problem in general, as the F_1

[4]The p values are included in the supplementary material.

	Top Group	Group Size	LR−PL					ATT−FL					CFSC				
			TP	FP	P	R	F_1	TP	FP	P	R	F_1	TP	FP	P	R	F_1
Credit	Total_Trans_Ct	691	636	293	.685	.920	.785	534	31	.945	.773	.850	609	43	.934	.881	.907
Company	Attr21	531	472	445	.515	.889	.652	321	21	.939	.605	.735	261	51	.837	.492	.619
Mobile	RAM	530	530	137	.795	1.000	.886	475	11	.977	.896	.935	513	85	.858	.968	.910
NHIS	Emphysema=No	240	136	23	.855	.567	.682	0	0	-	.000	.000	177	16	.917	.738	.818
Ride	Trip_Cost	691	382	356	.518	.553	.535	549	22	.961	.795	.870	630	54	.921	.912	.916

Table 3: Evaluation for CFSC, LR−PL, and ATT−FL on the top groups. Feature labels are generated via the evidence counterfactual strategy. CFSC had better or comparable F_1 measures for most datasets whereas LR−PL and ATT−FL had fluctuating performance.

	Top Group	Group Size	LR−PL					ATT−FL					CFSC				
			TP	FP	P	R	F_1	TP	FP	P	R	F_1	TP	FP	P	R	F_1
Credit	Total_Trans_Amt∧Total_Trans_Ct	430	389	2	.995	.905	.948	425	223	.656	.988	.788	421	10	.977	.979	.978
Company	Attr26∧Attr27∧Attr34	1131	1124	1	.999	.994	.996	0	0	-	.000	.000	1115	5	.996	.986	.991
Mobile	RAM	534	531	133	.800	.994	.886	424	3	.993	.794	.882	498	11	.978	.933	.955
NHIS	Emphysema=Yes∧Yrs_Since_Smk	322	311	1	.997	.966	.981	0	0	-	.000	.000	305	6	.981	.947	.964
Ride	Trip_Cost∧Trip_Seconds	632	567	91	.862	.897	.879	593	308	.658	.938	.774	551	99	.848	.872	.860
Synthetic1	Feature0∧Feature3∧Feature4	206	196	14	.933	.951	.942	39	4	.907	.189	.313	194	0	1.000	.942	.970
Synthetic2	Feature6	185	172	0	1.000	.930	.964	181	0	1.000	.978	.989	180	3	.984	.973	.978
Synthetic3	Feature11∧Feature3	252	198	80	.712	.786	.747	0	0	-	.000	.000	242	11	.957	.960	.958

Table 4: Evaluation for CFSC, LR−PL and ATT−FL on top groups. Feature labels are generated via decision trees. CFSC had better or comparable F_1 results to LR−PL, whereas ATT−FL had fluctuating performance

	LR	DT	RL−P	RL−N	ATT	RNP	LR−PL	ATT−FL	CFSC	*True*
Credit	34.4	4.3	1.0	1.2	15.4	10.8	1.2	1.6	2.0	2.1
Company	64.0	7.1	.1	.1	59.4	30.1	1.3	1.9	3.2	3.2
Mobile	19.6	1.5	.8	.7	1.7	6.5	1.0	1.3	1.2	1.2
NHIS	114.3	6.5	1.0	1.7	15.0	43.5	4.3	31.1	5.0	4.8
Ride	44.8	5.1	.8	1.2	1.7	14.1	1.6	1.9	2.3	2.6

Table 5: The density measure as defined in Equation 3. Feature labels are generated via the evidence counterfactual strategy. CFSC often had the closest density to the true values. ATT−FL and LR−PL had lower density than the true values in most cases.

		(λ_a=λ_a^*, γ_a=0.5)	(λ_a=0.5, γ_a=0.5)	(λ_a=1, γ_a=1)
Credit	*Clf.* F_1	.886	.892	.564
	Fea. F_1	.684	.666	.710
	Comb. F_1	.785	.779	.637
Company	*Clf.* F_1	.771	.782	.560
	Fea. F_1	.631	.570	.691
	Comb. F_1	.701	.676	.626
Mobile	*Clf.* F_1	.957	.955	.567
	Fea. F_1	.856	.857	.857
	Comb. F_1	.907	.906	.712
NHIS	*Clf.* F_1	.827	.824	.576
	Fea. F_1	.735	.687	.754
	Comb. F_1	.781	.756	.665
Ride	*Clf.* F_1	.832	.841	.438
	Fea. F_1	.697	.684	.694
	Comb. F_1	.765	.763	.566

Table 6: Comparison between different sets of λ_a and γ_a for CFSC. Feature labels are generated via the evidence counterfactual strategy.

values were often in the 0.7 range. Case 1, even though it balanced both classification and feature selection performance, had a reasonable feature selection performance in comparison to Case 3 (comparable on two datasets, within 0.02 for one, within 0.03 for one, and within 0.06 for the worst case).

