Incipient Anomaly Detection with Ensemble Learning

Baihong Jin

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2020-199
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-199.html

December 9, 2020

Incipient Anomaly Detection with Ensemble Learning

by

Baihong Jin

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alberto Sangiovanni-Vincentelli, Chair
Professor Kameshwar Poolla
Associate Professor Stefano Schiavon

Fall 2020

Incipient Anomaly Detection with Ensemble Learning

Abstract

Incipient Anomaly Detection with Ensemble Learning

by

Baihong Jin

Doctor of Philosophy in Engineering – Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Alberto Sangiovanni-Vincentelli, Chair

Anomaly detection techniques are important in system health monitoring applications (e.g., fault detection and disease diagnosis). By recognizing suspicious patterns in data, anomaly detection models can tell whether system has degraded from the normal operating condition into a faulty or diseased state. To avoid unnecessary losses, it is desirable to have a way to identify incipient anomalies, i.e. to detect potential problems in their early stages of development. In buildings, early detection of incipient faults can help reduce maintenance and repair costs, save energy, and enhance occupant comfort. In healthcare, if incipient diseases can be discovered early, effective treatments can be applied and can prevent diseases from progressing into more severe stages.

However, it is difficult to accurately identify incipient anomalies while at the same time not incurring into too many false alarms. Incipient anomalies present milder deviations compared to severe ones, and are difficult to detect and diagnose due to their close resemblance to normal operating conditions. Anomaly detection approaches based on supervised Machine Learning (ML) rely on high-quality labeled data to build accurate classifiers. However, the lack of incipient anomaly examples in the training data can pose severe risks to anomaly detection methods that are built upon ML techniques, because these anomalies can be easily mistaken as normal operating conditions.

Ensemble learning is widely applied in ML to improve model performance and to mitigate decision risks. In ensemble approaches, predictions from a diverse set of learners are combined to obtain a joint decision with lower bias and variance. Recently, various methods have been explored in literature for estimating prediction uncertainties using ensemble learning. To address this challenge of incipient anomalies, I propose in this dissertation to utilize the uncertainty information available from ensemble learning to identify potential misclassified incipient anomalies. We will show that ensemble learning methods can give improved performance on incipient anomalies and identify common pitfalls in these models through extensive experiments on two real-world applications—detection of chiller faults and diagnosing

diabetic retinopathy diseases. A theoretical analysis that compares the two popular strategies for extracting uncertainty information will also be given. We will also discuss how to design more effective ensemble models for detecting incipient anomalies.

To My Family.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my deepest appreciation to my advisor, Prof. Alberto Sangiovanni Vincentelli, for his persistent help and encouragement throughout my PhD study; he provided tremendous insights into my research and always made himself available whenever I needed guidance and help. Most importantly, he has taught me to be open-minded and to embrace new ideas and techniques. I am also grateful to Prof. Kameshwar Poolla, my co-advisor, for his guidance and support throughout my academic journey. Without the advice and support from Alberto and Kameshwar this dissertation would not have been possible.

I would like to thank Prof. Stefano Schiavon, another faculty member on my dissertation committee for his teaching and advice. Over the past few years, I have learned much about building science from Stefano from his lectures and also our personal interactions. I must also thank Prof. Costas Spanos who kindly guided me through this six-year journey, and served on my qualifying exam committee.

I would also like to thank other faculty members and researchers who have collaborated with me during my PhD study, including Prof. Yuxin Chen, Prof. Duncan Callaway, Prof. Baosen Zhang, Prof. Pierluigi Nuzzo, Dr. Paul Raftery, Prof. Sanjit Seshia, Prof. Edward Lee, Prof. Yasser Shoukry, Prof. Ravi Prasher, Dr. Sean Lubner, Dr. Mahmoud Elzouka, Dr. Seshadhri Srinivasan, Prof. Qi Zhu, as well as many others at EECS, SinBerBEST and LBNL. The collaborations with you greatly expanded the horizon of my view and understanding.

Over the past few years at UC Berkeley, I had the fortune to work in several different research fields and to learn from brilliant researchers worldwide. With Prof. Pierluigi Nuzzo, I researched and developed contract-based design methodologies and applied it to the domains of smart building and smart grids [39, 45, 35]. I have always been amazed by the broad knowledge of Dr. Yuxun Zhou in machine learning, optimization and energy domains, and enjoyed the time working with and learning from him in our joint works [107, 39, 35]. The friendship and collaboration with Dr. Pan Li and Dr. Dai Wang will always be a fortune in my life and career; I have truly learned much from their expertise when working together on voltage regulation problems in power systems [60, 59]. My wonderful journey in exploring anomaly detection methods started from my collaborations with Dr. Matt Weber and Dr. Dan Li. With Matt, we developed GORDIAN [94], a formal reasoning-based outlier detection approach for secure localization, a technique that merges the strength of formal reasoning and optimization for detecting outliers in localization systems. I was truly glad the paper got published after four years of hard work. As a PhD student in the SinBerBEST program, I was fortunate to have the chance to work with Dr. Dan Li, a true expert in fault detection and diagnosis methods. Dan's knowledge and experience in her areas inspired me much when we worked together to develop anomaly detection algorithms for Cyber-Physical System (CPS) [56, 44, 57, 40]. I cannot be luckier to have met Prof. Yuxin Chen in the Data-driven CPS Project. He is so knowledgeable in theoretical machine learning and has always been so patient with me in our collaborations [43, 44, 40, 41, 88, 89]. I cannot imagine a collaborator and mentor better than Yuxin. Lastly, I want to express my appreciation towards Prof. Guojie Luo, my undergraduate research advisor at Peking University. Guojie led me into the world of research, and continued to support me during my PhD career. I was fortunate to be working with him on a number of important topics in VLSI CAD, including static timing analysis [38] and circuit placement.

Next I would also like to extend my deepest gratitude towards my dear friends and cohorts at Berkeley and at other institutions, including Jieyi Lu, Mehdi Maasoumy, Liangpeng Guo, Chung-Wei Lin, Daniel Fremont, Marten Lohstroh, Xiangyu Yue, Yiyi He, Ruoxuan Xiong, Yingshui Tan, Antonio Iannopollo, Shromona Ghosh, Edward Kim, John Finn, Nikunj Bajaj, Wayne Lin, Zheng Liang, Sheng-Jung Yu, Alessio Iovine, Ruoxi Jia, Ming Jin, Han Zou, Hari Prasanna Das, Juyue Chen, Shuyang Li, Min Ting, and many others. You have made my time at Berkeley an enjoyable and memorable journey.

I also owe a huge thank-you to Shirley Salanio, Jessica Gamble, Zuraimi Sultan and Judy Huang for answering all my administrative questions over the past six years. Without your help, I would have spent much more time and effort sorting out those my administrative issues.

Finally, my deepest gratitude to my family—my parents, who have always been unconditionally supporting and encouraging me to pursue my dream since I was little, and have cultivated a desire for knowledge in my mind. This dissertation would not have been possible without their warm love, continued patience, and endless support.

Chapter 1

Introduction

1.1 Chapter Overview

This introductory chapter summarizes the motivation and key contributions of my dissertation research. The organization of contents in this dissertation will also be described.

1.2 Motivation

System degradation is a gradual and complicated process, and its development can in general be segmented into four discrete stages [76]: 1) *normal degradation*, 2) *transition region*, 3) *accelerated degradation*, and 4) *failure*; see Figure 1.1 for an illustration. We are interested in locating the transition region, a.k.a. the "knee" of the trajectory of the degrading health index; this is also where *incipient anomalies* often occur. In this dissertation, an "anomaly" can mean either a machine fault in industrial applications or a human disease in health applications. Incipient anomalies present milder symptoms compared to severe ones, and are more difficult to detect and diagnose due to their close resemblance to normal operating conditions. From a preventive maintenance perspective [5, 66], when a system reaches a transition region or when incipient anomalies happen, maintenance actions should be taken as soon as possible to stop the degradation process in order to prevent further damage [40].

In this dissertation, We will highlight the importance of incipient anomaly detection in the two following application domains.

Incipient Faults in Building Equipment Building faults whose impact is less perceivable and/or hinder regular operations are called *soft faults* [65, 102]. These soft faults, especially in their incipient stages, are difficult to detect as their signatures are not generally obvious due to their magnitudes, measurement/system noise, or feedback control actions [27, 93]. Nevertheless, they will impact energy consumption, occupants safety and well-being, system performance, running costs, and maintenance/repair costs adversely in the long-run if left undetected and unattended [35, 2]. Therefore, it is an important to develop methods to

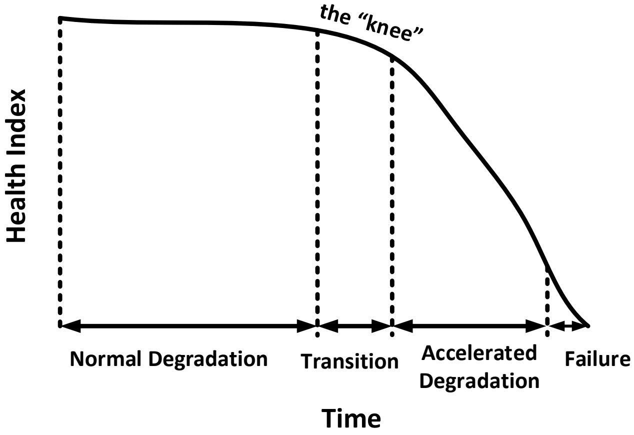


Figure 1.1: The four stages in a typical degradation process [76].

detect and diagnose such soft faults at their incipient stage, for various building systems such as chillers and Air Handling Units (AHUs).

Incipient Diseases in Healthcare Similar challenges also exist in the medical domain. Similar to the industrial machine faults described above, many chronic diseases naturally develop in a progressive manner—from an incipient stage to a severe stage. At the incipient stage, the symptom is often slight, making the disease difficult to detect. Due to this challenge, the best time for intervention and treatment is often missed. Take *diabetic retinopathy* (see Figure 1.2), a common complication of the diabetic disease and a leading cause of blindness in the working-age population of the developed world [23], for example. If the disease can be discovered and treated during its incipient stage, severe consequences such as blindness can usually be avoided. Therefore, an effective screening method that can identify people with incipient disease conditions will be very much desired for disease prevention.

1.3 Research Contributions

Due to the reasons stated above, it is important to find out when a system starts to shift away from its healthy condition. However, this is not easy with either conventional methods or data-driven methods, especially when there is a lack of labeled incipient anomaly examples in the available data. Without such examples, it is difficult to train and tune a well-performing model that can differentiate between normal conditions and incipient anomalies.

The solution proposed in this dissertation is based on ensemble learning [108], i.e., training multiple classifiers and leveraging their joint decisions to recognize incipient anomalies. In literature, a variety of ensemble methods have been proposed on the estimation of prediction uncertainties [52, 21, 51]. We will show in this dissertation that such uncertainty information

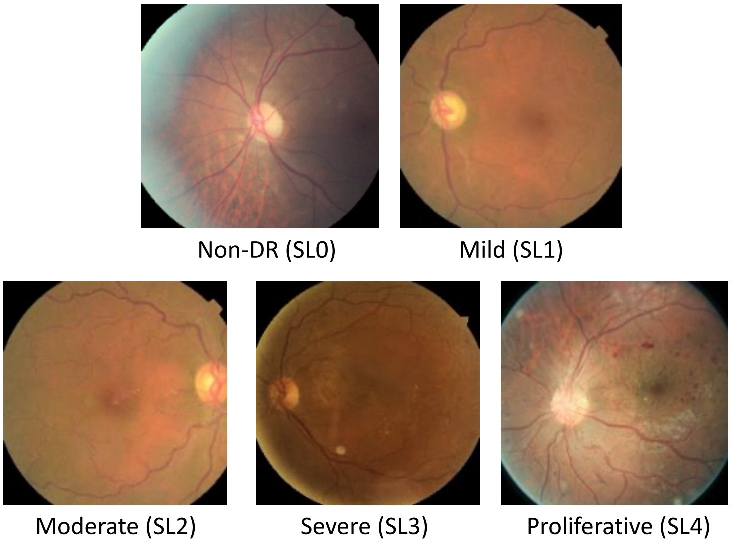


Figure 1.2: Example fundus images of diabetic retinopathy diseases.

can be leverage to indicate incipient anomalies that are prone to be misclassified as healthy condition data. Let us use the examples shown in Figure 1.3 to illustrate the idea.

In the two example applications shown in Figure 1.3, the system health conditions are both graded into five Severity Levels (SLs), from SL0 (healthy) to SL4 (most severe). If we train a classification model, it probably will have good performance on in-distribution data (SL0 & SL4). However, it may fail badly with identifying the incipient anomaly data. For example, the SL2 anomalies may be recognized as normal by any of the decision boundaries shown in Figure 1.3. More generally, classical supervised learning approaches designed for achieving maximal separation between labeled classes (e.g. margin-based classifiers, discriminative neural networks, etc), are less effective in detecting such low-severity, ambiguous, incipient anomaly data examples.

We can also see in Figure 1.3 that the individual classifiers have much disagreement on the SL2 data. If we train an ensemble of many diversified classifiers, the amount of disagreement among individual classifiers can be used to measure the prediction uncertainties, and is therefore useful for indicating incipient anomalies such as SL2. However, for SL1 data that are close to the normal cluster, the above approach will become less effective. We find this is a common phenomenon in our empirical studies. A remedy to this problem is to increase the *statistical power* of the base learners by moving the decision boundaries towards the normal cluster. Another question is how to properly combine the anomaly scores from ensemble members into an *uncertainty metric* to inform decision making. We will answer these questions in the subsequent chapters of this dissertation.

The proposed ensemble-based methodology for incipient anomaly detection is a useful

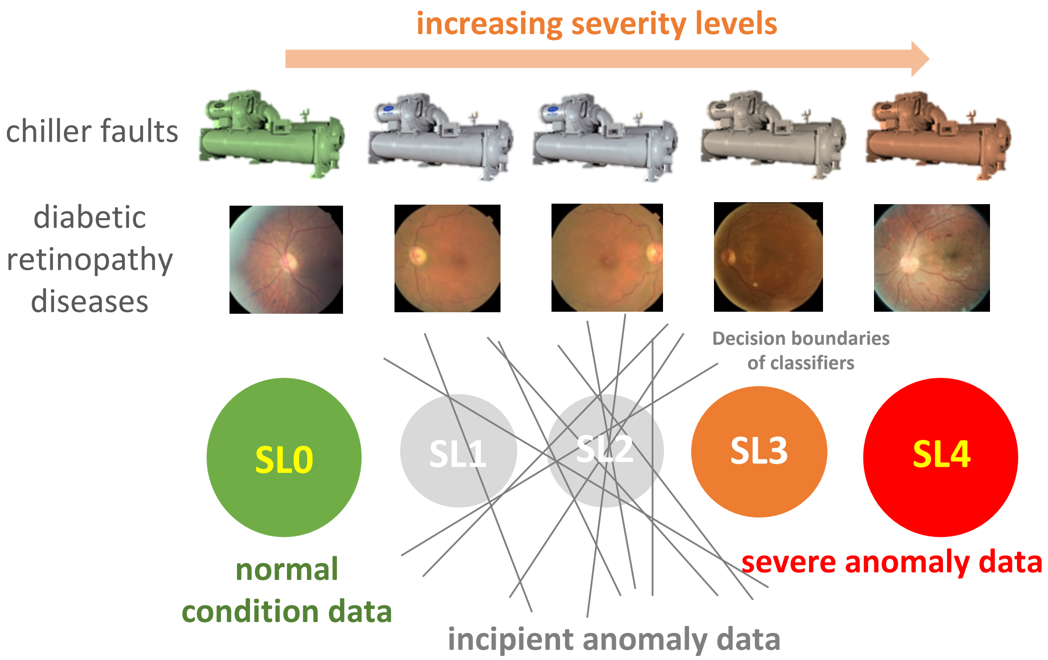


Figure 1.3: Illustration showing how an ensemble classifier can conceptually help detect incipient anomalies. The gray lines represent the decision boundaries of base learners in the ensemble.

complement to the existing methods in literature on multi-grade anomaly detection [50, 55, 54], specifically under cases where the available anomaly data for training are *insufficient* to cover the entire severity spectrum. In this dissertation, we will give an easy-to-use recipe for Machine Learning (ML) practitioners to develop ensemble anomaly detection models that can more effectively recognize incipient anomaly examples, and will provide recommendations to address the aforementioned issues that can help produce more effective ensemble models for anomaly detection applications. The contributions of this dissertation are summarized as follows:

- We have shown experimentally that incipient anomaly examples, when missing or underrepresented in the training distribution, can pose risks to popular supervised ML-based anomaly detection methods. Ensemble methods are in general helpful in improving the detection performance of both supervised and unsupervised models on such incipient anomalies.

- Two commonly used uncertainty metrics for ensemble learning, one based on ensemble mean (MEAN) and the other based on ensemble variance (VAR) are compared and analyzed. The theoretical analysis shows that the MEAN metric is more preferable to the VAR metric.

1.4 Organization of the Dissertation

This dissertation begins with an overview of the anomaly detection problem, and explains why incipient anomalies (including faults and diseases) cast a difficult challenge to existing anomaly detection techniques. Next, the ensemble methodology for incipient anomaly detection is proposed. Then, the proposed methodologies are demonstrated on data, with description of how the data are processed. Finally, this dissertation concludes with findings, recommendations, and future work. The breakdown by chapter is as follows:

Chapter 1 summarizes the motivation for the research and also the organization of chapters in this dissertation.

Chapter 2 gives the preliminaries on anomaly detection problems as well as relevant mathematical concepts.

Chapter 3 presents our proposed methodology on using ensemble learning to improve the detection performance on incipient anomalies.

Chapter 3 describes the datasets to be used for later experimental studies.

Chapter 4 describes the datasets to be used for later experimental studies.

Chapter 5 presents a case study on chiller fault detection. We will evaluate how ensembles made up of Decision Tree (DT) or Neural Network (NN) base learners help detect incipient faults.

Chapter 6 presents another case study on diabetic retinopathy diagnosis. In this case study, state-of-the-art Convolutional Neural Network (CNN) models are trained to differentiate between referable and non-referable diabetic retinopathy diseases. We will show how the ensembles of these models help identify incipient diseases.

Chapter 7 introduces an Stratification-Aware Cross-Validation (SACV) method for detection out-of-distribution (o.o.d.) faults, and will present experiment results.

Chapter 8 summarizes the results of the presented research projects, and discusses promising directions for future research.

Chapter 2

Preliminaries

2.1 Chapter Overview

In this chapter, we will formulate the anomaly detection problem in a formal way, and define notations and performance metrics that will be used throughout the rest of this dissertation. Some related work in literature will also be reviewed in this chapter.

2.2 Anomaly Detection

We formulate the anomaly detection problem in a *binary classification* setting. An anomaly detection model aims at differentiating fault conditions from the normal condition by monitoring the system state. Let $z \in \{0, 1\}$ be the ground-truth label of system state $\boldsymbol{x} \in \mathbb{R}^d$, where $z = 0$ stands for the normal condition and $z = 1$ the anomaly condition. An *anomaly detector* is some rule, or function, that assigns (predicts) a class label $\hat{z} \in \{0, 1\}$ to input x.

Let $\mathcal{X}$ be the set of data points, and $\mathcal{M}$ be a model class of classification models. Suppose a classification model $M \in \mathcal{M}$ defines an *anomaly score* function $s^M : \mathcal{X} \to \mathbb{R}$ that characterizes how likely a data point corresponds to an anomaly state; a larger $s^M(x)$ implies a higher chance of a data point $\boldsymbol{x}$ being an anomaly. The classifier's decision on whether or not x corresponds to an anomaly can be made by introducing a *decision threshold* τ^M to dichotomize the anomaly score $s^M(x)$. We can define the classifier's predicted label $\hat{z} = \mathbb{1}\{s^M(x) > \tau^M\}$, i.e. M predicts x to be an anomaly if and only if the anomaly score $s^M(x)$ is above the threshold τ^M. For evaluating the accuracy of anomaly detection, we can define the False Negative Rate (FNR) and False Positive Rate (FPR) of the model M on the test data distribution as follows:

$$\mathrm{FNR}(s^M, \tau^M) = \mathbb{P}\left[\hat{z} = 0 \mid z = 1\right], \tag{2.1}$$

$$\mathrm{FPR}(s^M, \tau^M) = \mathbb{P}\left[\hat{z} = 1 \mid z = 0\right]. \tag{2.2}$$

Let $\mathcal{X}^{\text{train}}$ be a subset of labeled data points for training. Ideally, our goal is to learn an anomaly score function s^* by minimizing its classification error on $\mathcal{X}^{\text{train}}$, and then decide a corresponding threshold τ, such that (s^*, τ) can optimally trade off the FNR and the FPR on unseen test data.

Setting the detection threshold τ We leverage the prediction uncertainties given by ensemble learners to make uncertainty-informed decisions. Consider an ensemble $\mathcal{E}$ comprising a diverse set of K binary classifiers, $\mathcal{M}^{(1)}, \mathcal{M}^{(2)}, \ldots, \mathcal{M}^{(K)}$, that have been trained for the same detection task. Let $z_i \in \{0, 1\}$ represent the ground-truth label of input x_i, and $\hat{y}_i^{(k)}$ denote the output of the kth classifier where $k \in \{1, \ldots, K\}$ and $\hat{y}_i^{(k)} \in [0, 1]$. By using a threshold τ to dichotomize the continuous output $y_i^{(k)}$, each classifier $\mathcal{M}^{(k)}$ produces a predicted class label $\hat{z}_i^{(k)}$ for input x_i.

As mentioned above, one always has to make a trade-off between FNR and FPR by setting an appropriate decision threshold τ (a.k.a. operating point). A simple approach is to directly set the decision threshold τ to a predefined value (e.g., 0.5); this is often not a bad approach if most data points are well separated and receive an anomaly score close to 0 or 1. However, such approach usually does not returns us a high-sensitivity classifier that satisfies a given FPR requirement. In real-practice, one often needs to decide a proper *operating point* on the Receiver Operating Characteristic (ROC) curve by taking FPR and FNR requirements into account. One way to do that is to set τ such that the FPR on the development set reaches a predefined level q. The rationale behind such scheme is to fix the FPR (type-1 errors) to a constant value on the development set while minimizing the number of false negatives (type-2 errors). Similar approaches are seen in other application domains. For example, in radar applications, this scheme is also known as Constant False Alarm Rate (CFAR) detection [78].

The decision scheme described above is illustrated in Figure 2.1 as the BASELINE scheme. The goal is to come up with a proper τ used for identifying positive examples. Under most cases, there will be false positives among the examples predicted as positive; however, these false positives are not the utmost concern if the FPR can be controlled to a low level. On the other hand, false negatives are anomalous instances mistaken as normal, which represents a more severe problem in anomaly detection. We propose utilizing prediction uncertainty information from ensemble classifiers to identify potential false negatives in an uncertainty-informed decision scheme.

Uncertainty-informed anomaly detection We consider an *uncertainty-informed* diagnostic scheme as an application of prediction uncertainties that fosters the collaboration between human and AI systems. In this scheme, an Machine Learning (ML) model is first used to screen the cases (operational data for industrial machines, medical images for humans, etc.). Cases diagnosed as positive will be referred to a human reviewer for further inspection, who will confirm the case as positive if she agrees with the ML model's decision. The BASELINE scheme suffers from the problem that false negatives from the ML model's diagnoses would never be reviewed by human diagnosticians. In an *uncertainty-informed* scheme, high-uncertainty negative examples will be identified and sent to human reviewers as well. The criterion used for picking out high-uncertainty examples does not have to be

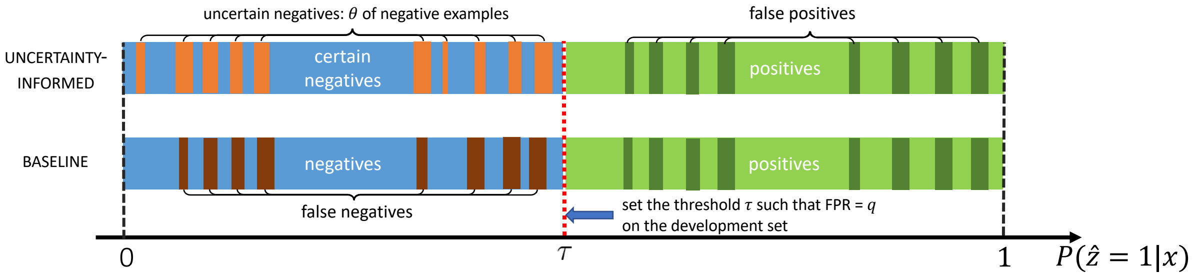


Figure 2.1: Illustration showing the concepts introduced in our uncertainty-informed decision framework.

based on the classifier confidence $\hat{y}$; in fact, we can use a variety of *uncertainty metrics* to be described below for ranking data examples by their associated uncertainties.

To identify false negatives in classification, we use an *uncertainty metric* U to rank the negative examples[1]. An uncertainty metric $U : \mathbb{R}^K \to \mathbb{R}$ takes as input the ensemble predictions $\{\hat{y}_i^{(k)}\}$ on x_i, and outputs an *uncertainty score* $u(x_i) \doteq U\left(y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(K)}\right)$. The interpretation of the uncertainty score $u(x_i)$ depends on the task. In our application, we seek to utilize prediction uncertainties for identifying false negative decisions: a higher $u(x_i)$ indicates higher prediction uncertainty associated with x_i, and hence more likely to be a false negative if $\hat{z}_i = 0$. In such situations, we may need human experts to join the decision process.

The uncertainty score $u(x_i)$ is a real-valued number, and to resolve a dichotomy between "uncertain" and "certain" we will need another threshold $\tilde{u}$. If $u(x_i) > \tilde{u}$ then x_i is deemed an uncertain input example and otherwise a certain one. Once uncertain examples are identified, we will need external resources (e.g., human experts) to inspect them and determine their true labels; however, such external resources are often limited (e.g., due to budget constraints) so we need to determine a proper threshold $\tilde{u}$ so that the number of uncertain negatives is controlled. We determine $\tilde{u}$ by setting the *uncertain negative ratio* on the development set to a pre-defined number $\theta \in (0, 1)$; the uncertain negative ratio is defined as the fraction of uncertain examples among those examples predicted as negative. Only the predicted negative examples that receive the highest uncertainty scores are deemed *uncertain negatives.* To evaluate how uncertain negatives overlap with the actual false negatives, we define the following performance measure.

Definition 1 (False Negative Precision)**.** *We define the false negative precision to be the fraction of false negative decisions among uncertain negative inputs, under a given uncertainty*

[1] Examples that are classified as negative by a classification model, i.e. $\{x_i \mid \hat{z}_i = 0\}$.

metric U and an uncertain negative ratio θ. Written in mathematical form,

$$\textit{FN-precision}(U, q) \doteq \frac{\left|\{x_i \mid i \in \mathcal{I}_q^-, z_i = 1\}\right|}{\left|\mathcal{I}_q^-\right|} \in [0, 1], \tag{2.3}$$

where $\mathcal{I}_q^-$ is the index set of uncertain negative examples.

The FN-precision metric can be interpreted as the ratio of identified uncertain examples being actual false negatives. The higher the FN-precision value, the fewer false alarms are likely to be raised by an algorithm that detects uncertain negatives. We can similarly define a "false negative recall" metric that measures the fraction of false negatives identified by the algorithm. However, in this dissertation we choose to directly report the total number of false negatives and the number of false negatives that are deemed "certain" by the evaluated uncertainty estimation algorithms, as we think it a more straightforward way to make the comparison.

One of our goals in this dissertation is to rigorously analyze and compare two commonly used uncertainty metrics MEAN and VAR, which will be detailed in the upcoming section. Formally, we seek the uncertainty metric U that maximizes FN-precision(U, q).

2.3 Related Work

Data-Driven Fault Detection Fault detection methods in the literature can be broadly classified into three categories: *(i)* model-based, *(ii)* signal-based, and *(iii)* data-driven [100, 106]. Model-based methods depend on explicit physical models at the device levels and use correlation tests on the input-output data to detect faults [36, 34, 37]. Authors in [97] point out that model-based methods are not as practical as data-driven methods in terms of applying the fault detection techniques to real buildings. Signal-based fault detection methods find sensor measurement signatures to indicate faults. Signal-based fault detection combining wavelet transformation and principal component analysis was presented in [61]. Although the methods achieved good performance, extracting relevant signatures and signals that indicate faulty condition is a daunting task for complex systems such as buildings.

In data-driven fault detection approaches, when labeled fault data are available, a fault detection task are usually modeled as a multi-class classification problem. Then a supervised learning method can be employed to learn a classifier to recognize the faults. Many supervised methods such as multivariate regression models [70], Bayes classifiers [28, 104, 99], neural networks (NN) [18, 109, 17], Fisher Discriminant Analysis (FDA) [16], Gaussion Mixture Models [32], Support Vector Data Description (SVDD) [103, 105], and Support Vector Machines (SVM) [63, 25, 9, 101, 69] have been proposed to classify the faults. Recently, Li et al. proposed a tree-structured learning method [55] that not only recognizes faults but also their severity levels; however, it is hard in practice to obtain such a well-labeled dataset that include incipient faults. Researchers have also proposed unsupervised approaches using Principle Component Analysis (PCA) [62], Statistical Process Control (SPC) [85], and autoencoders [80] for fault detection. Depending only on positive (healthy) class data, such unsupervised methods have found their use in detecting anomalies; however, they still lack the ability to diagnose these anomalies.

A review of the literature reveals that data-driven approaches relying on supervised learning are promising methods due to their ability to classify and differentiate data with multiple labels. However, in order to train a well-performing model, large amount of labeled data is typically needed, which is not always easy to obtain. Furthermore, although supervised learning tends to perform well on known (in-distribution) data patterns, the unseen (out-of-distribution) data may lead to unexpected prediction behaviors. In the context of fault detection, an incipient fault example not seen in the training phase may fool the classifier into wrong belief, which is certainly not desirable for fault detection applications. Although this problem can be conceptually alleviated by using a larger, more comprehensive training dataset, in practice it is technically infeasible to obtain fault data of all different fault types, and of all possible severity levels, especially for complex building systems such as chillers.

Out-of-Distribution Input Detection and Uncertainty Estimation In recent years, a number of research dissertations [51, 21] related to the detection of out-of-distribution (o.o.d.) data appeared in literature, especially in the deep learning community that has shown a strong and growing interest in utilizing ensemble methods in supervised learning to estimate the *uncertainty* behind the decisions on data points. Lakshminarayanan et al. [51] proposed using *random initialization* and *random shuffling* of training examples to diversify base learners of the same network architecture. Gal and Ghahramani proposed using MC-dropout [21] to estimate a network's prediction uncertainty by using dropout not only at training time but also at test time. By sampling a dropout model $\mathcal{M}$ using the same input for T times, we can obtain an ensemble of prediction results with T individual probability vectors. The dropout technique provides an inexpensive approximation to training and evaluating an ensemble of exponentially many similar yet different neural networks.

Although promising results from these ensemble approaches have been demonstrated on certain types of o.o.d. data such as dataset shift and unseen/unknown classes [51], it is difficult to evaluate their effectiveness in general, because the o.o.d. part of the world is obviously much "larger" than its in-distribution counterpart and is presumably much harder to analyze. In contrast, our work, although using similar algorithms to those on o.o.d. detection, still embraces a closed-world assumption and restricts the focus to incipient anomalies—a special type of data distribution that has a close connection to the training distribution. We speculate that some knowledge necessary for detecting incipient anomalies is already entailed in the training data, thus making the detection of incipient anomalies possible with supervised methods.

Model Calibration Another relevant line of work aims to produce good probability outputs using *model calibration* techniques [71, 24]. Calibration techniques are typically applied in a *post hoc* fashion, without affecting the parameters (weights) of the original model. For example, in temperature scaling [24], a temperature parameter T is learned on a separate calibration set so as to minimize some calibration error metric.

However, in binary classification tasks, temperature scaling will not affect the rankings among the output probabilities of data points. Let us denote by $y(x)$ the output of a model given input x indicating how likely the model thinks that input x belongs to the positive class. Consider any two input data points x_i and x_j, and suppose $y(x_i; T_1) \geq y(x_j; T_1)$ under temperature T_1. It is easy to prove that under a different temperature T_2 we still have $y(x_i; T_2) \geq y(x_j; T_2)$. This shows that the order among predictions will not change with the temperature, and as a result the decision results will not be affected under a fixed FPR.

Since we are dealing with anomaly detection, a binary classification task in this dissertation, we will not consider temperature scaling techniques in our upcoming experiments. It is worthy to note that in multi-class classification settings the order among maximum entry probabilities can be affected by temperature scaling, unlike in the binary classification case discussed above.

Chapter 3

Ensemble Methodology for Incipient Anomaly Detection

3.1 Chapter Overview

In the sequel, we describe our methodology of using ensemble learning for improving incipient anomaly detection performance. We will first give a brief introduction of a few popular ensemble learning methods including tree ensembles and neural network ensembles, and then discuss several design considerations for constructing ensemble classifiers for anomaly detection. Strategies for better combining ensemble predictions and for extracting uncertainty information will be discussed next. In the end, we will also present a theoretical analysis to compare the discussed two popular strategies for extracting uncertainty information, a core contribution of this dissertation.

3.2 Ensemble Learning

Ensemble learning [108] combines the predictions of multiple models to make a joint decision. Fast algorithms such as decision trees are commonly used in ensemble methods (e.g., random forests), although slower algorithms (e.g., neural networks) can benefit from ensemble techniques as well.

