Analytics-Enabled Quality and Safety Management Methods for High-Stakes Manufacturing Applications

by

Joshua Wilde

B.S., Tufts University (2011)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2023

Author . . . .
Sloan School of Management
November 1, 2022

Certified by . . . .
Retsef Levi
J. Spencer Standish (1945) Professor of Operations Management
Thesis Supervisor

Accepted by . . . .
Georgia Perakis
William F. Pounds Professor of Management Science
Co-Director, Operations Research Center

Analytics-Enabled Quality and Safety Management Methods for High-Stakes Manufacturing Applications

by

Joshua Wilde

Submitted to the Sloan School of Management
on November 1, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Quality management is a critical aspect of the management of manufacturing processes, particularly in industries where product reliability and safety are paramount. With increased digitization and automation, there is growing potential for analytical tools combined with ubiquitous data to aid the transition from quality management practices based on expert intuition and qualitative insights to more data-driven decision making. To assist in bridging the gap between this potential and current implementation practices, this thesis develops new methods for analytics-enabled quality and safety management.

Chapter 2 focuses on the problem of detecting clinically-relevant quality variation in pharmaceutical manufacturing of biologic drugs. Currently, both pre-market clinical trials and post-marketing studies focus on variability in safety outcomes due to individual patient-drug factors. However, the inherent complexity of biologic drug manufacturing and distribution raises potential risks that temporal variability in these systems could also impact clinical outcomes. The chapter describes a data-driven signal detection method using Hidden Markov models designed to monitor for manufacturing lot-dependent changes based on reported clinical outcomes. The method is tested on three lot sequences from a major biologic drug. The results suggest correlated lot-to-lot variability in two of the three, possibly related to changing manufacturing and supply chain conditions that may impact the per lot AE rates.

Chapter 3 explores the problem of creating structured access to unstructured quality data captured in free-text documents. Though operator reports and logs are ubiquitous in many manufacturing processes, one of the main barriers to their effective use in decision making is that unstructured data are often unclassified, which makes trend identification and other actionable analyses challenging. This chapter describes a machine learning and optimization-driven methodology to classify unstructured text in process environments into a known taxonomy of categories without access to an existing labeled training set. To accomplish this, the proposed method leverages information from existing reference documentation and formulates a linear program to select a set of key words that distinguish the categories from each other. Results from three test datasets with ground-truth labels indicate that the method delivers strong classification accuracy, both in absolute terms and relative to alternative methods.

Chapter 4 focuses on a quality test for an optical transceiver module, a high-tech hardware product, manufactured by an industrial partner. Currently, human experts review all test logs for quality problems. This chapter proposes a two-stage machine learning classification model that is able to automatically pass the vast majority of tested products and drastically reduces the need for manual review. Assessment on out of sample real test result data suggests that the two-stage model is able to reduce the manual review burden on the operator by 75-99% while on average satisfying the requirement to limit the number of passed defective modules.

Thesis Supervisor: Retsef Levi
Title: J. Spencer Standish (1945) Professor of Operations Management

Acknowledgments

I would like to start by thanking my advisor, Professor Retsef Levi, for his guidance through my time at MIT over the past four years. His optimism throughout the research process frequently outpaced my own, pushing me to accomplish more than I thought possible. Most importantly, I admire his drive to solve important problems by drawing from a vast toolbox of existing methods and creativity to develop new ones. I will strive to emulate this approach to research for the rest of my career. I also greatly appreciate the feedback from Professors Vivek Farias and Jónas Jónasson as members of my general exam and thesis committees.

Many academic and industrial collaborators have guided and inspired the work in this thesis. Evan Yao deserves particular thanks, and I have learned much from his creativity and persistence. Additionally, thanks to my collaborators at the MIT Center for Biomedical Innovation, including Stacy Springs, Jackie Wolfrum, Professor Tony Sinskey, and Professor Richard Braatz for sharing their wealth of biomanufacturing knowledge. Eli Arad and Oren Horvitz from Colorchip provided similar guidance and were incredibly generous with their time.

Though I learned a tremendous amount from my collaborators during my time at MIT, I owe much thanks to the my former colleagues at Fidelity. I would like to particuarly thank Lisa Emsbo-Mattingly and Jordan Alexiev for their support and mentoring that prepared me for my work at MIT. I learned how to do creative research from them, and perhaps most importantly, I saw ideal examples of how to contribute positively in a team environment. I could not ask for better professional and personal role models.

Academically, I am thoroughly indebted to my friends and colleagues from the ORC. One of the ORC's great strengths has always been the bonds that peers form, especially during the first two years. Through the best and worst periods of my experience at MIT, Kayla Cummings, Vassilis Digilakis, Georgia Dimaki, Sam Gilmour, and Galit Lukin were constant sources of laughter, commiseration, and support. Thank you, and I would not be here without you!

Finally, I reserve my most heartfelt thanks to my family and friends for their support. To my wife, Kristin, for holding me together when I needed it the most, to my parents and my sister Jocelyn for their unwavering encouragement, and to the rest of the Borrero family, Kellie, Kathy, George, and Chris, for pushing me forward while helping maintain my balance, I can only say thank you. I leaned on each of these people, and many more, heavily for support over the past four years, and not a single one ever failed to prop me up.

Contents

Chapter 1

Introduction

Quality management is a critical aspect of the management of manufacturing processes, particularly in industries where product reliability and safety are paramount. With increased digitization and automation, there is growing potential for analytical tools combined with ubiquitous data to aid the transition from quality management practices based merely on expert intuition and qualitative insights to more data-driven decision making [58]. Furthermore, data-driven quality management is becoming a competitive necessity in the face of increasing product and supply chain complexity and growing pressure to reduce time-to-market [21].

Yet a large gap remains between the potential for data-driven quality management practices and its current implementation scope in the field. According to a 2015 survey by the consultancy AT Kearney including experts and executives in the industrial, automotive, and complex consumer goods industries, 72% of respondents believed in the benefits of innovative quality management initiatives, but only 22% of companies were actually applying such methods [21]. More broadly, a recent 2020 survey of 1,320 manufacturing executives found that 62% of companies have not scaled analytics initiatives beyond a single manufacturing line, and the most frequently cited reason (26% of executives) was a lack of skills and capabilities [4]. To assist in bridging the implementation gap, this thesis develops new methods for analytics-enabled quality and safety management that integrate tools from optimization, statistics and machine learning and apply them to available data from both outside and inside the manufacturing plant.

Chapter 2 of this thesis focuses on the problem of detecting clinically-relevant quality variation in pharmaceutical manufacturing of biologic drugs. The regulatory approval of biologics products also includes the approval of a manufacturing process with tight specifications and control limits. However, approval of the processes and control limits is typically based on clinical trials that rely on results from very few manufacturing lots. In particular, lot-to-lot, manufacturing-related variability that affects patient outcomes is difficult to detect before the drug goes to market. Additionally, post-marketing research on drug safety is concentrated on uncovering novel drug-adverse event combinations with a particular focus on product-patient interaction [15, 22, 19, 39]. However, the inherent complexity of the manufacturing, distribution, and overall handling systems of biologic drugs raises potential risks that temporal variability in manufacturing and supply chain conditions also could impact clinical outcomes.

Chapter 2 aims to augment existing post-marketing surveillance efforts by addressing the challenge of monitoring the patient impact of manufacturing and supply chain variability. The chapter describes a data-driven signal detection method, called *HMMScan*, designed to monitor for manufacturing lot-dependent changes based on currently reported clinical outcomes, specifically the rate of adverse events (AEs) per final product lot. The proposed method posits that in the absence of clinically meaningful manufacturing and supply chain variability, it is expected that the variability in per lot AE rates is due solely to patient-drug interaction, and therefore is expected to be statistically independent across lots. In contrast, if temporal variability of the manufacturing and supply chain conditions impacts patient outcomes, then it is expected that the per lot AE rates will show temporal correlation between lots packaged at similar times. Thus, detecting the latter could signal that the underlying manufacturing and supply chain conditions potentially impact the observed AEs.

The HMMScan method takes as input a sequence of lots ordered based on their packaging (manufacturing) dates and models the respective sequence of per lot AE rates using a set of candidate probabilistic models. The candidate models, which fall in the family of Hidden Markov models, span a range of hypotheses that cover both a scenario in which per lot AE rates are independent across lots, as well as scenarios with temporal correlation across sequential lots. The best-fitting model is determined using the Bayesian Information Criterion that balances between model explanatory power and complexity. Whether there exists a positive signal of temporal correlation is determined based on the selected model. The method was applied to three distinct lot sequences of a major biologic drug using datasets readily available to manufacturers and regulators, and potential manufacturing and supply chain variation was detected in two of the three.

Chapter 3 of this thesis shifts the focus to data collected from the manufacturing line and explores the problem of creating structured access to unstructured quality data captured in free-text deviation report documents. Though operator reports and logs are ubiquitous in many manufacturing processes, particularly in highly regulated industries, companies are often unable to fully utilize these reports and logs to inform data-driven quality management decisions. One of the main barriers is that unstructured data are often unclassified, which makes analysis of trends and identification of underlying repeated root causes challenging. Indeed, in many practical use cases, the taxonomy of categories naturally arises from the process context and is known by the operator, but there is no seamless way to classify unstructured documents into this taxonomy without extensive manual review and annotation. For example, in a manufacturing context, the categories could naturally correspond to different process steps or pieces of equipment, and unstructured documents might capture deviation reports that occur during the course of the manufacturing process.

Chapter 3 describes a machine learning and optimization-driven methodology to classify unstructured text in process environments into a known taxonomy of categories without access to an existing labeled training set. To accomplish this, the proposed method, called the Document Classification with Reference Information (DCRI) method, leverages information from existing reference documentation called *category descriptions*. The category descriptions are integrated into a classification model via a novel optimization formulation. This formulation selects key words that distinguish the categories from each other in both the input documents and the category descriptions, accounting for the fact that word usage in the category descriptions could be different than word usage in the documents. The ultimate category predictions place higher weight on these key words.

Across three datasets, including a pharmaceutical quality deviation dataset and two known datasets from news websites, the DCRI method is able to deliver high classification accuracy (84 - 89%). The DCRI method's accuracy approaches (within 1.5%) the accuracy of a supervised classifier trained on the ground truth labels for two of the three datasets. Additionally, the DCRI method is significantly more accurate (4-20% improvement) than other unsupervised methods for incorporating category descriptions into document classification predictions. In the two of three datasets with the most informative category descriptions, the DCRI method also outperforms a semi-supervised benchmark method by 1.4-5.7%. Moreover, the optimized key word selection within the DCRI method demonstrates consistent benefit across all datasets and drives an improvement in prediction accuracy ranging from 0.7-2.5%. Finally, the results show that a substantial labeling effort, of at least 15-30% of the dataset, is necessary to achieve classification performance equivalent to the DCRI method, and the number of required labels is even higher to achieve statistically equivalent accuracy on unseen documents.

Chapter 4 also focuses on quality data from inside the manufacturing plant, specifically related to a quality test for an optical transceiver module, a high-tech hardware product. While this quality test is conducted using dedicated machines and equipment, the final determination of whether the test results signal a quality problem is performed by highly skilled human experts. The use of highly skilled personnel to conduct repetitive tasks is not only costly, but potentially leads to inconsistent outcomes that could depend on specific individuals and their respective knowledge, training and expertise. This motivates the need to develop machine learning enabled automation to aid review of the test results.

This chapter proposes a two-stage machine learning classification model that is able to automatically pass the vast majority of tested products and drastically reduces the need for manual review of test logs. The proposed approach codifies the operator's qualitative observation that some modules are much easier to classify than others in a custom, rules-based classifier with thresholds set by data-driven optimization. A second stage random forest classifier is trained to specifically identify passing modules that are not handled by the rules-based classifier, and modules that cannot be passed by either stage are designated for manual inspection.

Assessment on out of sample real test result data suggests that the two-stage model is able to reduce the manual review burden on the operator by 75-99% while on average satisfying the requirement to limit the number of passed defective modules. Compared to existing state-of-the-art tree-based algorithms, the two-stage model is superior in reducing manual review at the expense of slightly inferior error control.

Chapter 2

Surveillance of Adverse Event Variability across Manufacturing Lots in Biologics

2.1 Introduction

Methods for detecting post-marketing safety signals have long been the subject of active pharmacovigilance academic research, as well as regulatory and industrial work. These efforts have primarily focused on uncovering novel drug-adverse event combinations [15, 22, 19, 39], and specifically on the product-patient interaction as the primary source of variability in clinical outcomes.

However, there are also known examples of serious adverse events (AEs), including fatalities of patients, caused by pharmaceutical products with root causes linked to manufacturing and supply chain sources [3]. The inherent complexity of manufacturing, distribution, and overall handling systems of biologic drugs underscores the importance of risks related to temporal variability in manufacturing and supply chain conditions that could potentially impact clinical outcomes.

The manufacturing process and related control mechanisms are specified in detail during the regulatory assessment and approval and are expected to be closely followed during the post-approval phase. However, in 2019 the U.S. Food and Drug Administration (FDA) stated that monitoring the impact of manufacturing and supply chain variability on patients remains an open challenge for the pharmacovigilance community [45].

This chapter aims to augment existing post-marketing surveillance frameworks, specifically by addressing this challenge. The paper describes a new data-driven signal detection method, called HMMScan, inspired by the family of Hidden Markov models (HMMs) [33]. Relying on standard reported clinical outcomes and manufacturing attributes, it is designed to monitor for manufacturing and supply chain lot-dependent changes. Specifically, the newly proposed method relies on the rate of reported AEs per final product lot to flag potential safety signals that could be related to variability in manufacturing and supply chain conditions.

The detection method posits the hypothesis that in the absence of clinically meaningful variability in manufacturing and supply chain-related processes, the variability in per lot AE rates is expected to be driven solely by patient-drug interaction, and therefore is statistically independent across lots. In contrast, if temporal variability of the manufacturing and supply chain conditions impacts patient outcomes, then it is expected that the per lot AE rates will show temporal, or serial, correlation between lots manufactured at similar times. Thus, detecting statistical evidence supporting the latter could signal that the underlying manufacturing and supply chain conditions potentially impact the observed AEs.

The newly proposed method relies on a probabilistic modeling framework that can be used to signal when the AE rates in a collection of lots seem to show serial correlation. The serial correlation is assessed with respect to the series of lots ordered by packaging date, which is used as a proxy for the manufacturing timing of the respective lots. Specifically, the model provides an indication that the underlying manufacturing or supply chain condition might have ‘safe’ (baseline) and risky states. Beyond providing a statistical signal regarding the presence of serial correlation, the model also indicates which particular lots are more likely to be related to risky states of the manufacturing or supply chain condition. This can help guide further investigation of potential causal factors that drive the risky states.

The HMMScan method is applied to a single product at a time and takes as input a sequence of final product lots with their respective reported AE rates. HMMScan considers multiple competing probabilistic candidate models, each fitted to these input data, and selects the model that best explains the observed data of the sequenced lots and their respective AE rates.

One of the main challenges in identifying the manufacturing or supply chain related impact on the per lot AE rates is the fact that conditions of the manufacturing or supply chain systems may not be fully observable. This motivates the use of candidate models that fall into the broad category of Hidden Markov models (HMMs), each consisting of two major elements. The first element is the number of underlying hidden (unobserved) states, and the respective transition probabilities from each state to all other states. The model assumes that each lot is manufactured and handled under a hidden state that corresponds to a different state of the underlying manufacturing or supply chain conditions. The second element is a state-dependent binomial mixture distribution that captures the probabilistic pattern of the AEs per lot manufactured under the respective state. The dynamic transition between hidden states in the HMMs captures the potential variability in the underlying manufacturing or supply chain conditions, and their impact on the number of reported AEs per lot is captured through the respective state-dependent binomial mixture distribution.