We presented a few possible scenarios here. The optimal balance depends on the application and needs to be decided by the stakeholders by weighing the trade-off between high classification and high feature selection performance.

6 Limitations

While there are publicly available text classification datasets where pieces of text were highlighted as rationales, we could not find any tabular data with instance-level features were highlighted. Hence, we created simulated experts for tabular data. Creating simulated experts and users is not new; for example, Sharma and Bilgic [2018] created simulated experts for learning with rationales for text, Tanno et al. [2019] used simulated annotators for learning with label noise, Lei et al. [2020] used simulated users for recommender systems, Li et al. [2019] used simulated labelers for crowdsourcing.

While simulating experts has its advantages, such as the opportunity to experiment with many datasets and the ability to control and vary simulation settings, we also acknowledge that the empirical findings might not carry over to the real datasets completely. To mitigate this problem, we created two completely different experts: an evidence counterfactual expert and a decision tree expert. Furthermore, we experimented with several kinds of domains (e.g., credit, health, ride sharing, etc.) with varying number of features.

7 Conclusions

We proposed a joint model that can learn from both class labels and instance-level feature labels, to perform what we define as *context-aware feature selection and classification*: skim a given instance's full feature vector, focus on the relevant features for that instance, and make a final classification decision using only the selected features. We adapted several approaches from the literature to the context-aware feature selection and classification task. The empirical evaluations showed that the proposed model outperformed them on the combined classification and feature selection measures while also was able to better emulate the ground-truth instance-level feature selections.

References

[Adriaenssens *et al.*, 2004] Veronique Adriaenssens, Bernard De Baets, Peter LM Goethals, and Niels De Pauw. Fuzzy rule-based models for decision support in ecosystem management. *Science of the Total Environment*, 319(1-3):1–12, 2004.

[Arai *et al.*, 2016] Hiromasa Arai, Crystal Maung, Ke Xu, and Haim Schweitzer. Unsupervised feature selection by heuristic search with provable bounds on suboptimality. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Arya *et al.*, 2019] Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, et al. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*, 2019.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Barrett *et al.*, 2018] Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 302–312, 2018.

[Bastings and Filippova, 2020] Jasmijn Bastings and Katja Filippova. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? *arXiv preprint arXiv:2010.05607*, 2020.

[CDC, 2017] CDC. National health interview survey (nhis) dataset. https://www.cdc.gov/nchs/nhis/nhis_2017_data_release.htm, 2017. Accessed: 2023-01-14.

[City of Chicago, 2019] City of Chicago. Ride-sharing dataset. https://data.cityofchicago.org/Transportation/Transportation-Network-Providers-Trips-2019/iu3g-qa69, 2019. Accessed: 2023-01-14.

[Cohen, 1995] William W Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier, 1995.

[Das *et al.*, 2013] Shubhomoy Das, Travis Moore, Weng-Keen Wong, Simone Stumpf, Ian Oberst, Kevin McIntosh, and Margaret Burnett. End-user feature labeling: Supervised and semi-supervised approaches based on locally-weighted logistic regression. *Artificial Intelligence*, 204:56–74, 2013.

[De Laat, 2018] Paul B De Laat. Algorithmic decision-making based on machine learning from big data: can transparency restore accountability? *Philosophy & technology*, 31(4):525–541, 2018.

[Fürnkranz and Widmer, 1994] Johannes Fürnkranz and Gerhard Widmer. Incremental reduced error pruning. In *Machine Learning Proceedings 1994*, pages 70–77. Elsevier, 1994.

[Gao *et al.*, 2016] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Variational information maximization for feature selection. *arXiv preprint arXiv:1606.02827*, 2016.