Decision Tree Ensembles

Decision Trees (DTs) [6] make classification or regression decisions by learning simple decision rules inferred from the data features, and they are commonly used as base learners for constructing ensemble learners. We will focus on classification DTs in this chapter. There are multiple ways to combine individual DT classifiers into sequential or parallel ensembles. In sequential ensembles, base learners are created over iterations and there are dependencies among them [77]. Notable examples include various boosting methods, e.g., AdaBoost [20] and gradient tree boosting [48]. In parallel ensembles, individual base learners are created independently of each other [77]. Bootstrap Aggregation (or Bagging [7]) is one representative parallel ensemble approach.

In this dissertation, we will be using a parallel ensemble approach to construct DT ensembles. The variations among individual DT learners will be utilized to lower biases and inform decision uncertainties at test time.

Neural Network Ensembles

The large and flexible design space of deep learning models make them suitable candidates for building ensemble models. Ensembles of deep learning models can be broadly categorized into two types, *explicit ensembles* and *implicit ensembles*. We will give a brief overview of these two types below and describe several representative models.

Explicit Ensembles We give a brief overview of popular deep learning-based ensemble methods that can be used for estimating decision uncertainties. The two methods described below can be categorized into the class of *explicit ensemble* models, where a diverse set of individual models are combined into an ensemble in order to make a joint decision.

Deep Ensemble Proposed by Lakshminarayanan et al. [51, 19], a *deep ensemble* is made up of multiple neural networks of the same architecture; the individual learners are diversified by *random initialization* as well as random shuffling of training examples. The deep ensemble method has been shown to be effective in detecting out-of-distribution (o.o.d.) inputs from image datasets.

Hyperparameter Ensemble Hyperparameter ensemble [96] is a more general ensemble approach for deep learning, where the models generated from hyperparameter tuning procedures are combined into ensembles.

Implicit Ensembles The next two methods fall under the category of *implicit ensemble* methods [30] due to their "*train once get many for free*" nature, where multiple predictions can be generated from one single trained model; the diversity among predictions either comes from the stochasticity inherent to the network (as in MC-dropout) or from perturbations to the input data (as in Test-Time Augmentation (TTA)).

Monte Carlo Dropout (MC-dropout) Dropout [84] is a popular and powerful regularization technique to prevent overfitting neural network parameters. Recently, Gal and Ghahramani proposed using MC-dropout [21] to estimate a network's prediction uncertainty by using dropout not only at training time but also at test time. By sampling a dropout model $\mathcal{M}$ using the same input for T times, we can obtain an ensemble of prediction results with T individual probability vectors. The dropout technique provides an inexpensive approximation to training and evaluating an ensemble of exponentially many similar yet different neural networks.

Test-Time Augmentation (TTA) Similar to the MC-dropout technique, a network with TTA [3, 92] produces a different result each time we “sample” the same network with the same given input x. Different from MC-dropout networks, TTA adds randomness to the test input x through data augmentation as is often performed during training, e.g., adjustment of brightness, image cropping and image flipping. This creates an ensemble of exponentially many predictors as in MC-dropout networks.

Implicit ensemble methods are considered appealing due to the reduced training costs since only one model needs to be trained. However, the use of explicit ensembles itself does not incur much additional cost in reality [108], as compared to single learners or implicit ensembles. The development of Machine Learning (ML) models (including implicit ensembles) usually involves Design Space Exploration (DSE), e.g., architecture search [110], hyperparameter tuning [4], (training-time) data augmentation [98] and k-fold cross-validation [79]. The model instances generated during the DSE processes can be used to construct explicit ensembles; in this respect, the advantages of implicit ensembles over explicit ones is not so significant.

Inducing Diversity in an Ensemble

Diversity is recognized as one of the key factors that contribute to the success of ensemble approaches [8]. As illustrated in Figure 3.1, the diversity among ensemble members is crucial for improved detection performance on o.o.d. data instances. In our empirical study to be described later, we will employ bagging [7] to induce diversity among ensemble members. Bagging [7] (or bootstrap aggregation) is a classical approach for creating diversity among ensemble members. The core idea is to construct models from different training datasets using *randomization*. In the original bagging approach [7], a random subset of the training samples is selected for training each member classifier. A later variant, the so-called “feature bagging” (a.k.a. random subspace method [29]) selects a random subset of the features for training. One famous application of bagging in ML is the Random Forest (RF) model. In this study, we will only use sample bagging in our experiments to induce diversity among ensemble classifiers.

Combining Base Models into an Ensemble

In ensemble analysis, one challenge is that the anomaly scores given by different models may not be directly comparable. This is known as the *normalization issue* [1]. This issue is more common with unsupervised or semi-supervised models, because the outputs from these models (e.g., the reconstruction errors from autoencoders) are often naturally unbounded. If the scores from different models are directly combined without normalization (e.g., by calculating the average or the maximum), models that give higher anomaly scores may be inadvertently favored [1]. The normalization issue is less concerning for supervised classification models that use a softmax layer to produce probability vectors whose values are bounded within the $[0, 1]$ interval; still, there are still concerns about whether or not these probability estimates

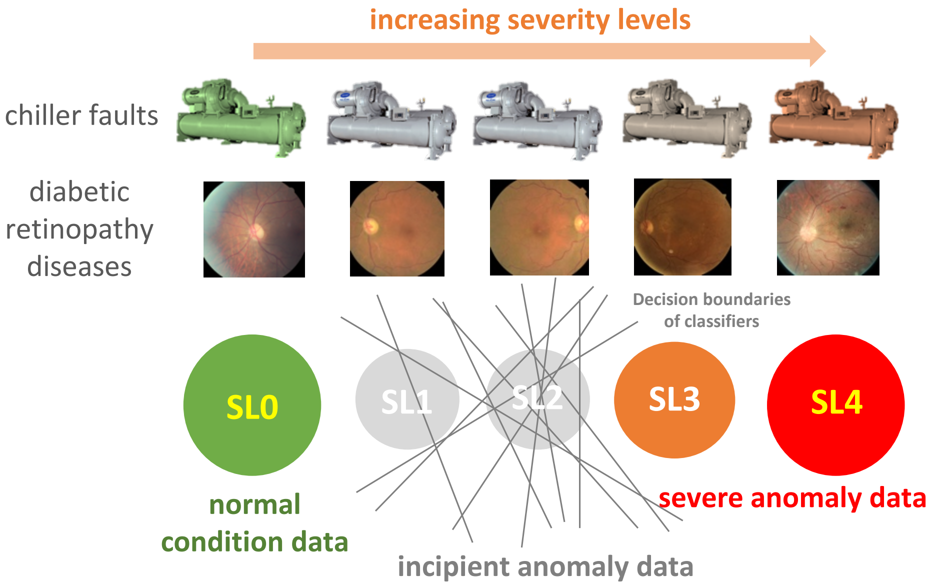


Figure 3.1: Illustration showing how an ensemble classifier can conceptually help detect incipient anomalies (i.e. SL1 & SL2 in this example). The gray lines represent the decision boundaries of base learners in the ensemble.

are well *calibrated* (known as the *calibration issue*). For the ensemble supervised classifiers in this study, we assume minimal impacts from the calibration issues.

On top of the normalization issue, how to properly aggregate the (normalized) anomaly scores from models in an ensemble, known as the *combination issue* [1], constitutes another major challenge in ensemble analysis. Depending on how the base learners are combined into an ensemble detector, we can classify the combination scheme into a 1) hard voting or a 2) soft voting scheme. In hard voting schemes, each base learner predicts a binary label $\hat{y} \in \{0, 1\}$ indicating whether an input example x is normal or not, while in soft voting schemes base learners outputs real-valued anomaly scores. In this work, we will mainly consider ensemble models made up of supervised classifiers, and *focus on how to properly obtain uncertainty estimates from the score vectors in order to better detect incipient anomalies.*

3.3 Uncertainty Estimation for Ensemble Learners

Suppose we have m data points for testing, and they are organized into a design matrix $\boldsymbol{X} \in \mathbb{R}^{m \times d}$. The outputs from the ensemble of detection models can be accordingly written as an $m \times K$ matrix $\hat{\boldsymbol{Y}}$, where K is the ensemble size. Note that entries in matrix $\hat{\boldsymbol{Y}}$ can either take discrete values from $\{0, 1\}$ (in a hard voting scheme) or take continuous values from $[0, 1]$ (in a soft voting scheme), depending on the nature of the underlying base learners.

Using superscripts to differentiate ensemble members and subscripts to differentiate data points, we can denote the rows and columns of matrix $\hat{\boldsymbol{Y}}$ as follows

$$\hat{\boldsymbol{Y}} \doteq \begin{bmatrix} | & | & \cdots & | \\ \hat{\boldsymbol{Y}}^{(1)} & \hat{\boldsymbol{Y}}^{(2)} & \cdots & \hat{\boldsymbol{Y}}^{(K)} \\ | & | & \cdots & | \end{bmatrix} = \begin{bmatrix} — & \hat{\boldsymbol{Y}}_1 & — \\ — & \hat{\boldsymbol{Y}}_2 & — \\ \cdots & \cdots & \cdots \\ — & \hat{\boldsymbol{Y}}_m & — \end{bmatrix} \tag{3.1}$$

where each $\hat{\mathbf{Y}}^{(k)} = \left[\hat{y}_1^{(k)} \; \hat{y}_2^{(k)} \cdots \hat{y}_m^{(k)}\right]^\intercal$ represents the predictions from the kth single learner ($k = 1, 2, \ldots, K$) on the m data points, and each $\hat{\mathbf{Y}}_i = \left[\hat{y}_i^{(1)} \; \hat{y}_i^{(2)} \cdots \hat{y}_i^{(K)}\right]$ represents the K predictions from the ensemble learner on x_i.

To come up with an uncertainty estimate for x_i, we calculate $U(\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \cdots, \hat{y}_i^{(K)})$ using U as the uncertainty metric. A number of metrics have been proposed in literature for estimating the prediction uncertainties of ensemble learners. In [51], the metrics are broadly classified into two categories: *confidence-based* and *disagreement-based* metrics. The former category is designed to capture the consensus of the individual learners in an ensemble, while the latter aims to measure the degree of disagreement among their predictions; however, the two seemingly unrelated goals can have a significant overlap. In this dissertation, we propose a rigorous categorization for these uncertainty metrics depending on their mathematical forms to unveil their differences and to enable further analyses. Some metrics (hereinafter referred to as type-1 metrics) rely only on the ensemble output $\hat{y}_i^e$, while others (referred to as type-2 metrics) take all single learner's outputs into account. Type-1 metrics use the ensemble output $\hat{y}_i^e$ to compute the confidence level, without the need to know what the individual predictions are. A negative aspect of these metrics is that the disagreement among individual learners can be hidden beneath the ensemble output $\hat{y}_i^e$.

Confidence Gap Metric (MEAN) An intuitive metric that measures the confidence of a classifier on input x is to see how close the prediction $\hat{y}^e$ is to the decision threshold τ^e. Here the superscripts in $\hat{y}^e$ and τ^e signify values associated with an ensemble classifier; in the special case where $K = 1$, the ensemble classifier degenerates to a single learner classifier. The smaller the gap $|\hat{y}_i^e - \tau^e|$ is, the higher the uncertainty with x_i. Since we prefer the convention that larger function values of $u^{\text{MEAN}}(x_i)$ corresponds to larger uncertainties, we define the uncertainty score under the margin metric can be formulated as

$$u^{\text{MEAN}}(x_i) \doteq 1 - |\hat{y}_i^e - \tau^e|\,, \tag{3.2}$$

where a constant 1 is added to the definition so that the uncertainty value $u^{\text{MEAN}}(x)$ is always positive. Since the ensemble prediction $\hat{y}_i^e$ is obtained by taking the average of the individual outputs of classifiers in the ensemble, we will hereinafter refer to this metric as MEAN.

Binary Cross-Entropy Metric (ENTROPY) The binary cross-entropy u^{ENTROPY} as a function of x_i takes the form

$$u^{\text{ENTROPY}}(x_i) \doteq -\left[\hat{y}_i^e \log \hat{y}_i^e + (1-\hat{y}_i^e)\log(1-\hat{y}_i^e)\right] \tag{3.3}$$

ENTROPY is equivalent to MEAN when the decision threshold $\tau^e = 0.5$. It can be easily proved that when $\tau^e = 0.5$,

$$u^{\text{MEAN}}(x_i) > u^{\text{MEAN}}(x_j) \Leftrightarrow u^{\text{ENTROPY}}(x_i) > u^{\text{ENTROPY}}(x_j). \tag{3.4}$$

In other words, when $\tau = 0.5$ the rankings assigned by u^{ENTROPY} and by $u^{\text{MEAN}}(x)$ to the data points are the same. Since we identify uncertain examples by finding the top-ranked data points, u^{ENTROPY} and $u^{\text{MEAN}}(x)$ are equivalent. The u^{ENTROPY} metric can be useful for evaluating prediction uncertainties when no decision threshold is a priori assigned.

Comparing to the type-1 metrics described above, type-2 metrics have the potential to give a more comprehensive characterization of the individual predictions (e.g., the disagreement among $\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \cdots, \hat{y}_i^{(K)}$). The following two existing type-2 metrics that are often used in literature, focus on quantifying the *disagreement* among individual learners in an ensemble and for this reason, may be able to address the shortcomings of type-1 metrics.

Variance Metric (VAR) The variance (or standard deviation) metric [52, 44] measures how spread out the individual learners' predictions are from the ensemble prediction $\hat{y}_i^e$. The uncertainty score of input x_i based on *sample variance* can be written as

$$u^{\text{VAR}}(x_i) \doteq \frac{1}{K-1}\sum_{k=1}^{K}\left[\hat{y}_i^{(k)} - \hat{y}_i^e\right] \tag{3.5}$$

Kullback–Leibler (KL) Divergence Metric (KL) Similar to the variance metric, the KL divergence metric [22] measures the deviation of individual learner's predictions from the ensemble output $\hat{y}_i^e$. The uncertainty score $s^{\text{KL}}(x_i)$ of input x_i under the KL divergence metric can be written as

$$u^{\text{KL}}(x_i) \doteq \frac{1}{K}\sum_{k=1}^{K} D_{\text{KL}}\left(y_i^{(k)} \middle\| \hat{y}_i^e\right) = \sum_{k=1}^{K}\hat{y}_i^{(k)} \log\frac{\hat{y}_i^{(k)}}{\hat{y}_i^e}. \tag{3.6}$$

A problem with VAR and KL is that they focus mainly on the disagreement among ensemble predictions but do not take in consideration the value of $\hat{y}_i^e$. Consider a scenario where the all ensemble members predict a probability of 0.5. Both VAR and KL will produce an uncertainty score of 0 and thus will not be able to capture any prediction uncertainties; in fact, this case where all learners give an output of 0.5 is highly uncertain. Next, we will compare two representative uncertainty metrics, MEAN and VAR, from a theoretical perspective.

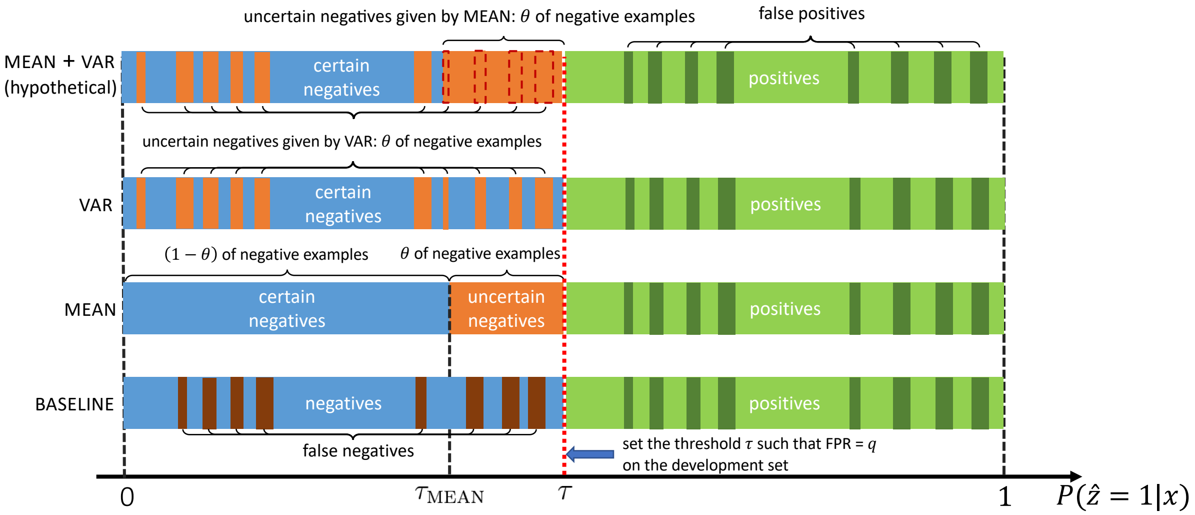

uncertain negatives given by MEAN: θ of negative examples
false positives
MEAN + VAR (hypothetical)
certain negatives
positives
uncertain negatives given by VAR: θ of negative examples
VAR
certain negatives
positives
$(1-\theta)$ of negative examples
θ of negative examples
MEAN
certain negatives
uncertain negatives
positives
BASELINE
negatives
positives
false negatives
set the threshold τ such that FPR = q on the development set
0
τ_{MEAN}
τ
1
$P(\hat{z}=1|x)$

Figure 3.2: Illustration showing the concepts in an uncertainty-informed decision framework.

We use Figure 3.2 to illustrate the relation between the various concepts introduced above. In the illustration, MEAN+VAR is a hypothetical uncertainty metric where the uncertain examples identified by MEAN+VAR are the *union* of the two sets of uncertain examples identified by MEAN and by VAR, not subject to the constraint imposed by q; see Figure 3.2 for an illustration. Therefore, it is at least as good as MEAN or VAR. If MEAN and VAR do not have much overlap, MEAN+VAR will identify many more false negatives than either of them alone.

3.4 Theoretical Analysis on Uncertainty Metrics MEAN and VAR

To model how different classifiers will respond to a given input x_i, we assume that the prediction $\hat{y}_i^{(k)}$ from classifier $\mathcal{M}^{(k)}$ is sampled from a beta distribution $\mathcal{B}(\alpha_i, \beta_i)$ that is characterized by two parameters by α_i and β_i. We further assume that $\alpha_i + \beta_i$ is fixed to the same constant value for all i's. Under this assumption, the Severity Level (SL) of the case represented by x_i can be characterized by a single parameter α_i, easing further analysis. The larger the value of α_i, the more severe the case of x_i is. When α_i and β_i are close, the case is ambiguous as the distribution shifts towards being symmetric (i.e. signifying much disagreement) rather than being one-sided.

The main theoretical contribution of this dissertation is presented in the following theorem, which implies that if x_i is more likely to be positive than x_j, then for ensemble learners of fixed size K, the upper bound on the probability of s_{MEAN} making a wrong decision is lower.

In other words, $s_{\textsc{mean}}$ is likely to be a more robust measure than $s_{\textsc{var}}$.

The choice of uncertainty metric U determines how examples are ranked and therefore affects the detection performance of false negatives. We expect the final ranking negative examples due to the uncertainty metric U matches the true severity ranking given by α_i. Taking a microscopic view into the ranking process, we consider two negative examples x_i and x_j, and assume x_i represents a less severe case than x_j. Under the above beta distribution assumption, we will have $\alpha_i < \alpha_j \leq \beta_j$. Our theoretical analysis will focus on the chance that x_i (the less ambiguous or more normal case) is considered more uncertain than x_j (the more ambiguous case). If the following theorem holds, then those correctly ranked by VAR are also likely to be correctly ranked by MEAN, indicating that MEAN is a preferable uncertainty metric to VAR.

Lemma 1. *Consider two inputs x_i, x_j with uncertainty score $s(x_i)$ and $s(x_j)$ estimated from K i.i.d. ensemble learners, and denote by $\Delta_{ij}(s) := \mathbb{E}[s(x_j) - s(x_i)]$ the difference of expected uncertainty score. If $\Delta_{ij}(s) > 0$, then $\Pr\left(s(x_i) > s(x_j)\right) = \mathcal{O}\left(\frac{Var(s(x_i)+Var(s(x_j)))}{\Delta_{ij}^2(s)}\right)$.*

The proof of Lemma 1 is provided in Section A.1 in the appendix. Intuitively, Lemma 1 states that if input x_j is more uncertain than x_i w.r.t. the expected uncertainty $\mathbb{E}[s(\cdot)]$, then the probability of the sample uncertain measure s making a wrong decision is bounded. Based on such result, we establish the following error bounds for uncertainty metrics MEAN and VAR.

Theorem 1. *Consider inputs x_i, x_j, with $y_i \sim \mathcal{B}(\alpha_i, \beta_i)$, $y_j \sim \mathcal{B}(\alpha_j, \beta_j)$, and $\alpha_i + \beta_i = \alpha_j + \beta_j$. Let $\Delta_{ij}(s) := \mathbb{E}[s(x_j) - s(x_i)]$ where $s(\cdot)$ denotes an uncertainty score estimated from K i.i.d. individual learners in an ensemble. If $\alpha_i < \alpha_j \leq \beta_j$, then*

$$\Delta_{ij}(s_{\textsc{mean}}) > \Delta_{ij}(s_{\textsc{var}}) > 0.$$

Furthermore, it holds that

$$\Pr\left(s_{\textsc{mean}}(x_i) > s_{\textsc{mean}}(x_j)\right) = \mathcal{O}\left(\frac{1}{K\Delta_{ij}^2(s_{\textsc{mean}})}\right) \tag{3.7}$$

$$\Pr\left(s_{\textsc{var}}(x_i) > s_{\textsc{var}}(x_j)\right) = \mathcal{O}\left(\frac{1}{K\Delta_{ij}^2(s_{\textsc{var}})}\right) \tag{3.8}$$

The proof of Theorem 1 is provided in Section A.2 in the appendix. A direct corollary of the above theorem states that under infinite ensemble size, using either MEAN or VAR as the uncertainty metric does not make a difference.

Corollary 1. *If the sample size is infinite, then under the conditions of Theorem 1, we have $s_{\textsc{mean}}(x_i) < s_{\textsc{mean}}(x_j) \Leftrightarrow s_{\textsc{var}}(x_i) < s_{\textsc{var}}(x_j)$.*

Chapter 4

Data

4.1 Chapter Overview

This chapter introduces the three datasets to be used for experimental studies in this dissertation. Two among the three are building fault datasets formatted as multivariate point data; each data point represents a single observation for the system states and sensor measurements. We also have a medical image dataset from Kaggle for diabetic retinopathy diseases, on which we will examine the proposed ensemble-based approach's performance for modern computer vision tasks.

4.2 RP-1043 Chiller Dataset

The RP-1043 Chiller Fault Dataset [12] (later also referred to as the "chiller dataset") is not publicly available for download but can be purchased from ASHRAE. In this dataset, sensor measurements of a typical cooling system—a 90-ton centrifugal water-cooled chiller—are recorded under fault-free and various fault conditions. The 90-ton chiller is representative of chillers used in larger installations [68], and consists of the following parts: evaporator, compressor, condenser, economizer, motor, pumps, fans, and distribution pipes etc. with multiple sensor mounted in the system. Figure 4.1 depicts the cooling system with sensors mounted in both evaporation and condensing circuits.

In the experimental data, eight different types of process faults were injected into the chiller, and each fault was introduced at four levels of severity (SL1 - SL4, from slightest to most severe). In our study, we only included the six faults shown in Table 4.2, because an earlier study by Reddy [87] found certain limitations with the excess oil and faulty TXV operation data. The condenser fouling (CF) fault was emulated by plugging tubes into condenser. The reduced condenser water flow rate (FWC) fault and reduced evaporator water flow rate (FWE) fault were emulated directly by reducing water flow rate in the condenser and evaporator. The refrigerant overcharge (RO) fault and refrigerant leakage (RL) fault were emulated by reducing or increasing the refrigerant charge respectively. The excess oil

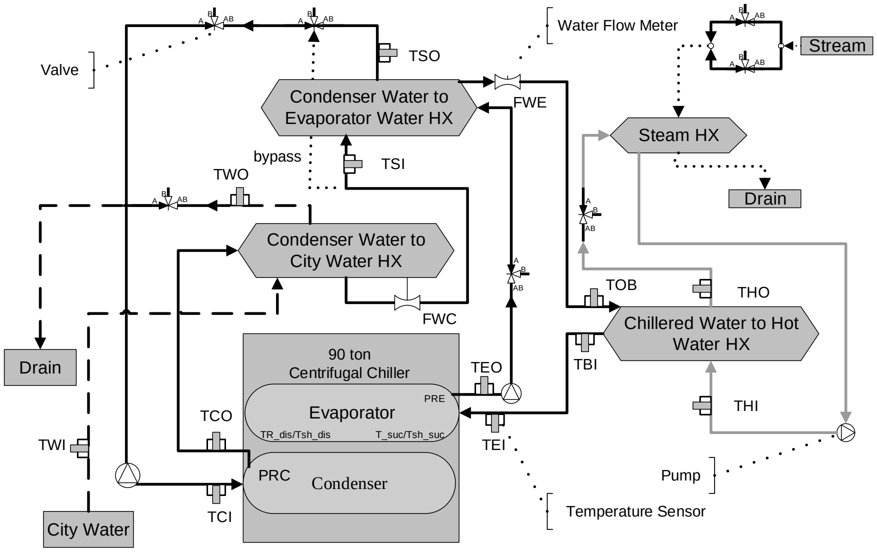


Figure 4.1: A schematic of the cooling system test facility and sensors mounted in the related water circuits [53].

(EO) fault was emulated by charging more oil than nominal. And the non-condensable in refrigerant (NC) fault was emulated by adding Nitrogen to the refrigerant.

All faults were tested at 27 different operating conditions with varying chiller thermal loads, chilled water outlet and inlet temperature settings. The data were collected at ten-second intervals, not only when the system had reached steady states, but also at transient states in between. We focused on only the steady-state data in this study. A more detailed review and discussion on the dataset can be found in [87].

4.3 Kaggle Diabetic Retinopathy (Kaggle-DR) Dataset

In addition to the RP-1043 Dataset, we also examined the efficacy of our proposed approach on medical images of diabetic retinopathy diseases. The Kaggle-DR dataset [13] comprises $88,702$ high-resolution images. Similar to chiller faults in the RP-1043 Dataset, the presence of diabetic retinopathy is also rated into five different SLs: *no-DR* (SL0), *mild* (SL1), *moderate* (SL2), *severe* (SL3), and *proliferate* (SL4), as illustrated in Figure 4.2. Again, SL3 and SL4 are considered severe anomalies. It is worthy to note one key difference between the chiller and the Kaggle-DR datasets: the SL1 cases in the Kaggle-DR dataset are considered a non-referable disease type and thus belong to the negative class, while in the chiller dataset SL1 data are considered positive (faulty).

Sensor	Description	Unit
TEI	Temperature of entering evaporator water	°F
TEO	Temperature of leaving evaporator water	°F
TCI	Temperature of entering condenser water	°F
TCO	Temperature of leaving condenser water	°F
Cond Tons	Calculated Condenser Heat Rejection Rate	Tons
Cooling Tons	Calculated City Water Cooling Rate	Tons
kW	Compressor motor power consumption	kW
FWC	Flow Rate of Condenser Water	gpm
FWE	Flow Rate of Evaporator Water	gpm
PRE	Pressure of refrigerant in evaporator	psig
PRC	Pressure of refrigerant in condenser	psig
TRC	Subcooling temperature	°F
T_suc	Refrigerant suction temperature	°F
Tsh_suc	Refrigerant suction superheat temperature	°F
TR_dis	Refrigerant discharge temperature	°F
Tsh_dis	Refrigerant discharge superheat temperature	°F

Table 4.1: Descriptions of variables used as features in the chiller dataset

Fault Types	**Identifier**	**Normal Operation**
Reduced Condenser Water Flow	*FT-FWC*	270 gpm
Reduced Evaporator Water Flow	*FT-FWE*	216 gpm
Refrigerant Leak	*FT-RL*	300 lb
Refrigerant Overcharge	*FT-RO*	300 lb
Condenser Fouling	*FT-CF*	164 tubes
Non-condensables in System	*FT-NC*	No nitrogen

Table 4.2: The six chiller faults used in our study

Data Formatting and Preprocessing The images in the Kaggle-DR dataset come in various resolutions, and each image consists of several million pixels. To save the time for loading data during model training, a preprocessing step was done before our experimental study to unify all images into square-shaped images with resolutions 224 × 224 or 384 × 384, two resolutions commonly used in computer vision that can keep much of the detailed information within the original image data.

The original image data comes in either of the two formats as exemplified in Figure 4.3. In the first format as shown in Figure 4.3a, the entire fundus is visible in the image, while in the second format as shown in Figure 4.3b, part of the fundus is cropped out by the frame and is thus not visible. By using a simple rule-based detector, we were able to differentiate

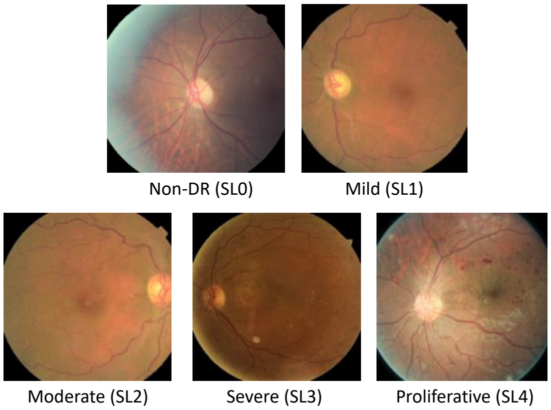
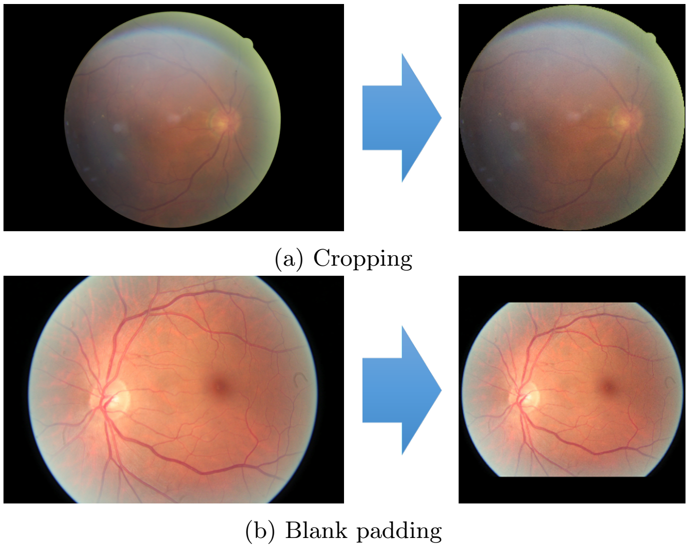


Figure 4.2: Example fundus images (preprocessed) that correspond to the five disease SLs.

between the two formats and treat them differently: for the first format, we cropped the original image such that the fundus will tightly fit inside the square, and for the second format blank strips were padded to make the image square-shaped and in a unified resolution.


(a) Cropping

(b) Blank padding

Figure 4.3: Preprocessing the fundus image data from the Kaggle-DR dataset [13].

Messidor-2 Diabetic Retinopathy Dataset The Messidor-2 Dataset [47] is another diabetic retinopathy image dataset that consists of 1, 748 images. The Messidor-2 Dataset is very similar to the Kaggle-DR dataset; Messidor-2 images are also graded into the five

SLs as in the Kaggle-DR dataset. The Messidor-2 Dataset was used in our experiment as an additional dataset to test the true generalization performance of the models trained on the Kaggle-DR dataset.

4.4 RP-1312 AHU Dataset

The Air Handling Unit (AHU) system is another important component of a building's Heating, Ventilation and Air Conditioning (HVAC) system. Its functionality is to regulate and circulate air to the indoor zones in a building. The schematic of a typical AHU system is depicted in Figure 4.4a that is configured for a Variable Air Volume (VAV) system that maintains the supply air temperature to the terminals for air-conditioning.

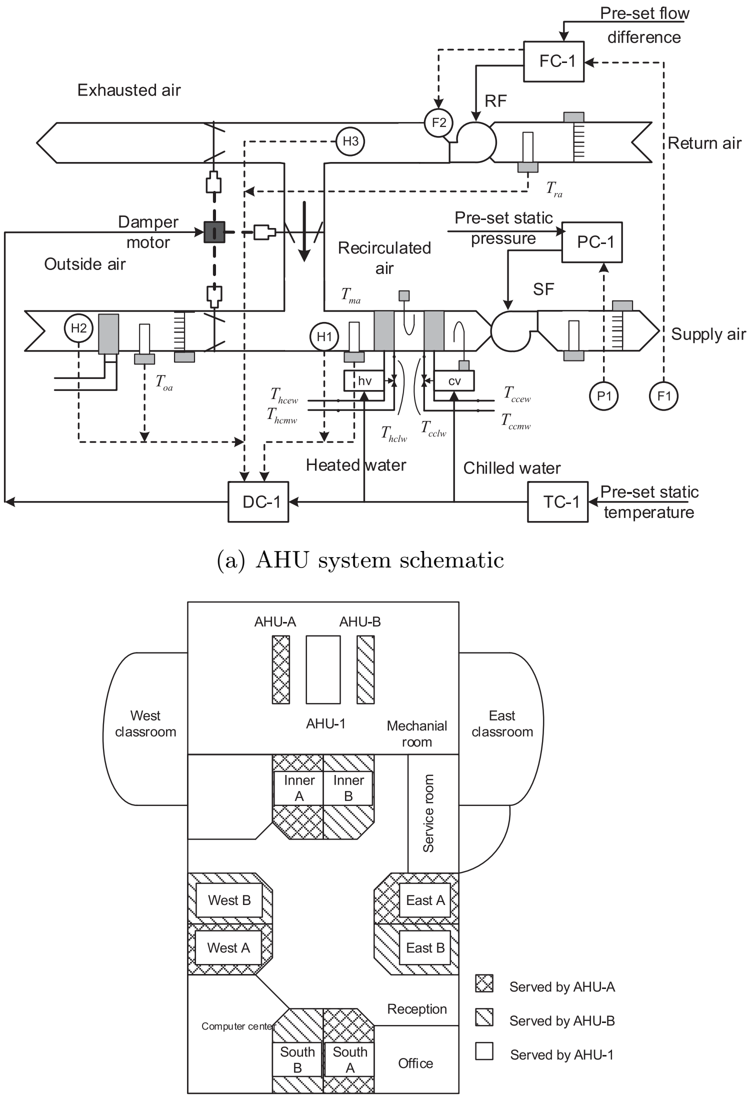
The testing site for creating the RP-1312 AHU Dataset [95] (later also referred to as the "AHU dataset") involved two AHUs, i.e., AHU-A and AHU-B as shown in Figure 4.4b. The two AHUs were operated under real weather and building load conditions. Faults were manually injected into the air-mixing box, the coils, and the fan sections of AHU-A (treatment group), while AHU-B (control group) was operated at nominal states.