The candidate models capture a range of hypotheses including independent AE rates across lots (i.e., a single state), as well as serial correlation of AE rates between lots produced at similar times (multiple states, each with state-dependent mixture distribution). The 'best' model is selected using Bayesian Information Criteria (BIC) [41], which weighs the explanatory power of the model with respect to the observed data against the complexity of the model (number of parameters).

To illustrate the application of the approach, data available from the FDA Adverse Event Reporting System (FAERS) database were used to analyze a biologic drug currently on the market.

The rest of the chapter is organized as follows. Section 2.2 below reviews the existing literature. Section 2.3 details the HMMScan method described above, and Section 2.4 characterizes the accuracy of HMMScan under various simulated scenarios. Section 2.5 describes the input data used to demonstrate these methods and provides the results of the case studies. To conclude, Section 2.6 discusses aspects related to the application of the methods and directions for future data gathering and analysis.

2.2 Literature Review

Several papers have designed statistical approaches that take as input a time series of monthly AE reports and identify points where either temporary or systematic changes in the rate of AE reports occur [12, 17]. However, these approaches operate on aggregated monthly AE data and are not designed to specifically identify sequences of lots with unusually high AE rates. Additionally, in practice multiple lots may be used in parallel to treat patients, and the overall AE rates capture the aggregated number of AEs across all lots that are on the market.

Mahaux *et al.* [24] apply a hierarchical statistical scanning method to simulated batch genealogy data to identify intermediate process steps that are associated with excess adverse events. The method relies on data that capture relationships between final product batches that share bulk intermediate product batches. Whereas this method could be used by manufacturers, particularly for detailed root cause analysis, it would likely be impractical for use by a regulator that does not often have access to such granular data consistently across multiple different products.

2.3 HMMScan Method Description

The goal of the HMMScan method is to provide an alert when the pattern of per lot AE rates in a time-ordered sequence of lots suggests that there might exist serial correlation in consecutive lots. In this chapter, the temporal ordering is determined based on the packaging date of each lot. However, more generally, the specific order of lots could be further refined using information about the source of the intermediate materials for each lot. As already discussed, the presence of such serial correlation points to a potential impact of changing underlying manufacturing and supply chain conditions on the observed per lot AE rates.

2.3.1 Modeling Approach

The per lot AE rates of an ordered sequence of L manufacturing lots, each with D doses, is assumed to vary as a function of two unobserved factors. The first factor is the heterogeneity of the patient population, and the second factor is changes in manufacturing and supply chain conditions. To capture the effects of these factors, the method leverages a stochastic model called the Hidden Markov model (HMM).

In particular, let $\mathcal{C} = \{1, 2, \ldots, C\}$ be the set of patient subpopulations that are exposed to a given sequence of drug lots. Additionally, let $\mathcal{S} = \{1, 2, \ldots, S\}$ be the set of possible states of the underlying manufacturing and supply chain conditions. For each state $s \in \mathcal{S}$ and subpopulation $c \in \mathcal{C}$, let p_{sc} be the average probability per dose of incurring an AE. Let w_{sc} be the likelihood that a lot in state s is used within a subpopulation $c \in \mathcal{C}$. For simplicity of exposition, it is assumed that each dose generates either zero or one AE. The state of each lot $\ell \in \{1, 2, \ldots, L\}$, which is unobserved (or hidden), is a random variable denoted by H_ℓ. The number of observed AEs for lot l where $H_\ell = s$ is captured through a state-dependent mixture of binomials (MB) distribution. That is, for each integer $a \in \{1, 2, \ldots, D\}$:

$$P(A_\ell = a | H_\ell = s) = \sum_{c=1}^{C} \big(w_{sc} \cdot \text{Binomial}(a; D, p_{sc}) \big)$$

For the remainder of the chapter, the state with the lowest (highest) mean AE rate will be referred to as the "low-risk" ("high-risk") state. Additionally, the sequence of states $\{H_\ell\}_{\ell \in \{1,2,\ldots,L\}}$ evolves according to a Markov transition matrix that captures the probability of moving from each state to any other state. Specifically, the transition matrix identifies, for each pair of states $s, s' \in \mathcal{S}$, the probability $P(H_{\ell+1} = s | H_\ell = s')$. Finally, note that the transition matrix induces a stationary distribution over the states that represents the long-run frequency of each state if the hidden Markov process were run on an infinitely long lot sequence.

2.3.2 Input Data

The input data of the HMMScan method include, for each lot in the sequence, the observed AE rates, i.e., the number of AEs per $\mathcal{D}$ doses. These AE rates are denoted by the vector $\mathbf{a} = (a_1, a_2, \ldots, a_L)$. When the lots have different numbers of doses, the AE rate is normalized accordingly.

It is important to acknowledge concerns regarding censoring of reported AEs. Underreporting of AEs to spontaneous reporting systems has been a well-documented but not well-understood concern, with some estimates of the underreporting rate over 90% [16]. More recent research by Alatawi and Hansen continues to find wide disparities in the estimated underreporting rate across products, though the authors notably do not find any statistically significant underreporting for biologics [1]. Regardless, the HMMScan method does not require a precise estimate of AE underreporting rates. In particular, if the reporting rate is constant over time or known in terms of relative magnitude over time, the ability for the HMMScan method to detect serial correlation is unaffected by the absolute level of this rate. Moreover, while sudden, short-term changes in the reporting rate could be mistaken as state transitions that affect the results of the HMMScan method, long-term, moderate trends, either positive or negative, should not meaningfully affect the ability of the method to detect local serial correlation.

2.3.3 Model Selection Procedure

This section describes how the HMMScan method selects the HMM model structure with the best fit to the observed sequence of per lot AE rates from a set of candidate model structures. The HMMScan model selection procedure in Figure 2-1 takes as input the observed sequence of AE rates, $\mathbf{a}$, and a set of candidate HMM models. The candidate models are obtained by varying the assumed number of states and subpopulations (i.e., the size of S and C) over a grid of potential values from 1 to

S_{max} and C_{max}, respectively.

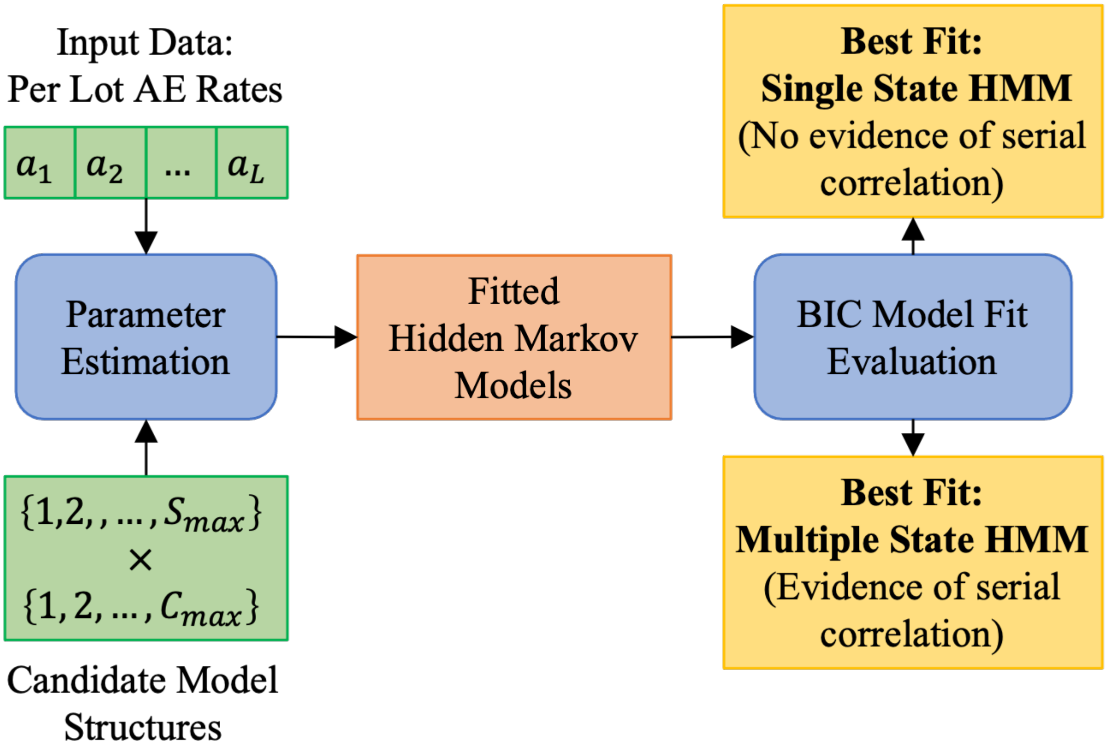

Input Data:
Per Lot AE Rates
a_1
a_2
...
a_L
Parameter Estimation
Fitted Hidden Markov Models
BIC Model Fit Evaluation
Best Fit: Single State HMM (No evidence of serial correlation)
Best Fit: Multiple State HMM (Evidence of serial correlation)
$\{1,2,,\ldots,S_{max}\}$
×
$\{1,2,\ldots,C_{max}\}$
Candidate Model Structures

Figure 2-1: HMMScan model selection procedure

Generally, the range of plausible HMM models in the typical use-cases considered in this paper can be covered by using small values for S_{max} and C_{max} (i.e., less than 10). The reason is that the number of relevant subpopulations is typically relatively small, and the manufacturing conditions can typically be aggregated into high-level states that capture the respective risk level for quality variation. Additionally, complex HMM structures with many hidden states and mixture components tend to overfit. Each candidate model corresponds to a hypothesis regarding the number of hidden states and patient subpopulations that best describes the observed AE rate sequence.

The HMMScan model selection procedure applies two sequential steps. The first step involves *Parameter Estimation* to calibrate the parameters of each candidate model. In the second step, *BIC Model Fit Evaluation* is used to determine which candidate models provide the best fit to the sequence of observed per lot AE rates.

During the Parameter Estimation step, maximum likelihood estimates for the HMM parameters are obtained via the Expectation Maximization (EM) algorithm [10]. For HMMs with $S = 1$, the EM algorithm uses closed form equations to iteratively optimize the binomial mixture weights and probabilities until convergence [27]. For HMMs with $S > 1$, the Baum-Welch algorithm, a variation of EM, optimizes both the transition probabilities and the state-specific distribution parameters [2]. The HMMScan implementation referenced in this paper relies on the implementations of EM and Baum-Welch in the pomegranate Python package [40]. Further details regarding parameter initialization can be found in Appendix A.1.

The second step of the HMMScan model selection procedure, BIC Model Fit Evaluation, compares the fitted candidate models using BIC and selects the model with the minimum BIC value. The BIC captures a tradeoff between the explanatory power of the model with respect to the data, and the complexity of the model in terms of the number of parameters. A detailed description of the BIC can be found in Appendix A.2, and a full derivation can be found in [34]. Pairwise differences in BIC values can also be translated into a more interpretable metric, the relative odds that one model fits the observed data better than the other. In [34], Raftery calculates that, for models fit on long input data sequences, a BIC difference of 10 or more indicates a greater than 99% probability that the model with the lower BIC value provides a stronger fit to the observed data.

2.3.4 Method Output

The HMMScan method outputs the best-fitting model according to the BIC, and this model can be used to detect whether there is statistical evidence in favor of serial correlation in the AE rates in the input lot sequence. If an HMM with $S > 1$ provides the best fit to the observed AE rates according to the BIC, then the HMMScan method signals that there is evidence in favor of serial correlation in AE rates for the input lot sequence. This is considered as a positive HMMScan signal for a serial correlation. On the other hand, if $S = 1$ provides the best fit, this is considered a negative HMMScan signal, i.e., no evidence of serial correlation.

In addition to indicating the potential presence of clinically relevant variation in manufacturing and supply chain conditions, the best-fitting HMM is used to identify the most likely sequence of hidden states associated with the input lot sequence. The mostly likely state sequence is calculated using the well-known and efficient Viterbi algorithm [33], which returns the path of hidden states that maximizes the joint likelihood of the hidden state sequence and the observed AE rates given the estimated maximum likelihood parameter values. These predicted hidden states can provide important temporal information as to what lots have been produced under high risk states, and this could be used to inform subsequent root cause analysis, as discussed in Section 2.5.4.

2.4 Method Validation

This section describes a validation and performance assessment of the HMMScan method through simulated synthetic data that capture different conditions and data input attributes. The selected conditions for the accuracy assessment are motivated by practical scenarios for true manufacturing and supply chain conditions. The specific instances for each respective scenario are captured through corresponding ground truth HMM models used to generate the synthetic data. Specifically, the scenarios vary in the number of hidden states, the degree of similarity of the state-dependent mixtures of binomial distributions, and the structure of the underlying transition matrix of the hidden states.

HMMScan is evaluated for its ability to detect the correct model structure for sample sequences of varying length generated by each ground truth model. For each sample sequence, the HMMScan method is applied according to the description in Section 2.3 above. Specifically, it fits a collection of candidate HMMs, each corresponding to a hypothesis about the structure of the ground truth HMM. This collection contains single-state models with up to six mixture components, two-state models with up to three mixture components, and three- and four-state models with up to two mixture components. Models with additional states and components did not provide the best BIC for any of the simulated sample sequences.

The performance of the HMMScan method is evaluated according to two metrics. The first metric is *detection accuracy*, which compares the structure of the lowest BIC model to the ground truth model. The second metric is *state prediction accuracy*, which evaluates the hidden state predictions. For a given sample sequence, the HMMScan method is deemed to have correctly detected the model structure if that sample is generated by a multiple-state (single-state) model and the model with the lowest BIC also has multiple states (a single state). The detection accuracy of HMMScan is defined for a particular ground truth model structure as the fraction of samples for which HMMScan correctly detects the model structure.

The state prediction accuracy for a single sample sequence is defined as the balanced accuracy of the per lot hidden state predictions from the model with the lowest BIC. Balanced accuracy is defined as the equally weighted average of the hidden state prediction accuracies for each hidden state. This metric is used to correct for imbalance in the ground truth frequency of the hidden states in a sample sequence. In instances with a ground truth model with multiple states, and where the model with the lowest BIC, selected by the HMMScan method, has a single-state structure, the ground truth state with the lowest mean AE rate (the low-risk state) will be predicted for all lots in the sequence. The state prediction accuracy for HMMScan for a particular ground truth model structure is defined as the mean of the state prediction accuracies across the samples generated by that model structure.

2.4.1 Simulated Scenario Instances

The primary accuracy assessment is performed using instances with ground truth HMMs models of one state or two states (low-risk and high-risk). The one-state ground truth HMMs have two binomial mixture components. The transition matrices associated with the two-state ground truth models are defined by three input parameters. The first parameter is the number of hidden states. The second parameter is the stationary probability of the low-risk state. Finally, the third parameter is the average number of consecutive lots in the high-risk state, often called the mean high-risk sojourn length. The different combinations of these inputs can be mapped to the following five practical motivating scenarios:

1. **No High-Risk Sojourns.** Sequences are generated by single-state HMMs, reflecting a process where per lot AE rates are not affected by manufacturing and supply chain variation.

2. **Short and Frequent High-Risk Sojourns.** The lots oscillate rapidly between the low-risk state and the high-risk state, simulating a manufacturing process that lacks proper control.