[Ghaeini *et al.*, 2018] Reza Ghaeini, Xiaoli Z Fern, and Prasad Tadepalli. Interpreting recurrent and attention-based neural models: a case study on natural language inference. *arXiv preprint arXiv:1808.03894*, 2018.

[Goyal, 2020] Goyal. Credit card customers dataset. https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers, 2020. Accessed: 2022-08-13.

[Guidotti, 2021] Riccardo Guidotti. Evaluating local explanation methods on ground truth. *Artificial Intelligence*, 291:103428, 2021.

[Hall, 2000] Mark A Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 359–366, 2000.

[Harradon *et al.*, 2018] Michael Harradon, Jeff Druce, and Brian Ruttenberg. Causal learning and explanation of deep neural networks via autoencoded activations. *arXiv preprint arXiv:1802.00541*, 2018.

[Holte, 1993] Robert C Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90, 1993.

[Jain and Wallace, 2019] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

[Jang *et al.*, 2016] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[Kohavi and John, 1997] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[Lage *et al.*, 2019] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1902.00006*, 2019.

[Lei *et al.*, 2016] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, November 2016. Association for Computational Linguistics.

[Lei *et al.*, 2020] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 304–312, 2020.

[Letham *et al.*, 2015] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[Li *et al.*, 2015] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*, 2015.

[Li *et al.*, 2019] Chaoqun Li, Liangxiao Jiang, and Wenqiang Xu. Noise correction to improve data and model quality for crowdsourcing. *Engineering Applications of Artificial Intelligence*, 82:184–191, 2019.

[Lundberg and Lee, 2017] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017.

[Martins and Astudillo, 2016] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016.

[Melville and Sindhwani, 2009] Prem Melville and Vikas Sindhwani. Active dual supervision: Reducing the cost of annotating examples and features. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, pages 49–57, 2009.

[Moeyersoms *et al.*, 2016] Julie Moeyersoms, Brian d'Alessandro, Foster Provost, and David Martens. Explaining classification models built on high-dimensional sparse data. *arXiv preprint arXiv:1607.06280*, 2016.

[Molnar, 2020] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[Narendra *et al.*, 2018] Tanmayee Narendra, Anush Sankaran, Deepak Vijaykeerthy, and Senthil Mani. Explaining deep learning models using causal inference. *arXiv preprint arXiv:1811.04376*, 2018.

[Purohit *et al.*, 2012] Seema U Purohit, Venkatesh Mahadevan, and Anjali N Kulkarni. Credit evaluation model of loan proposals for indian banks. *International Journal of Modeling and Optimization*, 2(4):529–534, 2012.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[Ribeiro *et al.*, 2018] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: high-precision model-agnostic explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 1527–1535, 2018.

[Seerat and Qamar, 2015] Bakhtawar Seerat and Usman Qamar. Rule induction using enhanced ripper algorithm for clinical decision support system. In *2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 83–91. IEEE, 2015.

[Serrano and Smith, 2019] Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019.

[Sharma and Bilgic, 2018] Manali Sharma and Mustafa Bilgic. Learning with rationales for document classification. *Machine Learning*, 107(5):797–824, 2018.

[Sharma, 2017] Sharma. Mobile price dataset. https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification, 2017. Accessed: 2022-08-13.

[Shrestha *et al.*, 2019] Yash Raj Shrestha, Shiko M Ben-Menahem, and Georg Von Krogh. Organizational decision-making structures in the age of artificial intelligence. *California Management Review*, 61(4):66–83, 2019.

[Tanno *et al.*, 2019] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11244–11253, 2019.

[Wachter *et al.*, 2017] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[Wang *et al.*, 2016] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.

[Wang *et al.*, 2022] Juanyan Wang, Manali Sharma, and Mustafa Bilgic. Ranking-constrained learning with rationales for text classification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2034–2046, 2022.

[Yu and Liu, 2003] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 856–863, 2003.

[Zaidan *et al.*, 2007] Omar Zaidan, Jason Eisner, and Christine Piatko. Using "annotator rationales" to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267, 2007.

[Zhang *et al.*, 2016] Ye Zhang, Iain Marshall, and Byron C Wallace. Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 795. NIH Public Access, 2016.

[Zieba *et al.*, 2016] Maciej Zieba, Sebastian K Tomczak, and Jakub M Tomczak. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert systems with applications*, 58:93–101, 2016.