The AHU dataset includes 16 commonly encountered AHU faults across three seasons, spring, summer, and winter. For the experimental study to be described in Chapter 7, we will treat the data from each season as an independent dataset. We will respectively name these three datasets as *AHU-spring*, *AHU-summer* and *AHU-winter*.

A detailed list of the 16 fault types studied by the AHU dataset is given in Table 4.3, where each fault is assigned a unique identifier. We can also see that the faults occurring in different seasons do not fully overlap. There are faults that exist only in spring but not in summer or winter, e.g., SP-FT-1. There are also faults that appear in all three seasons such as the "exhaust air damper stuck" fault.

Table 4.3: Fault Types Studied in the RP-1312 AHU Dataset

Fault Types	**Spring**	**Summer**	**Winter**
Outside air damper leak		*SU-FT-1*	*WT-FT-1*
Outside air temperature sensor bias	*SP-FT-1*		
Outside air damper stuck	*SP-FT-2*		*WT-FT-2*
Exhaust air damper stuck	*SP-FT-3*	*SU-FT-2*	*WT-FT-3*
Cooling coil valve control unstable	*SP-FT-4*	*SU-FT-3*	
Cooling coil valve reverse action		*SU-FT-4*	
Cooling coil valve stuck	*SP-FT-5*	*SU-FT-5*	*WT-FT-4*
Heating coil valve leaking		*SU-FT-6*	
Return fan at fixed speed	*SP-FT-6*	*SU-FT-7*	
Return fan complete failure	*SP-FT-7*	*SU-FT-8*	
Air filter area block fault	*SP-FT-8*		
Mixed air damper unstable	*SP-FT-9*		
Sequence of heating and cooling unstable	*SP-FT-10*		
Supply fan control unstable	*SP-FT-11*		
Heating coil fouling			*WT-FT-5*
Heating coil reduced capacity			*WT-FT-6*



(a) AHU system schematic

(b) RP-1312 testing site layout

Figure 4.4: (a) A typical single-duct VAV AHU system [53], and (b) the schematic of the testing site used for creating the RP-1312 AHU Dataset [95].

Chapter 5

Fault Detection for Chiller Systems

5.1 Chapter Overview

In anomaly detection applications, it is common to encounter anomaly data examples whose symptoms are graded into different Severity Levels (SLs). For example, in the chiller dataset and the Kaggle-DR dataset described earlier in Chapter 4, the anomalies (faults or diseases) are categorized into four different SLs, from SL1 (slightest) to SL4 (most severe).

As mentioned earlier, the ability of accurately assessing the severity of faults/diseases is important for anomaly detection applications, yet the task is very difficult especially on low-severity examples. As visualized in Figure 5.1, SL1 data clusters are much closer to the normal cluster than to their corresponding SL4 clusters. An anomaly detection system needs to be very sensitive so as to identify the low-severity faults; at the same time, it should keep the number of false positives (false alarms) low, which makes the design and implementation of such decision systems a challenging task.

This chapter will present and discuss experiment results on the chiller dataset described in the previous chapter. The presented research in this chapter is based on the author's recent papers [42, 88, 43]. The results from the proposed ensemble methods will be analyzed and compared from multiple perspectives, and be correlated with the theoretical results discussed earlier in Chapter 3.

5.2 Data Setup

ASHRAE RP-1043 Chiller Dataset

For our case study to be presented in this chapter, we used the ASHRAE RP-1043 Chiller Dataset to study the detection of incipient chiller faults, and studied the six faults (FWE, FWC, RO, RL, CF, NC) as listed in Table 4.2 as the anomaly (positive) class. Each fault in the dataset was introduced at four levels of severity (SL1–SL4, from the slightest to the most severe). We considered SL3 and SL4 cases as severe faults, and SL1 and SL2 cases

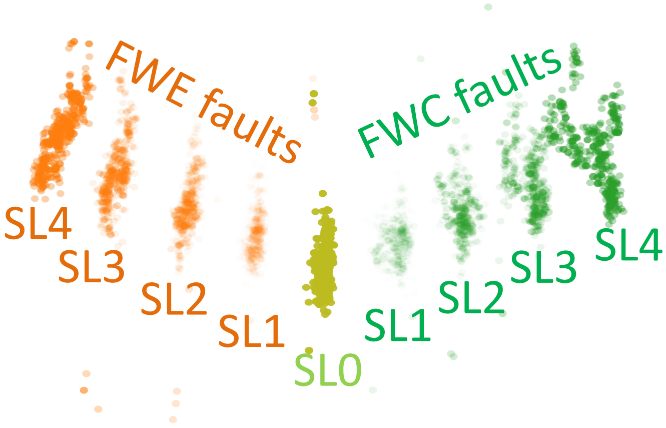


Figure 5.1: Visualization of part of the dimension-reduced RP-1043 chiller data [12] where the "severity spectra" for two fault conditions (FWE and FWC faults) are clearly visible. The normal condition and two fault conditions (each with four SLs) are shown.

as incipient faults. For feature selection, we followed our previous study [44] and used the sixteen key features as the same six faults therein for training our models.

Detailed descriptions of the sixteen selected features and the six fault types are given in Table 4.1 and Table 4.2 in the previous chapter. Since we are concerning fault detection (binary classification task) rather than fault classification (multi-class classification task), we put fault data of all fault types and SLs into one fault class.

To give the readers an intuitive view of how the chiller data are distributed, we employ the Linear Discriminant Analysis (LDA) algorithm [31] to reduce part of the data into two dimensions for visualization in Figure 5.1. We can observe a general trend in the visualization: data points will deviate further away from the normal cluster when the corresponding fault develops into a higher SL.

Before training, a few data points that are obvious outliers were first removed. Then the data are standardized before being used for training.

Data Partitioning

As common practice, we divide the chiller dataset into a *development set* and a *test set*. The test set can be further divided into two parts; one contains only the normal data (SL0) and the non-incipient anomalies (SL3 & 4), the other containing only the incipient anomalies; see Figure 5.2 for an illustration. All five SLs are present in the development set data. To model how the availability of incipient anomaly data affected the detection performance, we introduce a parameter, the incipient anomaly ratio ρ, to control the proportion of incipient anomaly data that enters the development set.

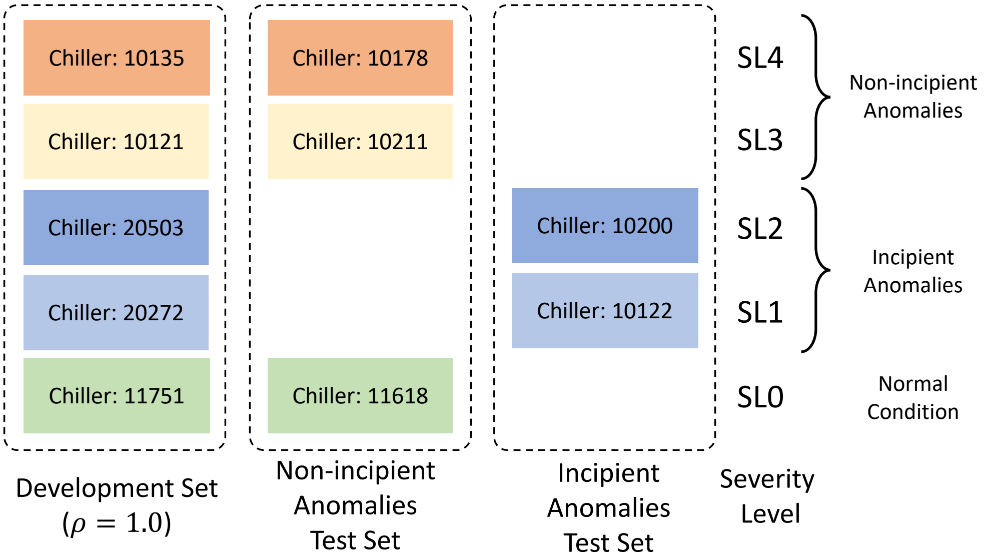


Figure 5.2: Layout of the development set and the test set data resulting from the partitioning the chiller data.

In our experiment, we test $\rho = 0, 0.2, 0.4, 0.6, 0.8, 1.0$. It is worthy to note that when $\rho = 0$, no incipient anomaly data appear in the development set; in other words, when $\rho = 0$ the incipient anomaly data become out-of-distribution (o.o.d.) because they are not present at training time. We specifically include this scenario in our study to see if the models can learn useful knowledge from only non-incipient anomalies that is useful for identifying incipient anomalies that are absent from the training distribution.

5.3 Model Setup and Training

Since the chiller data assume a multivariate point data format, common classification models like Decision Trees (DTs) and Neural Networks (NNs) can be employed for differentiating between the normal state and the fault states. In this study, we will build ensemble models using DT, NN and One-Class Support Vector Machine (OC-SVM) models as base learners. In this study, we use the `scikit-learn` package [74] for implementing the Machine Learning (ML) models used in our experiments. The base learners are all implemented by using existing modules in `scikit-learn`.

Creation of Ensemble Models

To build and evaluate ensemble models, we first will need a library of diversified base learners. The *model selection* process can be utilized to build such a library. Real-world ML practitioners perform extensive model selection to search for models using the development set data, and select those that are more likely to perform well on test sets. We employ a similar workflow in our empirical study. For each model class under study, we sweep over a wide range of hyperparameter settings, pick out a set of best-performing hyperparameters (and remove bad-performing ones), and assess whether or not our proposed ensemble method could deliver consistent performance improvement compared to the baseline scenarios.

In our empirical study, we evaluate ensembles of four different sizes: $K = 5, 10, 15, 25$, and compare their performance to the single learner case ($K = 1$). To carry out the hyperparameter search, we utilized the `GridSearchCV` module in `scikit-learn` to sweep over the prescribed hyperparameter space. For DT models, we swept the `max_depth` parameter over the range $\{8, 10, 12, 15, 20\}$ and attempted various parameters configurations such as `criterion` (measuring the quality of split) and `splitter` (strategy used to choose the split at each node). For NN models (multilayer perceptrons), we tried several different network topologies with depth ranging from 2 to 5, various batches sizes (32, 64, 128) and optimizer settings (`sgd` or `adam`). For OC-SVM models with Radial Basis Function (RBF) kernels, we conducted a grid search over parameters ν and γ [40].

After the grid search, the top R sets of hyperparameters for each model type are picked out and used for constructing the base learners for bagging ensembles. The bagging ensembles in this study are implemented using the `Bagging` module from `scikit-learn`, which enabled us to create bagging models with different types of base learners. The sizes of random sample subsets for training each base model can be specified through the `max_samples` argument.

5.4 Performance Evaluation

False Negative Rate (FNR)

For evaluating the detection performance on the chiller dataset, we first report the FNR on the test distribution data that include both incipient and non-incipient anomalies. Recall that the FNR is defined as the fraction of positive (fault) examples that are wrongly classified as negative (normal), i.e.,

$$\text{False Negative Rate (FNR)} = \frac{FN}{TP + FN}.$$

We examine the FNR for both incipient and non-incipient chiller anomalies under different settings of False Positive Rate (FPR) ratio q and the incipient anomaly ratio ρ, and show the results as box plots in Figure 5.3 for DT ensembles and in Figure 5.4 for NN ensembles. We show the FNR results on incipient anomaly cases for single learners ($K = 1$) and for ensemble learners ($K = 5, 25$); the FNR for non-incipient anomalies (see the left columns for both plots) are close to zero except for some single learner DT models, which indicates near-perfect classification performance between SL0 (normal conditions) and SL3 & SL4 (non-incipient anomalies).

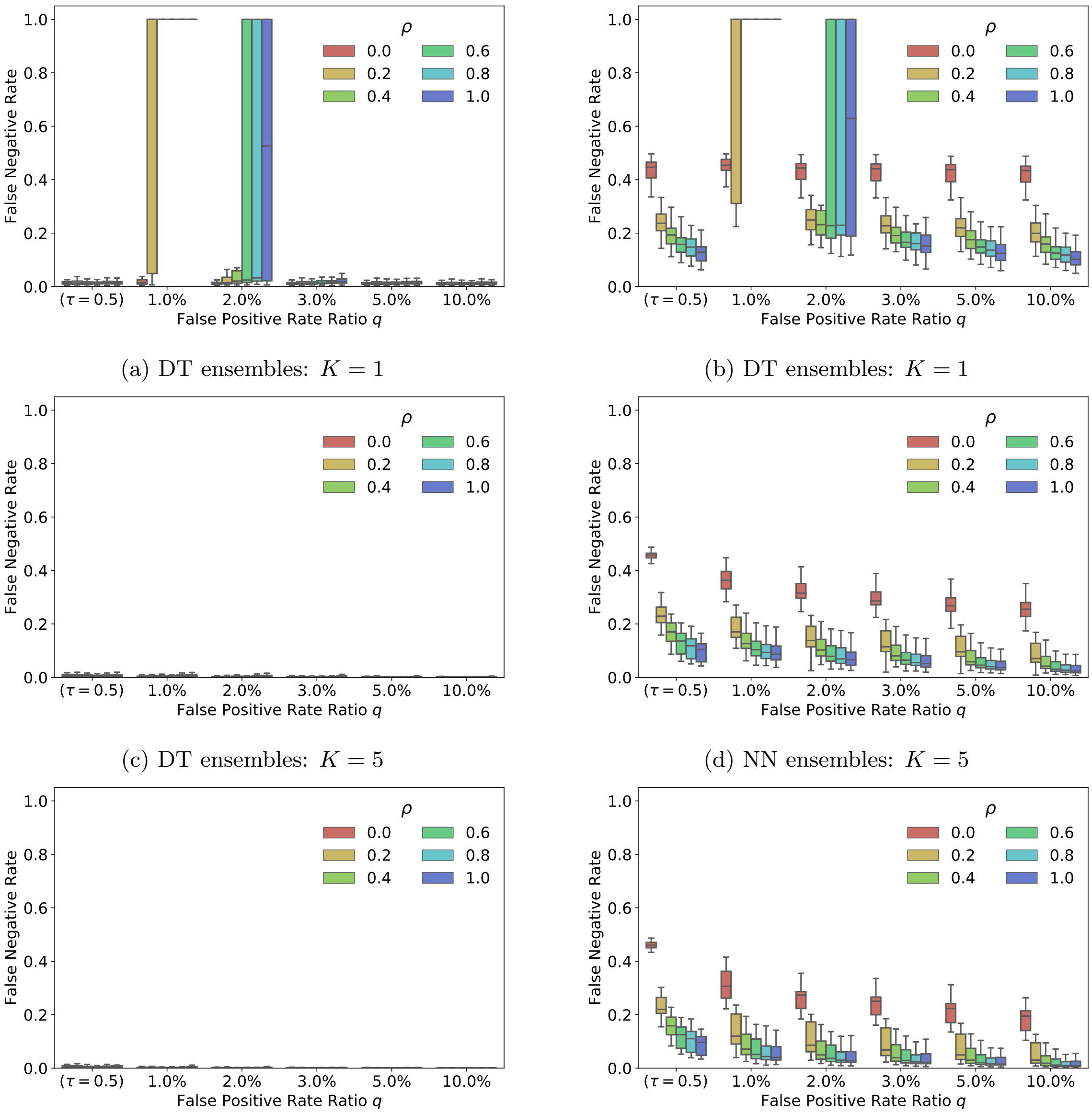
By comparing the single learner and ensemble models ($K = 1$ vs. $K = 5, 25$), we can immediately see the expected performance improvement for ensemble learners over single learners, especially for DT models since single DTs are relatively weak classifiers. In addition, we can observe a decreasing trend in FNR with increasing q in all cases, which indicates that an increased number of incipient anomalies can be detected when we lower the detection threshold τ (in other words, making the classifiers work at a more sensitive operating point). When the number of false negatives is not a big concern, increasing q (or lowering τ) is a practical way to improve the detection performance on incipient anomalies, without undermining the performance on non-incipient anomalies.

Number of Remaining False Negatives

The next performance index we evaluate is the number of remaining false negatives after applying uncertainty estimation. In our case study, we tested $\theta \in \{1\%, 2\%, 5\%, 10\%\}$. The numbers of remaining false negatives are obtained by assuming that all identified uncertain false negatives will receive corrected labels. We are interested in knowing the number of remaining false negatives because these are mistakes that the uncertainty estimation techniques fail to identify. We visualize the performance variations of the trained models as box plots in Figure 5.5 for DT models and in Figure 5.6 for NN models, respectively.

As displayed in the plots, besides MEAN and VAR we also include two other scenarios, BASELINE and MEAN+VAR, that respectively set the lower bound and the upper bound of performance of MEAN and VAR. Under BASELINE, no uncertainty information from output probabilities is utilized, i.e. $\theta = 0$. MEAN+VAR is a hypothetical uncertainty metric where the uncertain examples identified by MEAN+VAR are the *union* of the two sets of uncertain examples identified by MEAN and by VAR, not subject to the constraint imposed by q; see Figure 3.2 for an illustration. Therefore, it is at least as good as MEAN or VAR. If MEAN and VAR do not have much overlap, MEAN+VAR will identify many more false negatives than either of them alone; however, we can see from Figures 5.5 & 5.6 that this is not the case. The results given by MEAN+VAR do not have much improvement over those given by MEAN, indicating that many of the false negatives identified by VAR are also captured by MEAN, which matches the expectation of Theorem 1.

An immediate observation from Figures 5.5 & 5.6 is that ensemble learning can achieve substantial performance improvement even for small ensemble sizes ($K = 5$). For $K > 5$, we can still see significant improvement when K grows larger for tree ensembles; however, for NN ensembles the marginal improvement from increasing ensemble sizes is smaller, which is probably due to the fact that individual DT classifiers are relatively weak compared to individual NN classifiers. By comparing the performance of MEAN and that of VAR in the plots, we can see that MEAN leads to fewer remaining false negatives in general; in other words, the MEAN uncertainty metric can identify more false negatives than VAR.



(a) DT ensembles: $K = 1$

(b) DT ensembles: $K = 1$

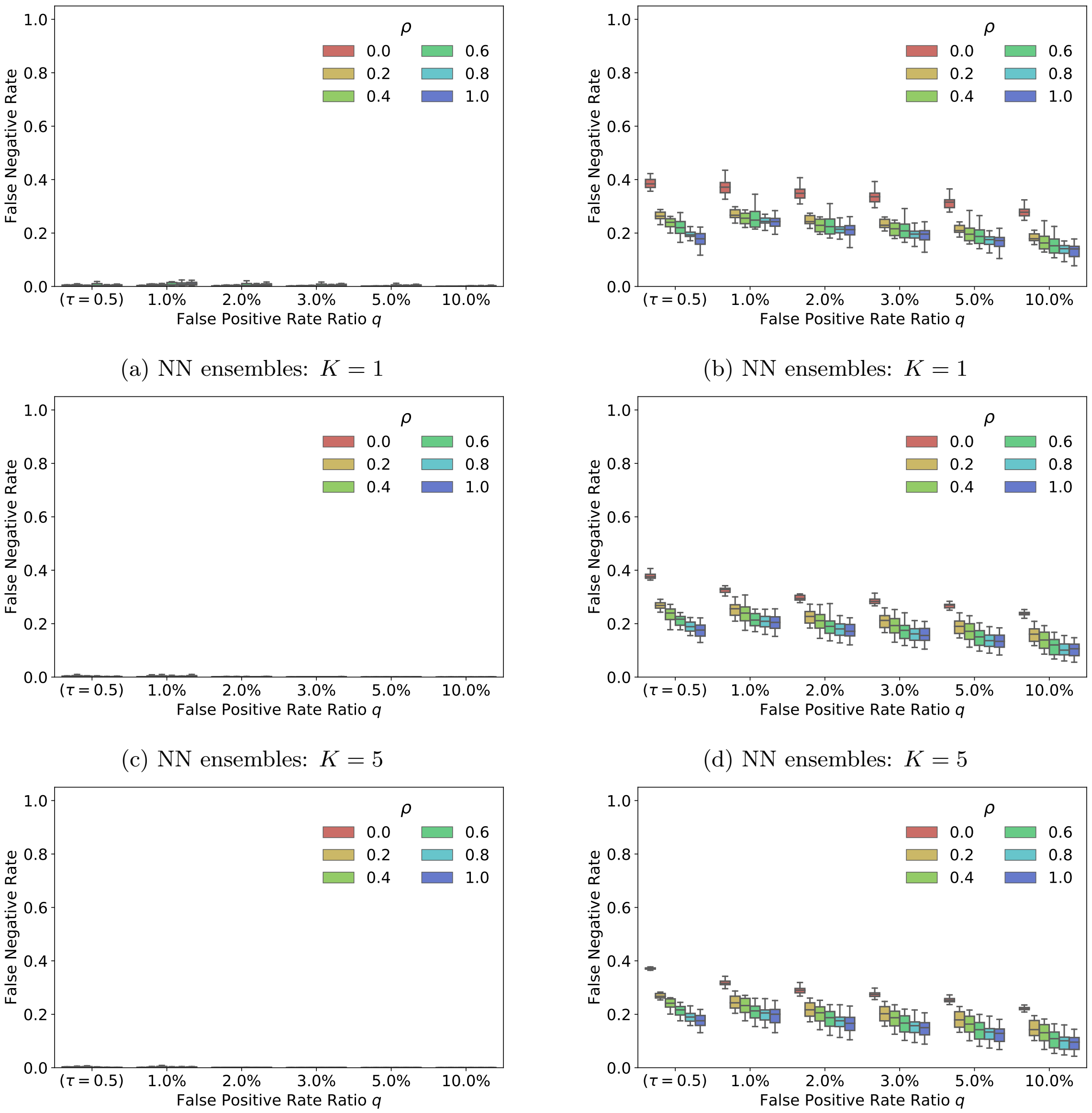
(c) DT ensembles: $K = 5$

(d) NN ensembles: $K = 5$

(e) DT ensembles: $K = 25$

(f) DT ensembles: $K = 25$

Figure 5.3: Detection performance in terms of FNR on incipient anomalies for single learners ($K = 1$) and for ensemble models ($K = 25$).



(a) NN ensembles: $K = 1$

(b) NN ensembles: $K = 1$

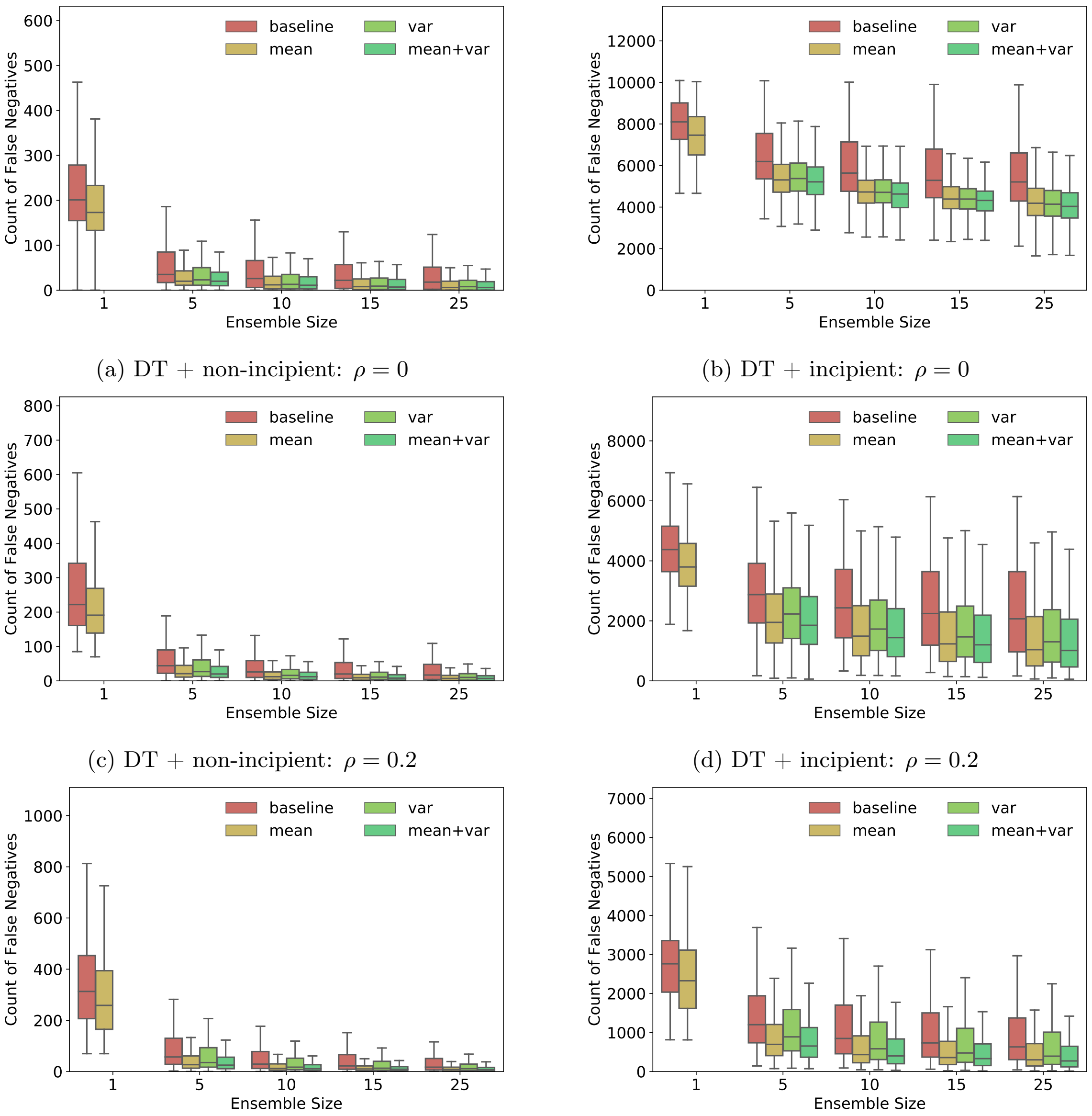
(c) NN ensembles: $K = 5$

(d) NN ensembles: $K = 5$

(e) NN ensembles: $K = 25$

(f) NN ensembles: $K = 25$

Figure 5.4: Detection performance in terms of FNR on incipient anomalies for single learners ($K = 1$) and for ensemble models ($K = 25$).



(a) DT + non-incipient: $\rho = 0$

(b) DT + incipient: $\rho = 0$

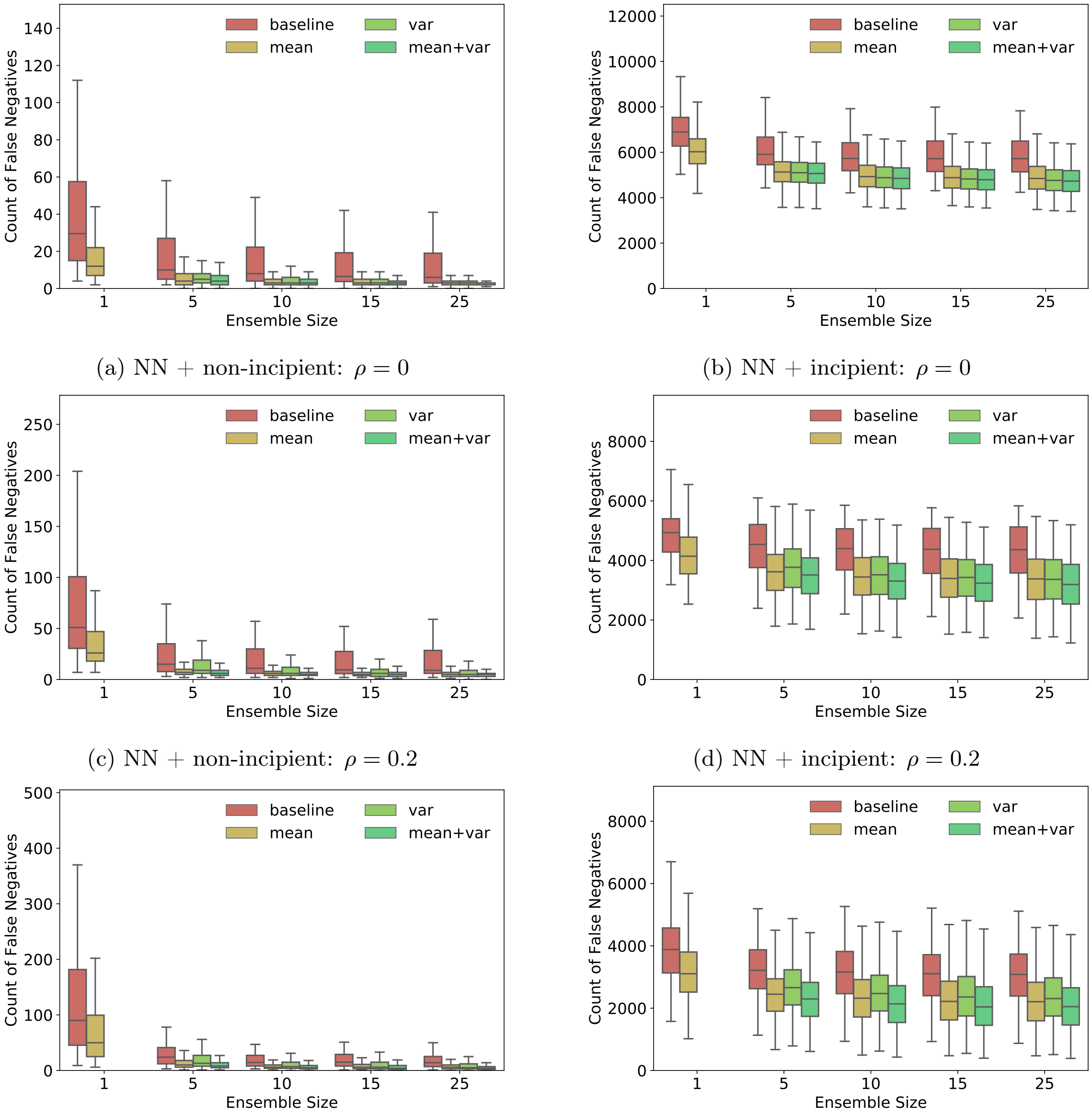
(c) DT + non-incipient: $\rho = 0.2$

(d) DT + incipient: $\rho = 0.2$

(e) DT + non-incipient: $\rho = 1.0$

(f) DT + incipient: $\rho = 1.0$

Figure 5.5: Box plots showing the number of certain false negatives (incipient anomalies wrongly classified as negative) after the rest are identified by uncertainty estimation for the chiller dataset.



(a) NN + non-incipient: $\rho = 0$

(b) NN + incipient: $\rho = 0$

(c) NN + non-incipient: $\rho = 0.2$

(d) NN + incipient: $\rho = 0.2$

(e) NN + non-incipient: $\rho = 1.0$

(f) NN + incipient: $\rho = 1.0$

Figure 5.6: Box plots showing the number of certain false negatives (incipient anomalies wrongly classified as negative) after the rest are identified by uncertainty estimation for the chiller dataset.

False Negative Precision (FN-precision)

Although the results and analyses above show that MEAN compared to VAR can identify more false negatives among incipient anomalies, it is not sufficient to show that MEAN is more preferable to VAR because the increased number of corrected false negatives may simply be a consequence of a large number of uncertain negatives being identified; in an extreme scenario, if all negative data points are marked as uncertain negatives, then all false negatives can be corrected. Therefore, we use the FN-precision metric to measure how precisely each model can identify the false negatives. As can be seen from Figure 5.7 (for DT ensembles) and Figure 5.7 (for NN ensembles), MEAN again outperforms VAR in terms of FN-precision.