3. **Short and Infrequent High-Risk Sojourns.** The process primarily operates in the low-risk state and occasionally moves into a high-risk state for a short period of time. The low sojourn time of the high-risk state indicates that the initially unobserved, or hidden, manufacturing or supply chain issues driving the differences in AE risk are resolved promptly, but the recurrence of the high-risk state indicates that the root cause is not fully resolved.

4. **Long and Frequent High-Risk Sojourns.** The process experiences many hidden issues that take an extended period of time to detect and resolve.

5. **Long and Infrequent High-Risk Sojourns.** The process experiences few hidden issues that take an extended period of time to detect and resolve.

Within each scenario, both the length of the sample sequence and the similarity between the mixture components (one-state models) or state-dependent distributions (two-state models) are varied. The similarity between two distributions is controlled by setting the binomial parameters to induce a particular value of the overlapping coefficient (OVL) [9, 52]. The OVL, which ranges between 0 and 1, measures the probability mass that is intersected by two probability mass functions. The length of the sample sequences is varied between 50 to 500. This range covers the sequences lengths observed in the use case data described in Section 2.5 (114-460 lots). Table 2.1 lists the specific parameter values used to define the ground truth models and sequences lengths.

Parameter	Description	Parameter Values
All HMMs		
Sequence Length	Length of sample sequences	{50,100,150,...,500}
One-State, Two-Component HMMs		
Overlapping Coefficient	Overlap between binomial components of the mixture distribution	{0.05, 0.25, 0.50}
Two-State, One-Component HMMs		
Overlapping Coefficient	Overlap between state-specific binomial distributions	{0.05, 0.25, 0.50}
Low-Risk State Stationary Probability	Long-term frequency of lots in low-risk state	{0.50, 0.75, 0.90}
High-Risk State Mean Sojourn Length (lots)	Average number of consecutive high-risk lots in an infinitely long sample	{1.25, 2, 4, 10, 25}

Table 2.1: Input parameters for two-state model validation simulated instances.

2.4.2 Method Validation Results

The results of the single-state model simulations, provided in Appendix A.3.1, show HMMScan detection accuracies above 0.97 for all sequence lengths and all degrees of mixture component overlap, indicating that HMMScan is able to detect sequences generated by Scenario 1 with very high sensitivity. The two-state simulation results (Figure 2-2) indicate that the HMMScan method has high detection accuracy for sequence lengths of 100-450 lots where both the low-risk and high-risk states have approximately the same long-term frequency (Scenarios 2 and 4) and are well-separated. The highest accuracy simulations for short sequences are characterized by either medium length sojourns in the high-risk state or rapid oscillation between low-risk and high-risk states. High detection accuracy on well-separated states is important for identifying large differences between high-risk and low-risk states that are likely to correspond to high-priority investigations.

Similarly, the state prediction accuracy for the simulated instances with two-state models is also highest for long samples generated from Scenario 2 and 4 models with well-separated states. Detailed state prediction accuracy results can be found in Appendix A.3.2.

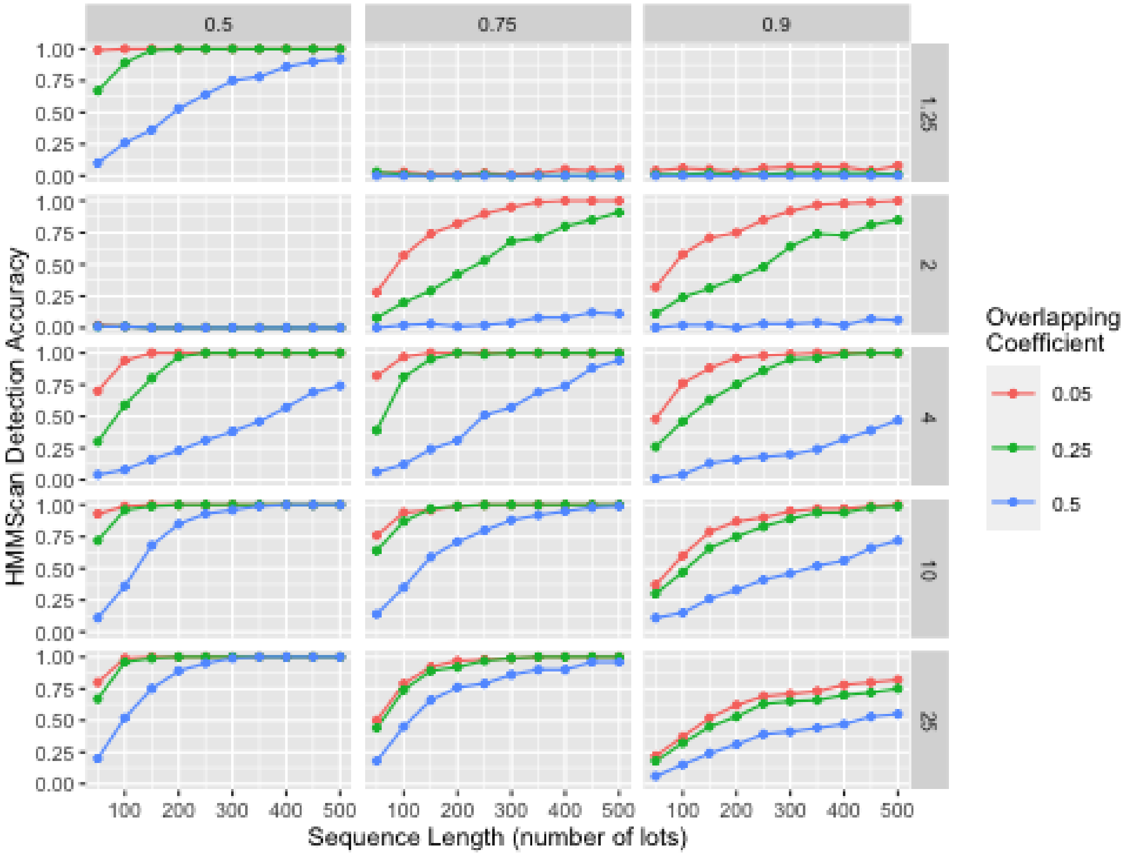


Figure 2-2: Method validation simulation results for HMMs with two states and one mixture component. Each point represents the HMMScan detection accuracy calculated on 100 sample sequences with the same length as denoted on the x-axis. The panels are organized in columns based on the low-risk state stationary probability and in rows by the mean high-risk state sojourn length. Each column value represents the expected fraction of lots from the low-risk state in a sample. Each row value represents the expected number of consecutive high-risk state lots observed each time the system moves to the high-risk state.

For the shorter sequence lengths, detection and prediction accuracy declines for very long high-risk sojourn lengths (Scenario 5) as the number of observed state transitions diminishes. Similarly, when the long-term frequencies of the hidden states are highly imbalanced (Scenario 3), transitions become more infrequent and accuracy declines. These results are unsurprising because these detecting the existence of multiple states is objectively more difficult in these scenario, particularly with shorter sequences.

Additional multiple-state simulations using instances with three- and four-state models and multiple mixture components directionally support the results described above, and the results can be found in Appendix A.3.3, Appendix A.3.4, and Appendix A.3.5. As expected, if two of the three states are very similar in a three-state generating model, the HMMScan method is frequently unable to distinguish between the similar states. Crucially, this does not impact HMMScan's ability to detect that these samples were drawn from a multiple-state model.

2.5 Use Case Data and Results

In this section, the HMMScan method is applied to real field data for three sequences of lots, each consisting of a different dose form of the same drug. The three dose forms each have different manufacturing and supply chain attributes and different mean levels of AEs. The lots for each dose form were considered as a temporal sequence based on the packaging date.

2.5.1 Data

The data regarding the packaging dates was provided by the corresponding industry partner. The industry partner also provided the number of doses for each lot. The reported AE counts per lot were obtained from the U.S. FDA Adverse Event Reporting System (FAERS) database [48], which aggregates spontaneous AE reports from manufacturers, patients, and health care providers primarily based in the United States. Each AE report consists of one or more reactions for a single patient incident. The industry partner data were matched to the AE information from FAERS using the lot numbers provided in both sources.

Table 2.2 summarizes the effects of applying a set of inclusion and exclusion criteria on the AE reports from FAERS. First, AE reports with a missing lot number in FAERS are excluded from the analysis, as are reports with an invalid lot number that does not appear in the industrial partner's records. AEs related solely to drug administration reactions (e.g., "wrong dose administered") or unrelated reactions (e.g., "dog bite") that are highly unlikely to reflect product quality issues are also excluded. A full list of excluded reactions can be found in Appendix A.4.

	Dose Form A (463 lots)	**Form B (271 lots)**	**Form C (119 lots)**	**Missing Lot Num.**	**Invalid Lot Num.**
Raw AEs from FAERS	71,890	13,582	2,789	283,888	8,653
Excluding drug administration AEs	67,402	13,184	2,562		
Relevant (known + other serious) AEs	**21,628**	**4,950**	**884**		
Expedited AEs	**7,798**	**2,051**	**437**		

Table 2.2: Count of adverse event reports by inclusion/exclusion criteria.

The primary analysis further limits the set of relevant AEs to those with at least one reaction that is either known to be associated with the drug or involves a serious reaction. These restrictions reflect a desire to minimize the number of included AEs that are not directly related to the product without omitting any very serious AEs. A list of known reactions is obtained from the drug's package label. Chest pains, pneumonia, fungal infections, malignancies, and relapse of prescribed indications are examples of known reaction categories included in the analysis. This list is augmented with the following serious reactions: loss of consciousness, arrythmia, hospitalization, and death.

A secondary robustness analysis is conducted using only AEs from expedited reports, which were deemed both serious and unexpected by the manufacturers that reported the events. Rather than minimizing the number of AEs unrelated to the product, the expedited reports capture events that are most likely to be concerning to manufacturers and regulators.

After restricting the set of eligible AEs, the raw AE counts and the number of doses per lot provided by the manufacturer are used to create per lot AE rates based on a normalized lot size of $D = 100,000$ doses. The final preprocessing step removes 17 lots (1.9%) with outlier AE rates from the dataset[1]. When the outlier lots are removed from a lot sequence, the lots on either side of the outliers are treated as consecutive, a method known as "gluing". Prior research indicates that applying the gluing procedure with less than 8% of lots designated as missing does not affect the likelihood or magnitude of HMM parameter estimates [32]. Table 2.3 shows the distribution of the AE rates per lot for each modeled dose form.

	All Lots			**Outliers Removed**		
	Dose Form A	**Dose Form B**	**Dose Form C**	**Dose Form A**	**Dose Form B**	**Dose Form C**
Min	0	0	0	0	0	0
25th	27	9	3	27	9	2
50th	41	18	11	41	18	10
75th	63	29	19	62	28	18
95th	85	44	47	84	39	23
Max	151	280	248	113	51	30
Mean	45	22	21	44	18	11
Lot Count	*463*	*271*	*119*	*459*	*264*	*113*

Table 2.3: Adverse event rates per lot. 25th, 50th, 75th, and 90th refer to percentiles.

In addition to the use case discussed in this section, the HMMScan method was applied to several vaccine products using only publicly available AE information from the U.S. Vaccine Adverse Events Reporting System (VAERS) database [47], a similar dataset to the FAERS database. Since lot sizes and packaging dates were not available for these products, the vaccine analysis required additional assumptions before applying the HMMScan method. These assumptions and the results of the vaccine HMMScan analysis are provided in Appendix A.6.

[1] An outlier is defined as an AE rate less than the 25th percentile minus 1.5 times the interquartile range (IQR) or greater than the 75th percentile plus 1.5 times the IQR [50].

2.5.2 Detection Results

For each dose form, the grid of candidate model structures is constructed by setting $S_{max} = 4$ and $C_{max} = 9$. The BIC values decline monotonically outside the chosen hyperparameter ranges, indicating that the complexity penalty is outweighing the likelihood gains and the models are overfitting the data. Each of the candidate models is fit with 50 random initializations and the results corresponding to the parameter estimates with the highest likelihood are retained. The BIC values for the fitted candidate models are shown in Figure 2-3.

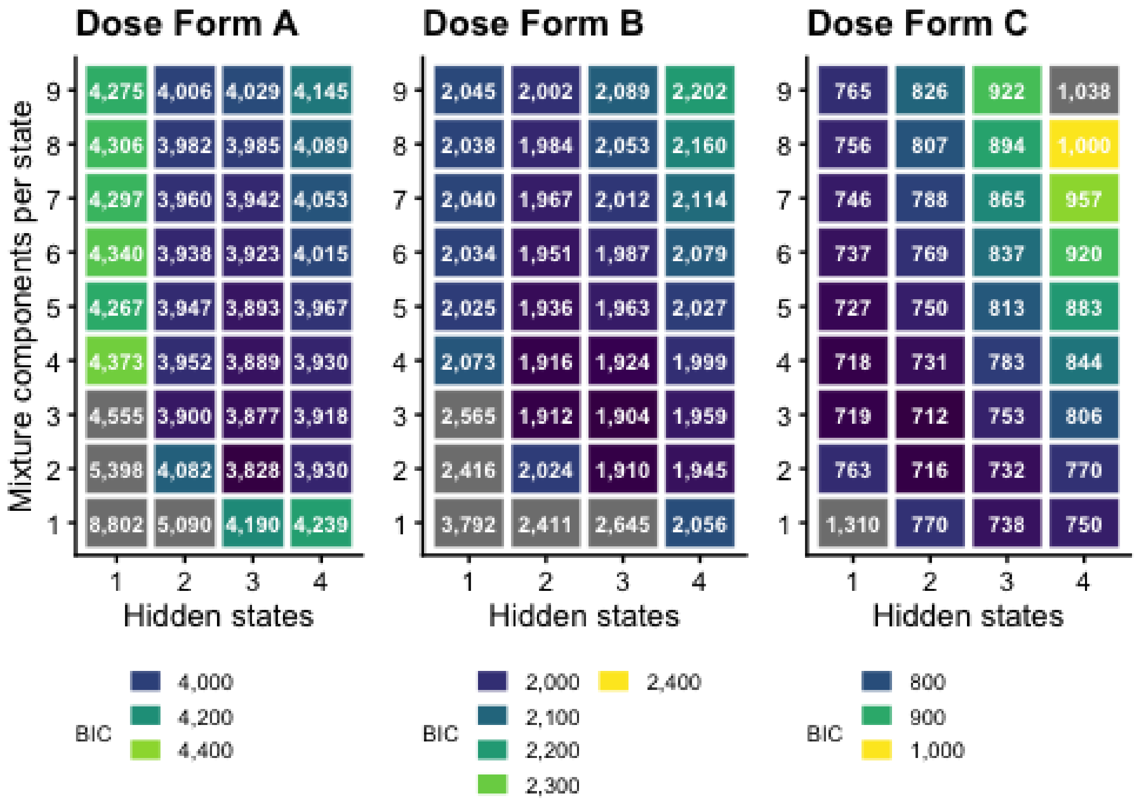


Figure 2-3: BIC values for the candidate HMMs for each dose form. Each tile indicates the BIC value for a fitted HMM with the number of states denoted on the x-axis and the number of binomial components per state-specific mixture distribution on the y-axis. Lower BIC values indicate a better fit of the model to the data.

Dose Forms A and B. Multiple state HMMs have the best fit as measured by BIC for dose forms A and B ($S = 3$, $C = 2$ for dose form A and $S = 3$,$C = 3$ for dose form B). The BIC difference between the best-fitting multiple state model and the best-fitting single state model is larger than 10 for both dose forms, suggesting significantly stronger fit for a multiple state model and related serial correlation in the per lot AE rates.