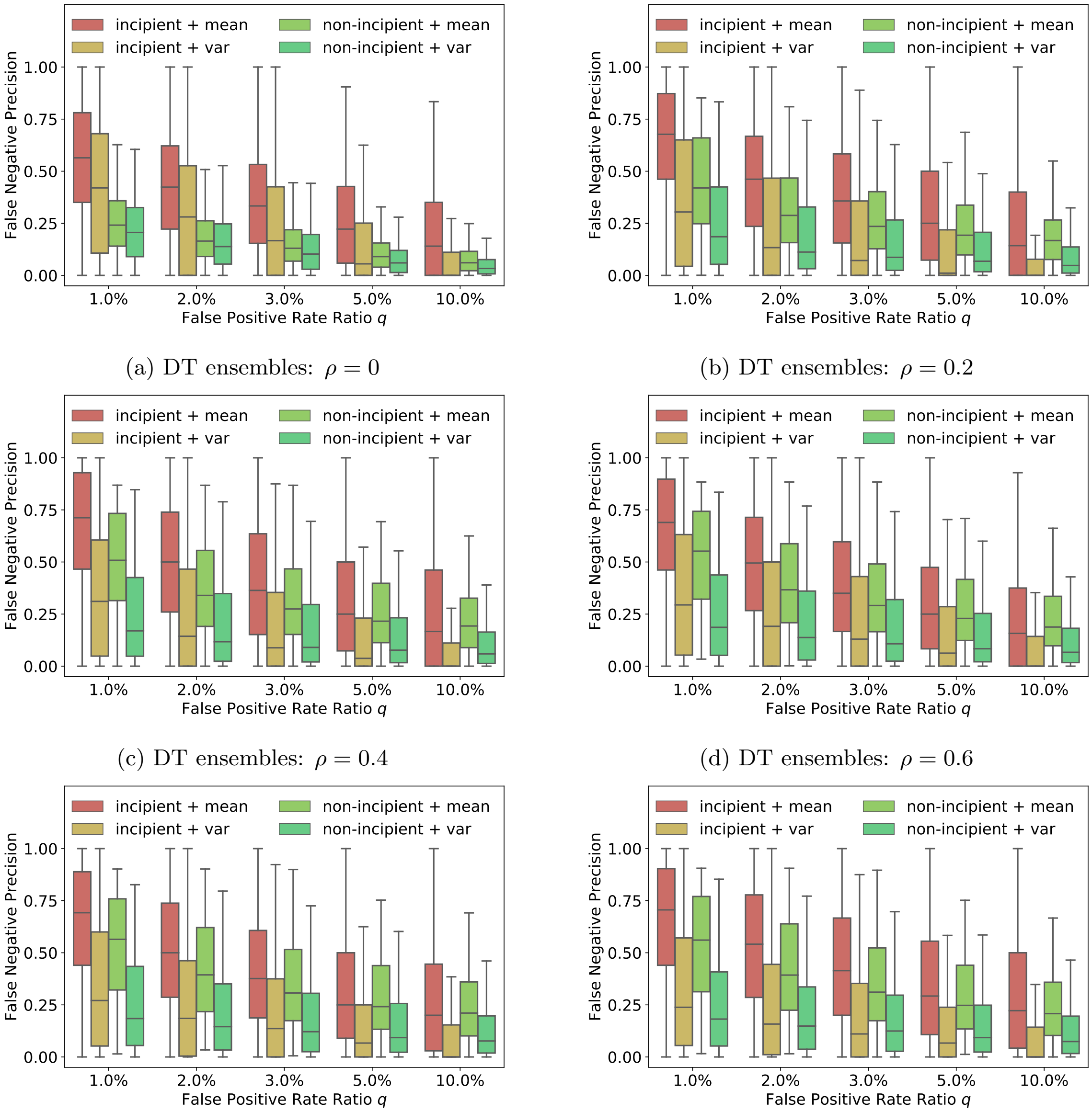
5.5 Detection Performance of One-Class Classifiers

As a comparative study, we also experimented using OC-SVM [82, 40], a popular one-class model for semi-supervised and unsupervised learning, to learn a boundary of the normal data points (i.e., the inliers) that can be used to separate the inliers from the outliers for the chiller dataset. Again, we conducted a grid search over various hyperparameter settings and picked out the best-performing OC-SVM models.

In Figure 5.9, we visualize the performance of OC-SVMs ensembles of three different sizes $K = 1, 5, 25$, and show how the detection performance in terms of FNR varies with the FPR ratio q. As with other learners for the chiller dataset, we used sample bagging to induce diversity among ensemble OC-SVM learners. The experiment results for ensemble learners, however, did not demonstrate much improvement over the single learner cases. By comparing the results for OC-SVM to those for DT and NN ensembles, we can see that OC-SVM gives inferior detection performance for both incipient and non-incipient anomalies. A detailed discussion on OC-SVM and other one-class methods (e.g., autoencoders) is beyond the scope of this dissertation. We believe there is still potential in using one-class methods for anomaly detection, which may become promising directions for future research.

5.6 Summary

We show in this chapter that, incipient chiller anomalies (faults) can pose critical challenges to supervised anomaly detection systems built upon ML techniques, especially under situations where incipient anomaly examples are absent from the training data. The resulting ML models (including DTs and NNs) can easily mistake incipient anomalies for normal ones, which can lead to costly consequences if the ideal time for intervention or treatment is missed. To address this challenge, we study how to exploit the uncertainty information from ensemble learners to identify incipient anomalies that are potentially wrongly classified.



(a) DT ensembles: $\rho = 0$

(b) DT ensembles: $\rho = 0.2$

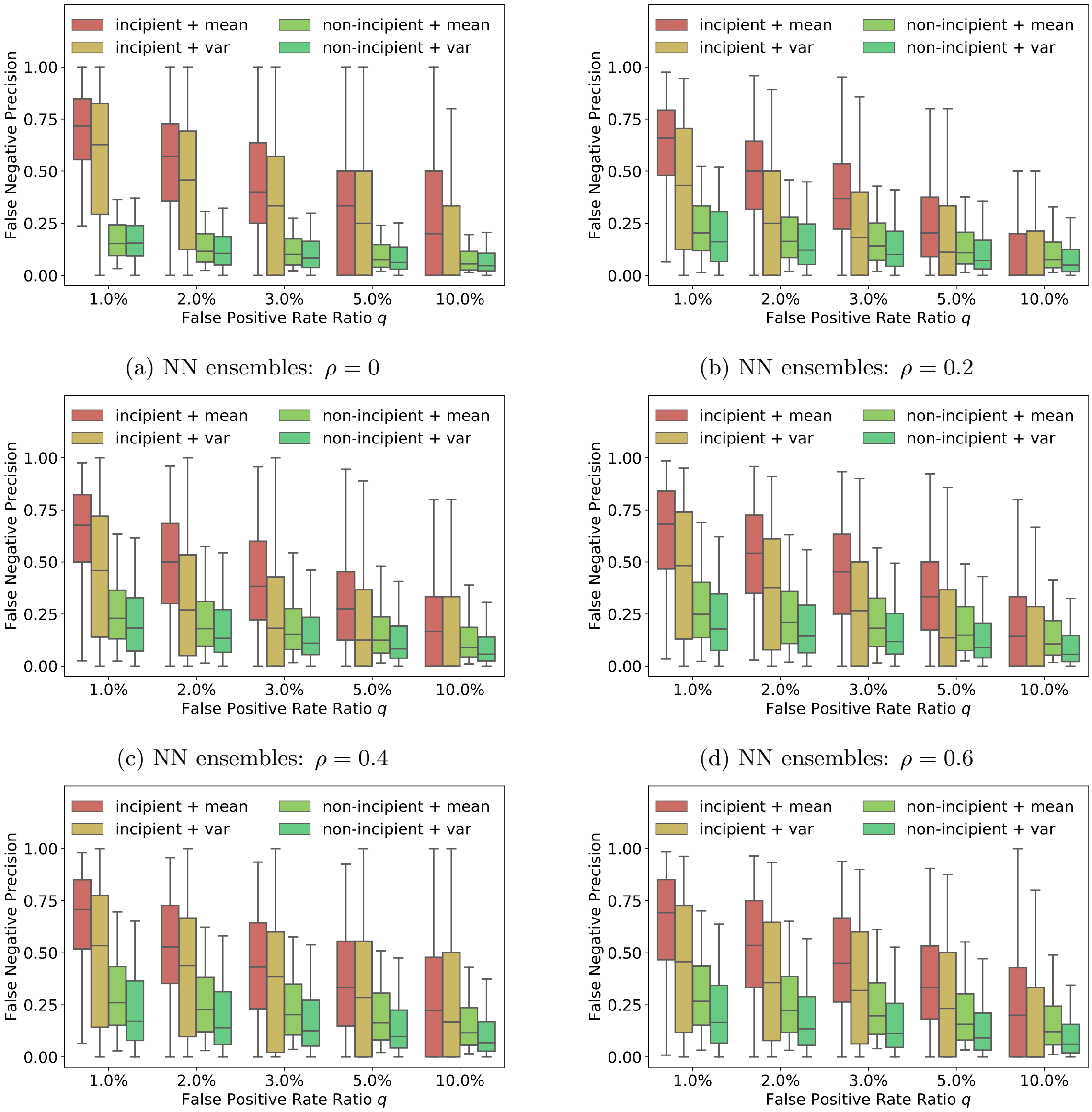
(c) DT ensembles: $\rho = 0.4$

(d) DT ensembles: $\rho = 0.6$

(e) DT ensembles: $\rho = 0.8$

(f) DT ensembles: $\rho = 1.0$

Figure 5.7: Box plots showing the FN-precision metric for DT ensemble classifiers ($K = 5, 10, 15, 25$) under different settings of the FPR percentile q for the two datasets. Different colors indicate performance indices given by MEAN and VAR for the incipient and the non-incipient data.



(a) NN ensembles: $\rho = 0$

(b) NN ensembles: $\rho = 0.2$

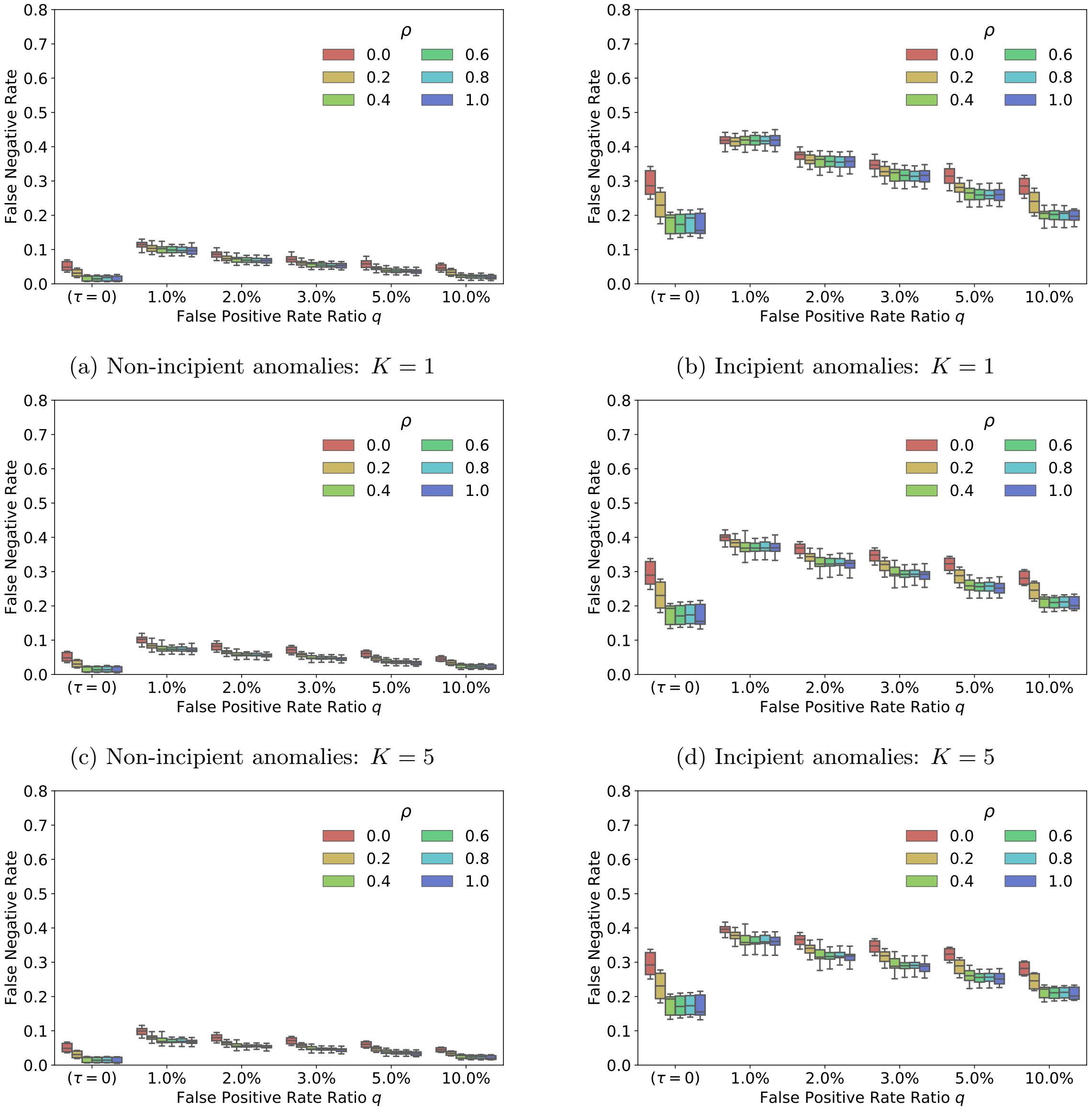
(c) NN ensembles: $\rho = 0.4$

(d) NN ensembles: $\rho = 0.6$

(e) NN ensembles: $\rho = 0.8$

(f) NN ensembles: $\rho = 1.0$

Figure 5.8: Box plots showing the FN-precision metric for ensemble NN classifiers ($K = 5, 10, 15, 25$) under different settings of the FPR ratio q for the two datasets. Different colors indicate performance indices given by MEAN and VAR for the incipient and the non-incipient data.



(a) Non-incipient anomalies: $K = 1$

(b) Incipient anomalies: $K = 1$

(c) Non-incipient anomalies: $K = 5$

(d) Incipient anomalies: $K = 5$

(e) Non-incipient anomalies: $K = 25$

(f) Incipient anomalies: $K = 25$

Figure 5.9: The performance of OC-SVMs classifiers on non-incipient anomalies (top panel) and incipient anomalies (bottom panel). Box plots for ensembles of three different sizes $K = 1, 5, 25$ are displayed.

Chapter 6

Diabetic Retinopathy Diagnosis

6.1 Chapter Overview

This chapter will present and discuss experiment results on the diabetic retinopathy data described in Chapter 4. The presented research in this chapter is based on the author's recent papers [42, 88]. The structure of our experimental study will largely follow the study on chiller data described in Chapter 5. In this study, we will be using Convolutional Neural Network (CNN) base learners to build ensembles. The results from our proposed ensemble methods will be analyzed from multiple perspectives, and be correlated with the theoretical results discussed earlier in Chapter 3.

6.2 Data Setup

Data Partitioning

Like what we have done with the chiller data, we divided the Kaggle-DR dataset into a *development set* and a *test set*. The test set was further divided into two parts; one contained only the normal data (SL0) and the non-incipient anomalies (SL3 & 4), the other containing only the incipient anomalies; see Figure 6.1 for an illustration. All five SLs were present in the development set data. To model how the availability of incipient anomaly data affected the detection performance, we introduced the incipient anomaly ratio ρ, a parameter that controls the proportion of incipient anomaly data that enters the development set.

In our experiment, we tested $\rho = 0, 0.2, 0.4, 0.6, 0.8, 1.0$. It is worthy to note that when $\rho = 0$, no incipient anomaly data appeared in the development set; in other words, the incipient anomaly data became o.o.d. when $\rho = 0$ because they were not present at training time. We specifically included this scenario in our study to see if the models could learn useful knowledge for identifying incipient anomalies from only the normal condition data and the non-incipient anomalies.

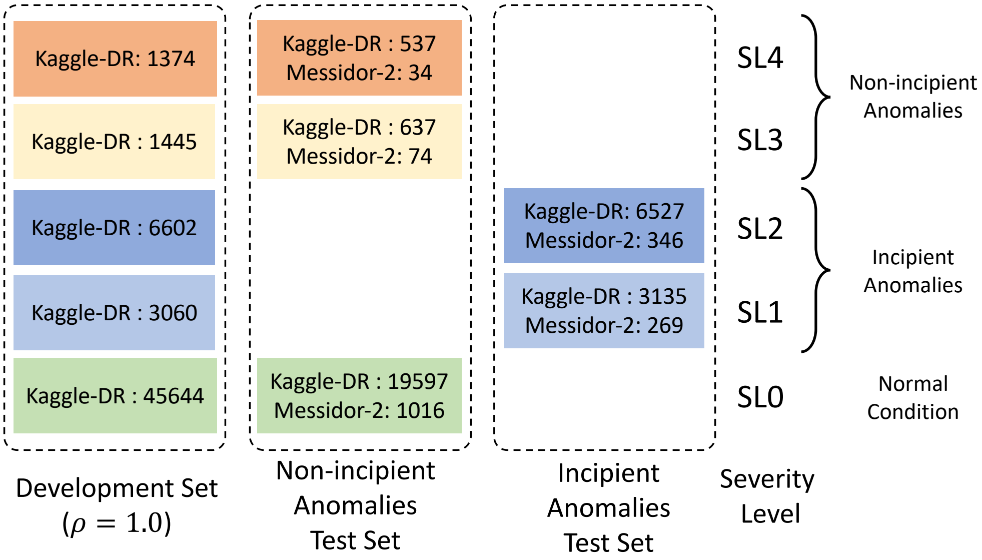


Figure 6.1: Layout of the development set and the test sets resulting from the partitioning the diabetic retinopathy data.

6.3 Model Setup

We conducted a case study on diagnosing diabetic retinopathy with ensembles of deep learning models. For benchmarking the performance of our ensemble-based solutions on medical imaging data, we used two popular collections of diabetic retinopathy image data described in Chapter 4, i.e. the Kaggle-DR dataset [13] and the Messidor-2 Dataset [14].

As mentioned earlier, the diabetic retinopathy disease is graded into five SLs, displayed in Figure 6.2. Following the problem setup used in prior literature [23], we trained models in order to distinguish the referable (SL2 & SL3 & SL4) cases from the non-referable ones (SL0 & SL1).

Three types of ensemble methods (hyperparameter ensemble, MC-dropout and Test-Time Augmentation (TTA)) described earlier in Chapter 3 were evaluated in our experiments. The deep ensemble approach [51] was not tested in our case study because we did not manage to train the networks from scratch (random initialization). Our solution was to train all the models on top of a pretrained ImageNet model, which proved to work well but also prevented us from implementing the original deep ensemble approach.

Hyperparameter Ensembles For the Kaggle-DR dataset, we trained multiple CNN models using different architectures and data augmentation settings, and randomly combined them into hyperparameter ensembles. Each ensemble model only consisted of base learners of the same type. We used CNN models for classifying image models in the Kaggle-DR dataset. The CNN models were implemented using the `pytorch` [73] framework. The deep learning models used to construct our ensembles varied in their architecture, image data resolution, training set selection, number of training epochs and data augmentation strengths. Two different CNN architectures, ResNet34 [26] and VGG16 [83], were used in our experiments. We used the binary-crossentropy loss function and the Adam [49] optimizer during training. All network parameters were initialized with the weights from pretrained models provided by the `torchvision` [67] package that were created for classifying objects from the ImageNet database [15].

Since our experiments involved scanning various ρ values, to reduce the total training effort, we first trained our models with non-incipient disease data (only SL0 & SL3 & SL4) for 130 epochs, and then continued to train the resulting networks with all training data (SL0 to SL4) till convergence. Most trained models reached an Area Under the Receiver Operating Characteristic curve (AUROC) of above 0.98 on both the training and the validation sets. We discarded the bad performing models and put the rest into a pool. The retained models in the pool were then used as base learners for constructing ensembles. To create an ensemble model instance, we randomly picked K single learners from the pool.

Data augmentation [75] has proved to be an important technique for training deep learning models that can prevent overfitting and can enhance model's generalization ability. We utilized several different types of data augmentation operations at training time that were available from the `torchvision` package [67]. These operations included `RandomResizedCrop`, `adjust_brightness`, `adjust_saturation` and `adjust_contrast` that could randomly adjust the aspect ratio, the brightness, the saturation and the contrast respectively. The strength of data augmentation in our experiments was controlled by a multiplier factor $\gamma \in \{0.1, 0.3, 0.5, 0.7\}$.

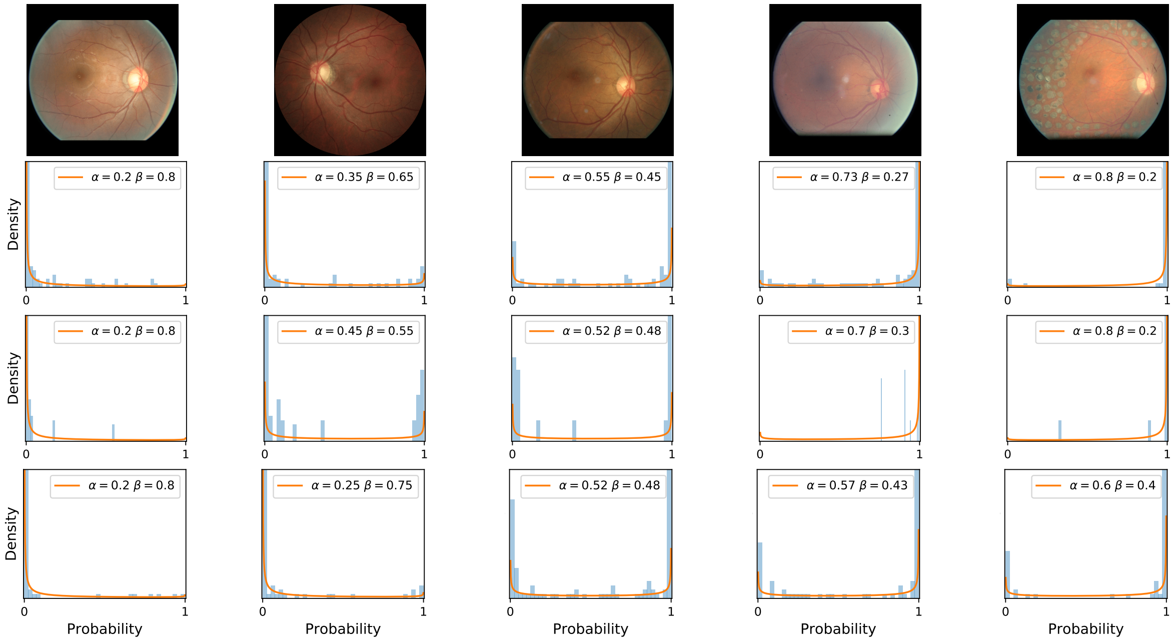
MC-Dropout Ensembles To construct MC-dropout ensembles, we repeatedly sampled the trained models that have dropout layers. Each model were sampled for $K = 50$ times, and the results were then combined and grouped for later analysis.

TTA Ensembles For TTA [3, 91] ensembles, the diversity comes from the stochasticity injected to the inputs at test time. TTA ensembles can then be obtained by repeatedly sampling the same network with stochastic inputs, as with MC-dropout models. We used a test-time data augmentation of different strengths $\gamma_{\text{test}} \in \{0.1, 0.3, 0.5, 0.7\}$, and repeatedly sampled each model for $K = 50$ times.

6.4 Experiment Results

Distribution of Ensemble Outputs

In our theoretical analysis in Section 3.4, we made an assumption that the individual predictions in an ensemble learner assumed a beta distribution $\mathcal{B}(\alpha, \beta)$ where $\alpha + \beta$ was held constant. We first performed an observational study to validate this assumption. For



(a) SL0 (No-DR) (b) SL1 (Mild) (c) SL2 (Moderate) (d) SL3 (Severe) (e) SL4 (Proliferate)

Figure 6.2: Fundus images (top panel) of the five SLs of diabetic retinopathy diseases, and the distributions (shown as histograms) of their corresponding classifier predictions under hyperparameter ensemble (second panel), MC-dropout (third panel) and TTA (fourth panel).

each image in the two datasets, we used models in the library to make predictions. For MC-dropout and TTA models, we created samples by repeatedly sampling the networks.

Five example images and their corresponding ensemble outputs are visualized in Figure 6.2. A beta distribution displayed by the orange curve is fitted to each distribution. Here we assume that $\alpha + \beta = 1$ for the fitted beta distribution $\mathcal{B}(\alpha, \beta)$; see the plots for the fitted α and β values, and Chapter B in the appendix for additional examples.

Performance Evaluation

As in the previous chapter on chiller data, we again report the detection performance in terms of the following performance metrics. We will mainly focus on the results from hyperparameter ensembles in this section. Some additional results for the two types of implicit ensembles will be also given, and a comparison between explicit and implicit ensemble approaches will be made.

False Negative Rate (FNR) We first report the detection performance of the trained ensembles on the Kaggle-DR datasets in terms of FNR. We examine the FNR for both incipient and non-incipient anomalies under different settings of the FPR ratio q and the incipient anomaly ratio ρ, and show the results as box plots in Figure 6.5 for three different ensemble sizes $K = 1, 5, 25$.

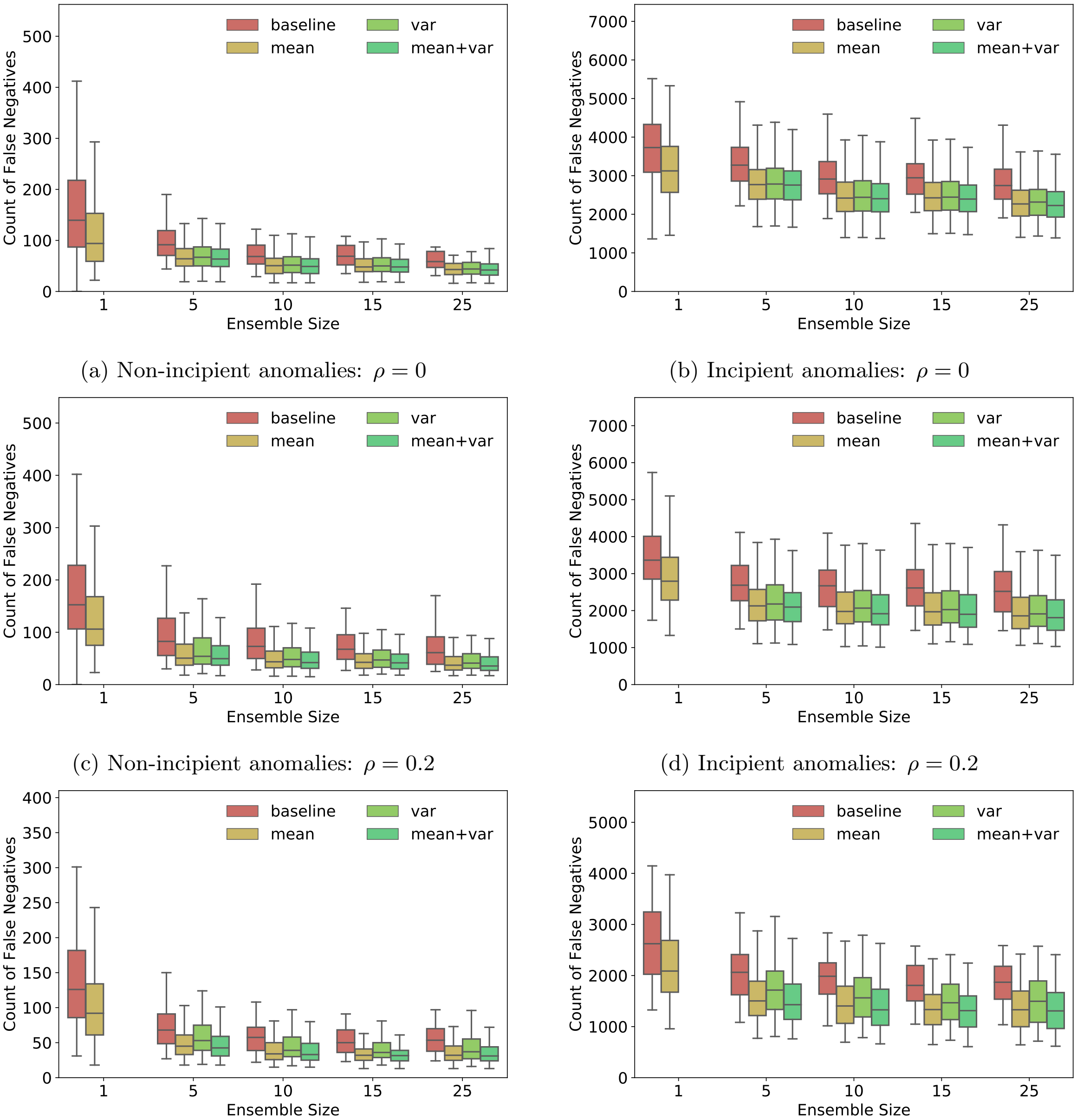
As can be seen from the plots, the FNRs for non-incipient anomalies are all close to zero, which indicates near-perfect classification performance between SL0 (normal conditions) and SL3 & SL4 (non-incipient anomalies). The results for non-incipient anomalies are not displayed here due to limited space. By comparing the two cases ($K = 1$ vs. $K = 25$), we can immediately see performance improvement for ensemble learners over single learners. In addition, we can observe a decreasing trend in FNR with increasing q, which indicates that more incipient anomalies can be detected when we lower the detection threshold τ; in other words, more incipient anomalies can be detected when the classifiers are working at more sensitive operating points.

Remaining False Negatives As in the previous chapter, we evaluate the number of remaining false negatives for the Kaggle-DR dataset after applying uncertainty estimation, under different uncertainty metrics and ensemble sizes; see Figure 6.3 for details. We are interested in knowing the number of remaining false negatives because these are mistakes that the uncertainty estimation techniques fail to identify. As can be seen from the plots, MEAN leads to fewer remaining false negatives for all ensemble sizes $K > 1$ and for $\rho = 0, 0.2, 1.0$. We can also see from Figure 6.3 that the results given by MEAN+VAR do not have much improvement over those given by MEAN, indicating that many of the false negatives identified by VAR are also captured by MEAN, which again matches the expectation of Theorem 1.

An immediate observation from Figure 6.3 is that CNN ensembles can achieve substantial performance improvement even for small ensemble sizes (e.g., $K = 5$); further improvement from increasing ensemble sizes, however, is smaller, indicating a diminishing return. By comparing the performance of MEAN and that of VAR, we can see that MEAN leads to fewer remaining false negatives in general, for all three ρ values; in other words, the MEAN uncertainty metric can identify more false negatives than VAR.

False Negative Precision (FN-precision) Although the above analysis shows that MEAN compared to VAR can identify more false negatives among incipient anomalies, it is not sufficient to show that MEAN is more preferable to VAR because the increased number of corrected false negatives may simply be a consequence of more uncertain negatives being identified. In an extreme scenario, if all negative data points are marked as uncertain negatives, then all false negatives can be corrected. Therefore, we use the FN-precision metric to measure how precisely each model can identify the false negatives. As with the previous study on chiller data, we again tested $\theta \in \{1\%, 2\%, 5\%, 10\%\}$ in this study. As can be seen from Figure 6.4, MEAN again outperforms VAR in terms of FN-precision.

Comparing the three types of ensembles From the results, we can see that all three ensemble methods have high uncertainty on o.o.d. datasets (ImageNet and CIFAR-10), indicating that all three methods have good performance in detecting o.o.d. tasks. On



(a) Non-incipient anomalies: $\rho = 0$

(b) Incipient anomalies: $\rho = 0$

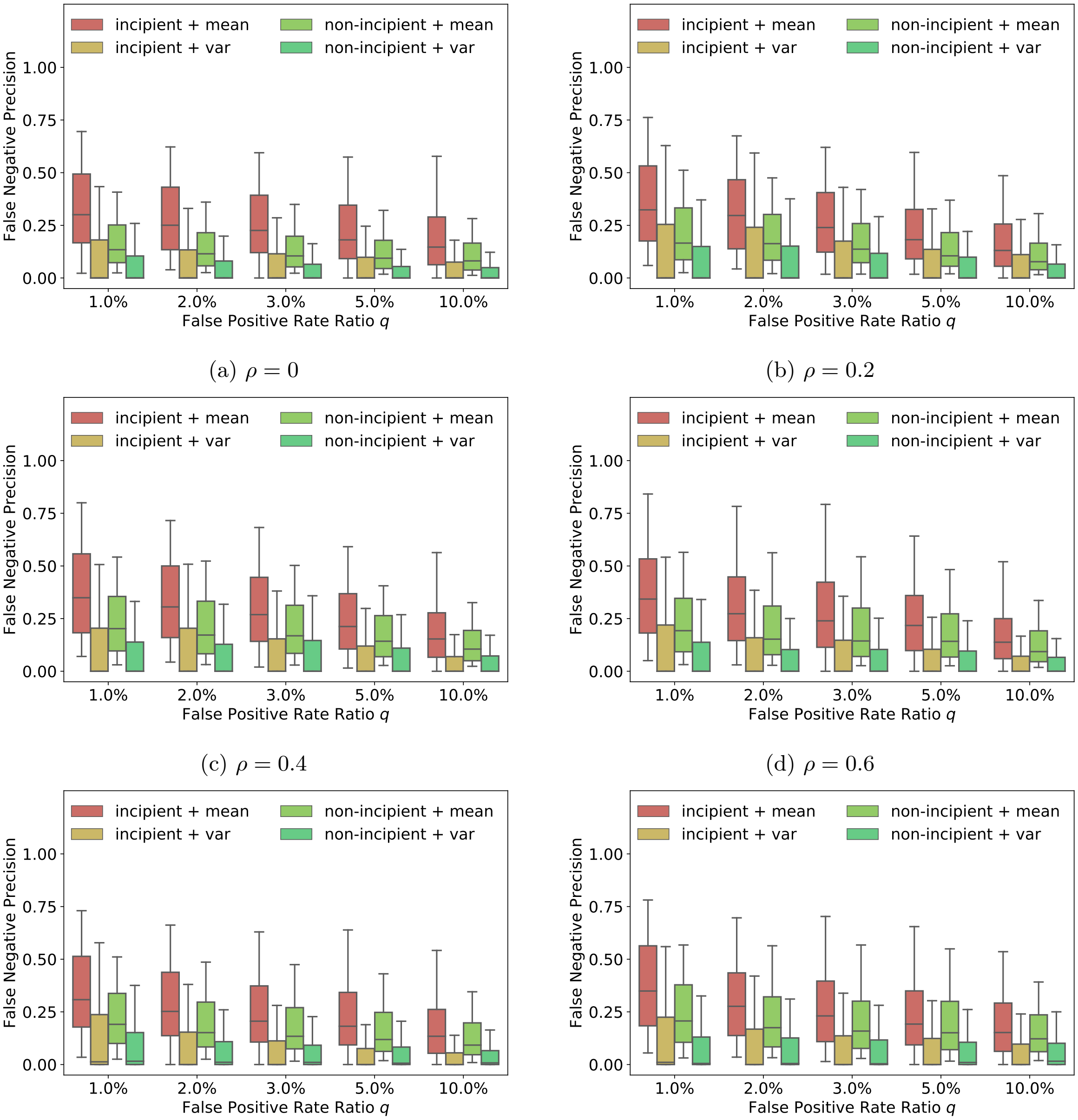
(c) Non-incipient anomalies: $\rho = 0.2$

(d) Incipient anomalies: $\rho = 0.2$

(e) Non-incipient anomalies: $\rho = 1.0$

(f) Incipient anomalies: $\rho = 1.0$

Figure 6.3: Box plots showing the number of remaining/certain false negatives (incipient anomalies wrongly classified as negative) after the rest are identified by uncertainty estimation for the Diabetic retinopathy dataset.