Dose Form C. A multiple-state HMM with $S = 2$ and $C = 3$ provides the lowest BIC for dose form C, but the BIC difference between this model structure and a single state model with $C = 3$ is lower than 10, indicating weaker evidence of serial correlation in the per lot AE rates.

2.5.3 Identifying States with High AE Risk

Figure 2-4 and Table 2.4 illustrate the maximum likelihood estimated parameters for the HMM with the lowest BIC value for dose forms A and B. This includes the state transition matrix, the stationary distribution of the time spend in each state, and the state-dependent mixture distribution. Due to the relatively weak evidence in favor of a multiple-state state model for dose form C, the maximum likelihood parameters are included in Appendix A.5. In Figure 2-4, a clear separation exists for dose form A between state 3, where the lots tend to be associated with a high number of AEs, and state 1 where AE rates are lower on average. State 2 represents a medium-risk state. Similarly, there is clear separation between the high-risk state 3 and the low-risk state 1 for dose form B.

	Dose Form A					**Dose Form B**				
	Transition Probs. (from row state to column state)					**Transition Probs. (from row state to column state)**				
Hidden State	To: State 1	To: State 2	To: State 3	Mean AE Rate (90% CI)	Stat. Prob.	To: State 1	To: State 2	To: State 3	Mean AE Rate (90% CI)	Stat. Prob.
1	0.76	0.19	0.05	9.8 (9.4-12.8)	0.14	0.92	0.08	0.00	6.9 (0.0-8.3)	0.25
2	0.06	0.90	0.04	32.9 (29.4-32.6)	0.43	0.06	0.85	0.09	14.5 (9.1-17.8)	0.30
3	0.02	0.03	0.95	66.4 (60.2-64.5)	0.43	0.00	0.06	0.94	26.4 (23.0-30.3)	0.45

Table 2.4: Estimated transition matrix and state-specific mean AE rates for best-fitting HMMs, with mean AE rate 90% CIs (confidence intervals) estimated via parametric bootstrap [35, 49]

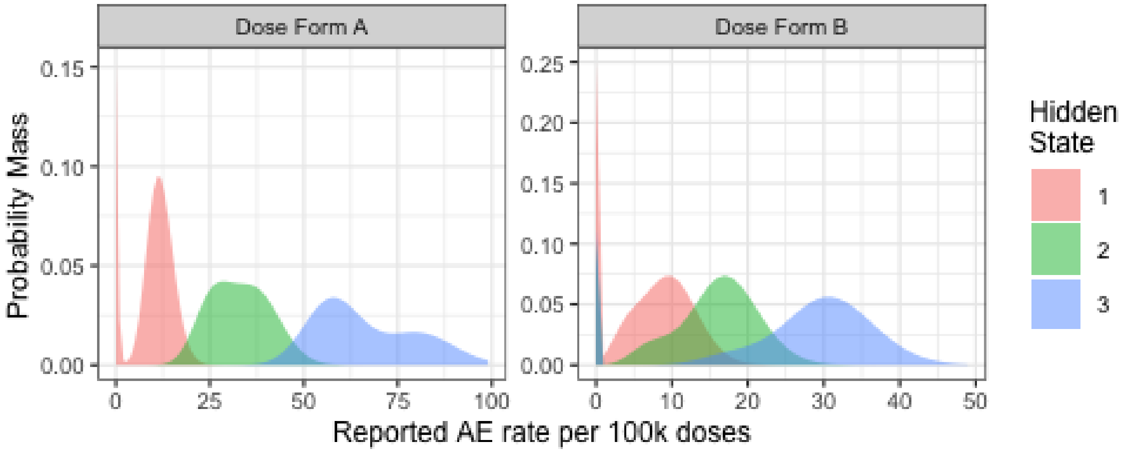


Figure 2-4: Fitted state-specific binomial mixture distributions for the best-fitting HMMs for dose forms A and B. Each panel shows the distribution for the state-specific distribution associated with each hidden state.

2.5.4 Interpreting the Use Case Results

The estimated state-specific mean AE rates in Table 2.4 demonstrate that the ordering of the states by AE risk is robust. The estimated transition matrices both have high probabilities on the diagonal, indicating that the hidden states are all very persistent. This suggests that high-risk and low-risk AE states tend to form long contiguous regions.

In fact, these regions are observable in Figure 2-5 for both dose form A and dose form B. This figure orders the lots by packaging date for both dose forms and colors the AE rate for each lot by its most likely hidden state. Both dose forms have two clearly identifiable regions of high-risk lots as well as multiple low-risk regions at the beginning and end of the sequences (Figure 2-5a and Figure 2-5b). Furthermore, when the HMMScan method is performed using AE rates based solely on expedited reports, the best-fitting HMMs indicate nearly identical high-risk regions (Figure 2-5d and Figure 2-5e). These regions are visually reasonable despite the presence of occasional lots with low AE rates in the high-risk regions.

Similar persistent high-risk regions are visible for dose form C in Figure 2-5c. However, the results on the expedited AE reports indicate that a single-state model has the lowest BIC, further suggesting only weak evidence in favor of multiple states in the ground truth model for this lot sequence.

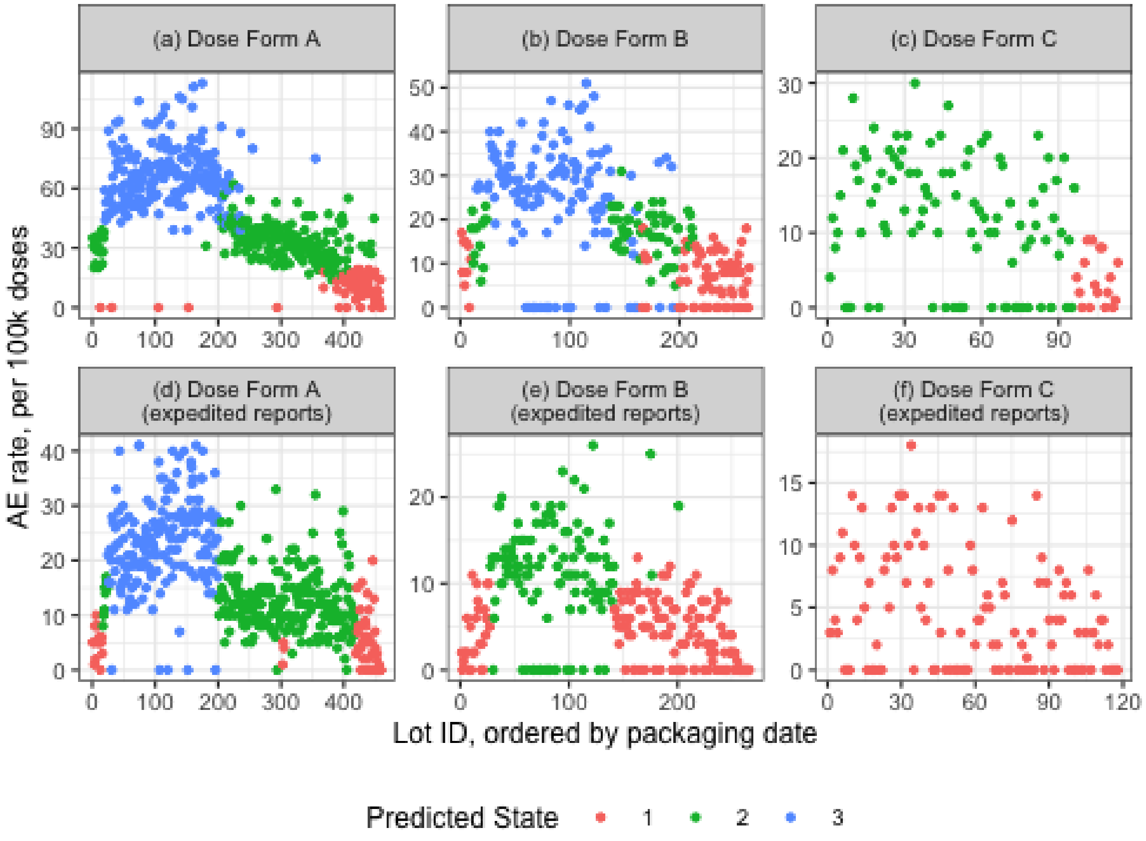


Figure 2-5: Per lot AE rates. The top row of plots calculates per lot AE rates based on the known and serious definition, while the bottom row includes only expedited AE reports. The lots are shaded by most likely hidden state according to the HMM with the lowest BIC.

2.5.5 Use Case Method Validation

This section applies a similar approach to the one described in Section 4 to gauge the likelihood that HMMScan method has accurately identified the correct model structures for the use case datasets. Specifically, the goal is to estimate the probability than an HMM with structure (S_{BIC}, C_{BIC}), i.e., S_{BIC} states and C_{BIC} mixture components, would have been chosen if the observed lot sequence were generated by an HMM with a different structure.

To obtain this estimate, 100 sample sequences with the same length as the observed sequence are generated from each fitted candidate HMM not selected by the BIC. Consider a sample sequence generated by a specific candidate HMM with structure $(S_{sampling}, C_{sampling})$. A sample sequence is considered misidentified if an HMM with structure (S_{BIC}, C_{BIC}) has the lowest BIC of all candidate models fit to that sequence. The fraction of misidentified sample sequences gives an estimate of the probability of misidentifying a sequence generated by a $(S_{sampling}, C_{sampling})$ HMM as a sequence from a (S_{BIC}, C_{BIC}) HMM.

Figure 2-6 shows the estimated misidentification probability for each candidate model structure for each of the three lot sequences in the use case. First, note that across all three lot sequences, the sample sequences generated by less complex HMMs than the BIC-selected HMM (i.e., $S_{sampling} \leq S_{BIC}$ and $C_{sampling} \leq C_{BIC}$) are rarely misidentified. Furthermore, only a small fraction of sample sequences generated by single-state HMMs are identified as having S_{BIC} states and C_{BIC} mixture components by HMMScan. This result implies that the observed HMMScan signals for these dose forms are reliable.

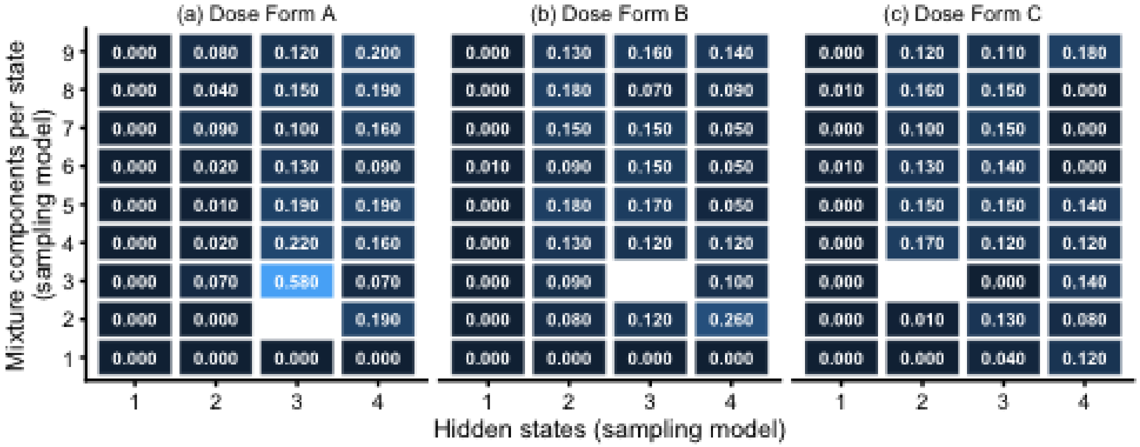


Figure 2-6: Estimated misidentification probabilities for the use case method validation. Each tile indicates the misidentification probability for a given sampling model with respect to S_{BIC} and C_{BIC}. For dose form A, $S_{BIC} = 3$ and $C_{BIC} = 2$, for dose form B, $S_{BIC} = 3$ and $C_{BIC} = 3$, and for dose form C, $S_{BIC} = 2$ and $C_{BIC} = 3$.

2.6 Discussion and Conclusions

This paper presents HMMScan, a novel pharmacovigilance method for detecting patterns in AE rates across manufacturing lots using probabilistic modeling techniques. HMMScan is a method that could be utilized by both manufacturers and regulators to automate lot variability monitoring and inform targeted root cause analysis. Specifically, HMMScan provides: (1) a reliable signal when serial correlation is detected in an observed AE rate sequence, and (2) a model to identify individual lot subsequences where variation in manufacturing and supply conditions may have contributed to higher AE rates. In a case study of three lot sequences corresponding to three dose forms of a major biologic, the strong evidence of serial correlation was detected for two of three dose forms.

The HMMScan method is proposed as an initial signal detection tool to identify lot sequences where serial correlation in AE rates suggests the potential presence of clinically relevant variation in manufacturing and supply chain conditions. Root cause analysis utilizing additional, and likely proprietary, features of the manufacturing lots would be essential to confirm a causal relationship between manufacturing and supply chain conditions and AE rate variation. An investigator could start by examining the lots in the vicinity of hidden state transitions, since these are periods during which the HMMScan method suggests clinically meaningful changes to manufacturing and supply chain conditions might have occurred. Only via this detailed root cause analysis can an investigator rule out other factors unrelated to manufacturing and supply chain conditions and determine if a truly causal relationship between such conditions and safety outcomes actually exists.

An implementation of the HMMScan method in R and Python is available as a GitHub repository [53]. This repository includes a tutorial for generating results for a new use case, as well as instructions for reproducing the use case and simulation results presented in this paper. The relevant data are also stored in a public repository [54].

A possible direction for future methodological research is to increase the complexity of the candidate model structures that HMMScan considers by allowing the hidden state of lot ℓ to depend on a prior history of states before lot $\ell-1$. Limited dependence on only the most recent hidden state is useful because it yields the fast and well-understood Baum-Welch algorithm for maximum likelihood parameter estimation. However, EM-based parameter estimation algorithms for variable length HMMs, which allow state dependence on history prior to the most recent state, have been proposed [11]. More recently, a Bayesian model for variable length Markov chains was introduced [20], though this model has not been studied in a hidden Markov setting.

In principle, a primary benefit of HMMScan is the potential to apply the method broadly across a range of pharmaceutical products. Such broad application of HMMScan would rely on a well-developed data input pipeline to gather the following information for each lot: packaging date, relevant AE counts, number of doses, and dose form. This is the minimum required data input for the method as currently constructed, though in principle the model could take additional information about the distribution patterns by lot, including more granular regional distribution information and patient characteristics. Additional information about the lot-to-lot differences in patient populations could be used to adjust the AE counts to account for these differences. In this case, a positive signal of serial correlation in AE rates would be even more likely to correspond to variation in manufacturing and supply chain conditions. Further collaboration between regulators, manufacturers, and academics to collect and format these data is the first step toward realizing this opportunity to augment drug safety monitoring to improve patient outcomes.

Chapter 3

Unsupervised Text Classification with Reference Information

3.1 Introduction

Quality control and quality assurance are critical in manufacturing and service operations. With increasing digitization and automation, these processes generate more data than ever, which creates opportunities to enhance the process quality. That said, much of the relevant data are still generated by human operators and customers and are often kept as unstructured text. For example, on manufacturing lines operator reports and logs often include exclusive and important information that are not captured elsewhere. These documents include updates regarding line operations, as well as records of quality deviations, their investigations, and resolution. More broadly, creating structured access to a large corpus of documents is also important for enhancing the scope and quality of operations in a variety of service-related settings. A frequent barrier to tapping the potential of these unstructured data is the lack of a labeled training set, that is, a set of documents that have already been assigned to a topic or category within a structured taxonomy.