(a) $\rho = 0$

(b) $\rho = 0.2$

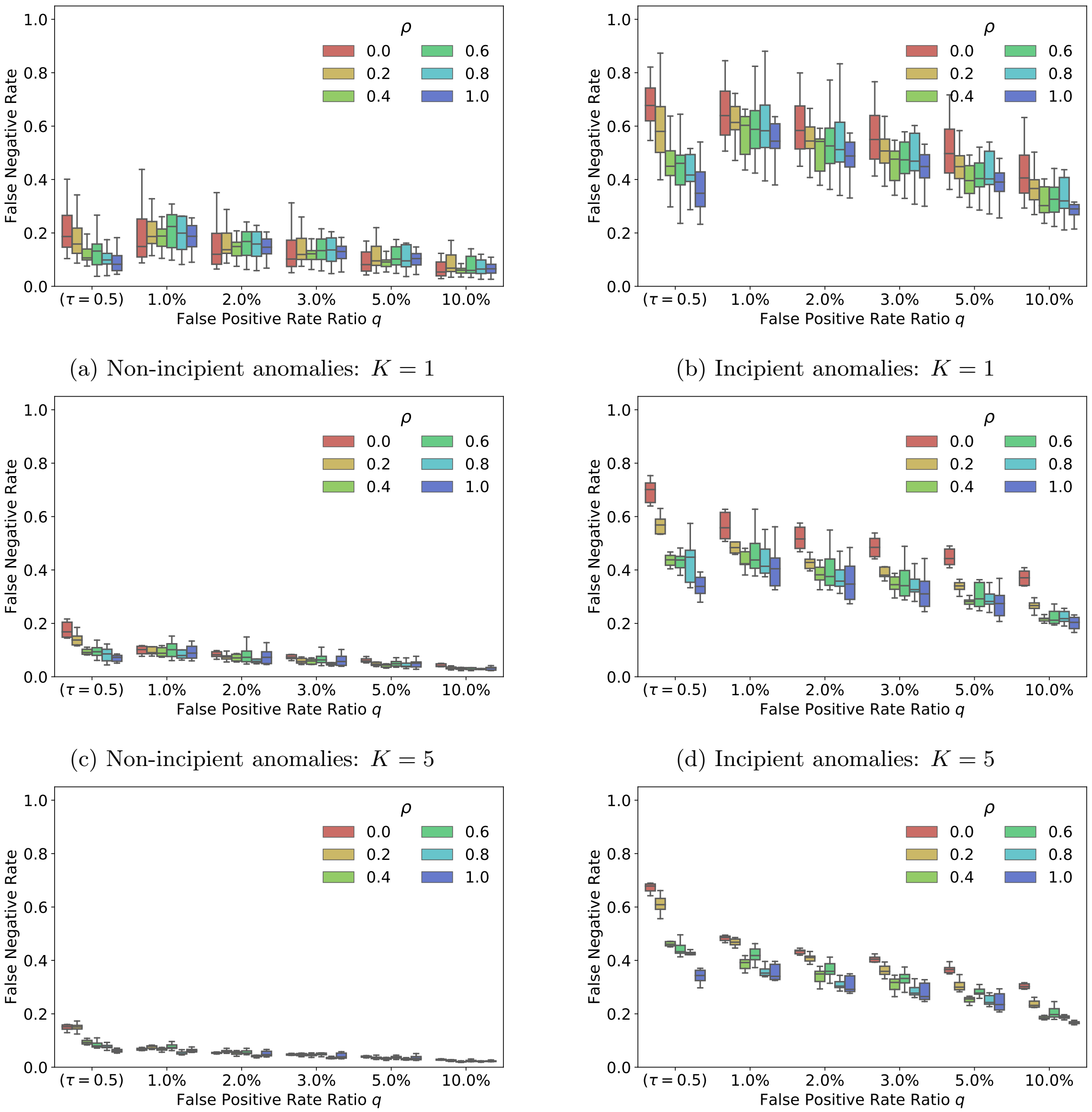
(c) $\rho = 0.4$

(d) $\rho = 0.6$

(e) $\rho = 0.8$

(f) $\rho = 1.0$

Figure 6.4: Box plots showing the FN-precision metric for all ensemble classifiers ($K > 1$) under different settings of the FPR percentile q for the Kaggle-DR dataset. Boxes of different colors indicate performance indices given by MEAN and VAR for the incipient and the non-incipient data.



(a) Non-incipient anomalies: $K = 1$

(b) Incipient anomalies: $K = 1$

(c) Non-incipient anomalies: $K = 5$

(d) Incipient anomalies: $K = 5$

(e) Non-incipient anomalies: $K = 25$

(f) Incipient anomalies: $K = 25$

Figure 6.5: FNR metrics on non-incipient (left column) and incipient (right column) anomalies from the Kaggle-DR dataset, for ensemble sizes $K = 1, 5, 25$.

in-distribution (i.d.) dataset, what can be observed from the results is that the uncertain scores have more concentrated distributions around zero while still some data points have high uncertainty scores. To explore the component of these high-uncertainty data, we defined a breakdown of the high-uncertainty data points by fixed percentile and calculated the respective proportion of data of different severity levels, from SL0 to SL4, among all high-uncertainty data. The results are shown in Table 6.1. We also added the severity ratios given by single learners and from the raw datasets for comparison.

From the results, we can see that, compared with the other two uncertain metrics, the MEAN metric have lower proportion in SL0 & SL3 & SL4 that are supposed to be less ambiguous and higher proportion in SL1 & SL2 that are supposed to be the real ambiguous data. This indicates that MEAN have higher specificity on the real ambiguous data. In addition, among the three ensemble methods, hyperparameter ensembles have greatly higher ratios on SL1 & SL2, indicating that the hyperparameter ensemble is much more precise in detecting ambiguous data.

Here detailed results for the Kaggle-DR dataset and the Messidor-2 Dataset will be given in Table B.3 and Table B.4, respectively. Tables B.1 & B.2 show fractions of the false negatives and the total uncertain negatives across different ρ values and different FPR ratio q in Kaggle-DR dataset and Messidor-2 Dataset respectively. The percentage of false negatives in all uncertainty negatives are shown in parentheses.

6.5 Summary

In this chapter, we show that the challenges in incipient anomaly detection that are identified in Chapter 5 also exist in modern computer vision tasks. Incipient diabetic retinopathy diseases can easily be mistaken as non-referable cases, and therefore many patients with such conditions will miss the best time for treatment if they trust the diagnosis given by some AI models that seem to be well-trained but fail to handle these challenging cases. Our proposed ensemble approach again helps address this challenge by alleviating the problem.

The three main takeaways from the experimental studies in Chapter 5 and Chapter 6 are summarized as follows:

- Without sacrificing the detection performance on non-incipient anomalies, we can improve the classifier's performance on incipient anomalies by using models of higher sensitivity; this can be done by tuning down the a classifier's detection threshold τ.

- The detection performance on incipient anomalies can be greatly improved by incorporating some incipient anomaly data, even in small amount, into the training distribution (i.e. the development set).

- MEAN is a more preferable uncertainty metric to VAR, as proved by our theoretical analysis and shown by our empirical results.

The three recommendations above are complementary and can lead to better results when applied together. It is worthy to note that in this dissertation we mainly focus on supervised ML models and their ensembles. One-class methods such as OC-SVMs and autoencoders are also promising and interesting directions for further investigations.

Table 6.1: A breakdown of the high-uncertainty data points across different SLs. The five percentage numbers in each entry show the respective proportion of the data of the five severity levels, SL0 to SL4, among all high-uncertainty data points.

Dataset	θ (%)	Uncertainty metric	Hyperparameter ensemble	MC-dropout	Test time augmentation	Single learner
Kaggle-DR		Raw Data Distribution	73.6, 7.0, 14.8, 2.3, 2.1			
	10	MEAN	8.24, 15.76, 69.82, 3.47, 2.71	44.27, 7.32, 37.94, 5.63, 4.85	30.83, 15.07, 43.69, 4.9, 5.51	66.9, 8.6, 17.3, 3.7, 3.6
		VAR	15.9, 16.41, 61.37, 2.98, 3.34	57.8, 6.42, 28.27, 3.78, 3.73	35.73, 13.73, 41.75, 3.81, 4.98	—
		KL	15.67, 16.38, 61.21, 3.94, 2.8	49.08, 8.17, 33.62, 4.61, 4.52	34.2, 16.62, 38.28, 5.21, 5.7	—
	5	MEAN	5.09, 17.77, 73.41, 1.2, 2.53	30.57, 9.31, 52.33, 3.61, 4.18	21.75, 19.38, 53.41, 3.89, 1.57	61.5, 9.7, 22.1, 3.4, 3.3
		VAR	11.44, 17.25, 66.91, 1.54, 2.87	36.83, 9.62, 45.25, 3.81, 4.49	24.84, 20.51, 49.61, 3.19, 1.85	—
		KL	11.19, 17.79, 67.17, 1.63, 2.22	33.91, 8.69, 42.83, 6.64, 7.93	28.5, 19.11, 44.32, 4.27, 3.81	—
	2	MEAN	1.65, 19.09, 77.52, 1.28, 0.47	14.68, 12.6, 67.66, 2.55, 2.51	8.74, 23.49, 65.73, 1.04, 1	68.7, 8.8, 17.7, 2.1, 2.7
		VAR	1.73, 16.88, 76.19, 2.94, 2.26	14.67, 11.31, 66.7, 0.96, 6.36	9.7, 22.43, 62.46, 2.59, 2.82	—
		KL	1.8, 18.72, 76.69, 1.75, 1.03	16.25, 10.74, 64.87, 3.65, 4.49	11.16, 23.23, 61.16, 2.48, 1.97	—
	1	MEAN	1.4, 17.23, 79.95, 0.19, 1.23	10.77, 12.01, 71.45, 2.5, 3.26	6.49, 21.93, 66.38, 1.9, 3.31	61.5, 7.7, 19.9, 2.8, 3.0
		VAR	1.48, 18.09, 78.83, 1.52, 0.08	11.12, 13.39, 68.53, 0.42, 6.54	6.18, 26.11, 63.16, 1.58, 2.97	—
		KL	1.03, 18.41, 78.6, 0.86, 1.1	13.9, 11.56, 67.62, 3.27, 3.65	8.77, 24.81, 62.89, 1.92, 1.61	—
Messidor-2		Raw Data Distribution	58.2, 15.4, 21.4, 4.3, 2.0			
	10	MEAN	11.07, 17.15, 65.83, 2.98, 2.98	42.39, 13.82, 36.54, 4.08, 3.16	33.9, 26.72, 32.55, 4.66, 2.17	57.0, 16.3, 20.5, 4.0, 2.2
		VAR	20.26, 11.3, 63.58, 2.24, 2.61	50.74, 12.8, 26.82, 6.3, 3.35	42.83, 29.24, 20.85, 4.22, 2.86	—
		KL	20.72, 12.04, 63.01, 2.33, 1.89	50.3, 13.64, 28.15, 4.52, 3.39	41.98, 28.63, 21.21, 5.81, 2.37	—
	5	MEAN	7.32, 20.39, 68.35, 1.67, 2.26	33.34, 14.9, 43.98, 3.91, 3.87	21.59, 35.06, 38, 2.98, 2.36	61.5, 16.0, 18.8, 1.8, 1.9
		VAR	9.87, 18.64, 66.33, 2.88, 2.28	36.41, 14.85, 42.14, 3.25, 3.34	29.47, 31.16, 32.48, 3.38, 3.51	—
		KL	9.53, 18.8, 66.09, 2.78, 2.8	41.35, 14.61, 37.43, 2.9, 3.7	29.67, 32.73, 29.8, 4.23, 3.57	—
	2	MEAN	4.51, 23.23, 69.68, 1.58, 1	28.61, 14.08, 47.09, 5.28, 4.93	9.69, 38.29, 49.41, 1.24, 1.37	60.6, 19.7, 15.4, 2.4, 1.9
		VAR	6.05, 21.51, 67.69, 2.13, 2.62	30.48, 15.8, 44.04, 6.09, 3.59	14.64, 36.6, 44.19, 1.64, 2.93	—
		KL	6.71, 22.07, 67.32, 2.12, 1.79	31.36, 15.56, 45.45, 4.58, 3.05	18.85, 36.28, 39.91, 2.06, 2.9	—
	1	MEAN	1.55, 25.31, 72.17, 0.53, 0.44	19.73, 16.3, 60.36, 1.08, 2.54	8.14, 38.56, 52.24, 1.03, 0.02	60.4, 15.5, 19.0, 3.0, 2.1
		VAR	1.28, 24, 72.15, 1.75, 0.83	25.39, 15.11, 51.8, 2.59, 5.11	10.6, 39.71, 47.79, 0.78, 1.13	—
		KL	2.06, 23.15, 72.72, 2, 0.08	23.17, 14.03, 59.74, 1.81, 1.25	11.49, 40.22, 45.21, 1.92, 1.16	—

Chapter 7

Out-of-Distribution Fault Detection with Stratification-Aware Cross-Validation

7.1 Chapter Overview

In this chapter, we will describe a Stratification-Aware Cross-Validation (SACV) approach for detecting out-of-distribution (o.o.d.) faults that are absent from the training distribution. Although this work targets o.o.d. faults which differs from the main subject of this dissertation—incipient anomalies, the approach to be presented next shares many commonalities with the ensemble-based methodology described in the previous chapters, and provides a new insight into a related research topic—o.o.d. detection. The presented research in this chapter is based on the author's recent paper [89].

The rest of this chapter will be organized as follows. We will explain the motivation for this work in Section 7.2. The o.o.d. detection problem formulation, as well as necessary background, will be given in Section 7.3. Next, we will describe the SACV approach in details in Section 7.4. Two case studies, including the experiment results, will be presented in Section 7.5. We will discuss related work in Section 7.6, and later summarize this work in Section 7.7 to conclude this chapter.

7.2 Motivation

As mentioned in previous chapters, it is highly appealing to have an end-to-end approach that can directly learn from system operational data and then produce well-performing fault detection models. However, the domain shift [64] (a.k.a. distribution shift [86], concept drift [90]) problem presents a major challenge for the adoption of data-driven methods in practice. Although models trained with supervised learning methods tend to perform well on in-distribution data patterns, the unseen, o.o.d. data may cause unexpected prediction behaviors. In order to train a well-performing model, large amounts of labeled, diversified data are typically needed, which are not always easy to obtain, especially for fault detection tasks where the fault data usually constitute only a small fraction of the collected data.

In fault detection applications, the prediction task is usually to differentiate a "normal" class (hereinafter referred to as the negative class) from a set of fault classes (hereinafter referred to as the the positive class), which is often cast as a binary classification problem. Since a system can have multiple fault states, the corresponding positive class data are often *stratified*. Severe consequences may arise if the trained model fail to detect some of the strata, especially for safety critical applications. These failures may hide behind the seemingly high detection accuracy numbers. Worse still, if some strata are missing from the training distribution but appear in the test distribution (a.k.a. o.o.d.), regular Machine Learning (ML) training pipelines offer no guarantee on such o.o.d. data. In other words, many false negative decisions may occur for such o.o.d. inputs. For example, if an unseen fault type occurs or if an industrial machine is operating under a different environment, a fault detection model may fail to identify such fault conditions.

The unseen nature of domain shifts presents a major challenge to training ML models that can generalize, especially in the lack of domain knowledge. To address this issue, we wish to make best use of available data (although not comprehensive enough to capture all possible variations) to obtain ML models as robust as possible against domain shifts. Our solution is to use a stratification-aware cross-validation strategy during model selection, which helps *reject* those models that are not likely to perform well at test time. We believe our proposed strategy is an easy-to-use recipe for developing supervised ensemble fault detection models that are more immune to the above-mentioned domain shift phenomena. We summarize our contributions in this work as follows.

- We propose a Stratification-Aware Cross-Validation (SACV) strategy for training ML models on stratified data to improve robustness against unknown domain shift in test distribution.

- The efficacy of the proposed method is demonstrated in two case studies: a commercial building chiller system and a commercial building Air Handling Unit (AHU) system. The results show that our SACV strategy can lead to substantial improvement in detecting o.o.d. faults.

- On top of the SACV strategy, we applied ensemble learning in an *uncertainty-informed* fault detection framework to identify false negatives which demonstrated significant performance boost when domain experts can help correct the decisions on the high-uncertainty negative examples identified by our algorithm.

7.3 Background and Problem Formulation

Fault Detection on Stratified Data

As in Chapter 2, we again formulate the fault detection problem under a *binary classification* setting. Let $\mathcal{X}$ be the set system states and sensor measurements, and $\mathcal{M}$ be a model class of classification models. Suppose a fault detection model $M \in \mathcal{M}$ defines an *anomaly score* function $s^M : \mathcal{X} \to \mathbb{R}$ that characterizes how likely the system state x corresponds to a fault state; a larger $s^M(x)$ implies a higher chance of a data point x being a fault. The classifier's decision on whether or not x corresponds to a fault can be made by introducing a *decision threshold* τ^M that dichotomizes the anomaly score $s^M(x)$. We can decide the classifier's predicted label as follows,

$$\hat{z} = \mathbb{1}\{s^M(x) > \tau^M\}, \tag{7.1}$$

where $\mathbb{1}\{\cdot\}$ is the identity function. For evaluating the performance of M, we can define the False Negative Rate (FNR) and False Positive Rate (FPR) of the model M on the test data distribution as follows:

$$\mathrm{FNR}(s^M, \tau^M) = \mathbb{E}[\hat{z} = 0 \mid z = 1]\,, \tag{7.2}$$

$$\mathrm{FPR}(s^M, \tau^M) = \mathbb{E}[\hat{z} = 1 \mid z = 0]\,. \tag{7.3}$$

Let $\mathcal{X}^{\text{dev}}$ be the subset of labeled training data points that are available to us at training time. Ideally, the goal is to learn an anomaly score function s^* by minimizing the classification error on $\mathcal{X}^{\text{dev}}$, and then decide a corresponding threshold τ^*, such that the resulting model $M \doteq (s^*, \tau^*)$ can achieve an optimal trade-off between FNR and FPR on the (unseen) test data distribution $\mathcal{D}_{\text{test}}$.

Out-of-distribution Data Different from the usual assumption that the training set and the test set data are sampled from the same distribution, in this chapter we assume that the test data distribution $\mathcal{D}_{\text{test}}$ not only comprises of in-distribution (i.d.) data $\mathcal{D}^{\text{i.d.}}_{\text{test}}$ but also some o.o.d. data $\mathcal{D}^{\text{o.o.d.}}_{\text{test}}$ with domain shift. Our goal is to train a binary classification model M using the development set data $\mathcal{D}_{\text{dev}}$ such that M achieves the best precision-recall trade-off on the test data $\mathcal{D}_{\text{test}}$ that includes both the i.d. and the o.o.d. portions. In this study, we also assume that the data distributions follow a *stratified* structure; in other words, the fault data are structured as a set of *subgroups* (a.k.a. strata). Suppose that the development set data consist of $K^{\text{i.d.}}$ subgroups in total; the i.d. test distribution $\mathcal{D}^{\text{i.d.}}_{\text{test}}$ consists of the same data subgroups. The o.o.d. test distribution $\mathcal{D}^{\text{o.o.d.}}_{\text{test}}$ contains $K^{\text{o.o.d.}}$ subgroups that do not appear in the development set.

7.4 Methodology

Our proposed methodology combines the strengths of 1) Stratification-Aware Cross-Validation (SACV), a novel validation approach for model selection, and 2) the ensemble-based uncertainty estimation method described in previous chapters, for o.o.d. detection. We will give more details below.

Validation and Model Selection Validation is a classic and almost a must-have procedure for model selection in a modern ML pipeline. The goal of validation is to obtain an accurate estimate of a trained model's prediction performance on the test set, under the typical assumption that the training set and the test set are sampled from the same data distribution. By using validation during a model selection procedure, we can reject model instances that overfit to the training data or lead to unsatisfactory performance.

Holdout validation (hereinafter abbreviated as "holdout") is one of the simplest validation strategies in ML. Part of the development set data is held out as the validation set, and the rest is used for training the models. The holdout validation involves only a single run, and hence part of the data is never used for training and may cause misleading results. Cross-validation alleviates the problem by involving multiple validation runs, and then combine the results of the runs together (to be discussed in details shortly). The k-fold cross-validation method (hereinafter abbreviated as "k-fold") partitions the development set data into k equal-sized folds. In a rotated fashion, each time a fold is held out as the validation set and the rest is used for training. Under both holdout and k-fold strategies, the development set is split *randomly* into a training set and a validation set. Since the split is random, we can expect that the $K^{\text{i.d.}}$ subgroups of the development set will all be represented in both the training and the validation set. If the cross-validation procedure is properly implemented, we can expect the resulting model will perform well on the i.d. data, i.e. these $K^{\text{i.d.}}$ subgroups in the development set. However, such cross-validation strategy does not take into account the resulting model's generalization behavior on o.o.d. test data, and therefore the resulting classifier may not perform well on $\mathcal{D}^{\text{o.o.d.}}_{\text{test}}$.

Uncertainty-informed Decision Making In addition to using cross-validation methods, we can also leverage ensemble learning as in previous chapter to identify high-uncertainty data examples and let experts decide whether or not they are o.o.d. faults that are wronly classified as in-distribution data. Similar to what we have done with incipient anomalies, for detecting o.o.d. faults we use an *uncertainty metric* U to rank the negative examples[1], in order to identify high-uncertainty examples that are likely to be false negative decisions. We suppose that an ensemble model of size T is used, and denote the predictions of individual ensemble members on x_i as $y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(T)}$. The uncertainty metric $U : \mathbb{R}^K \to \mathbb{R}$ takes as input the ensemble predictions $\{\hat{y}_i^{(k)}\}$ on x_i, and outputs an real-valued *uncertainty score* $u(x_i) \doteq U\left(y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(T)}\right)$. To resolve a dichotomy between "uncertain" and "certain", we introduce a threshold $\tilde{u}$ on $u(x)$: if $u(x) > \tilde{u}$ then x is deemed an uncertain input example and otherwise a certain one. As we have done in Chapter 2, we select the value of $\tilde{u}$ so as to bound the uncertain negative ratio to be below a level of θ on the development set. To

[1] Examples that are classified as negative by a classification model, i.e. $\{x_i \mid \hat{z}_i = 0\}$.

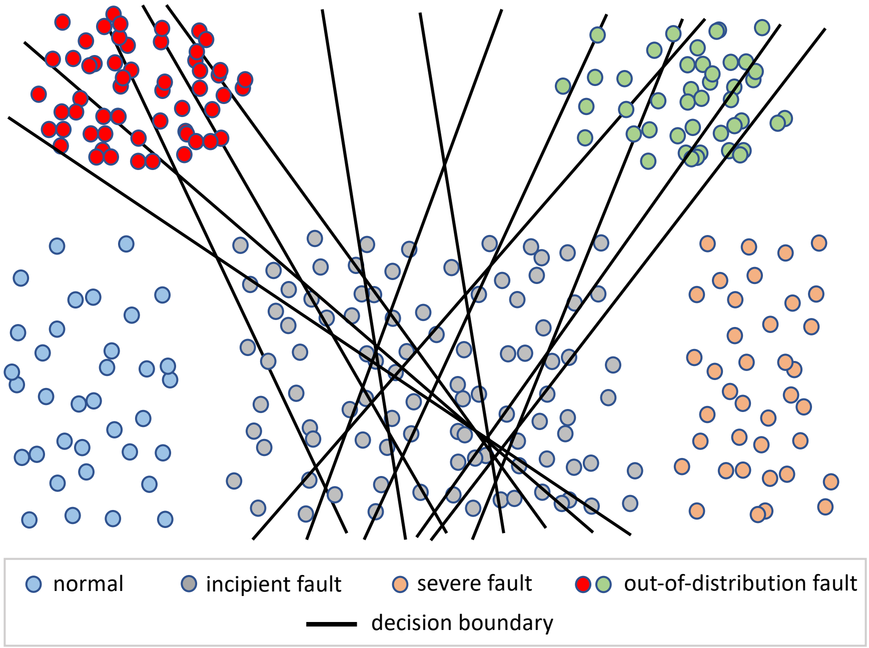


Figure 7.1: An illustration showing how ensemble classifiers help detect incipient fault data [46, 88] and o.o.d. fault data.

evaluate how the identified uncertain negatives overlap with the actual false negatives, we again use the false negative precision metric as defined in Definition 1.

As mentioned in previous chapters, diversity is recognized as one of the key factors that contribute to the success of ensemble approaches [8]; the diversity allows individual classifiers to generate different decision boundaries. As illustrated in Figure 7.1, the diversity among ensemble members is crucial for improving the detection performance on o.o.d. data instances. For the ensemble methods to work, the individual classifiers must exhibit *diversity* among themselves, such that the resulting ensemble can hopefully give a high prediction uncertainty on o.o.d. data points.

Ensemble-based SACV Strategy for Model Selection

To address the issue mentioned above, we propose a Stratification-Aware Cross-Validation (SACV) strategy that explicitly emphasizes and prioritizes the model's generalization performance on test data under domain shift. When an SACV strategy is employed, one by one, a subgroup (stratum) of the development set data is selected as the o.o.d. validation set; then part of the rest $K^{\text{i.d.}} - 1$ subgroups will be used as the training set, and the remaining portion will be used as the i.d. validation set, as illustrated in Figure 7.2.

A different technique with similar name is the *stratified k-fold cross-validation*, which also deals with stratified data but should not be confused with our proposed SACV strategy. In stratified k-fold cross-validation, the folds are made by preserving the portion of samples for each class (or stratum). As a result, instead of returning randomly sampled folds, stratified k-fold cross-validation returns *stratified folds*. Similar to stratified k-fold cross-validation, our proposed strategy also takes data stratification into consideration; however, we deliberately exclude one or more stratum from the training set and keep them solely in the validation set so that we can directly measure a trained model's generalization performance at training time.

The primary objectives of cross-validation are 1) assessing model validity and 2) hyperparameter tuning. During cross-validation, we search through the hyperparameter space and evaluate the performance of each configuration. Suppose a total of R hyperparameter configurations, respectively denoted by $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_R$, are evaluated and ranked during cross-validation. In our empirical study, we will retain the top-r hyperparameter configurations, instead of the single best-performing one, and report their performance indices.

Combining Results from Multiple Validation Runs

To finalize model selection, the conventional method (hereinafter referred to as REFIT-ALL) is to refit the model using the entire development set data and the selected hyperparameter configuration $\mathcal{H}^*$. Another method is to combine the $K^{\text{i.d.}}$ models, e.g., by using simple average, that are created during cross-validation in a ensemble. The idea is similar to sample Bagging [7]; as a result, we will name this approach COMBINE. Later, we will compare REFIT-ALL and COMBINE in our empirical study.

Ensemble Learning and Uncertainty Estimation In our empirical study to be described later, we employed the Bagging [7] (or bootstrap aggregation) approach for creating diversity among ensemble members. The core idea is to construct a family of models by randomly subsetting the development set (a.k.a. *sample bagging* [7]). A later variant called *feature bagging* [29] selects a random subset of the features for training each member classifier in an ensemble. One famous application of Bagging in ML is the Random Forest (RF) model. In our empirical study, we only used sample bagging for inducing diversity among ensemble classifiers. In this study, only *homogeneous* base learners, i.e. models of the same type, are used to construct ensembles. The case of heterogeneous ensembles is an interesting setting and we leave it for future investigation.

A theoretical analysis for comparing the two uncertainty metrics MEAN and VAR is given earlier in Chapter 3, but on uncertain examples known as *incipient anomalies* that exhibit mild symptoms of known anomaly (faults or diseases) types. The results showed that MEAN is a more *robust* uncertainty metric than VAR in the sense that the performance lower bound given by MEAN is higher than that of VAR. It is still unclear which uncertainty metric is likely to perform better on o.o.d. strata. In our empirical study to be presented later, we

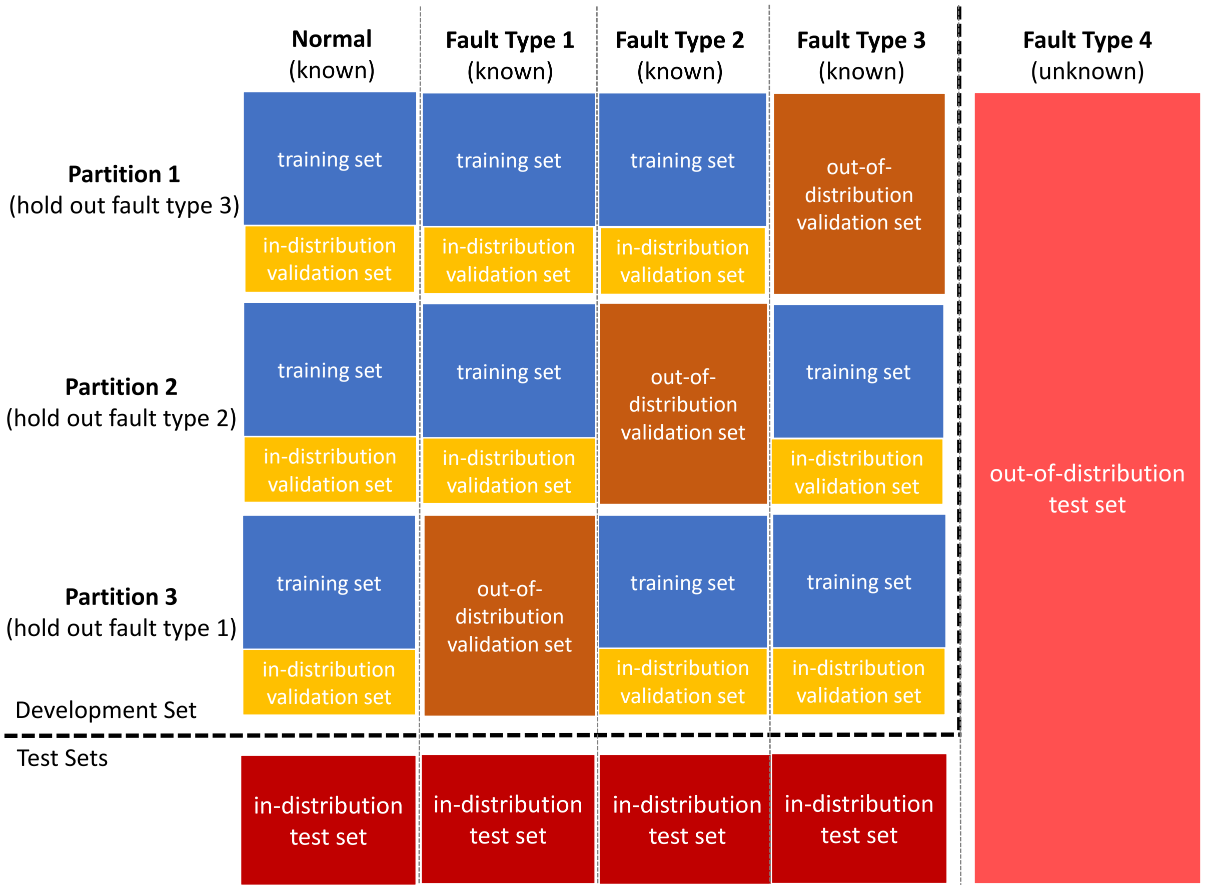

Normal (known)
Fault Type 1 (known)
Fault Type 2 (known)
Fault Type 3 (known)
Fault Type 4 (unknown)
Partition 1 (hold out fault type 3)
training set
training set
training set
out-of-distribution validation set
in-distribution validation set
in-distribution validation set
in-distribution validation set
Partition 2 (hold out fault type 2)
training set
training set
out-of-distribution validation set
training set
in-distribution validation set
in-distribution validation set
in-distribution validation set
Partition 3 (hold out fault type 1)
training set
out-of-distribution validation set
training set
training set
in-distribution validation set
in-distribution validation set
in-distribution validation set
out-of-distribution test set
Development Set
Test Sets
in-distribution test set
in-distribution test set
in-distribution test set
in-distribution test set

Figure 7.2: An illustration showing how SACV partitions a dataset during cross-validation. In this example, the dataset is made up of four fault types (subgroups), and three out of the four appear in the development set. Our goal is to train a classifier using the development set data to achieve good detection performance on both the unseen i.d. (dark red) and the o.o.d. (light red) test data.

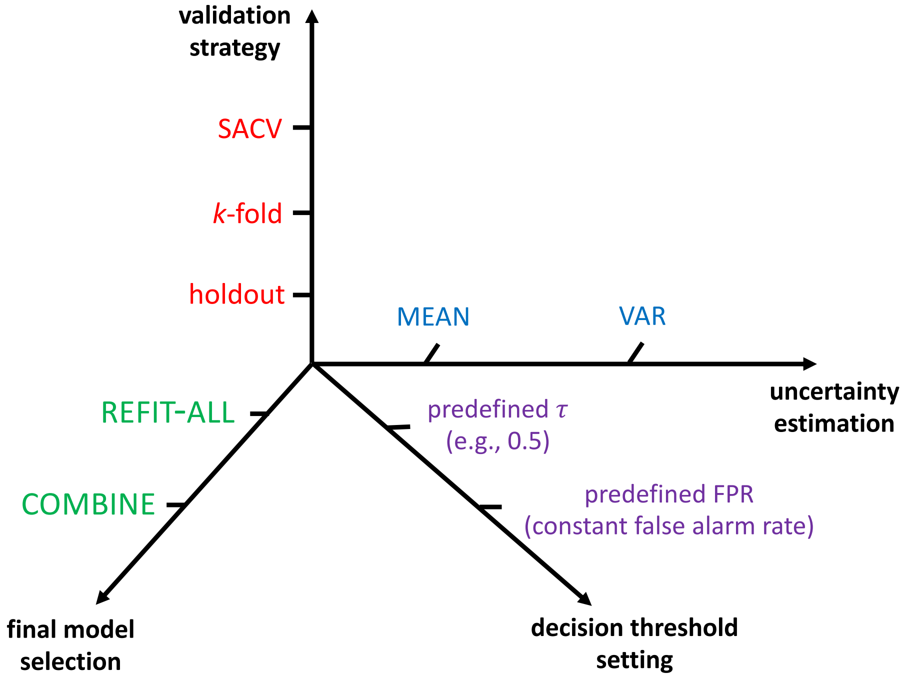


Figure 7.3: An illustration showing the concepts and techniques compared in this study. Orthogonal concepts are put onto different axes.

will again evaluate and compare the two uncertainty metrics MEAN and VAR, and see which performs better in o.o.d. detection under an SACV framework.

We show in Figure 7.3 the relationship among the various concepts introduced above. Note that techniques on different axes are orthogonal, and thus can be applied together.

7.5 Experimental Study

Data Preparation

In this section, we give a brief overview of the two datasets to be used in our empirical study. further details about them can be find in Chapter 4. We will also describe how we partitioned the datasets in our experiments into development sets and test sets.

Chiller Faults Dataset We used the ASHRAE RP-1043 Dataset [12] ("chiller dataset") as introduced in Section 4.2 to examine the proposed approach. In the chiller dataset, sensor measurements of a 90-ton centrifugal water-cooled chiller are recorded under fault-free and various fault conditions. In this study, we included the six faults (FT-FWE, FT-FWC, FT-RO, FT-RL, FT-CF, FT-NC) used in our previous study [44] as the fault classes. Each fault class (type) includes fault conditions of all four Severity Levels (SLs).

AHU Faults Dataset We used the ASHRAE RP-1312 Dataset [95] ("AHU dataset") to examine our proposed approach. Our study included 25 commonly encountered AHU faults (eleven in spring, eight in summer, and six in winter). By treating data from each season as an independent dataset, we ended up having three sub-datasets, namely *AHU-spring*, *AHU-summer* and *AHU-winter*, for our experimental study. We adopted the features selected by Li et al.'s previous work [58] in our experiments.

Dataset Partitioning To study the generalization performance of different cross-validation methods, we performed a series of experiments on each dataset. For each dataset consisting of K subgroups, we repeated the experiment for K times, each time leaving out a different subgroup as the o.o.d. test set. The i.d. test set is then partitioned out of the rest $K^{\text{i.d.}} = K - 1$ subsets. The remaining data will make up the development set.

Experiment Setup

We conducted the experiments on the chiller and AHU faults datasets described above. Decision Tree (DT) and Neural Network (NN) models were used as base learners in our experimental study, and then combined them together into Bagging ensembles [7]. We built Bagging ensembles of two different sizes 5 and 10, and used the single learner cases (i.e., ensemble size 1) as the baseline. For each experiment, we excluded one subgroup from the whole dataset and use it as the o.o.d. test data, as described earlier. To induce diversity, we swept a wide range of hyperparameters settings, and selected the five best-performing sets of hyperparameters.

Result Analysis

Comparing Final Model Selection Methods: REFIT-ALL vs. COMBINE We first compared the two "final model selection" methods, REFIT-ALL and COMBINE, by examining their performance differences on the three datasets (including all of their sub-datasets). Both give similar performance on i.d. data, and we further assessed their performance in terms of the FNR on the o.o.d. data 1) under different configurations of q (i.e. the predefined FPR level on the development set): 1%, 2%, 3%, 5%, 10% and also 2) under $\tau = 0.5$.

For the three AHU sub-datasets, we only noticed significant performance differences when SP-FT-8, SU-FT-4, WT-FT-4 are used as o.o.d. data, and COMBINE performed much better than REFIT-ALL. When the rest are used as the o.o.d. data, both REFIT-ALL and COMBINE gave very low FNR. We observed similar phenomena with the chiller dataset. For the chiller dataset, performance difference is only significant when RL and CF are held out as o.o.d. test set. Again, COMBINE outperformed REFIT-ALL. In Figure 7.4, we only displayed results for the above-mentioned cases where there was significant performance gap between REFIT-ALL and COMBINE with either NN or DT ensemble models, and omitted the rest. The low FNR in the omitted cases may be a result of the held-out subgroups not being enough "out-of-distribution"; in other words, the held-out subgroup may resemble one or more of the i.d. subgroups, and thus leads to near-perfect detection performance on o.o.d. subgroups. In the analysis to be presented next, we will omit these cases as well, and focus on the challenging cases where the held-out test set presents real o.o.d. challenges to fault detection models.

To sum up, it is clear that the COMBINE method gives lower FNR compared with the REFIT-ALL, indicating that the COMBINE has a better performance in improving the models' generalization ability. Therefore, in our next experiments, we will only display results from COMBINE.

Comparing Validation Strategies: SACV vs. k-fold vs. Holdout Next, we evaluated the ensemble methods' performance on the o.o.d. data when different validation strategies are used. As in the previous experiment, we examined the FNRs across different configurations of τ (by directly setting $\tau = 0.5$ or varying q). For comparison, we used the holdout validation and the k-fold cross-validation as our baselines. The number of splits used in k-fold cross-validation is set to be equal to the number of classes of the development set, i.e. $k = K^{\text{i.d.}}$. We visualize the results from same subgroups as introduced in the previous analysis. The results can be found in Figure 7.5.

Comparing the three validation strategies shown in Figure 7.5, we can clearly see that SACV achieved significant improvement in FNR over the other two validation strategies, indicating that SACV is indeed effective in improving the models' generalization performance. In Figure 7.5, we only showed the results for a selected number of cases where baseline methods performed poorly on the held out o.o.d. data, and omitted the rest since the baseline FNRs for these cases are already close to zero. In addition, we can also see from the results that the FNRs decrease with the increment of q.

Comparing Uncertainty Metrics: MEAN vs. VAR Finally, we compared the different metrics used for uncertainty estimating including: 1) MEAN, 2) VAR. For both methods, we examined $\theta \in \{1\%, 2\%, 5\%, 10\%\}$. In addition, we tested a baseline case $\theta = 0$ in which no uncertainty information was exploited. We evaluated our models' generalization performance by calculating the number of remaining false negatives after applying uncertainty estimation, assuming all of the identified false negatives can be corrected perfectly by human experts.

The results are displayed in Figure 7.6. As illustrated in the plots, it is clear that both MEAN and VAR metrics have decent improvement in identifying false negatives over BASELINE. Specifically, comparing MEAN and VAR, we also found that VAR outperformed MEAN, indicating that VAR excelled at estimating o.o.d. data. Another finding was that

DT ensemble models gave more significant performance improvement as the ensemble size grew, compared to NN models. One possible reason for this is that single NN classifiers have stronger classification abilities over single DT classifiers.

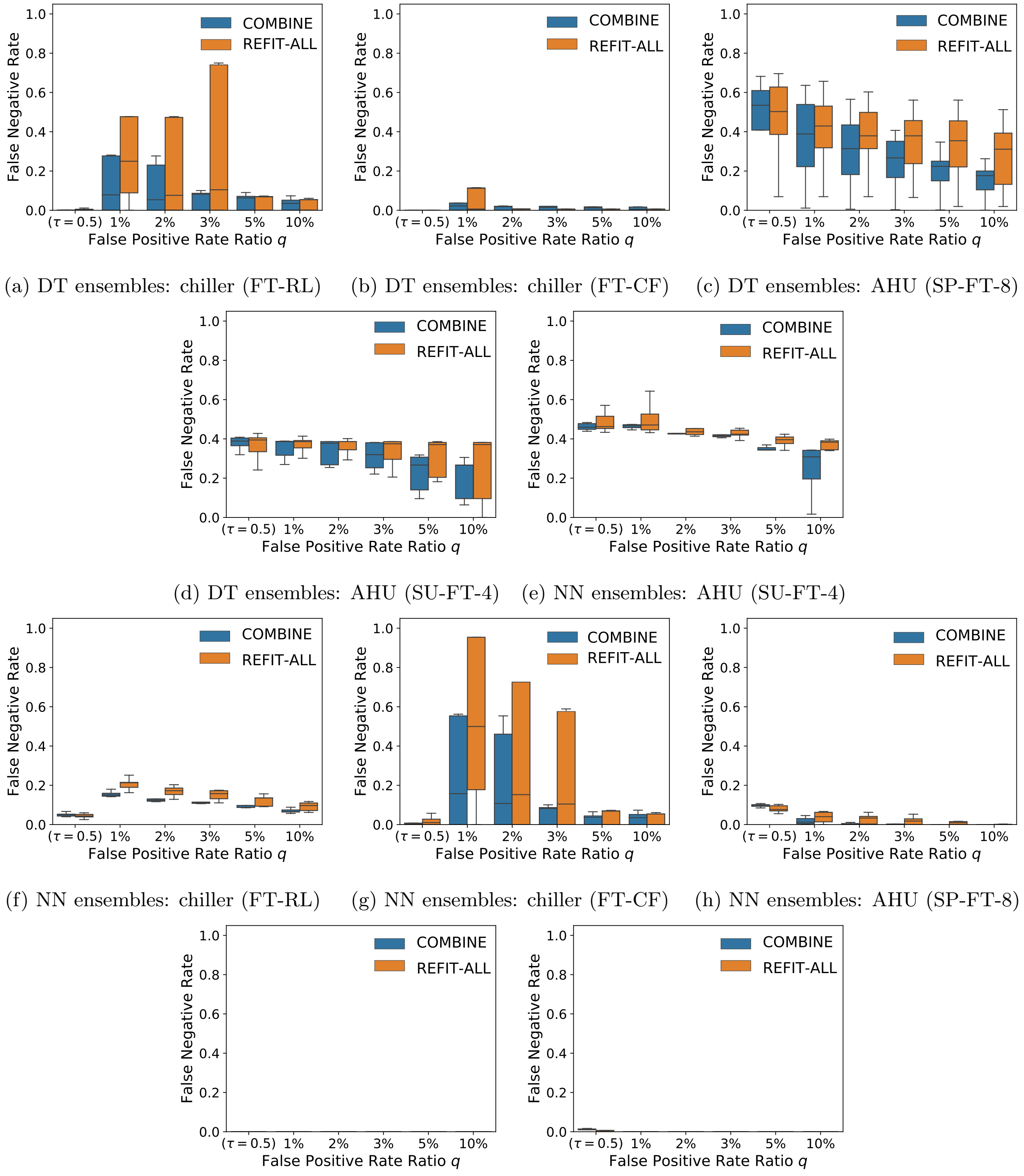
The above results seem to contradict the conclusion made in earlier chapters, where we showed that MEAN is more preferable to VAR for identifying incipient anomalies (faults or diseases). It is worth mentioning that our focus in this chapter is o.o.d. fault data that are not included in the development set during training, rather than incipient faults. We illustrate the differences between the two scenarios in Figure 7.1, and how ensemble methods can help with fault detection in both scenarios. It will be interesting future work to understand why VAR excels at identifying o.o.d. data.

7.6 Related Work

Out-of-Distribution Data Detection In recent years, a number of research papers [51, 21] related to o.o.d. detection are seen in literature. Lakshminarayanan et al. [51] proposed using *random initialization* and *random shuffling* of training examples to diversify base learners of the same network architecture. Gal and Ghahramani proposed using MC-dropout [21] to estimate a network's prediction uncertainty by using dropout not only at training time but also at test time. By repeatedly sampling a dropout model $\mathcal{M}$ for T times using the same input, we can obtain an ensemble of T prediction results. The MC-dropout technique provides an inexpensive approximation to training and evaluating an ensemble of exponentially many similar yet different neural networks.

Adversarial Validation A closely related technique that also deals with the domain shift phenomenon between training and test distributions is the *adversarial validation* approach [72] whose goal is to detect and address the difference between the training and the test datasets. The idea of adversarial validation is to create an adversarial validation set as a *proxy* of the test set, such that the resulting model from model selection can achieve satisfactory performance on the adversarial validation set (and hopefully on the test set as well).

The creation of effective adversarial validation sets, however, will usually require prior information about the test data distribution. In some occasions, for example in Kaggle competitions, part of the test set data is made public at training time while the rest is used as a "private test set". Such setting makes it possible to apply adversarial validation approaches. A classifier is trained to distinguish the training and the (public) test set data, and then part of the training data (e.g., the difficult-to-classify ones) that resembles the test data can be held out as an adversarial validation set. Such approach is described as the "validation data selection" method in Pan et al.'s recent work [72], which also describes other types of adversarial validation methods; see details therein for further information.



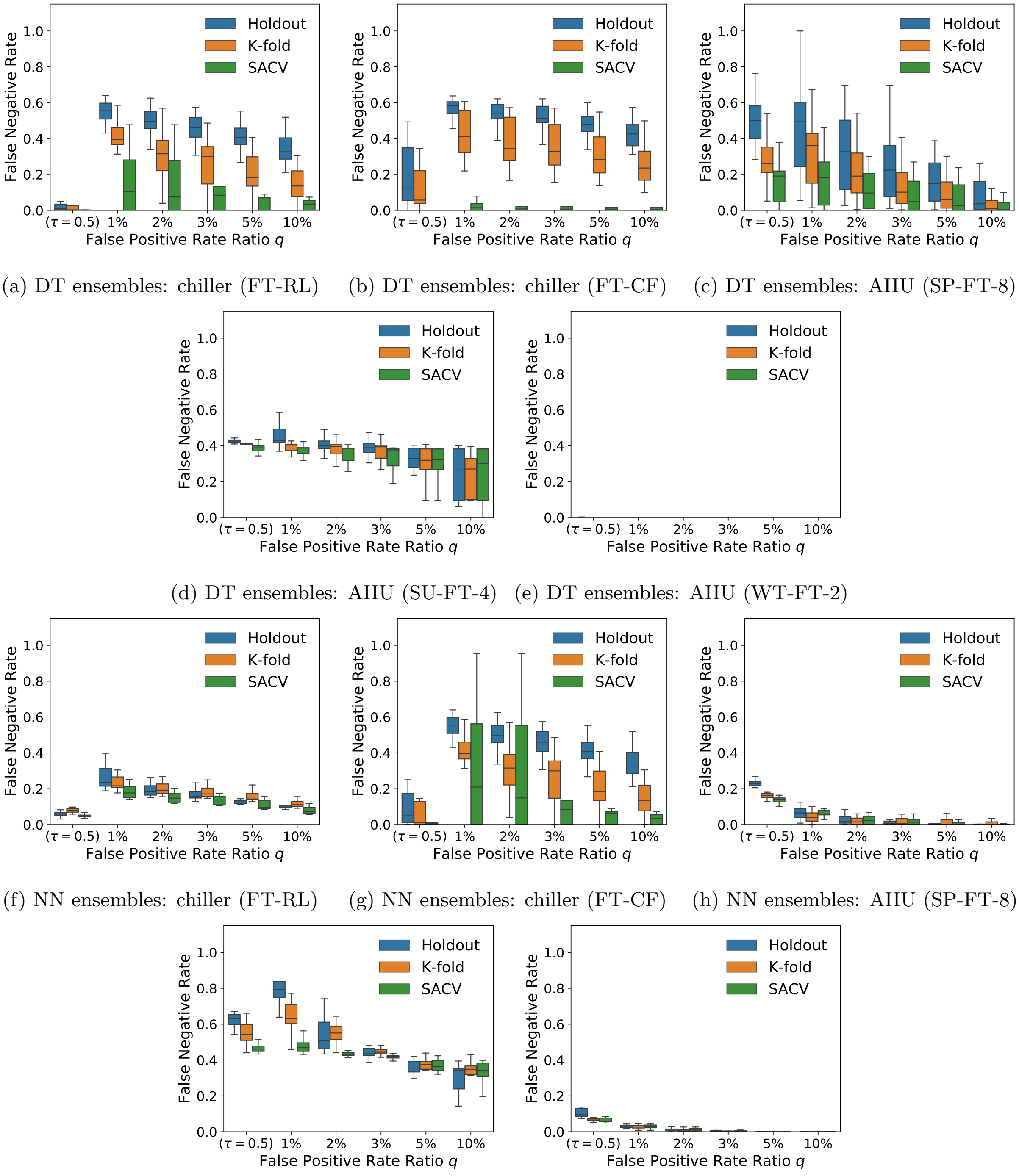
(a) DT ensembles: chiller (FT-RL) (b) DT ensembles: chiller (FT-CF) (c) DT ensembles: AHU (SP-FT-8)

(d) DT ensembles: AHU (SU-FT-4) (e) NN ensembles: AHU (SU-FT-4)

(f) NN ensembles: chiller (FT-RL) (g) NN ensembles: chiller (FT-CF) (h) NN ensembles: AHU (SP-FT-8)

(i) DT ensembles: AHU (WT-FT-2) (j) NN ensembles: AHU (WT-FT-2)

Figure 7.4: Performance comparison between the REFIT-ALL and the COMBINE methods in terms of their FNR on different datasets are presented: 1) the chiller dataset and 2) the AHU dataset. The excluded subgroup that is used as the o.o.d. test set and SACV is used as the cross-validation method.



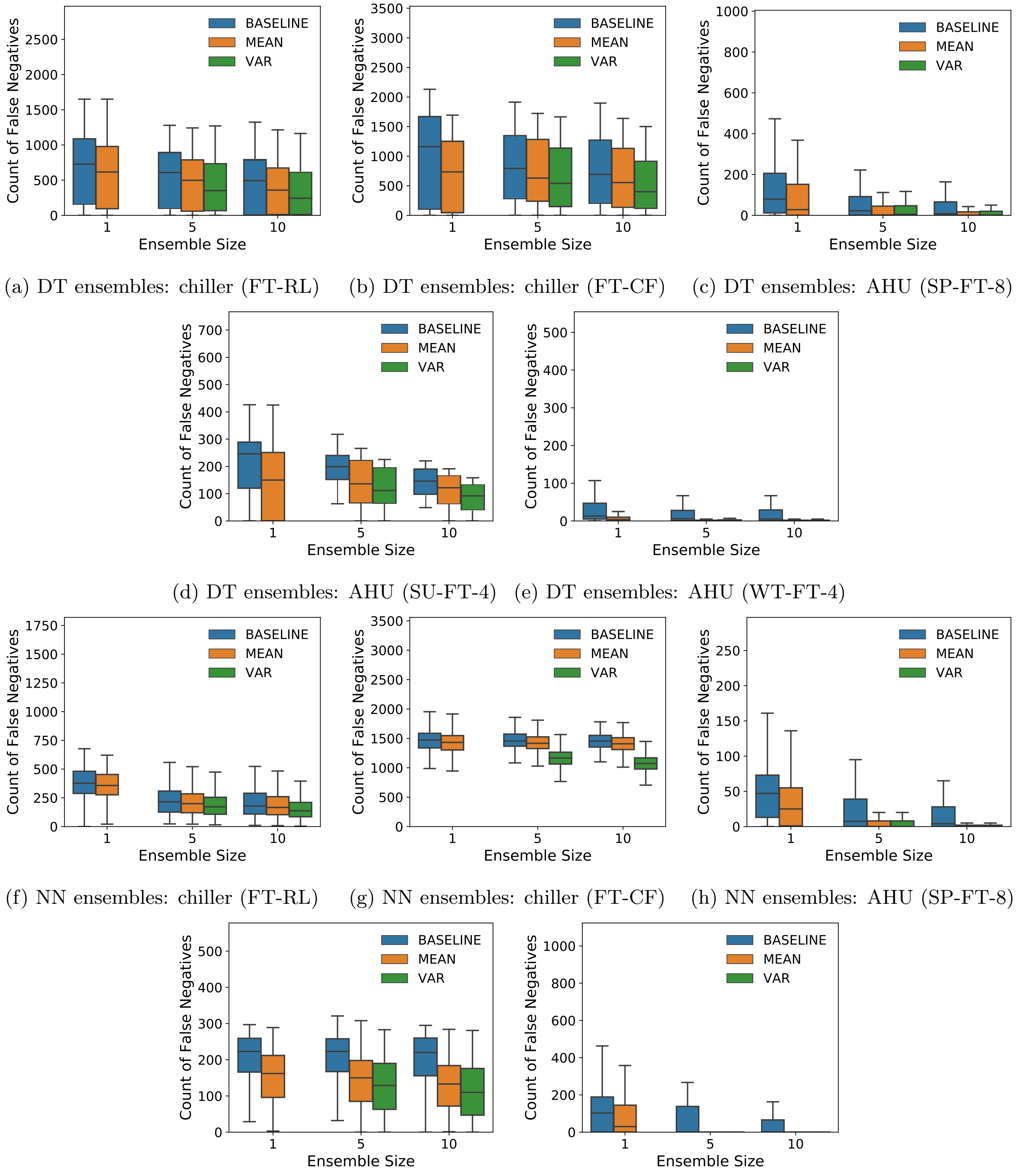
(a) DT ensembles: chiller (FT-RL) (b) DT ensembles: chiller (FT-CF) (c) DT ensembles: AHU (SP-FT-8)

(d) DT ensembles: AHU (SU-FT-4) (e) DT ensembles: AHU (WT-FT-2)

(f) NN ensembles: chiller (FT-RL) (g) NN ensembles: chiller (FT-CF) (h) NN ensembles: AHU (SP-FT-8)

(i) NN ensembles: AHU (SU-FT-4) (j) NN ensembles: AHU (WT-FT-2)

Figure 7.5: The FNR given by different (cross-)validation methods: 1) holdout, 2) k-fold, and 3) SACV. Results from DT and NN ensembles on the chiller and AHU datasets are presented.



(a) DT ensembles: chiller (FT-RL) (b) DT ensembles: chiller (FT-CF) (c) DT ensembles: AHU (SP-FT-8)

(d) DT ensembles: AHU (SU-FT-4) (e) DT ensembles: AHU (WT-FT-4)

(f) NN ensembles: chiller (FT-RL) (g) NN ensembles: chiller (FT-CF) (h) NN ensembles: AHU (SP-FT-8)

(i) NN ensembles: AHU (SU-FT-4) (j) NN ensembles: AHU (WT-FT-4)

Figure 7.6: The count of remaining false negatives under different uncertainty metrics: 1) BASELINE ($\theta = 0$, i.e. no uncertainty information is exploited), 2) MEAN and 3) VAR. The results from DT ensembles and NN ensembles on the three datasets are presented.

The "public test set" data in the above Kaggle example is in fact part of the development set (because these "test" data are available at training time), and thus not actually a "real" test set. In situations where little information about the test data distribution is available, adversarial validation will not be applicable. Our SACV approach does not require prior information about the unseen test distribution. Instead, our approach relies only on the available development set data.

7.7 Summary

In this chapter, we have shown that the domain shift problem can undermine fault detection performance in stratified data, when some subgroups (strata) appear in the test data distribution but not in the training distribution. To address this issue, an easy-to-use cross-validation method is proposed to mitigate the issue and demonstrated its efficacy on two representative Cyber-Physical System (CPS) datasets. In the experimental study, our proposed SACV approach achieved significant performance improvement over traditional holdout and k-fold validation methods on o.o.d. data, in the meantime without sacrificing its performance on i.d. data. For future work, we plan to extend the proposed methodology to datasets of different modalities, such as image data.

Chapter 8

Conclusions and Future Research

8.1 Conclusions from Incipient Anomaly Detection

We can see from the presented research that incipient anomalies represent a real challenge in Cyber-Physical System (CPS) and healthcare AI. When it is difficult tell whether the system is in an anomalous state or not, AI diagnostic tools should have the ability to say "*I am not sure*" instead of being overly confident towards either side. Such uncertain response can be useful in indicating the possibility of incipient anomalies that can be easily misdetected as normal conditions.

Our presented ensemble-based approach utilizes the prediction uncertainty information from ensemble classifiers to identify incipient anomaly examples that are wrongly classified as negatives (false negatives). The proposed technique is applied in uncertainty-informed decision schemes for two real-world applications: 1) the detection of chiller faults, and 2) the diagnosis of referable diabetic retinopathy diseases. Both applications demonstrate the effectiveness of our proposed approach.

Impact on Machine Learning (ML) Fields Our work contributes to the theoretical discussion of several important sub-fields of ML and AI, including active learning [10, 33], anomaly detection[44, 41], and ensemble learning [108]. Our research opens up a new path towards gaining a deeper understanding of the interactions between ensemble learning and uncertainty estimation. Prior works in this area are mostly empirical. Although various ensemble methods have demonstrated appealing improvement on a diverse set of tasks, it is very important to understand the theoretical underpinnings of such approaches, which can guide us in designing more effective algorithms and avoid unnecessary trial-and-errors.

Impact on Energy and Smart Buildings Besides its direct impact on anomaly and fault detection algorithms, our work also contributes to the smart building domain because the proposed techniques can help identify hard-to-detect incipient faults that are common in building equipment. Such faults often lead to increased energy consumption of building equipment, and our proposed research can help mitigate the loss of energy efficiency by detecting these soft faults during their early stages.

Impact on Healthcare AI One direct impact of our research on healthcare AI is accuracy improvement. By selecting high-uncertainty negative examples to send for human diagnosticians, our proposed method can help reduce the number of false negatives. On top of that, our uncertainty estimation technique is critical to the adoption of AI-based disease screening and diagnosis schemes, since it can assess the decision risks in addition to accuracy improvement, which is important for both healthcare providers and patients.

8.2 Opportunities for Further Research

At the end of my dissertation, I would like to highlight several potential directions that is worth future research.

Detecting Incipient Anomalies in Temporal Data In health monitoring applications for humans and industrial machines, it is natural to encounter time series data (e.g., sensor data streams) or sequential observations made over time (e.g., CT scans for the same person over several years). The temporal correlations within such data can reveal useful information about system degradation; however, the solution we propose earlier for incipient anomaly detection does not exploit the temporal correlations, leaving room for future research and improvement.

One major obstacle to the above-mentioned research direction is the lack of public domain data that make sequential observations on real degradation processes. The RP-1043 and RP-1312 datasets only provide synthetic fault data that are artificially injected into chiller and Air Handling Unit (AHU) systems. The Kaggle-DR dataset only captures one-time measurements on the patients, instead of making multiple observations on each patient over time. Therefore, these above-mentioned datasets are not suitable for this line of research. The C-MAPSS dataset [81] does provide time series data produced by simulation that captures the degradation of flight engines. However, the ground truth labeling information for indicating the exact onset of faults is not made public, making it difficult to validate and benchmark detection algorithms. I thus call for the academia and industry to release useful datasets to open up this direction of research.

Probabilistic Risk Assessment In real-world diagnostic scenarios, an accurate assessment of a model's confidence is important, especially when the prediction results are used to assist human decision makers. Our presented approach measures the amount of uncertainty by using uncertainty metrics (MEAN, VAR, etc.) to inform decision making. On the other hand, probability as a natural measure of uncertainty is more desirable from a mathematical point of view and also easier for human decision makers to interpret. The connections and differences between the two line of approaches are still unclear, making it a meaningful direction for future investigation.

Bibliography

[1] Charu C Aggarwal. “Outlier ensembles: position paper”. In: *ACM SIGKDD Explorations Newsletter* 14.2 (2013), pp. 49–58.

[2] Sergio Altomonte et al. “Ten questions concerning well-being in the built environment”. In: *Building and Environment* (2020), p. 106949.

[3] Murat Seckin Ayhan and Philipp Berens. “Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks”. In: (2018).

[4] Rémi Bardenet et al. “Collaborative hyperparameter tuning”. In: *International conference on machine learning*. 2013, pp. 199–207.

[5] Richard Barlow and Larry Hunter. “Optimum preventive maintenance policies”. In: *Operations research* 8.1 (1960), pp. 90–100.

[6] L. Breiman et al. “Classification and Regression Trees Wadsworth”. In: 1984.

[7] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24.2 (1996), pp. 123–140.

[8] Gavin Brown et al. “Diversity creation methods: a survey and categorisation”. In: *Information Fusion* 6.1 (2005), pp. 5–20.

[9] Kai-Ying Chen et al. “Using SVM based method for equipment fault detection in a thermal power plant”. In: *Computers in industry* 62.1 (2011), pp. 42–50.

[10] Yuxin Chen et al. “Active Detection via Adaptive Submodularity”. In: *Proc. International Conference on Machine Learning (ICML)*. June 2014.

[11] Eungchum Cho and Moon Jung Cho. “Variance of sample variance”. In: *Section on Survey Research Methods–JSM* 2 (2008), pp. 1291–1293.

[12] MC Comstock and JE Braun. “Development of analysis tools for the evaluation of fault detection and diagnostics in chillers. ASHRAE Research Project RP-1043”. In: *American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., Atlanta. Also, Report HL* (1999), pp. 99–20.

[13] Jorge Cuadros and George Bresnick. “EyePACS: an adaptable telemedicine system for diabetic retinopathy screening”. In: *Journal of diabetes science and technology* 3.3 (2009), pp. 509–516.

[14] Etienne Decencière et al. "Feedback on a publicly distributed database: the Messidor database". en. In: *Image Analysis & Stereology* 33.3 (Aug. 2014), pp. 231–234. ISSN: 1854-5165. DOI: `10.5566/ias.1155`. URL: `http://www.ias-iss.org/ojs/IAS/article/view/1155`.

[15] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[16] Zhimin Du and Xinqiao Jin. "Multiple faults diagnosis for sensors in air handling unit using Fisher discriminant analysis". In: *Energy Conversion and Management.* 49.12 (2008), pp. 3654–3665.

[17] Zhimin Du et al. "Fault detection and diagnosis for buildings and HVAC systems using combined neural networks and subtractive clustering analysis". In: *Building and Environment* 73 (2014), pp. 1–11.

[18] Bo Fan et al. "A hybrid FDD strategy for local system of AHU based on artificial neural network and wavelet analysis". In: *Building and environment* 45.12 (2010), pp. 2698–2708.

[19] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. "Deep Ensembles: A Loss Landscape Perspective". In: *arXiv preprint arXiv:1912.02757* (2019).

[20] Y. Freund and R. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *EuroCOLT*. 1995.

[21] Yarin Gal. "Uncertainty in deep learning". In: *University of Cambridge* (2016).

[22] Jacob Goldberger, Shiri Gordon, and Hayit Greenspan. "An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures". In: *null*. Ieee. 2003, p. 487.

[23] Varun Gulshan et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs". In: *Jama* 316.22 (2016), pp. 2402–2410.

[24] Chuan Guo et al. "On calibration of modern neural networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1321–1330.

[25] Hua Han et al. "PCA-SVM-based automated fault detection and diagnosis (AFDD) for vapor-compression refrigeration systems". In: *HVAC & R Research* 16.3 (2010), pp. 295–313.

[26] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[27] Zhangming He et al. "An incipient fault detection approach via detrending and denoising". In: *Control Engineering Practice* 74 (2018), pp. 1–12.

[28] David J Hill, Barbara S Minsker, and Eyal Amir. "Real-time Bayesian anomaly detection for environmental sensor data". In: *Proceedings of the Congress-International Association for Hydraulic Research*. Vol. 32. Citeseer, 2007, p. 503.

[29] Tin Kam Ho. "The random subspace method for constructing decision forests". In: *IEEE transactions on pattern analysis and machine intelligence* 20.8 (1998), pp. 832–844.

[30] Gao Huang et al. "Snapshot ensembles: Train 1, get M for free". In: *arXiv preprint arXiv:1704.00109* (2017).

[31] Alan Julian Izenman. "Linear discriminant analysis". In: *Modern multivariate statistical techniques*. Springer, 2013, pp. 237–280.

[32] Padmini Jaikumar et al. "Detection of anomalous events from unlabeled sensor data in smart building environments". In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Ieee, 2011, pp. 2268–2271.

[33] Shervin Javdani et al. "Near-Optimal Bayesian Active Learning for Decision Making". In: *In Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*. Apr. 2014.

[34] Saththasivam Jayaprakash and Ng Kim Choon. "Predictive and diagnostic methods for centrifugal chillers". In: *ASHRAE Trans* 114.1 (2008), pp. 282–287.

[35] R. Jia et al. "Design Automation for Smart Building Systems". In: *Proceedings of the IEEE* 106.9 (Sept. 2018), pp. 1680–1699. ISSN: 0018-9219. DOI: 10.1109/jproc.2018.2856932.

[36] Qin Jianying and Wang Shengwei. "A fault detection and diagnosis strategy of VAV air-conditioning systems for improved energy and control performances". In: *Energy and Buildings* 37.10 (2005), pp. 1035–1048. ISSN: 0378-7788.

[37] Ru Jifeng and Li X Rong. "Variable-structure multiple-model approach to fault detection, identification, and estimation". In: *IEEE Transactions on Control Systems Technology* 16.5 (2008), pp. 1029–1038. ISSN: 1063-6536.

[38] Baihong Jin, Guojie Luo, and Wentai Zhang. "A fast and accurate approach for common path pessimism removal in static timing analysis". In: *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2016, pp. 2623–2626.

[39] Baihong Jin et al. "A contract-based framework for integrated demand response management in smart grids". In: *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. 2015, pp. 167–176.

[40] Baihong Jin et al. "A One-Class Support Vector Machine Calibration Method for Time Series Change Point Detection". In: *arXiv preprint arXiv:1902.06361* (2019).