In many use practical use cases, the taxonomy of categories naturally arises from the process context and is known to the operator. For example, in the manufacturing use case, categories could correspond to different process steps or pieces of equipment. Moreover, in these use cases, written documents that describe each category and provide side information are often easily obtained. Yet there is often no seamless way to classify unstructured documents into the taxonomy without extensive manual review and annotation. This chapter describes an innovative machine learning and optimization-driven methodology to classify unstructured data in process environments without access to an existing labeled training set.

The newly proposed approach developed in this chapter is illustrated through a primary use case in a manufacturing setting, and it is further validated on two datasets from news websites with known classification labels, which shows the potential for the method to be used in other document classification applications. The primary use case concerns categorizing documents related to quality deviation reports on an existing pharmaceutical manufacturing line for biologic drugs. The deviation reports and associated investigation notes, provided by an industrial partner, are recorded as free text without much detailed structured metadata to identify the specific process step where the deviation occurs.

Current state-of-the-art approaches implemented by the industry partner leverage unsupervised clustering methods to group similar deviations together for the purposes of trend analysis and automated searches for similar deviations. However, these approaches do not enable the user to pre-specify the content of each of the clusters and ensure that the deviations associated with specific process steps are indeed grouped together. Moreover, the ability to classify any new deviations into a pre-specified taxonomy of categories corresponding to process steps is important to enable more actionable and targeted deviation tracking methods. This use case has natural analogs in other industrial manufacturing processes, such as automotive and aircraft manufacturing or oil and gas refining, where extensive text logs and detailed written process documentation are common and sometimes even required by regulatory authorities.

In addition to the primary manufacturing use case, the newly developed approach is validated on two news-related datasets commonly used to test automated document classification algorithms. The validation datasets include a subset of the 20 Newsgroups dataset [36], a collection of online comments on news topics, and the

BBC News dataset [14], which includes online news stories related to five specified topics. The documents in these datasets have assigned topic (category) labels, which can be used to evaluate the predictive accuracy of the described approach relative to other benchmark methods. Furthermore, similar to process steps in a manufacturing environment, the topics in these datasets (e.g., sports, technology, automobiles) can be described objectively using easily obtained outside factual sources like encyclopedias. These datasets demonstrate the potential utility of the proposed approach in service applications, such as classification of customer feedback reports, automated organization of legal documents, or medical diagnosis prediction from physician notes.

This chapter proposes a method called *Document Classification with Reference Information (DCRI)* to classify unlabeled input documents into a prespecified taxonomy of categories without the use of a labeled training set. Unlike standard existing unsupervised methods like Latent Dirichlet Allocation (LDA) [5], the DCRI method predicts document labels within the context of a prespecified taxonomy of categories. Instead of relying on manually labeled training documents within a supervised algorithmic approach to make these predictions, the DCRI method uses existing reference information in the form of *category descriptions* to distinguish between the categories. These category descriptions are assumed to be rich enough to describe each category of the taxonomy. As already discussed, in many practical settings such documentation is often readily accessible. For example, in the pharmaceutical manufacturing use case, all process steps are described in detailed written standard operating procedures (SOPs). For the news datasets, reference information about common news topics is accessible via encyclopedia sources such as Wikipedia.

Crucially, the DCRI method does not require that the category descriptions are written by the same author or for the same purpose as the unlabeled input documents. The DCRI method uses an optimization model to select key words that distinguish the categories from each other in both the input documents and the category descriptions, and the ultimate category predictions place higher weight on these words. This explicit adjustment for the differences between the input documents and category descriptions differentiates the DCRI method from existing semi-supervised methods such as semi-supervised Multinomial Naive Bayes [30].

3.1.1 Contributions and Key Results

This chapter describes a novel approach that enables document classification into a prespecified taxonomy without a labeled training set. Instead, the approach leverages available reference information in the form of *category descriptions*. A naïve approach could use all the significant words appearing in the input documents and reference category descriptions to make document classification predictions. However, to mitigate the impact of word usage differences between the category descriptions and the documents, the DCRI method utilizes a new linear programming-based optimization formulation that ensures the predictions are robust to these differences. More specifically, the linear program guides the selection of a sparser subset of important words.

The DCRI method is tested on three datasets, including a dataset from the primary manufacturing use case with manually-assigned ground truth category labels and two validation news datasets described above. The method delivers consistently high classification accuracy (84 - 89%) across all three datasets. For the manufacturing and Newsgroups datasets, the DCRI method's accuracy approaches (within 1.5%) the accuracy of a supervised classifier trained on the ground truth labels.

Additionally, the DCRI method is compared to several benchmark approaches. The DCRI method is significantly more accurate (4-20% improvement) compared to alternative, LDA-based methods for incorporating category descriptions into document classification predictions. The DCRI method outperforms semi-supervised Multinomial Naive Bayes by 1.4-5.7% on the two of three datasets with the most informative category descriptions. A critical driver of the DCRI method's performance improvement is optimized key word selection, which directly improves predictive accuracy across all three datasets by up to 2.5% compared to the naïve approach. Furthermore, the results show that to achieve classification performance equivalent to the DCRI method requires a substantial labeling effort, 15-30% of the dataset, and the number of required labels is even higher to achieve statistically equivalent accuracy on unseen documents.

The rest of the chapter is organized as follows. Section 3.2 details the experimental datasets, Section 3.3 describes the DCRI method, Section 3.4 provides the experimental results, and Section 3.5 concludes.

3.2 Data Description

This section introduces the three datasets considered in this chapter to motivate the development of the DCRI method. Section 3.2.1 describes the primary manufacturing use case dataset and Section 3.2.2 details the news-related datasets.

3.2.1 Pharmaceutical Deviation Dataset

The input documents for the primary use case dataset, referred to as the Pharma dataset in this chapter, consist of a corpus of unlabeled free text descriptions of deviations from standard operating procedures that occurred at a pharmaceutical manufacturing plant for biologic drugs between 2011 and 2016. While this dataset was compiled in collaboration with a specific industrial partner, such deviation data are common across the pharmaceutical industry because U.S. Food and Drug Association regulations require that all product deviations from established specifications and standards are recorded [46]. The deviation descriptions contain short and long descriptions of the deviation event, as well as any root cause analysis and recommended follow-up actions from subsequent investigation. A description of the deviation description components can be found in Appendix B.1.

The Pharma dataset consists of 563 deviations, representing a small subset of over 10,000 major and minor deviations that the industrial partner might wish to categorize. The deviations in the Pharma dataset all relate to the production bioreactor stage at a single manufacturing facility. These deviations were identified using existing structured metadata, which are detailed enough to determine the high-level manufacturing stage of the deviation, such as the production bioreactor, but not detailed enough to establish the precise process step. Only the production bioreactor deviations were included in the Pharma dataset in order to make manual labeling of each input document feasible. The availability of ground truth labels allows for evaluation of the predictive performance of the DCRI method and comparison to other benchmark methods.

The category taxonomy of the production bioreactor process steps is described below. A production bioreactor is a vessel where specially cultivated living cells, called inoculum, are mixed with a growth medium for the purpose of producing a particular pharmaceutical ingredient. A schema of five categories of process steps was developed in collaboration with process experts at the industrial partner. The categories include *bioreactor additions*, *process monitoring*, *sample processing*, *maintenance*, and *filter integrity testing.* This schema was designed to be specific enough to enable meaningful analysis without introducing categories likely to be sparsely populated by deviations. The manually assigned labels indicate significant class imbalance in the document dataset. In descending order, the distribution of document labels across the five categories is 0.50, 0.19, 0.18, 0.9, and 0.04. The category descriptions were compiled from relevant excerpts from over 30 standard operating procedures that describe the production bioreactor process. A further description of the categories, including examples of typical deviations, can be found in Appendix B.2.

3.2.2 News-Related Datasets

In addition to the Pharma dataset, the DCRI algorithm was evaluated on two news-related datasets, 20 Newgroups and BBC News. These datasets are included to assess how the performance of the algorithm generalizes to other service-related applications, such as classification of customer feedback reports. Both datasets are are widely used in the natural language processing community for document classification research because true topic labels have been manually assigned to the documents.

The Newsgroups (NG) dataset is a collection of labeled documents, which consist of free text online comment posts that have each been classified by topic category [36]. The dataset consists of 3,314 documents that span four categories: *religion* (alt.atheism), *computer graphics* (comp.graphics), *cars* (rec.autos), and *space* (sci.space). The BBC dataset consists of 1,490 labeled documents from the BBC News website corresponding to news stories in five topical categories: *Business*, *Entertainment*, *Politics*, *Sports*, and *Technology* [14]. For both datasets, Wikipedia articles associated with the headline topics for each category are used as category descriptions. The Wikipedia entries for the BBC categories correspond exactly to the titles of the categories, while the Wikipedia entries for the NG categories are *Atheism* and *Religion*, *Computer Graphics* and *Pixel*, *Automobile* and *Cars*, and *Astronomy* and *Spacecraft*, respectively. Both the NG and BBC documents are approximately evenly balanced across classes.

Table 3.1 below shows the summary statistics for the three experimental datasets, where D denotes the number of documents, C is the number of categories, and V the number of unique terms (single words and two-word phrases) that appear in at least one document and at least one category description. The datasets vary in terms of the number of documents and size of the vocabulary, but in all cases the average category description is long, more than 1,800 words after removing stop words. In contrast, the average unlabeled document is shorter than the average category description by at least one order of magnitude. This discrepancy is consistent with the assumption that the category descriptions are rich and detailed enough to provide meaningful term frequency estimates in lieu of labeled input documents.

	D	C	V	Mean Document Length (words)	Mean Category Desc. Length (words)
Pharma	563	5	1924	129	3585
NG	3314	4	1684	46	3660
BBC	1490	5	1485	94	1806

Table 3.1: Dataset summary statistics. D denotes the number of documents, C is the number of categories, and V the number of terms (single words and two-word phrases) that appear in at least one document and at least one category description.

3.3 DCRI Method Description

This thesis chapter studies the general problem of classifying an input set of D free text unlabeled documents into a pre-specified set of C categories. Each category is associated with a separate *category description* containing reference information that details the content and themes related to the category.

This section describes the Document Classification with Reference Information (DCRI) method, beginning with a high-level description of the proposed method and providing more details in Sections 3.3.1 - 3.3.3. Figure 3-1 below illustrates the three primary algorithmic steps of the DCRI method, *Vectorization*, *Preliminary Labeling*, and *Label Augmentation*. These steps ultimately produce a predicted category label for each of the D input documents. The Vectorization step converts the input raw text into numerical features that represent the frequency count of each term in each document. Additionally, the terms in the category descriptions are assigned *importance scores* that indicate how unique each term is to each category. In the Preliminary Labeling step, an initial category prediction is produced, for each document, by calculating document-category scores from a linear combination of the document term frequencies weighted by the category importance scores. The Label Augmentation step uses optimization and machine learning to intelligently revise the preliminary labels and output a final predicted category label for each document. Importantly, this step also provides weights for the terms that can be used to classify future documents not included in the initial set of input documents.

3.3.1 Vectorization

The Vectorization procedure (Figure 3-2) takes as an input the raw text of the D documents and C category descriptions. The procedure outputs numerical feature matrices that encode these text inputs using a bag of words (BoW) representation [26]. The BoW model, which has been used for decades in the natural language processing community [42], represents a corpus of documents using a feature for each term in a prespecified vocabulary. Each feature captures the frequency of occurrence of the

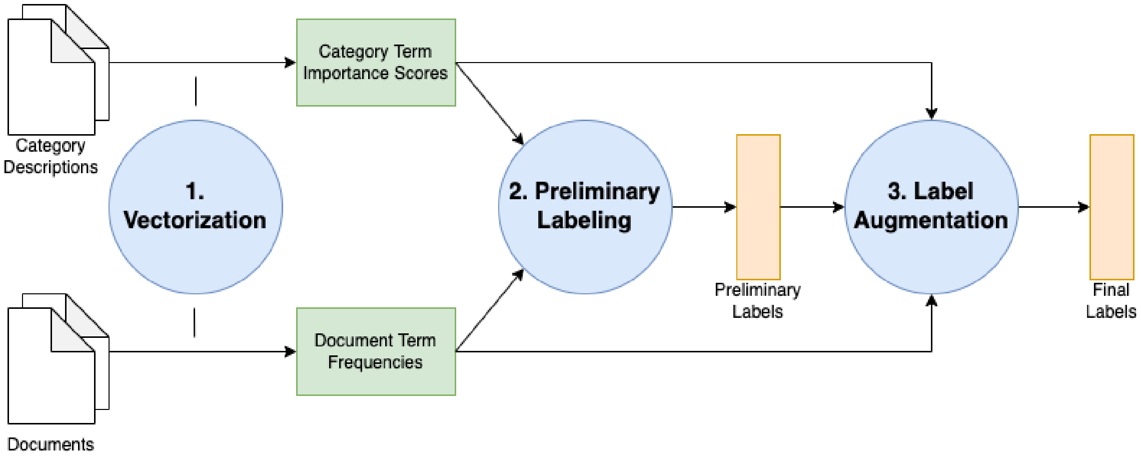


Figure 3-1: Document Classification with Reference Information (DCRI). Blue circles indicate the primary steps of the document classification procedure. The green rectangles indicate the vectorized inputs to the label prediction task used to generate predicted category labels. Preliminary and final predicted label outputs are denoted by orange rectangles.

respective term in each document in the corpus.

After removing English stop words, dates, times, and purely numerical terms, the vocabulary is formed by including all single words (unigrams) that appear in at least one category description and at least one input document, as well as the top 30% of two-word phrases (bigrams) with the highest category importance scores. The vocabulary dimension is denoted by V. The input document feature matrix, denoted $\mathbf{F} \in \mathbb{Z}^{D\times V}$, consists of the respective frequencies of the terms in the vocabulary, and each entry f_{dv} corresponds to the count of term v in document d. The term frequencies are adjusted to account for the presence of both unigrams and bigrams in the vocabulary by decrementing the frequency of each unigram by the number of appearances in bigrams included in the vocabulary. The details of this term frequency adjustment can be found in Appendix B.3.

The category description vectorization forms a similar matrix $\mathbf{S} \in \mathbb{R}^{C\times V}$ where each entry represents the importance score of each term to each category. The elements of $\mathbf{S}$ are term frequency inverse document frequency (TF-IDF) scores [38], a transformation of raw term frequencies common in information retrieval and natural language processing. The TF-IDF score s_{cv} is high when the term v represents both

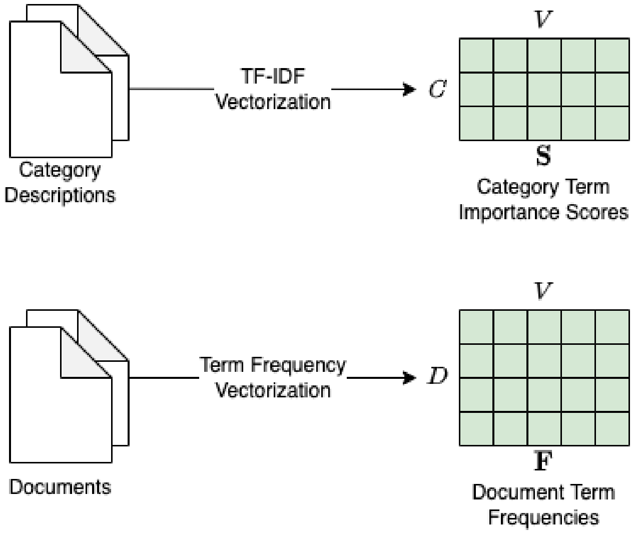


Figure 3-2: Vectorization step of the DCRI method. During Vectorization, the raw text of the documents and category descriptions are transformed into feature matrices **F** and **S**, respectively. **F** represents the raw count of each term in each document, while **S** represents the TF-IDF importance score of each term to each category.

a high fraction of the terms used in the category c description and is also used in few other category descriptions. The specific TF-IDF version used in this chapter, which is based on a version with demonstrated favorable document classification performance [37] and incorporates additional normalization to account for the presence of bigrams, is detailed in Appendix B.4. A high TF-IDF score indicates that a term is specific to a particular category and therefore is an important feature for document classification.