[41] Baihong Jin et al. "An Encoder-Decoder Based Approach for Anomaly Detection with Application in Additive Manufacturing". In: *arXiv preprint arXiv:1907.11778* (2019).

[42] Baihong Jin et al. "Are Ensemble Classifiers Powerful Enough for the Detection and Diagnosis of Intermediate-Severity Faults?" In: *arXiv preprint arXiv:2007.03167* (2020).

[43] Baihong Jin et al. "Augmenting Monte Carlo Dropout Classification Models with Unsupervised Learning Tasks for Detecting and Diagnosing Out-of-Distribution Faults". In: *arXiv preprint arXiv:1909.04202* (2019).

[44] Baihong Jin et al. "Detecting and diagnosing incipient building faults using uncertainty information from deep neural networks". In: *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. Ieee. 2019, pp. 1–8.

[45] Baihong Jin et al. "Online computation of polytopic flexibility models for demand shifting applications". In: *2017 13th IEEE Conference on Automation Science and Engineering (CASE)* (2017), pp. 900–905.

[46] Baihong Jin et al. "Using Ensemble Classifiers to Detect Incipient Anomalies". In: *ACM Transactions on Cyber-Physical Systems (under review)* (2020).

[47] Kaggle. *MESSIDOR-2 DR Grades.* July 2018. URL: `https://www.kaggle.com/google-brain/messidor2-dr-grades`.

[48] Guolin Ke et al. "Lightgbm: A highly efficient gradient boosting decision tree". In: *Advances in neural information processing systems*. 2017, pp. 3146–3154.

[49] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2015).

[50] Jonathan Krause et al. "Grader variability and the importance of reference standards for evaluating machine learning models for diabetic retinopathy". In: *Ophthalmology* 125.8 (2018), pp. 1264–1272.

[51] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6402–6413.

[52] Christian Leibig et al. "Leveraging uncertainty information from deep neural networks for disease detection". In: *Scientific reports* 7.1 (2017), p. 17816.

[53] Dan Li. "Fault detection and diagnosis for chillers and AHUs of building ACMV systems". PhD thesis. 2017.

[54] Dan Li, Guoqiang Hu, and Costas J Spanos. "A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis". In: *Energy and Buildings* 128 (2016), pp. 519–529.

[55] Dan Li et al. "Fault detection and diagnosis for building cooling system with a tree-structured learning method". In: *Energy and Buildings* 127 (2016), pp. 540–551.

[56] Dan Li et al. "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks". In: *International Conference on Artificial Neural Networks*. Springer. 2019, pp. 703–716.

[57] Dan Li et al. “Multivariate anomaly detection for time series data with generative adversarial networks”. In: *arXiv preprint arXiv:1901.04997* (2019).

[58] Dan Li et al. “Optimal Sensor Configuration and Feature Selection for AHU Fault Detection and Diagnosis”. In: *IEEE Transactions on Industrial Informatics* 13.3 (2017), pp. 1369–1380.

[59] Pan Li et al. “A tractable ellipsoidal approximation for voltage regulation problems”. In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 1301–1306.

[60] Pan Li et al. “Distribution system voltage control under uncertainties using tractable chance constraints”. In: *IEEE Transactions on Power Systems* 34.6 (2018), pp. 5208–5216.

[61] Shun Li and Jin Wen. “A model-based fault detection and diagnostic methodology based on PCA method and wavelet transform”. In: *Energy and Buildings* 68 (2014), pp. 63–71. ISSN: 0378-7788.

[62] Shun Li and Jin Wen. “A model-based fault detection and diagnostic methodology based on PCA method and wavelet transform”. In: *Energy and Buildings* 68 (2014), pp. 63–71.

[63] Jian Liang and Ruxu Du. “Model-based fault detection and diagnosis of HVAC systems using support vector machine method”. In: *International Journal of refrigeration* 30.6 (2007), pp. 1104–1114.

[64] Yawei Luo et al. “Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2507–2516.

[65] J. Braun M. Comstock and E. Groll. “The sensitivity of chiller performance to common faults”. In: *HVAC & R Res* 7.3 (2001), pp. 263–279.

[66] Mazhar Ali Khan Malik. “Reliable preventive maintenance scheduling”. In: *AIIE transactions* 11.3 (1979), pp. 221–228.

[67] Sébastien Marcel and Yann Rodriguez. “Torchvision the Machine-Vision Package of Torch”. In: *Proceedings of the 18th ACM International Conference on Multimedia*. MM ’10. Firenze, Italy: Association for Computing Machinery, 2010, pp. 1485–1488.

[68] Comstock MC and Braun JE. “Fault detection and diagnostic (FDD) requirements and evaluation tools for chillers”. In: *Ashrae* (2002).

[69] Timothy Mulumba et al. “Robust model-based fault diagnosis for air handling units”. In: *Energy and Buildings.* 86 (2015), pp. 698–707.

[70] G. Mustafaraj, J. Chen, and G. Lowry. “Development of room temperature and relative humidity linear parametric models for an open office using BMS data”. In: *Energy and Buildings* 42 (Aug. 2010), pp. 348–356.

[71] Alexandru Niculescu-Mizil and Rich Caruana. “Predicting good probabilities with supervised learning”. In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 625–632.

[72] Jing Pan et al. “Adversarial Validation Approach to Concept Drift Problem in Automated Machine Learning Systems”. In: *arXiv preprint arXiv:2004.03045* (2020).

[73] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[74] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[75] Luis Perez and Jason Wang. “The Effectiveness of Data Augmentation in Image Classification using Deep Learning”. In: *ArXiv* abs/1712.04621 (2017).

[76] Emmanuel Ramasso and Abhinav Saxena. “Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset”. In: *Annual Conference of the Prognostics and Health Management Society 2014*. 2014.

[77] Shebuti Rayana, Wen Zhong, and Leman Akoglu. “Sequential ensemble learning for outlier detection: A bias-variance perspective”. In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE. 2016, pp. 1167–1172.

[78] Mark A Richards. *Fundamentals of radar signal processing*. Tata McGraw-Hill Education, 2005.

[79] Juan D Rodriguez, Aritz Perez, and Jose A Lozano. “Sensitivity analysis of k-fold cross validation in prediction error estimation”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.3 (2009), pp. 569–575.

[80] Mayu Sakurada and Takehisa Yairi. “Anomaly detection using autoencoders with nonlinear dimensionality reduction”. In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. Acm. 2014, p. 4.

[81] Abhinav Saxena and Kai Goebel. “C-MAPSS data set”. In: *NASA Ames Prognostics Data Repository* (2008).

[82] Bernhard Schölkopf et al. “Estimating the support of a high-dimensional distribution”. In: *Neural computation* 13.7 (2001), pp. 1443–1471.

[83] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[84] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[85] Biao Sun et al. "Building energy doctors: An SPC and Kalman filter-based method for system-level fault detection in HVAC systems". In: *IEEE Transactions on Automation Science and Engineering.* 11.1 (2014), pp. 215–229.

[86] Yu Sun et al. "Test-time training with self-supervision for generalization under distribution shifts". In: *International Conference on Machine Learning (ICML)*. 2020.

[87] PE T Agami Reddy PhD. "Development and Evaluation of a Simple Model-Based Automated Fault Detection and Diagnosis (FDD) Method Suitable for Process Faults of Large Chillers/DISCUSSION". In: *ASHRAE Transactions* 113 (2007), p. 27.

[88] Yingshui Tan et al. "Exploiting Uncertainties from Ensemble Learners to Improve Decision-Making in Healthcare AI". In: *ArXiv* abs/2007.06063 (2020).

[89] Yingshui Tan et al. "Generalizing Fault Detection Against Domain Shifts Using Stratification-Aware Cross-Validation". In: *ACM Transactions on Cyber-Physical Systems (under review)* (2020).

[90] Peter Vorburger and Abraham Bernstein. "Entropy-based concept shift detection". In: *Sixth International Conference on Data Mining (ICDM'06)*. Ieee. 2006, pp. 1113–1118.

[91] Guotai Wang et al. "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks". In: *Neurocomputing* 338 (2019), pp. 34–45.

[92] Guotai Wang et al. "Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation". In: *International MICCAI Brainlesion Workshop*. Springer. 2018, pp. 61–72.

[93] Kajiro Watanabe et al. "Incipient fault diagnosis of chemical processes via artificial neural networks". In: *AIChE journal* 35.11 (1989), pp. 1803–1812.

[94] Matthew Weber et al. "Gordian: Formal Reasoning-based Outlier Detection for Secure Localization". In: *ACM Transactions on Cyber-Physical Systems* 4.4 (2020), pp. 1–27.

[95] J Wen and S Li. *RP-1312–Tools for evaluating fault detection and diagnostic methods for air-handling units*. Tech. rep. ASHRAE, Tech. Rep, 2012.

[96] F. Wenzel et al. "Hyperparameter Ensembles for Robustness and Uncertainty Quantification". In: *ArXiv* abs/2006.13570 (2020).

[97] Dumidu Wijayasekara et al. "Mining building energy management system data using fuzzy anomaly detection and linguistic descriptions". In: *Industrial Informatics, IEEE Transactions on* 10.3 (2014), pp. 1829–1840.

[98] Sebastien C Wong et al. "Understanding data augmentation for classification: when to warp?" In: *2016 international conference on digital image computing: techniques and applications (DICTA)*. Ieee. 2016, pp. 1–6.

[99] Fu Xiao et al. "Bayesian network based FDD strategy for variable air volume terminals". In: *Automation in Construction* 41 (2014), pp. 106–118.

[100] Dai Xuewu and Gao Zhiwei. “From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis”. In: *IEEE Transactions on Industrial Informatics* 9.4 (2013), pp. 2226–2238. ISSN: 1551-3203.

[101] Ke Yan et al. “ARX model based fault detection and diagnosis for chillers using support vector machines”. In: *Energy and Buildings* 81 (2014), pp. 287–295.

[102] Wen Shen Yan Ke and Timothy Mulumba. “ARX model based fault detection and diagnosis for chillers using support vector machines”. In: *Energy and Buildings* 81 (2014), pp. 287–295.

[103] Yang Zhao, Shengwei Wang, and Fu Xiao. “Pattern recognition-based chillers fault detection method using Support Vector Data Description (SVDD)”. In: *Applied Energy.* 112 (2013), pp. 1041–1048.

[104] Yang Zhao, Fu Xiao, and Shengwei Wang. “An intelligent chiller fault detection and diagnosis methodology using Bayesian belief network”. In: *Energy and Buildings* 57 (2013), pp. 278–288.

[105] Yang Zhao et al. “A robust pattern recognition-based fault detection and diagnosis (FDD) method for chillers”. In: *HVAC&R Research* 20.7 (2014), pp. 798–809.

[106] Gao Zhiwei, Carlo Cecati, and Ding Steven X. “A survey of fault diagnosis and fault-tolerant techniques-Part I: fault diagnosis With model-based and signal-based approaches”. In: *IEEE Transactions on Industrial Electronics* 62.6 (2015), pp. 3757–3767. ISSN: 0278-0046.

[107] Yuxun Zhou, Baihong Jin, and Costas J Spanos. “Learning convex piecewise linear machine for data-driven optimal control”. In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2015, pp. 966–972.

[108] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms.* Chapman and Hall/CRC, 2012.

[109] Yonghua Zhu, Xinqiao Jin, and Zhimin Du. “Fault diagnosis for sensors in air handling unit based on neural network pre-processed by wavelet and fractal”. In: *Energy and buildings* 44 (2012), pp. 7–16.

[110] Barret Zoph and Quoc V Le. “Neural architecture search with reinforcement learning”. In: *arXiv preprint arXiv:1611.01578* (2016).

Appendix A

Supporting Materials for Chapter 3

A.1 Proof of Lemma 1

Proof. Let $Z = s(x_j) - s(x_i)$ be a random variable, where $s(x_i)$ and $s(x_j)$ denotes the uncertainty score of x_i and x_j estimated from K i.i.d. ensemble learners. Therefore $\Delta_{ij}(s) = \mathbb{E}[Z] > 0$. By Chebyshev's Inequality, we obtain

$$\Pr\left(|Z - \mathbb{E}[Z]| \geq \Delta_{ij}(s)\right) \leq \frac{\mathrm{Var}(Z)}{\Delta_{ij}^2(s)} \tag{A.1}$$

which implies that

$$\Pr\left(Z - \mathbb{E}[Z] \leq -\Delta_{ij}(s)\right) = \Pr\left(Z - \Delta_{ij}(s) \leq -\Delta_{ij}(s)\right) = \Pr\left(s(x_j) - s(x_i) \leq 0\right) \leq \frac{\mathrm{Var}(Z)}{\Delta_{ij}^2(s)} \tag{A.2}$$

Further noticing that $\mathrm{Var}(Z) = \mathrm{Var}(s(x_j) - s(x_i)) = \mathrm{Var}(s(x_j)) + \mathrm{Var}(s(x_i))$, we conclude that

$$\Pr\left(s(x_i) > s(x_j)\right) = \mathcal{O}\left(\frac{\mathrm{Var}(s(x_i)) + \mathrm{Var}(s(x_j))}{\Delta_{ij}^2(s)}\right) \tag{A.3}$$

which completes the proof. □

A.2 Proof of Theorem 1

Based on Lemma 1, below we provide the proof of Theorem 1.

Proof. To prove the first statement, i.e. $\Delta_{ij}(s_{\text{MEAN}}) > \Delta_{ij}(s_{\text{VAR}}) > 0$, we consider the following properties of a beta distribution $\mathcal{B}(\alpha, \beta)$.

$$\mu = \frac{\alpha}{\alpha+\beta}, \tag{A.4}$$

$$\sigma = \frac{\alpha\beta}{(\alpha+\beta)^2(1+\alpha+\beta)} \tag{A.5}$$

$$= \frac{\mu(1-\mu)}{1+\alpha+\beta} \tag{A.6}$$

where μ_i and σ_i respectively represent the mean and variance of the beta distribution $\mathcal{B}(\alpha_i, \beta_i)$.

Let $\alpha_i + \beta_i = \alpha_j + \beta_j = c$. Since $\alpha_i < \alpha_j \leq \beta_j$, we know

$$\mu_i = \frac{\alpha_i}{\alpha_i+\beta_i} < \frac{\alpha_j}{\alpha_j+\beta_j} = \mu_j \leq \frac{1}{2}, \tag{A.7}$$

$$\sigma_i = \frac{\mu_i(1-\mu_i)}{1+\alpha_i+\beta_i} < \frac{\mu_j(1-\mu_j)}{1+\alpha_j+\beta_j} = \sigma_j. \tag{A.8}$$

Therefore, we have

$$\Delta_{ij}(s_{\text{MEAN}}) = \mathbb{E}\left[s_{\text{MEAN}}(x_j) - s_{\text{MEAN}}(x_i)\right] = \mu_j - \mu_i > 0, \tag{A.9}$$

$$\Delta_{ij}(s_{\text{VAR}}) = \mathbb{E}\left[s_{\text{VAR}}(x_j) - s_{\text{VAR}}(x_i)\right] = \sigma_j - \sigma_i > 0. \tag{A.10}$$

Furthermore, notice that

$$\Delta_{ij}(s_{\text{VAR}}) = \frac{\mu_j(1-\mu_j) - \mu_i(1-\mu_i)}{1+c} \tag{A.11}$$

$$< \mu_j(1-\mu_j) - \mu_i(1-\mu_i) \tag{A.12}$$

$$= \mu_j - \mu_i - (\mu_j^2 - \mu_i^2) \tag{A.13}$$

$$< \Delta_{ij}(s_{\text{MEAN}}), \tag{A.14}$$

which proves the first statement of Theorem 1.

To prove the second statement, i.e., to provide an upper bound on the errors of s_{MEAN} and s_{VAR}, we plug in the definition of s_{MEAN} and s_{VAR} to Lemma 1:

$$\Pr\left(s_{\text{MEAN}}(x_i) > s_{\text{MEAN}}(x_j)\right) = \mathcal{O}\left(\frac{\text{Var}(s_{\text{MEAN}}(x_i)) + \text{Var}(s_{\text{MEAN}}(x_j))}{\Delta_{ij}^2(s_{\text{MEAN}})}\right) \tag{A.15}$$

$$\overset{(a)}{=} \mathcal{O}\left(\frac{\sigma_j + \sigma_i}{K\Delta_{ij}^2(s_{\text{MEAN}})}\right) \tag{A.16}$$

$$= \mathcal{O}\left(\frac{1}{K\Delta_{ij}^2(s_{\text{MEAN}})}\right) \tag{A.17}$$

where step (a) is due to $\mathrm{Var}(s_{\text{MEAN}}(x_i)) = \sigma_i/n$. Similarly,

$$\begin{aligned}
\Pr\left(s_{\text{VAR}}(x_i) > s_{\text{VAR}}(x_j)\right) &= \mathcal{O}\left(\frac{\mathrm{Var}(s_{\text{VAR}}(x_i)) + \mathrm{Var}(s_{\text{VAR}}(x_j))}{\Delta_{ij}^2(s_{\text{VAR}})}\right) && \text{(A.18)}\\
&\overset{(b)}{=} \mathcal{O}\left(\frac{1}{K\Delta_{ij}^2(s_{\text{VAR}})}\right). && \text{(A.19)}
\end{aligned}$$

Here, step (b) is due to the variance of sample variance $\mathrm{Var}(s_{\text{VAR}}(x_i)) = \frac{1}{K}(\mu_4 - \sigma^2(x_i)) + \mathcal{O}(n^{-2}) = \mathcal{O}\left(\frac{1}{K}\right)$ [11] where μ_4 is the Kurtosis of the beta distribution $\mathcal{B}(\alpha_i, \beta_i)$. □

Appendix B

Supporting Materials for Chapter 6

Table B.1: Performance in terms of FN-precision numbers for the Messidor-2 Dataset. The fraction in each entry shows the number of false negatives (the numerator) and the number of uncertain negatives (the denominator). The percentage numbers in the parentheses are the corresponding FN-precision values.

Ensemble Method	ρ	Uncertainty Metric	$q = 1$	$q = 2$	$q = 5$	$q = 10$	$q = 15$
Hyperparameter Ensemble	0.2	MEAN	286.45 / 580.59 (49.34%)	459.35 / 1088.36 (42.21%)	820.2 / 2285.32 (35.89%)	1163.8 / 3843.18 (30.28%)	1401.9 / 5169.09 (27.12%)
		VAR	175.45 / 471.55 (37.21%)	322.91 / 923.86 (34.95%)	669.68 / 2161.73 (30.98%)	1011.09 / 3768.86 (26.83%)	1238.5 / 5106.95 (24.25%)
		MEAN+VAR	305.85 / 704.86 (43.39%)	523.45 / 1321.68 (39.6%)	877.35 / 2569.23 (34.15%)	1192.6 / 4049.45 (29.45%)	1423.55 / 5352.45 (26.6%)
	0.4	MEAN	257.70 / 560.09 (46.01%)	457.50 / 1073.45 (42.62%)	814.35 / 2340.55 (34.79%)	1131.65 / 3974.82 (28.47%)	1336.05 / 5335.91 (25.04%)
		VAR	129.27 / 348.95 (37.05%)	276.09 / 795.59 (34.7%)	619.86 / 2088.91 (29.67%)	949.59 / 3784.23 (25.09%)	1165.23 / 5216 (22.34%)
		MEAN+VAR	300.55 / 690.95 (43.5%)	511.70 / 1282.36 (39.9%)	867.00 / 2657.14 (32.63%)	1169.55 / 4292.95 (27.24%)	1361.6 / 5589.18 (24.36%)
	0.6	MEAN	182.05 / 404.36 (45.13%)	337.18 / 779.09 (43.36%)	757.50 / 2101.32 (36.05%)	1119.10 / 3889.55 (28.77%)	1332.40 / 5316.82 (25.06%)
		VAR	67.41 / 193.55 (34.83%)	165.50 / 489.64 (33.8%)	490.36 / 1705.36 (28.75%)	864.18 / 3510.23 (24.62%)	1112.32 / 5014.59 (22.18%)
		MEAN+VAR	216.00 / 507.82 (42.53%)	391.30 / 967.32 (40.45%)	819.03 / 2490.36 (32.9%)	1180.25 / 4405.14 (26.79%)	1387.55 / 5840.77 (23.76%)
	0.8	MEAN	184.75 / 455.68 (40.54%)	313.40 / 828.64 (37.82%)	601.00 / 1838.32 (32.69%)	912.45 / 3304.68 (27.61%)	1108.36 / 4570.09 (24.26%)
		VAR	73.86 / 252.50 (29.25%)	156.45 / 557.55 (28.06%)	367.27 / 1460.82 (25.14%)	600.95 / 2802.82 (21.44%)	790.00 / 4085.05 (19.34%)
		MEAN+VAR	215.75 / 588.45 (36.66%)	361.85 / 1053.82 (34.34%)	653.85 / 2184.50 (29.93%)	975.40 / 3932 (24.81%)	1161.70 / 5303.55 (21.9%)
	1.0	MEAN	139.35 / 297.05 (46.91%)	257.12 / 582.59 (44.11%)	594.50 / 1572.32 (37.78%)	1052.10 / 3648.09 (28.84%)	1293.40 / 5302.5 (24.39%)
		VAR	38.36 / 123.68 (31.02%)	89.09 / 290.91 (30.63%)	306.50 / 1159.18 (26.44%)	758.68 / 3215.27 (23.6%)	1064.59 / 5042.23 (21.11%)
		MEAN+VAR	163.85 / 377.59 (43.39%)	301.38 / 743.32 (40.6%)	665.50 / 1993.09 (33.39%)	1111.80 / 4165.68 (26.69%)	1354.75 / 5911.32 (22.92%)
MC-Dropout	0.2	MEAN	156.00 / 672.00 (23.21%)	259.00 / 1196 00(21.66%)	468.50 / 2404.00 (19.49%)	697.25 / 4143.50 (16.83%)	864.75 / 5682.00 (15.22%)
		VAR	47.00 / 342.00 (13.74%)	68.50 / 589.00 (11.63%)	102.50 / 1284.50 (7.98%)	118.50 / 1833.50 (6.46%)	154.05 / 2332.98 (6.60%)
		MEAN+VAR	198.75 / 996.00 (19.95%)	318.75 / 1737.00 (18.35%)	546.75 / 3553.00 (15.39%)	780.75 / 5763.00 (13.55%)	938.50 / 7220.08 (13.00%)
	0.4	MEAN	127.75 / 442.00 (28.9%)	278.00 / 1013.50 (27.43%)	588.50 / 2593.00 (22.7%)	840.75 / 4461.00 (18.85%)	1015.53 / 6070.00 (16.73%)
		VAR	26.00 / 128.00 (20.31%)	175.50 / 817.00 (21.48%)	234.05 / 1578.00 (14.86%)	302.75 / 2856.50 (10.60%)	336.50 / 3998.00 (8.42%)
		MEAN+VAR	148.75 / 543.58 (27.37%)	369.5 / 1529.00 (24.17%)	687.75 / 3641.00 (18.89%)	975.00 / 6531.00 (14.93%)	1164.75 / 9097.50 (12.8%)
	0.6	MEAN	99.25 / 368.00 (26.97%)	182.50 / 709.00 (25.74%)	386.50 / 1718.00 (22.50%)	651.50 / 3326.50 (19.59%)	846.00 / 4824.50 (17.54%)
		VAR	80.00 / 310.00 (25.81%)	110.25 / 480.50 (22.94%)	213.00 / 1357.00 (15.7%)	293.50 / 2597.50 (11.3%)	366.00 / 4017.50 (9.11%)
		MEAN+VAR	108.54 / 421.66 (26.97%)	213.75 / 877.50 (24.36%)	489.25 / 2598.50 (18.83%)	823.5 / 5382.00 (15.30%)	1060.25 / 8051.00 (13.17%)
	0.8	MEAN	80.25 / 275.00 (29.18%)	166.25 / 580.50 (28.64%)	420.50 / 1639.50 (25.65%)	758.25 / 3742.00 (20.26%)	967.75 / 5573.00 (17.36%)
		VAR	20.00 / 115.40 (17.33%)	102.50 / 360.50 (28.43%)	238.75 / 1286.50 (18.56%)	321.75 / 2755.00 (11.68%)	353.75 / 3894.50 (9.08%)
		MEAN+VAR	89.14 / 317.89 (29.18%)	225.00 / 815.50 (27.59%)	496.00 / 2295.00 (21.61%)	867.75 / 5566.00 (15.59%)	1096.25 / 8399.50 (13.05%)
	1.0	MEAN	97.50 / 356.50 (27.35%)	183.50 / 671.00 (27.35%)	390.75 / 1664.00 (23.48%)	649.25 / 3252.00 (19.96%)	846.41 / 4830.50 (17.51%)
		VAR	51.75 / 211.00 (24.53%)	95.00 / 509.50 (18.65%)	164.00 / 1345.50 (12.19%)	212.00 / 2514.50 (8.43%)	242.25 / 3620.00 (6.69%)
		MEAN+VAR	130.50 / 499.00 (26.15%)	248.75 / 1070.50 (23.24%)	507.75 / 2828.50 (17.95%)	778.25 / 5352.5 (14.54%)	984.50 / 7836.00 (12.56%)
TTA	0.2	MEAN	160.73 / 442.53 (36.32%)	300.73 / 867.40 (34.67%)	628.15 / 2043.80 (30.73%)	1001.05 / 3798.60 (26.35%)	1268.52 / 5405.95 (23.47%)
		VAR	101.92 / 439.09 (23.21%)	188.45 / 871.03 (21.64%)	384.03 / 2053.84 (18.7%)	603.42 / 3807.60 (15.85%)	761.99 / 5403.30 (14.10%)
		MEAN+VAR	225.89 / 615.50 (36.7%)	381.36 / 1105.61 (34.49%)	712.02 / 2366.18 (30.09%)	1072.93 / 4171.33 (25.72%)	1330.81 / 5796.02 (22.96%)
	0.4	MEAN	145.64 / 446.73 (32.60%)	270.80 / 863.79 (31.35%)	556.17 / 2005.39 (27.73%)	865.48 / 3684.75 (23.49%)	1076.09 / 5198.03 (20.70%)
		VAR	98.44 / 405.14 (24.30%)	183.54 / 800.00 (22.94%)	390.42 / 1915.01 (20.38%)	624.40 / 3583.47 (17.42%)	785.10 / 5113.27 (15.35%)
		MEAN+VAR	216.05 / 672.65 (32.12%)	357.82 / 1174.99 (30.45%)	635.78 / 2379.68 (26.72%)	927.29 / 4097.41 (22.63%)	1127.04 / 5660.57 (19.91%)
	0.6	MEAN	136.05 / 400.18 (34%)	254.99 / 790.19 (32.27%)	541.09 / 1888.11 (28.66%)	880.71 / 3601.50 (24.45%)	1123.41 / 5190.18 (21.64%)
		VAR	94.30 / 402.53(23.43%)	173.96 / 776.89(22.39%)	374.77 / 1863.93(20.11%)	630.69 / 3574.91(17.64%)	828.19 / 5161.17(16.05%)
		MEAN+VAR	204.16 / 649.81 (31.42%)	348.87 / 1168.24 (29.86%)	661.88 / 2488.02 (26.6%)	987.76 / 4306.54 (22.94%)	1208.71 / 5883.95 (20.54%)
	0.8	MEAN	125.70 / 340.94 (36.87%)	237.19 / 671.14 (35.34%)	518.29 / 1631.14 (31.77%)	871.00 / 3115.51 (27.96%)	1149.79 / 4553.01 (25.25%)
		VAR	55.64 / 282.56 (19.69%)	112.50 / 563.72 (19.96%)	281.28 / 1396.29 (20.14%)	562.35 / 2789.61 (20.16%)	852.66 / 4199.69 (20.30%)
		MEAN+VAR	174.40 / 583.68 (29.88%)	323.77 / 1114.29 (29.06%)	654.62 / 2433.87 (26.90%)	1003.66 / 4082.39 (24.59%)	1264.61 / 5528.58 (22.87%)
	1.0	MEAN	112.15 / 331.71 (33.81%)	214.10 / 655.97 (32.64%)	473.00 / 1595.31 (29.65%)	806.60 / 3091.35 (26.09%)	1076.09 / 4534.76 (23.73%)
		VAR	57.26 / 300.19 (19.08%)	110.43 / 573.19 (19.27%)	265.93 / 1393.10 (19.09%)	512.16 / 2784.15 (18.4%)	750.04 / 4197.76 (17.87%)
		MEAN+VAR	160.47 / 566.43 (28.33%)	287.13 / 1037.14 (27.68%)	572.94 / 2213.89 (25.88%)	908.01 / 3817.26 (23.79%)	1177.10 / 5311.63 (22.16%)

Table B.2: Performance in terms of FN-precision numbers for the Messidor-2 Dataset. The fraction in each entry shows the number of false negatives (the numerator) and the number of uncertain negatives (the denominator). The percentage numbers in the parentheses are the corresponding FN-precision values.