3.3.2 Preliminary Labeling

In the Preliminary Labeling step (Figure 3-3), the input consists of the category importance scores and document term frequencies, which are used to calculate a preliminary category label for each input document. First, preliminary document-category scores, denoted $\mathbf{P}^{AT} \in \mathbb{R}^{D\times C}$, are obtained using all the terms. Specifically, the preliminary score for each document d and category c is obtained via the dot product between the document's term frequency vector and the log of the category c

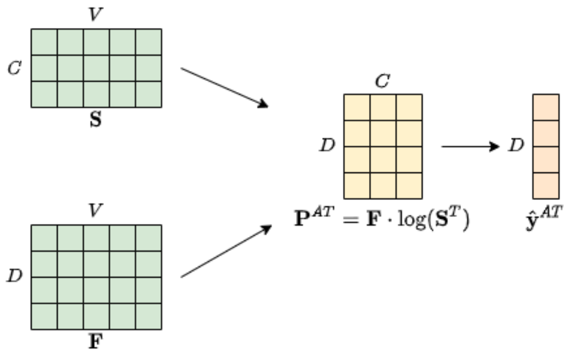


Figure 3-3: Preliminary Labeling step of DCRI method. During the Preliminary Labeling step, the document term frequencies ($\mathbf{F}$) and the category importance scores ($\mathbf{S}$) are combined to form preliminary document-category scores ($\mathbf{P}^{AT}$), with rows $\mathbf{p}_d^{AT}$, and labels ($\hat{y}^{AT}$), where $\hat{y}_d^{AT} = \text{argmax}_{c \in \{1,2,\ldots,C\}} \mathbf{p}_d^{AT}$.

term importance score vector:

$$\mathbf{P}^{AT} = \mathbf{F} \cdot \log(\mathbf{S}^T) \tag{3.1}$$

For a given document, the preliminary label corresponds to the category with the highest preliminary document-category score. Specifically, the output of the Preliminary Labeling step is a category label vector $\hat{y}^{AT}$, where each element $\hat{y}_d^{AT}$ represents the category with the maximum score in the row vector of document-category scores $\mathbf{p}_d^{AT} \in \mathbb{R}^C$:

$$\hat{y}_d^{AT} = \text{argmax}_{c \in \{1,2,\ldots,C\}} \mathbf{p}_d^{AT} \tag{3.2}$$

3.3.3 Label Augmentation

The linear document-category scoring function in equation (3.1) above is related to the well-known Multinomial Naive Bayes (MNB) classification algorithm [23]. In the MNB setting, the category term importance score parameters $\mathbf{S}$ are learned from a training set of labeled documents. Without access to such a labeled training set, the preliminary labels are obtained by effectively assuming that each category description is the only labeled training document associated with its respective category. This is a strong assumption since the category descriptions are assumed to be written as reference material, so their purpose, authorship, and writing style likely differ substantially from the documents.

This section describes the Label Augmentation step, which revises the preliminary labels to minimize the number of classification errors that might be induced by statistical differences between the category descriptions and documents. This step takes as input the preliminary labels $\hat{y}^{AT}$, the category-term importance scores $\mathbf{S}$, and the document-term frequencies $\mathbf{F}$, and outputs a new set of improved category labels. The step consists of two sequential procedures, *Key Term Labeling* and *Supervised Labeling.*

The Key Term Labeling procedure calculates document-specific category scores that rely on a sparse set of key terms rather than the full vocabulary. The set of key terms is selected to induce category predictions that are more robust to differences in word usage between the category descriptions and document corpus. These differences arise from the fact that, in most cases, the category descriptions and documents have been written for entirely different purposes, though both refer to the same underlying activities. The connection between sparsity (and other regularization methods) in linear models and robustness to data noise has been well-documented [56, 55], though this existing literature primarily focuses on noise generated by measurement error or mislabeling.

To this end, the Key Term Labeling procedure leverages a suitably designed optimization model. The primary decision variables are denoted by a vector $\mathbf{x} \in [0,1]^V$, where x_v corresponds to a weight assigned to each term in the vocabulary. Auxiliary decision variables are denoted by the matrix $\mathbf{P} \in \mathbb{R}^{D \times C}$ represent weighted document-category scores with the weights $\mathbf{x}$ applied to the terms. Each score in $\mathbf{P}$ is denoted p_{dc} for document d and category c, and in particular $p_{d,\hat{y}_d^{AT}}$ represents the score for document d corresponding to the preliminary category label $\hat{y}_d^{AT}$. Additionally, each element h_d of the auxiliary variable vector $\mathbf{h} \in \mathbb{R}^D$ represents the highest weighted document-category score for document d corresponding to any category other than $\hat{y}_d^{AT}$. Finally, the input data in the optimization problem consist of document-term frequencies $\mathbf{F} \in \mathbb{Z}^{D \times V}$ and category-term importance scores $\mathbf{S} \in \mathbb{R}^{C \times V}$.

Using these decision variables and data, the following linear program is formulated:

$$\begin{aligned}
\max_{\mathbf{x},\mathbf{h},\mathbf{P}} \quad & \sum_{d=1}^{D} \left(p_{d,\hat{y}_d^{AT}} - h_d \right) \\
\text{s.t.} \quad & p_{dc} = \sum_{v=1}^{V} f_{dv} \cdot x_v \cdot \log(s_{cv}), && \forall d \in [D], c \in [C] \\
& h_d \geq p_{dc}, && \forall d \in [D], \forall c \neq \hat{y}_d^{AT} \\
& \mathbf{x} \in [0,1]^V, \mathbf{h} \in \mathbb{R}^D, \mathbf{P} \in \mathbb{R}^{D \times C}
\end{aligned} \tag{3.3}$$

The constraints in problem (3.3) enforce the interpretation of the auxiliary variables described above. The objective, called the *maximum separation objective*, is designed to place high weights on terms that increase the confidence of the preliminary category predictions, i.e., the score of the preliminary predicted category is much higher than the scores of the other categories.

The maximum separation objective incentivizes high weights on terms with both of the following properties. First, the maximum category-term importance score should be significantly higher than the remaining category-term importance scores. This property measures a term's *specificity* to a single category based on the category descriptions. Second, a high-weight term should appear primarily in documents where the preliminary label matches the category associated with the maximum category-term importance score. This second property measures the *consistency* of a term's usage when comparing the document corpus to the category descriptions.

Terms that are both specific and consistent tend to increase $p_{d,\hat{y}_d^{AT}}$ relative to the scores for other categories, thereby increasing the positive separation between $p_{d,\hat{y}_d^{AT}}$ and h_d. However, terms that are either specific or consistent, but do not have both properties, are less likely to receive high weights. Terms with low specificity are not able to meaningfully add to the difference between $p_{d,\hat{y}_d^{AT}}$ and h_d for any document, and terms with low consistency may meaningfully decrease the separation objective for certain documents.

A critical aspect of the formulation of problem (3.3) is that $p_{d,\hat{y}_d^{AT}}$ is not required to be the highest score for document d. In order to increase the separation objective for the majority of documents, the optimal term weights may induce category predictions for the remaining documents that do not agree with the preliminary labels. For these documents, the difference $p_{d,\hat{y}_d^{AT}} - h_d$ would be negative, suggesting that the preliminary label for document d should be revised to the category associated with the score h_d. The premise of the Key Term Labeling procedure is that the revised label predictions, based primarily on specific and consistent terms, are more likely to reflect the ground truth than the preliminary label predictions.

If the optimal term weights obtained by solving problem (3.3), denoted $\tilde{\mathbf{x}}$, are integral, then this solution vector has a clear interpretation. Any term v where $\tilde{x}_v = 1$ is chosen for inclusion in the revised document-category predictions, and the rest of the terms are excluded. Since integrality is not guaranteed, $\tilde{\mathbf{x}}$ is converted to an integral solution $\mathbf{x}^*$ by setting all non-integral values equal to 0.

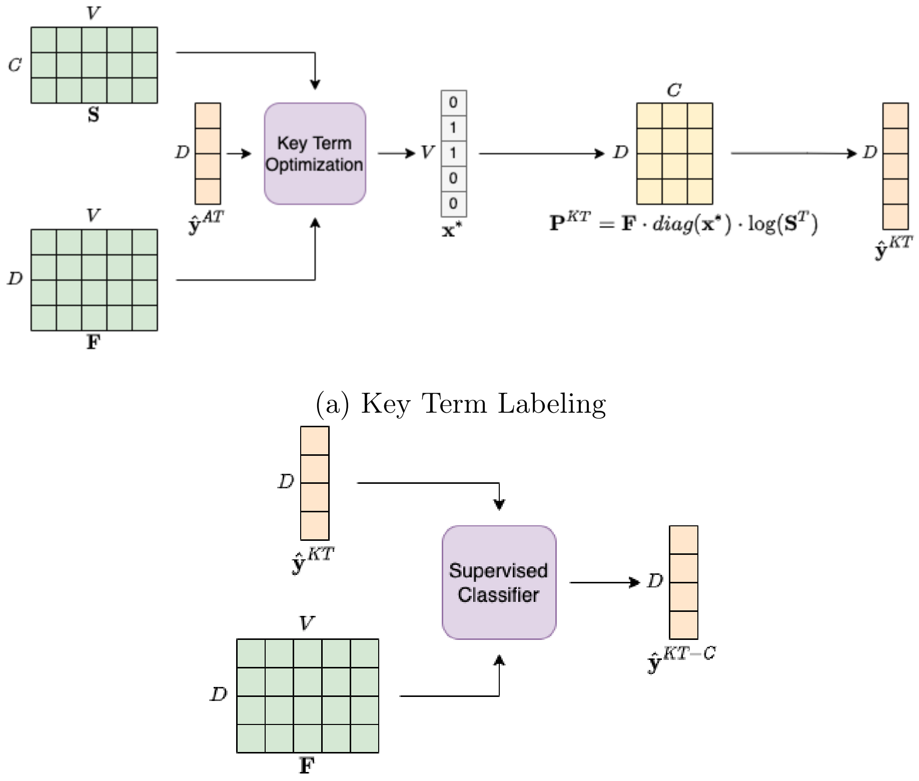
The choice of $\mathbf{x}^*$ enables the calculation of key term category scores $\mathbf{P}^{KT} \in \mathbb{R}^{D \times C}$ and the vector of revised label predictions $\hat{y}^{KT}$ (Figure 3-4a):

$$\mathbf{P}^{KT} = \mathbf{F} \cdot diag(\mathbf{x}^*) \cdot \log(\mathbf{S}^T) \tag{3.4}$$

$$\hat{y}^{KT} = \operatorname{argmax}_c \mathbf{P}^{KT} \tag{3.5}$$

In the second stage of the Label Augmentation step, a supervised classifier is fit using $\hat{y}^{KT}$ as labels for the input documents and $\mathbf{F}$ as the feature matrix. The final predicted labels from the Label Augmentation step are the "in-sample" predictions of the supervised classifier on $\mathbf{F}$, which we denote $\hat{y}^{KT-C}$. Additionally, the fitted supervised classifier could be used to make category predictions for a new batch of unseen input documents. This scenario would be most relevant when the original input documents used to fit the supervised classifier are unavailable, and thus the DCRI method cannot be rerun on a combined dataset of the original and unseen documents.

Note that any classification algorithm could be employed during the Supervised Labeling stage. The specific one used in the proposed algorithm is the linear Complement Naive Bayes algorithm [37]. This classifier has been shown to exhibit performance competitive with other standard classifiers on document classification tasks with the added benefit of providing particularly strong performance on datasets with class imbalance [37]. Though the key term predicted labels are likely to contain errors, existing literature has shown Naive Bayes classifiers to be highly robust to label noise [28]. Therefore, it is reasonable to expect that the accuracy of the final predicted labels $\hat{y}^{KT-C}$ could approach the accuracy of a supervised classifier trained on $\mathbf{F}$ and the (unavailable) true category labels for the input documents.



(a) Key Term Labeling

(b) Supervised Classification

Figure 3-4: Label Augmentation step of the DCRI method. Label Augmentation consists of two sequential procedures. Figure 3-4a illustrates the first procedure, Key Term Labeling, which generates a new set of category predictions for each document ($\hat{y}^{KT}$) based on an optimized subset of key terms from the vocabulary. Figure 3-4b shows the second procedure, which yields the final category predictions ($\hat{y}^{KT-C}$) by training a supervised classifier on the document term frequencies and estimated labels $\hat{y}^{KT}$.

To provide further intuition regarding the value of the Label Augmentation step, it is instructive to consider the NG dataset in detail. Figure 3-5 plots each term in the vocabulary of the NG dataset as a point. The x-axis value estimates the "true" specificity of each term within the document corpus using a metric called the *document corpus importance score.* Similar to the category-term importance scores ($\mathbf{S}$), the document corpus importance scores are TF-IDF scores calculated for each term and for each category. However, instead of the category descriptions, the document corpus importance scores utilize as the text for each category c the concatenated set of documents whose ground-truth label is c. The x-axis value for a term v is the difference between the highest document corpus importance score, which corresponds to category c_v^*, and the second highest score, which corresponds to category $c_v^{'}$. The y-axis value for term v measures the difference between the category-term importance score (from $\mathbf{S}$) for category c_v^* and the highest remaining category-term importance score, corresponding to category $\hat{c}_v$.

Figure 3-5 demonstrates that some words in the NG vocabulary are much more useful for making category predictions than others due to differences between the category descriptions and the document corpus. To show this, three types of terms are highlighted in Figure 3-5. Terms with high positive values on both axes are both highly specific to category c_v^*, since this category has a much higher importance score than the other categories, and also highly consistent, since c_v^* has the highest score in both the document corpus and the category descriptions. Examples of such terms, which are called simply *specific terms*, are highlighted in blue. For these examples, $c^* = space$, and qualitatively it is clear the presence of any of these terms in a document strongly indicates that the document discusses a space-related topic.

Alternatively, y-axis values close to zero, such as the terms highlighted in green, are called *low-signal terms.* The lack of separation in category-term importance scores means that these terms do not contribute meaningfully to category predictions. Finally, terms with negative y-axis values are called *misleading terms.* These terms are not consistent, since the category with the highest importance score differs between the category descriptions and the document corpus. Misleading terms, such as the examples highlighted in orange, pose a problem for the category predictions since

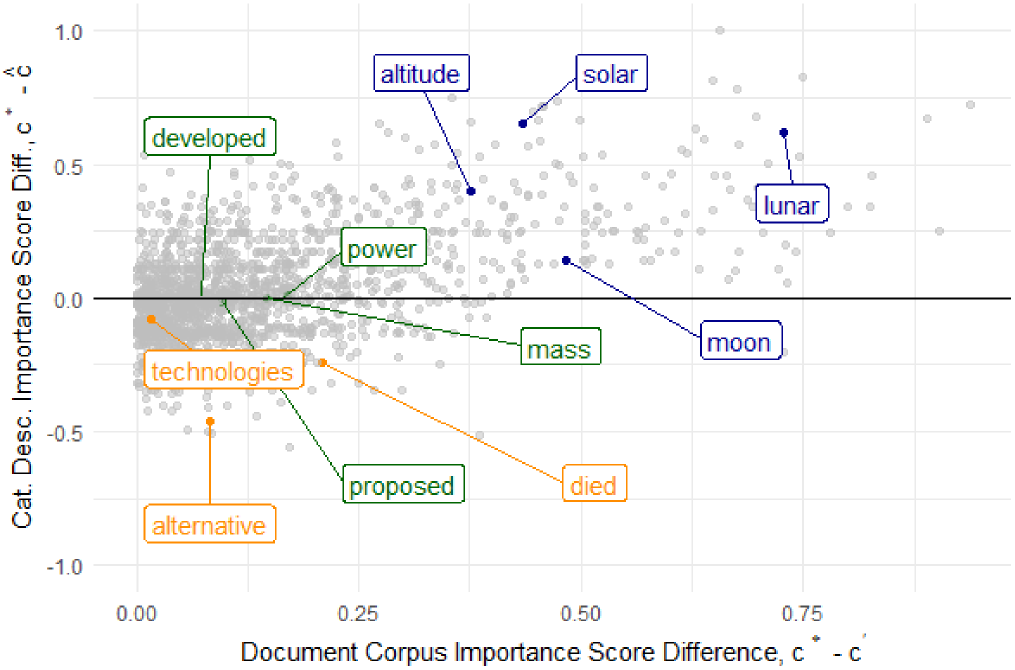


Figure 3-5: Term specificity and consistency between documents and category descriptions in the NG dataset. Each point represents a term v in the vocabulary. Examples of specific and consistent terms for the *space* category are highlighted in blue. Examples of low-signal terms are highlighted in green, and misleading terms are shown in orange.

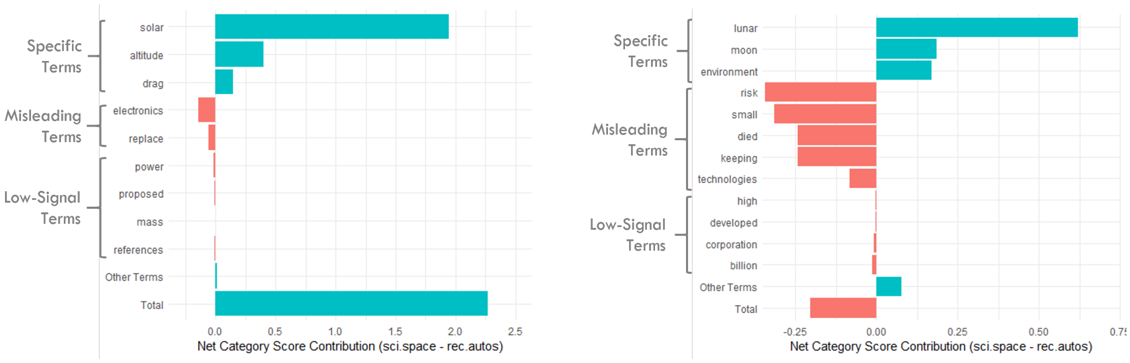
they appear to provide a signal for a particular category based on the category descriptions, but this signal is not correct. These terms are often not easily qualitatively associated with a particular category, and the apparent specificity of the category-term importance scores occurs due to random variation in word choice across category description authors.

The goal of the Label Augmentation step, and the Key Term Labeling procedure in particular, is to remove the impact of misleading terms from the category predictions while retaining the specific terms. Figure 3-6 shows an example of how this strategy can strictly improve upon the preliminary label predictions for two documents in the NG dataset, both of which have *space* as the ground-truth category label. The figure shows the difference between the *space* category importance score and the *cars* category importance score for a subset of words that have been manually identified as specific, low-signal, and misleading. The last row indicates the net difference summed across all words, so the sign of this difference indicates the preliminary category label for the document. In Figure 3-6a, the effect of specific terms outweighs the effect of the misleading terms, leading to a correct preliminary label. However, in Figure 3-6b, several very misleading terms are present, and the cumulative impact is large enough to induce an incorrect preliminary label. Note that if the category prediction were made based on the specific terms alone, both category predictions would have been correct.

3.4 DCRI Performance Evaluation

This section presents the experimental results on the three datasets described in Section 3.2. Recall that these datasets contain ground-truth labels for the documents that allow for evaluation of the classification accuracy of the DCRI method and competing methods. The results presented in this section test the following aspects of the DCRI method:

1. Classification accuracy of the final DCRI method category predictions



(a) Document 7: Correct Preliminary Label (b) Document 770: Incorrect Prelim. Label

Figure 3-6: Impact of specific, low-signal, and misleading terms on the preliminary label predictions. In both examples, the true label is *space*. The Total row in each subfigure indicates the difference in the preliminary label category scores for the *space* and *cars* categories. A positive value indicates that the preliminary category score for *space* is highest, and a negative value indicates that the preliminary category score for *cars* is highest.

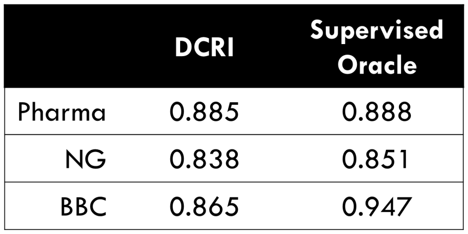
2. Classification performance relative to alternative methods for incorporating category description information, specifically Latent Dirichlet Allocation (LDA) and semi-supervised Multinomial Naive Bayes (MNB)

3. The value of the Key Term Labeling optimization procedure, measured by the accuracy of the revisions that the key term label predictions make to the preliminary label predictions

4. The degree of manual labeling effort required for a fully supervised approach to achieve equivalent classification accuracy to the DCRI method

3.4.1 DCRI Method Classification Accuracy

Figure 3-7 shows that the final DCRI method category predictions are highly accurate with respect to the ground-truth labels for the three datasets. To put these results in context, the DCRI method is compared to a theoretical supervised classifier trained on the input documents and the ground truth category labels, referred to as the Oracle classifier. The form of the Oracle classifier is an L2-regularized logistic regression, and 5-fold cross validation is used to obtain an out-of-sample predicted label for each



	DCRI	Supervised Oracle
Pharma	0.885	0.888
NG	0.838	0.851
BBC	0.865	0.947

Figure 3-7: DCRI accuracy, relative to the supervised Oracle classifier. The supervised Oracle classifier is a logistic regression trained on the ground-truth labels. The out-of-sample accuracies are shown in this figure, calculated using 5-fold cross validation.

document in the dataset. The out-of-sample classification accuracy of the Oracle is shown in Figure 3-7, and this accuracy can be thought of as an approximate upper bound on the accuracy that a user could expect from an unsupervised method like the DCRI method. For the Pharma and NG datasets, the accuracy of the DCRI method is within 1.5% of the Oracle accuracy, while the gap is larger, approximately 8%, on the BBC dataset.

3.4.2 Benchmark Method Comparison

In this section, the DCRI method is compared to several alternative machine learning methods based on either the unsupervised Latent Dirichlet Allocation (LDA) model [5] or the semi-supervised Multinomial Naive Bayes (SSMNB) model [31].

The standard unsupervised LDA model is unable to consistently partition the documents into clusters that represent the desired categories, highlighting the need for the category descriptions. Two LDA-based approaches that incorporate the category descriptions are considered, Matched LDA and Informed LDA, but the classification accuracies of these approaches underperform the DCRI method's accuracy by between 4-20% across the three datasets. The DCRI method also outperforms SSMNB by 1.4-5.7% on the two datasets with the most informative category descriptions, though the accuracy of SSMNB is 8% higher on the BBC dataset where the category descriptions are less informative.

Latent Dirichlet Allocation

LDA is an unsupervised probabilistic method for document clustering which takes an unlabeled corpus of input documents and a prespecified number of topics k as input. The underlying generative model assumes the existence of k unobserved topics parameterized by topic-specific multinomial models over the words in the vocabulary, and each input document is represented as a mixture over these topics. Each word in each input document is assumed to be sampled independently from this document-specific mixture distribution. Therefore, the LDA model parameters consist of the multinomial parameters for each topic as well as the topic mixture weights for each input document. The parameters are typically estimated using variational methods [18], and the maximum topic mixture weight can be used to assign a topic prediction to each document after fitting.

The drawback of LDA for the settings studied in this chapter is that LDA is unable to incorporate prior knowledge about the content of the topics into the standard parameter estimation procedure. This deficiency is evident when LDA is performed on the documents in the Pharma dataset, specifying $k = 5$ to match the known number of categories. Ideally, each of the five topics would correspond to a unique category. However, Figure 3-8 shows that topics 1, 2, and 3 contain a meaningful fraction of documents from at least two different categories. Furthermore, topics 4 and 5 are both primarily comprised of documents from the *process monitoring* category. This example shows that the topics obtained by the standard LDA model are not guaranteed to map one-to-one onto the desired categories, a general problem that renders unsupervised clustering approaches unsuitable for this setting.

Matched and Informed LDA

Instead of a fully unsupervised approach, two alternative methods that incorporate the category description information into the LDA model, *Matched LDA* and *Informed LDA*, are proposed as benchmarks for the DCRI method. Matched LDA performs a post-processing step on the fitted LDA parameters and matches the topic-term multi-

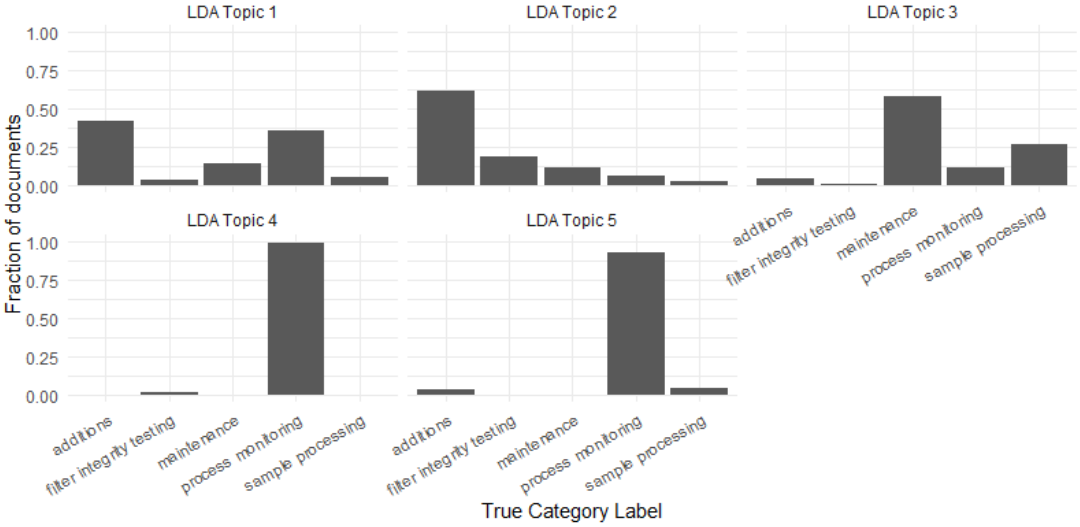
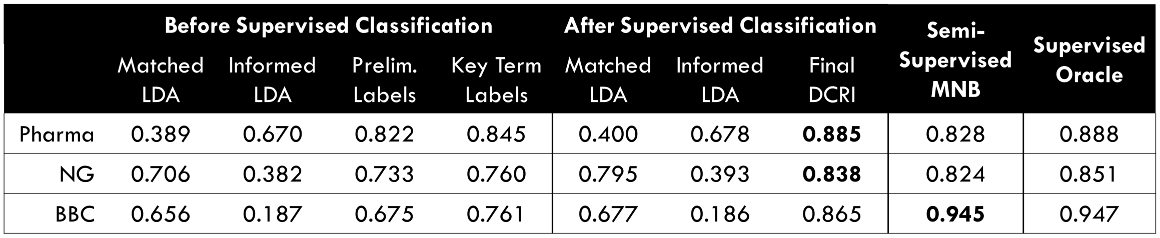


Figure 3-8: Distribution of documents across true category labels for each LDA predicted topic in the Pharma dataset. LDA topics 1, 2, and 3 contain documents from multiple categories, while topics 4 and 5 both primarily contain documents from the *process monitoring* category.

nomial distributions to the empirical category-term distributions estimated from the category descriptions. Every topic-category assignment is considered and the matching with the lowest root mean squared error (RMSE) is retained. Note that while this matching step forces a one-to-one correspondence between the topics and categories, the LDA parameter estimation procedure remains divorced from the category definitions, so there is no guarantee that a high-quality matching exists.

Informed LDA, which was proposed in [25], incorporates the category information directly into the LDA parameter estimation procedure, alleviating the need for the post-processing step in Matched LDA. The Informed LDA method proposes to use the empirical category-term distributions from the category descriptions as priors for the topic-term multinomial distributions, creating the necessary one-to-one correspondence between the LDA topics and the categories. These priors are incorporated directly into the variational procedure for parameter estimation.

In the experimental results, the Matched LDA and Informed LDA category predictions are considered as benchmarks for the key term label predictions made by the DCRI method during the Label Augmentation step before applying the Supervised



	Before Supervised Classification				After Supervised Classification			Semi-Supervised MNB	Supervised Oracle
	Matched LDA	Informed LDA	Prelim. Labels	Key Term Labels	Matched LDA	Informed LDA	Final DCRI		
Pharma	0.389	0.670	0.822	0.845	0.400	0.678	**0.885**	0.828	0.888
NG	0.706	0.382	0.733	0.760	0.795	0.393	**0.838**	0.824	0.851
BBC	0.656	0.187	0.675	0.761	0.677	0.186	0.865	**0.945**	0.947

Figure 3-9: DCRI accuracy comparison to benchmarks. Each cell represents the prediction accuracy for a particular model on a particular dataset relative to the ground-truth labels. The Preliminary Labels, Key Term Labels, and Final DCRI columns represent the accuracies of the initial, intermediate, and final predictions of the DCRI method, respectively. Bold cells indicate the predictions with the highest accuracy. All accuracy differences within a delineated section for a single dataset are statistically significant with $p < 0.001$. Additionally, the accuracy differences between the Final DCRI and emi-supervised MNB predictions are also statistically significant with $p < 0.001$.

Labeling procedure. The first four columns of Figure 3-9 demonstrate that Matched LDA and Informed LDA perform poorly relative to the DCRI method predictions. Both LDA methods consistently underperform the key term label predictions by a margin of at least 5% and up to 57%, and the classification accuracy of both methods varies massively across the three datasets. In fact, the LDA methods also underperform the preliminary label predictions made by the DCRI method before the Label Augmentation step. Note that all accuracy differences between the the LDA-based methods and the DCRI method are statistically significant with $p < 0.001$ calculated from 30 bootstrap samples [13].

Recall that the final label predictions from the DCRI method are obtained from a CNB supervised classifier trained using the document term frequencies as features and the key term label predictions as labels. The classification accuracy of the DCRI method's final predictions is shown in the seventh column of Figure 3-9. The fifth and sixth columns of Figure 3-9 represent the accuracies of the CNB supervised classifier trained using label predictions from Matched LDA and Informed LDA, respectively, during training. Though the CNB classification layer is almost universally beneficial, the LDA methods still exhibit meaningful accuracy variation and underperform the DCRI method across all three datasets. After the application of the CNB classifier, the accuracy differences between the LDA-based methods and the DCRI method remain statistically significant with $p < 0.001$.