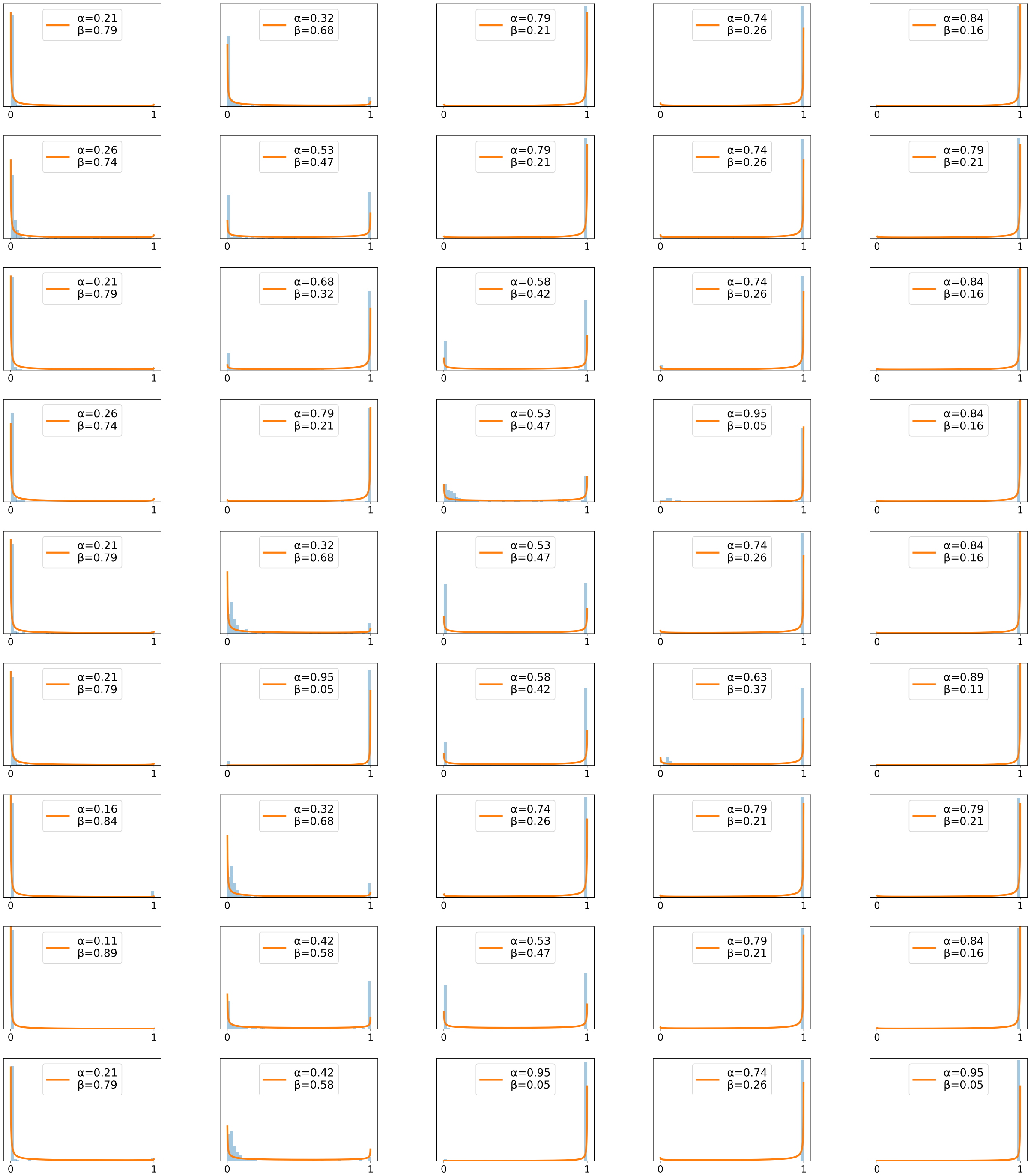
Ensemble Method	ρ	**Uncertainty Metric**	$q = 1$	$q = 2$	$q = 5$	$q = 10$	$q = 15$
Hyperparameter Ensemble	0.2	MEAN	34.05 / 46.67 (72.96%)	34.25 / 48.24 (71%)	68.45 / 103.77 (65.96%)	101.61 / 172.67 (58.85%)	121.06 / 227.24 (53.28%)
		VAR	29.17 / 40.49 (72.06%)	26.27 / 37.87 (69.37%)	49.01 / 75.83 (64.64%)	72.36 / 125.09 (57.85%)	87.29 / 164.57 (53.04%)
		MEAN+VAR	37.06 / 51.65 (71.75%)	47.59 / 68.51 (69.47%)	80.77 / 126.23 (63.99%)	108.63 / 191.37 (56.76%)	125.1 / 242.12 (51.67%)
	0.4	MEAN	22.48 / 32.66 (68.83%)	37.4 / 54.7 (68.37%)	70.75 / 113.8 (62.17%)	97.45 / 180.65 (53.94%)	112.8 / 228.75 (49.31%)
		VAR	20.14 / 30.63 (65.74%)	33.3 / 51.75 (64.35%)	51.95 / 85.10 (60.62%)	89 / 166.25 (53.53%)	105.05 / 213.45 (49.22%)
		MEAN+VAR	22.88 / 34.86 (65.82%)	51.85 / 79.8 (64.97%)	80.55 / 134 (60.11%)	101.65 / 194.05 (52.38%)	114.7 / 238.8 (48.03%)
	0.6	MEAN	20.5 / 26.26 (78.06%)	28.7 / 37.9 (76.56%)	63.9 / 86.9 (73.53%)	90.75 / 135.10(67.17%)	104.75 / 173.26 (60.46%)
		VAR	20.14 / 26.31 (76.54%)	28.5 / 37.87 (74.25%)	58.1 / 84.17 (69.02%)	85.1 / 135.92 (62.61%)	140.45 / 173.81 (57.79%)
		MEAN+VAR	24.19 / 31.54 (76.69%)	34.8 / 46.26 (75.22%)	74.35 / 105.76 (70.3%)	100.3 / 158.68 (63.21%)	142.65 / 252.3 (56.54%)
	0.8	MEAN	17.1 / 22.1 (77.37%)	27.35 / 37.51 (72.95%)	60.15 / 88.23 (68.17%)	100.65 / 170.5 (59.03%)	117.35 / 222.05 (52.85%)
		VAR	14.77 / 19.05 (77.55%)	25.15 / 34.45 (73%)	53.1 / 78.39 (67.74%)	96.25 / 161.6 (59.56%)	118.05 / 221.45 (53.31%)
		MEAN+VAR	18.02 / 23.72 (75.97%)	30.15 / 42.38 (71.15%)	63.5 / 96.7 (65.67%)	108.45 / 190.05 (57.06%)	123.9 / 242.15 (51.17%)
	1.0	MEAN	17.52 / 22.02 (79.55%)	35.4 / 45.97 (77.01%)	88.1 / 125.05 (70.45%)	123.75 / 194.8 (63.53%)	93.7 / 252.05 (57.01%)
		VAR	17.52 / 21.34 (76.27%)	30.75 / 42.36 (72.6%)	86.8 / 131.2 (66.16%)	121.3 / 201.2 (60.29%)	140.8 / 253.4 (55.56%)
		MEAN+VAR	20.27 / 22.66 (76.22%)	40.9 / 56.15 (72.88%)	112.45 / 170.7 (65.88%)	135.95 / 226.95 (59.9%)	149.6 / 274.45 (54.51%)
MC-Dropout	0.2	MEAN	8.89 / 13.24 (67.15%)	17.14 / 26.23 (65.35%)	39.12 / 67.2 (58.21%)	62.22 / 128.38 (48.47%)	82.14 / 178.22 (46.09%)
		VAR	10.07 / 16.22 (62.05%)	14.18 / 30.34 (46.73%)	20.11 / 59.2(33.98%)	23.15 / 93.23 (24.83%)	26.2 / 120.28 (21.78%)
		MEAN+VAR	13.19 / 25.24 (52.25%)	26.18 / 50.3 (52.05%)	50.22 / 111.3 (45.12%)	73.12 / 197.23 (37.07%)	92.21 / 263.28 (35.02%)
	0.4	MEAN	15.66 / 32.24 (48.58%)	33.19 / 54.33 (61.08%)	57.17 / 97.32 (58.74%)	78.13 / 152.21 (51.33%)	89.15 / 200.23 (44.52%)
		VAR	5.17 / 7.22 (41.66%)	5.21 / 7.29 (71.52%)	29.05 / 54.19 (53.6%)	30.16 / 82.29 (36.65%)	34.17 / 109.19 (31.3%)
		MEAN+VAR	16.38 / 32.26 (50.79%)	33.71 / 55.16 (61.12%)	61.19 / 110.31 (55.47%)	82.18 / 191.24 (42.97%)	95.15 / 258.3 (36.83%)
	0.6	MEAN	15.04 / 22.15 (67.88%)	23.07 / 43.16 (53.44%)	27.19 / 53.74(50.6%)	33.13 / 48.44 (45.69%)	56.21 / 134.44 (41.81%)
		VAR	1.18 / 1.77 (66.69%)	4.09 / 8.17 (50.08%)	13.12 / 46.22 (28.38%)	14.18 / 49.5 (28.59%)	24.9 / 97.25 (25.6%)
		MEAN+VAR	15.1 / 22.17 (68.15%)	24.14 / 45.25 (53.34%)	30.23 / 71.79 (42.11%)	37.18 / 83.38 (44.59%)	66.41 / 207.33 (32.03%)
	0.8	MEAN	7.16 / 13.28 (53.92%)	9.13 / 16.21 (56.3%)	18.06 / 35.16 (51.36%)	33.25 / 69.28 (47.99%)	64.18 / 130.34 (49.24%)
		VAR	4.25 / 6.9 (38.76%)	6.18 / 9.25 (66.76%)	18.13 / 40.28 (45%)	37.24 / 93.3 (39.92%)	50.16 / 126.22 (39.74%)
		MEAN+VAR	7.2 / 13.34 (53.96%)	15.15 / 25.28 (59.93%)	36.19 / 75.24 (48.1%)	68.26 / 158.38 (43.1%)	108.22 / 247.38 (43.75%)
	1.0	MEAN	5.14 / 12.18 (42.19%)	6.11 / 17.15 (35.61%)	37.16 / 75.2 (49.42%)	58.17 / 128.22 (45.37%)	69.16 / 168.22 (41.11%)
		VAR	4.15 / 12.26 (33.85%)	4.25 / 16.32 (26.03%)	6.15 / 26.22 (23.47%)	7.13 / 42.23 (16.88%)	9.26 / 73.32 (12.63%)
		MEAN+VAR	8.09 / 21.19 (38.16%)	9.2 / 30.34 (30.33%)	39.13 / 90.2 (43.38%)	61.15 / 156.3 (39.13%)	73.17 / 223.28 (32.77%)
TTA	0.2	MEAN	15.42 / 25.84 (59.6%)	24.28 / 41.65 (58.29%)	49.56 / 93.46 (53.03%)	75.78 / 163.84 (46.25%)	92.52 / 223.52 (41.39%)
		VAR	14.27 / 23.79 (59.99%)	23.43 / 40.3 (58.13%)	48.1 / 91.33 (52.66%)	75.23 / 163.15 (46.11%)	92.13 / 223.53 (41.21%)
		MEAN+VAR	16.79 / 28.33 (59.26%)	26 / 45.05 (57.72%)	50.36 / 96.42 (52.23%)	77.63 / 169.9 (45.69%)	94.22 / 231.09 (40.77%)
	0.4	MEAN	12.27 / 22.39 (54.8%)	20.26 / 37.93 (53.42%)	42.14 / 82.11 (51.32%)	73.4 / 153.43 (47.84%)	93.67 / 212.11 (44.16%)
		VAR	7.02 / 14.36 (48.87%)	11.47 / 24.61 (46.61%)	25.91 / 60.49 (42.84%)	48.88 / 122.81 (39.8%)	70.06 / 182.98 (38.29%)
		MEAN+VAR	17.27 / 33.19 (52.04%)	29.33 / 58.16 (50.42%)	61.22 / 128.63 (47.6%)	89.44 / 204.85 (43.66%)	101.75 / 248.21 (40.99%)
	0.6	MEAN	11.48 / 22.35 (51.37%)	19.57 / 38.61 (50.68%)	38.65 / 78.44 (49.28%)	69.91 / 153.38 (45.58%)	94.44 / 224.03 (42.16%)
		VAR	6.95 / 13.58 (51.21%)	12.68 / 25.33 (50.08%)	35.43 / 71.75 (49.38%)	69.64 / 152.8 (45.58%)	94.13 / 224 (42.02%)
		MEAN+VAR	16.51 / 32.3 (51.11%)	28.31 / 56.46 (50.14%)	51.71 / 105.96 (48.8%)	75.71 / 168.65 (44.89%)	97.85 / 235.1 (41.62%)
	0.8	MEAN	10.81 / 20.18 (53.57%)	17.55 / 33.84 (51.86%)	33.78 / 72.11 (46.84%)	55.53 / 142.75 (38.9%)	69.61 / 199.8 (34.84%)
		VAR	12.14 / 24.09 (50.42%)	19.05 / 38.48 (49.51%)	33.84 / 73 (46.35%)	54.5 / 137.78 (39.56%)	68.35 / 194.13 (35.21%)
		MEAN+VAR	16.04 / 31.03 (51.7%)	23.57 / 47.18 (49.96%)	37.97 / 83.46 (45.49%)	57.9 / 151.68 (38.18%)	71.66 / 208.88 (34.31%)
	1.0	MEAN	7.02 / 13.89 (50.54%)	11.82 / 22.86 (51.69%)	28.39 / 57.41 (49.46%)	54.8 / 122.8 (44.62%)	77.41 / 183.14(42.27%)
		VAR	8.69 / 21.26 (40.88%)	14.39 / 34.35 (41.9%)	28.24 / 70.05 (40.31%)	51.41 / 130.38 (39.44%)	72.1 / 185.7(38.82%)
		MEAN+VAR	14.08 / 31 (45.41%)	24.58 / 53.06 (46.32%)	54.95 / 123.21 (44.6%)	95.78 / 230.74 (41.51%)	117.39 / 292.46(40.14%)

Table B.3: Number of the remaining false negative predictions from uncertainty-informed diagnosis schemes for the Kaggle-DR dataset. The reduction from the baseline (no uncertainty information is exploited) is shown as percentage numbers in the parentheses.

Ensemble Method	ρ	**Uncertainty Metric**	$q = 1$	$q = 2$	$q = 5$	$q = 10$	$q = 15$
Hyperparameter Ensemble	0.2	MEAN	2449.65 (-9.43%)	2265 (-16.26%)	1918.15 (-29.08%)	1586.8 (-41.33%)	1359.2 (-49.75%)
		VAR	2519.95 (-6.83%)	2363.6 (-12.61%)	1997.65 (-26.14%)	1636.15 (-39.51%)	1394.1 (-48.46%)
		MEAN+VAR	2410.05 (-10.89%)	2201.8 (-18.59%)	1860.95 (-31.2%)	1558.85 (-42.37%)	1338.7 (-50.5%)
	0.4	MEAN	2120.85 (-10.42%)	1929.25 (-18.51%)	1587.1 (-32.97%)	1281.25 (-45.88%)	1086.95 (-54.09%)
		VAR	2230.95 (-5.77%)	2076.65 (-12.29%)	1713.45 (-27.63%)	1365.75 (-42.32%)	1134.75 (-52.07%)
		MEAN+VAR	2079.25 (-12.18%)	1875.35 (-20.79%)	1533.6 (-35.23%)	1244.5 (-47.44%)	1057.85 (-55.32%)
	0.6	MEAN	2217.6 (-7.27%)	2068.15 (-13.52%)	1663.85 (-30.43%)	1316.5 (-44.95%)	1113.05 (-53.46%)
		VAR	2320.5 (-2.97%)	2215.9 (-7.34%)	1874.6 (-21.61%)	1477.1 (-38.24%)	1215.15 (-49.19%)
		MEAN+VAR	2183.55 (-8.7%)	2014.7 (-15.76%)	1605.05 (-32.89%)	1259.6 (-47.33%)	1060.5 (-55.66%)
	0.8	MEAN	2074.05 (-7.84%)	1949.45 (-13.38%)	1674.25 (-25.61%)	1373.55 (-38.97%)	1188.55 (-47.19%)
		VAR	2172.35 (-3.47%)	2085.7 (-7.32%)	1864.25 (-17.16%)	1615.75 (-28.21%)	1412.45 (-37.24%)
		MEAN+VAR	2042.7 (-9.24%)	1904.35 (-15.38%)	1626.45 (-27.73%)	1316.6 (-41.5%)	1132.65 (-49.67%)
	1.0	MEAN	2066.75 (-6.06%)	1952.4 (-11.26%)	1629.55 (-25.93%)	1189.95 (-45.91%)	958.75 (-56.42%)
		VAR	2159.55 (-1.84%)	2106.7 (-4.25%)	1875.7 (-14.74%)	1398.5 (-36.43%)	1076.2 (-51.08%)
		MEAN+VAR	2042.2 (-7.18%)	1909.2 (-13.22%)	1561.6 (-29.02%)	1129.95 (-48.64%)	896.45 (-59.25%)
MC-Dropout	0.2	MEAN	4496.1 (-0.39%)	4486.9 (-0.59%)	4454.9 (-1.3%)	4413.1 (-2.22%)	4374.3 (-3.08%)
		VAR	4509.5 (-0.09%)	4507.7 (-0.13%)	4498.5 (-0.33%)	4498.3 (-0.34%)	4499.5 (-0.31%)
		MEAN+VAR	4493.7 (-0.44%)	4482.7 (-0.68%)	4443.7 (-1.55%)	4399.1 (-2.53%)	4364.5 (-3.3%)
	0.4	MEAN	4035.2 (-0.27%)	4026 (-0.49%)	3993.2 (-1.3%)	3980.2 (-1.63%)	3965.5 (-1.99%)
		VAR	4043.9 (-0.05%)	4041.5 (-0.11%)	4036.8 (-0.23%)	4039.3 (-0.17%)	4038.5 (-0.19%)
		MEAN+VAR	4033.7 (-0.3%)	4023.9 (-0.55%)	3989.5 (-1.4%)	3972.3 (-1.82%)	3956 (-2.22%)
	0.6	MEAN	3678.4 (-0.35%)	3671.8 (-0.53%)	3657 (-0.93%)	3640.9 (-1.37%)	3622.1 (-1.88%)
		VAR	3687.3 (-0.11%)	3687 (-0.12%)	3685.2 (-0.17%)	3682.2 (-0.25%)	3683.2 (-0.22%)
		MEAN+VAR	3674.1 (-0.47%)	3665.1 (-0.72%)	3648 (-1.18%)	3632.3 (-1.6%)	3615.8 (-2.05%)
	0.8	MEAN	3794.5 (-0.17%)	3785.5 (-0.41%)	3765 (-0.95%)	3741.5 (-1.57%)	3721.8 (-2.08%)
		VAR	3798.8 (-0.06%)	3797.8 (-0.08%)	3797.1 (-0.1%)	3795.1 (-0.16%)	3796 (-0.13%)
		MEAN+VAR	3791.1 (-0.26%)	3783.8 (-0.45%)	3763.1 (-1%)	3733.7 (-1.77%)	3725.1 (-2%)
	1.0	MEAN	3517.8 (-0.2%)	3511.8 (-0.37%)	3490.6 (-0.98%)	3469.3 (-1.58%)	3455.4 (-1.97%)
		VAR	3522.2 (-0.08%)	3519.7 (-0.15%)	3516.7 (-0.24%)	3518.1 (-0.2%)	3519.5 (-0.16%)
		MEAN+VAR	3513.2 (-0.33%)	3507.4 (-0.5%)	3488.1 (-1.05%)	3465.3 (-1.69%)	3450 (-2.13%)
TTA	0.2	MEAN	3645.64 (-0.54%)	3629.61 (-0.98%)	3590.94 (-2.03%)	3549.78 (-3.15%)	3520.88 (-3.94%)
		VAR	3655.58 (-0.27%)	3647.67 (-0.48%)	3629.36 (-0.98%)	3607.78 (-1.57%)	3593.35 (-1.96%)
		MEAN+VAR	3641.16 (-0.66%)	3624.16 (-1.12%)	3585.15 (-2.19%)	3545.99 (-3.26%)	3517.06 (-4.05%)
	0.4	MEAN	3618.64 (-0.43%)	3606.51 (-0.77%)	3574.31 (-1.65%)	3539.63 (-2.61%)	3511.74 (-3.37%)
		VAR	3629.76 (-0.13%)	3624.83 (-0.26%)	3612.01 (-0.62%)	3594.3 (-1.1%)	3580.51 (-1.48%)
		MEAN+VAR	3613.65 (-0.57%)	3596.8 (-1.03%)	3560.31 (-2.04%)	3521.91 (-3.09%)	3496.29 (-3.8%)
	0.6	MEAN	3298.46 (-0.15%)	3289.08 (-0.3%)	3253.57 (-0.86%)	3210.81 (-1.49%)	3173.9 (-2.04%)
		VAR	3306.92 (-0.02%)	3306 (-0.03%)	3302 (-0.1%)	3289.69 (-0.29%)	3268.08 (-0.63%)
		MEAN+VAR	3295.75 (-0.2%)	3281.79 (-0.42%)	3249.22 (-0.91%)	3206.09 (-1.57%)	3155.75 (-2.31%)
	0.8	MEAN	3058.86 (-0.55%)	3044.11 (-1.02%)	3011.75 (-2.08%)	2981.05 (-3.07%)	2954.78 (-3.93%)
		VAR	3068.14 (-0.24%)	3061.46 (-0.46%)	3044.88 (-1%)	3027.69 (-1.56%)	3015.8 (-1.95%)
		MEAN+VAR	3052.75 (-0.74%)	3038.33 (-1.21%)	3004.99 (-2.3%)	2972.66 (-3.35%)	2951.89 (-4.02%)
	1.0	MEAN	2777.21 (-0.25%)	2709.25 (-1.39%)	2719.44 (-1.22%)	2663.14 (-2.12%)	2631.25 (-2.62%)
		VAR	2774.09 (-0.31%)	2786.42 (-0.1%)	2774.25 (-0.31%)	2751.93 (-0.68%)	2744.39 (-0.81%)
		MEAN+VAR	2749.96 (-0.71%)	2714.22 (-1.3%)	2726.37 (-1.1%)	2667.96 (-2.05%)	2566.55 (-3.6%)

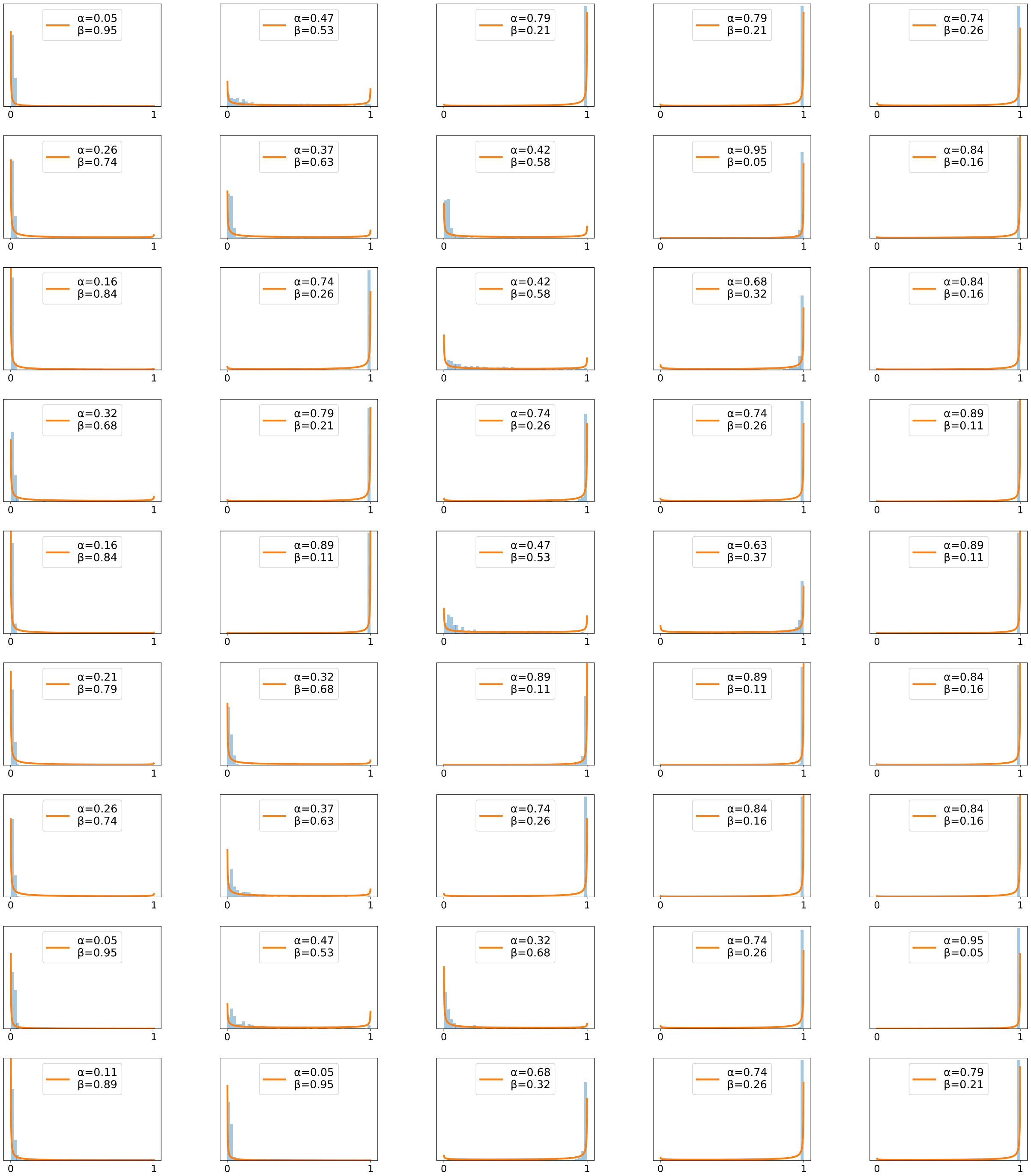
Table B.4: Number of the remaining false negative predictions from uncertainty-informed diagnosis schemes for the Messdior-2 dataset. The reduction from the baseline (where no uncertainty information is exploited) is shown as percentage numbers in the parentheses.

Ensemble Method	ρ	**Uncertainty Metric**	$q = 1$	$q = 2$	$q = 5$	$q = 10$	$q = 15$
Hyperparameter Ensemble	0.2	MEAN	201.95 (-7.93%)	192.75 (-12.13%)	160.75 (-26.72%)	118.95 (-45.77%)	80.15 (-63.46%)
		VAR	205.58(-7.64%)	193.52 (-11.77%)	173.39 (-20.95%)	138.34(-36.93%)	95.68 (-56.38%)
		MEAN+VAR	199.55 (-9.03%)	188.55 (-14.04%)	149.55 (-31.82%)	104.95 (-52.15%)	70.35 (-67.93%)
	0.4	MEAN	164.4 (-4.47%)	160.5 (-6.74%)	154.5 (-10.23%)	139.3 (-19.06%)	135.7 (-21.15%)
		VAR	164.30 (-4.53%)	162.97 (-5.30%)	159.6 (-7.25%)	146.78 (-14.71%)	142.05(-17.46%)
		MEAN+VAR	163.3 (-5.11%)	156.9 (-8.83%)	142.5 (-17.2%)	143.7 (-16.5%)	129.3 (-24.87%)
	0.6	MEAN	190.84 (-4.13%)	188.76 (-5.17%)	186.69 (-6.21%)	181.18 (-8.98%)	1174.97 (-12.28%)
		VAR	191.24 (-3.92%)	190.44 (-4.32%)	188.12 (-5.49%)	181.98 (-8.58%)	174.79 (-12.1%)
		MEAN+VAR	189.88 (-4.61%)	188.12 (-5.49%)	185.96 (-6.58%)	180.18 (-9.48%)	172.23 (-13.48%)
	0.8	MEAN	165.1 (-4.67%)	164.78 (-4.86%)	163.98 (-5.33%)	157.78 (-8.9%)	152.67 (-11.85%)
		VAR	165.25 (-4.59%)	165.42 (-4.5%)	163.26 (-5.74%)	160.17 (-7.52%)	153.28 (-11.5%)
		MEAN+VAR	164.78 (-4.86%)	163.34 (-5.7%)	160.46 (-7.36%)	150.68 (-13%)	141.8 (-18.13%)
	1.0	MEAN	164.45 (-1.62%)	161.15 (-3.59%)	157.7 (-5.65%)	156.35 (-6.46%)	152.45 (-8.79%)
		VAR	165.25 (-1.14%)	163.85 (-1.97%)	161.25 (-3.53%)	158.95 (-4.91%)	159.15 (-4.79%)
		MEAN+VAR	164.1 (-1.88%)	159.2 (-4.76%)	152.15 (-8.97%)	151.55 (-9.33%)	147.5 (-11.76%)
MC-Dropout	0.2	MEAN	316.39 (-0.41%)	315.73 (-0.62%)	314.25 (-1.09%)	312.64 (-1.59%)	310.76 (-2.18%)
		VAR	317.28 (-0.13%)	317.25 (-0.14%)	317.07 (-0.2%)	316.77 (-0.29%)	316.87 (-0.26%)
		MEAN+VAR	315.96 (-0.55%)	315.06 (-0.83%)	313.35 (-1.37%)	311.78 (-1.86%)	310.13 (-2.38%)
	0.4	MEAN	294.32 (-0.37%)	293.4 (-0.68%)	290.12 (-1.79%)	288.82 (-2.23%)	287.35 (-2.73%)
		VAR	295.19 (-0.07%)	294.95 (-0.15%)	294.48 (-0.31%)	294.73 (-0.23%)	294.65 (-0.25%)
		MEAN+VAR	294.17 (-0.42%)	293.19 (-0.75%)	289.75 (-1.91%)	288.03 (-2.49%)	286.4 (-3.05%)
	0.6	MEAN	279.96 (-0.62%)	279.04 (-0.94%)	275.84 (-2.08%)	271.66 (-3.56%)	267.78 (-4.94%)
		VAR	281.3 (-0.14%)	281.12 (-0.21%)	280.2 (-0.53%)	280.18 (-0.54%)	280.3 (-0.5%)
		MEAN+VAR	279.72 (-0.7%)	278.62 (-1.09%)	274.72 (-2.48%)	270.26 (-4.06%)	266.8 (-5.29%)
	0.8	MEAN	266.75 (-0.24%)	265.85 (-0.58%)	263.8 (-1.35%)	261.45 (-2.23%)	259.48 (-2.96%)
		VAR	267.18 (-0.08%)	267.08 (-0.12%)	267.01 (-0.15%)	266.81 (-0.22%)	266.9 (-0.19%)
		MEAN+VAR	266.41 (-0.37%)	265.68 (-0.64%)	263.61 (-1.42%)	260.67 (-2.52%)	259.81 (-2.84%)
	1.0	MEAN	251.78 (-0.29%)	251.18 (-0.52%)	249.06 (-1.36%)	246.93 (-2.21%)	245.54 (-2.76%)
		VAR	252.22 (-0.11%)	251.97 (-0.21%)	251.67 (-0.33%)	251.81 (-0.27%)	251.95 (-0.22%)
		MEAN+VAR	251.32 (-0.47%)	250.74 (-0.7%)	248.81 (-1.46%)	246.53 (-2.36%)	245 (-2.97%)
TTA	0.2	MEAN	350.36 (-0.19%)	349.91 (-0.31%)	348.44 (-0.73%)	345.99 (-1.43%)	343.69 (-2.09%)
		VAR	350.74 (-0.08%)	350.2 (-0.23%)	349.61 (-0.4%)	348.48 (-0.72%)	347.84 (-0.9%)
		MEAN+VAR	349.56 (-0.41%)	348.76 (-0.64%)	345.71 (-1.51%)	341.31 (-2.76%)	340.66 (-2.95%)
	0.4	MEAN	207.1 (-0.59%)	206.13 (-1.06%)	203.53 (-2.3%)	200.9 (-3.56%)	198.93 (-4.51%)
		VAR	207.94 (-0.19%)	207.68 (-0.31%)	207.14 (-0.57%)	205.89 (-1.17%)	205.13 (-1.54%)
		MEAN+VAR	206.03 (-1.12%)	205.63 (-1.3%)	202.3 (-2.89%)	199.08 (-4.48%)	198.3 (-4.81%)
	0.6	MEAN	197.06 (-0.61%)	196.26 (-1.02%)	193.37 (-2.47%)	190.44 (-3.95%)	188.97 (-4.69%)
		VAR	197.57 (-0.35%)	197.04 (-0.62%)	195.78 (-1.26%)	194.24 (-2.03%)	193.8 (-2.25%)
		MEAN+VAR	196.42 (-0.93%)	195.81 (-1.24%)	193.36 (-2.48%)	189.72 (-4.31%)	188.44 (-4.96%)
	0.8	MEAN	202.48 (-0.49%)	201.2 (-1.12%)	200.15 (-1.63%)	196.4 (-3.48%)	193.8 (-4.75%)
		VAR	203.16 (-0.15%)	202.96 (-0.25%)	201.73 (-0.86%)	200.19 (-1.62%)	198.74 (-2.33%)
		MEAN+VAR	201.93 (-0.76%)	201 (-1.22%)	197.73 (-2.83%)	196.83 (-3.27%)	194.53 (-4.4%)
	1.0	MEAN	162.23 (-0.55%)	161.28 (-1.13%)	160.1 (-1.85%)	156.98 (-3.77%)	156.33 (-4.17%)
		VAR	162.58 (-0.34%)	162.18 (-0.58%)	161.74 (-0.85%)	160.61 (-1.54%)	159.61 (-2.15%)
		MEAN+VAR	161.85 (-0.78%)	161.18 (-1.2%)	159.95 (-1.95%)	157.23 (-3.62%)	157.35 (-3.54%)



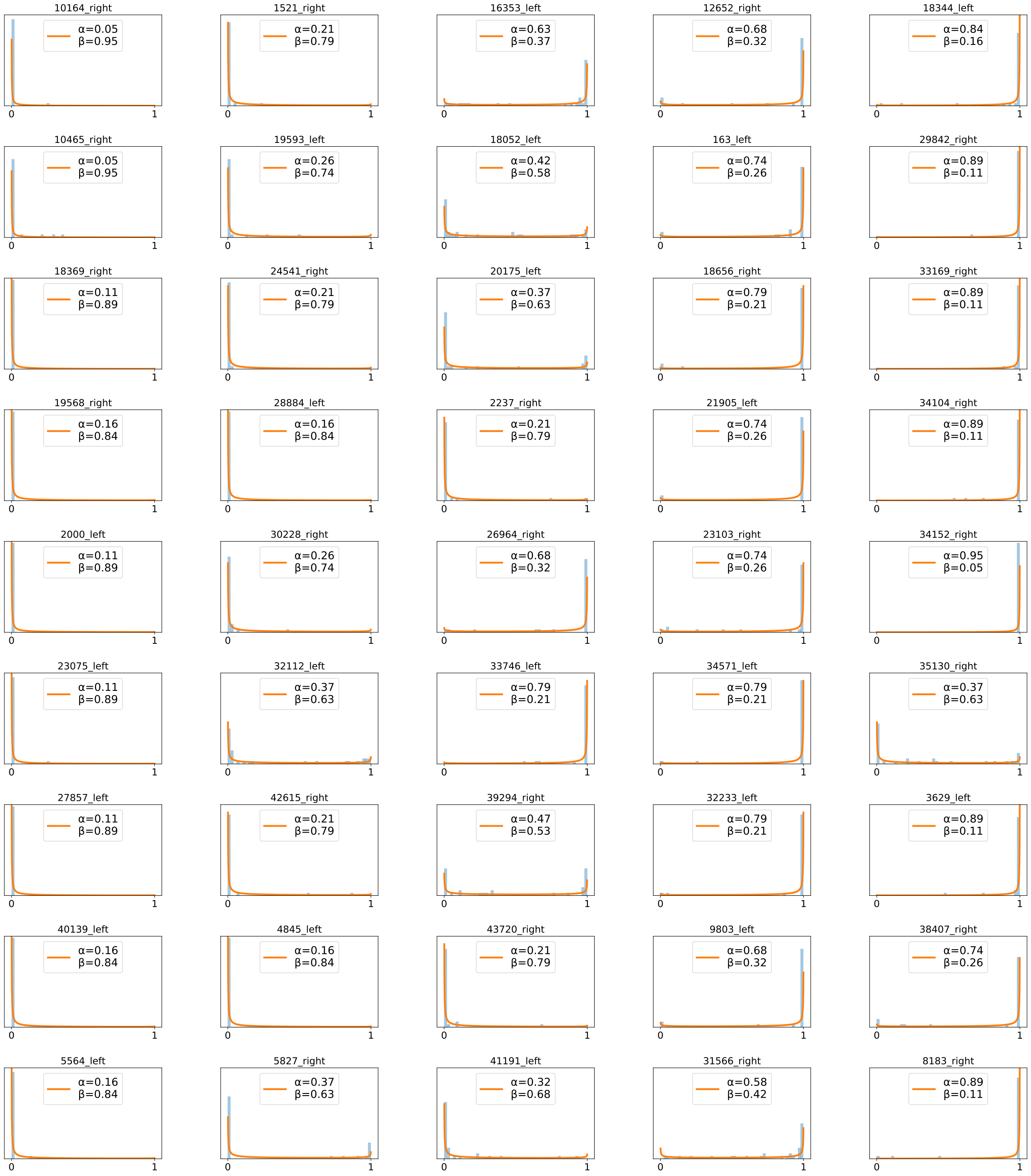
(a) SL0 (b) SL1 (c) SL2 (d) SL3 (e) SL4

Figure B.1: Histograms showing the spread of trained (with $\rho = 0.2$) DT models' predictions on a selected numbers of data examples from the chiller dataset, and a fitted beta distribution $\mathcal{B}(\alpha, \beta)$ for each example.



(a) SL0 (b) SL1 (c) SL2 (d) SL3 (e) SL4

Figure B.2: Histograms showing the spread of trained (with $\rho = 0.2$) NN models' predictions on a selected numbers of data examples from the chiller dataset, and a fitted beta distribution $\mathcal{B}(\alpha, \beta)$ for each example.



(a) SL0 (b) SL1 (c) SL2 (d) SL3 (e) SL4

Figure B.3: Histograms showing the spread of trained (with $\rho = 0.2$) NN models' predictions on a selected numbers of data examples from the diabetic dataset, and a fitted beta distribution $\mathcal{B}(\alpha, \beta)$ for each example.