Semi-Supervised Multinomial Naive Bayes

The Multinomial Naive Bayes (MNB) classifier is a popular document classification algorithm that can be applied in a supervised or semi-supervised setting. The underlying generative model for the MNB classifier assumes that each input document has a single associated category, and each word in each document is sampled independently according to a category-specific multinomial distribution. In the supervised setting, the maximum likelihood estimates for the multinomial parameters yield a linear classifier that motivates the form of the DCRI preliminary label scores (equation 3.1).

In a semi-supervised setting with both labeled and unlabeled input documents, the iterative Expectation Maximization (EM) algorithm can be used to obtain maximum likelihood parameter estimates [31]. SSMNB is implemented as a benchmark algorithm for the DCRI method by treating the category descriptions as individual labeled documents along with the unlabeled input documents. SSMNB is similar to the DCRI method in its initial treatment of the category descriptions as labeled input documents. However, the DCRI method explicitly corrects for the difference between the category descriptions and input documents in the Label Augmentation step, while SSMNB relies on the EM iterations to converge to parameter estimates that represent the input documents. An empirical comparison demonstrates the relative efficacy of these two correction mechanisms.

The DCRI method outperforms the SSMNB predictions on the Pharma and NG datasets, but SSMNB exhibits the best performance on the BBC dataset (Figure 3-9, second to last column). These mixed results can be attributed to two factors, (1) the quality of the category descriptions is higher for the Pharma and NG datasets, and (2) the effect of the EM procedure on classification accuracy is inconsistent and highly sensitive to violations of the underlying MNB model assumptions. The two factors are discussed in more detail below.

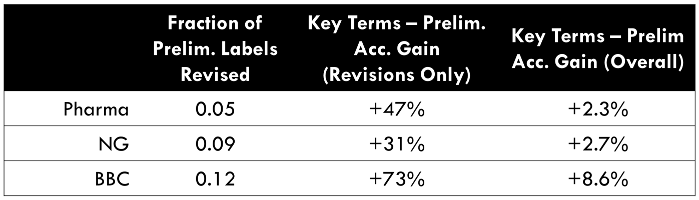
First, the Pharma and NG category descriptions provide superior information for document classification relative to the BBC dataset. This claim is substantiated by calculating the *document corpus importance scores* referred to in Section 3.3.3, which are TF-IDF scores for each term and each category calculated using the document corpus and the true labels. The document corpus importance scores are compared to **S**, the category term importance scores based on the category descriptions. Also recall from Section 3.3.3 that a term is denoted as *consistent* if the category with the maximum true category importance score matches the category with the highest importance score based on the category descriptions. The fraction of consistent terms is substantially higher for the Pharma and NG datasets (0.44 and 0.51, respectively) than the BBC dataset (0.31). The relative consistency between the documents and category descriptions in the Pharma and NG datasets enables the DCRI method to achieve accuracy close to that of the Oracle classifier, as discussed previously.

Second, the EM procedure is detrimental to classification accuracy for the Pharma and NG datasets but useful on the BBC dataset. The inconsistency of EM is a drawback of the SSMNB algorithm that has been observed in prior literature, particularly when a latent sub-category structure exists in the true document-generating distribution at a higher level of granularity than the categories that are modeled [30]. This sub-category structure is a violation of the assumed MNB generative model, which posits that the only latent category structure in the true document-generating distribution is the structure that is modeled. If the latent sub-categories span multiple modeled categories, then the EM procedure can yield poor classification results when measured against the modeled categories.

While the presence of latent sub-categories is difficult to measure directly, a high degree of separation between the modeled categories suggests that any latent sub-categories are unlikely to span multiple modeled categories. To test the separation between the modeled categories in each dataset, the documents are assigned to clusters based on their ground-truth category labels. According to the Calinski-Harabasz Index (CHI) [7], a clustering evaluation metric that measures the ratio of between-category document distances to within-category document distances, the BBC categories are more than twice as well-separated (CHI = 24.7) than the Pharma (CHI = 10.5) and NG (CHI = 11.4) categories. The high cluster separation for the BBC dataset is likely associated with the relative success of EM on this dataset.

3.4.3 Value of Key Term Optimization

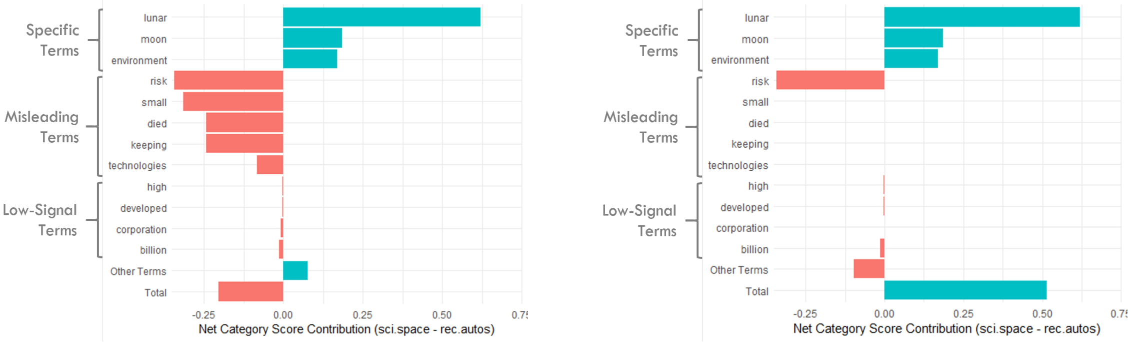
This section specifically illustrates the value of the Key Term Labeling optimization procedure within the DCRI method. Figure 3-10 shows the difference in accuracy between the preliminary label predictions $\hat{y}^{AT}$ and the key term label predictions $\hat{y}^{KT}$ made before the Supervised Labeling procedure is applied. Overall, the key term predicted labels outperform the preliminary labels on all three datasets by 2.3-8.6%. The key term predictions achieve this performance improvement by revising a modest fraction of observations, 5-12%, with a very meaningful increase in predictive accuracy (31 - 73%) on these revisions.



	Fraction of Prelim. Labels Revised	Key Terms – Prelim. Acc. Gain (Revisions Only)	Key Terms – Prelim Acc. Gain (Overall)
Pharma	0.05	+47%	+2.3%
NG	0.09	+31%	+2.7%
BBC	0.12	+73%	+8.6%

Figure 3-10: Accuracy difference between preliminary predicted labels using all terms and the revised key terms predictions after the Key Terms Labeling procedure. The second column indicates the accuracy difference on only the documents where the predictions differ. The third column is the product of the first and second columns.

To illustrate that the Key Term Labeling procedure achieves these improvements by retaining specific terms and ignores misleading generic terms as desired, Figure 3-11 revisits document 770 from the NG dataset discussed in Section 3.3.3. The Preliminary Labeling procedure predicts the incorrect label (*cars*) for this document due to the effects of misleading generic terms (Figure 3-11a). After applying the Key Terms Labeling procedure, Figure 3-11b shows that three of the four misleading generic terms, “small”, “died”, and “keeping” are ignored, while the terms specific to the correct category (space), “lunar”, ”moon”, and “environment”, are retained. Thus, the key term label prediction for document 770 is correct.



(a) Preliminary Category Scores

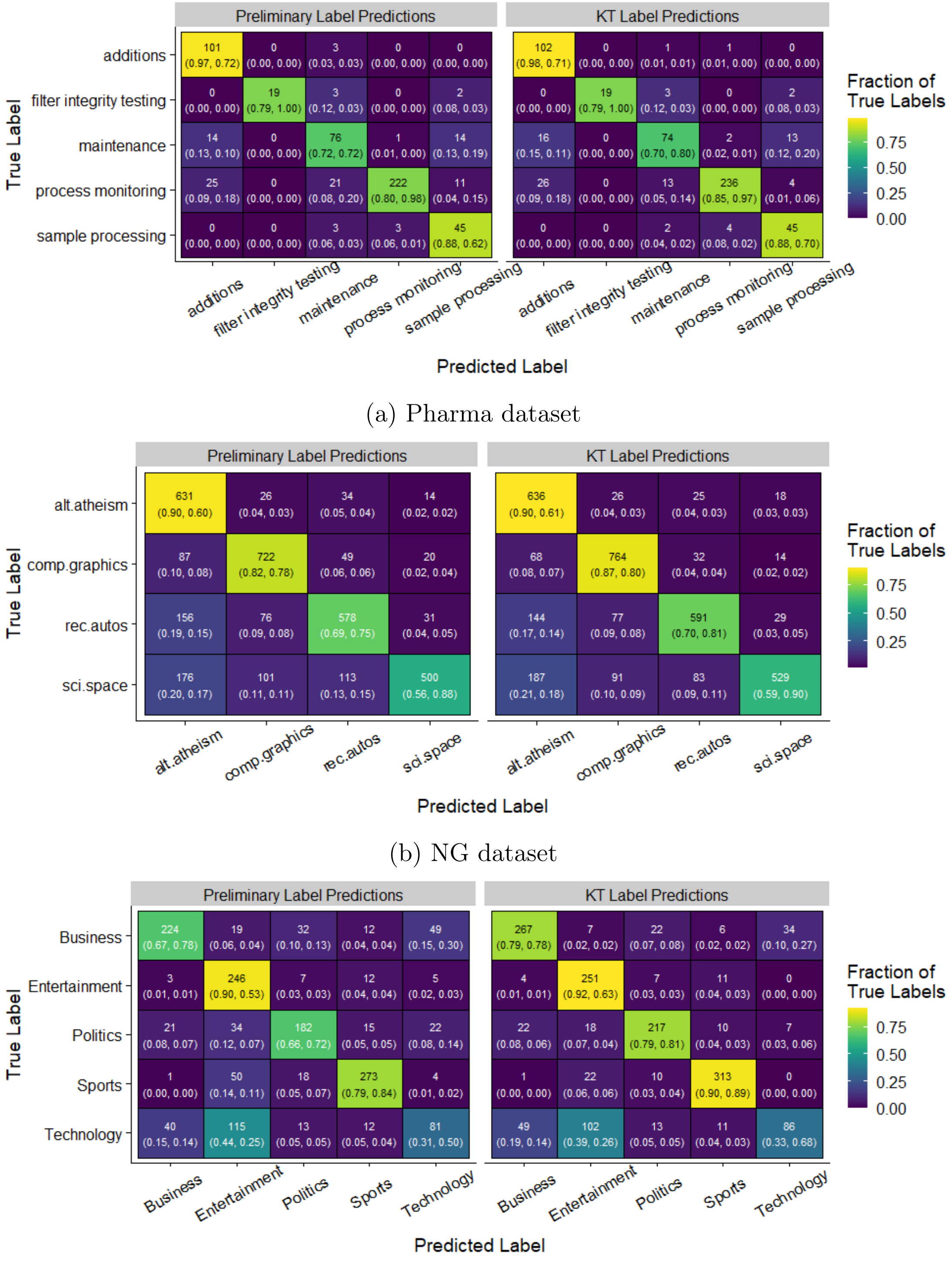
(b) Key Terms Category Scores

Figure 3-11: Preliminary vs Key Terms Category Scores for Document 770 in the NG dataset. The true label is sci.space. This figure shows that the Key Terms Labeling procedure successfully retains all specific terms while discarding four of five misleading generic terms.

Finally, Figure 3-12 compares the confusion matrices for the preliminary and key term predictions across the three datasets. Figures 3-12b and 3-12c indicate that Key Term Labeling yields a broad-based accuracy improvement across all categories for the NG and BBC datasets. For the Pharma dataset (Figure 3-12a) the accuracy improvement is narrowly concentrated in distinguishing the process monitoring category from the maintenance and sample processing categories.

Further examination of the key terms selected by the key term optimization in the Pharma dataset indicates that the optimization ignores the names of sensors and sample collection equipment. This is expected because the names of the sensors and sample collection equipment are not used consistently. Among the category descriptions, the sensors are specifically referenced by name primarily in the *process monitoring* description. However, the sensor names are mentioned in the documents associated with both *maintenance* and *process monitoring*. Similarly, sample collection equipment is referenced primarily in the *sample processing* category description, but since *sample processing* and *process monitoring* both occur while the bioreactor is running, references to sampling equipment can appear in *process monitoring* documents.

Similar confusion matrices analyzing the impact of the Supervised Classification procedure can be found in Appendix B.5.



Preliminary Label Predictions

True Label \ Predicted Label	additions	filter integrity testing	maintenance	process monitoring	sample processing
additions	101 (0.97, 0.72)	0 (0.00, 0.00)	3 (0.03, 0.03)	0 (0.00, 0.00)	0 (0.00, 0.00)
filter integrity testing	0 (0.00, 0.00)	19 (0.79, 1.00)	3 (0.12, 0.03)	0 (0.00, 0.00)	2 (0.08, 0.03)
maintenance	14 (0.13, 0.10)	0 (0.00, 0.00)	76 (0.72, 0.72)	1 (0.01, 0.00)	14 (0.13, 0.19)
process monitoring	25 (0.09, 0.18)	0 (0.00, 0.00)	21 (0.08, 0.20)	222 (0.80, 0.98)	11 (0.04, 0.15)
sample processing	0 (0.00, 0.00)	0 (0.00, 0.00)	3 (0.06, 0.03)	3 (0.06, 0.01)	45 (0.88, 0.62)

KT Label Predictions

True Label \ Predicted Label	additions	filter integrity testing	maintenance	process monitoring	sample processing
additions	102 (0.98, 0.71)	0 (0.00, 0.00)	1 (0.01, 0.01)	1 (0.01, 0.00)	0 (0.00, 0.00)
filter integrity testing	0 (0.00, 0.00)	19 (0.79, 1.00)	3 (0.12, 0.03)	0 (0.00, 0.00)	2 (0.08, 0.03)
maintenance	16 (0.15, 0.11)	0 (0.00, 0.00)	74 (0.70, 0.80)	2 (0.02, 0.01)	13 (0.12, 0.20)
process monitoring	26 (0.09, 0.18)	0 (0.00, 0.00)	13 (0.05, 0.14)	236 (0.85, 0.97)	4 (0.01, 0.06)
sample processing	0 (0.00, 0.00)	0 (0.00, 0.00)	2 (0.04, 0.02)	4 (0.08, 0.02)	45 (0.88, 0.70)

(a) Pharma dataset

Preliminary Label Predictions

True Label \ Predicted Label	alt.atheism	comp.graphics	rec.autos	sci.space
alt.atheism	631 (0.90, 0.60)	26 (0.04, 0.03)	34 (0.05, 0.04)	14 (0.02, 0.02)
comp.graphics	87 (0.10, 0.08)	722 (0.82, 0.78)	49 (0.06, 0.06)	20 (0.02, 0.04)
rec.autos	156 (0.19, 0.15)	76 (0.09, 0.08)	578 (0.69, 0.75)	31 (0.04, 0.05)
sci.space	176 (0.20, 0.17)	101 (0.11, 0.11)	113 (0.13, 0.15)	500 (0.56, 0.88)

KT Label Predictions

True Label \ Predicted Label	alt.atheism	comp.graphics	rec.autos	sci.space
alt.atheism	636 (0.90, 0.61)	26 (0.04, 0.03)	25 (0.04, 0.03)	18 (0.03, 0.03)
comp.graphics	68 (0.08, 0.07)	764 (0.87, 0.80)	32 (0.04, 0.04)	14 (0.02, 0.02)
rec.autos	144 (0.17, 0.14)	77 (0.09, 0.08)	591 (0.70, 0.81)	29 (0.03, 0.05)
sci.space	187 (0.21, 0.18)	91 (0.10, 0.09)	83 (0.09, 0.11)	529 (0.59, 0.90)

(b) NG dataset

Preliminary Label Predictions

True Label \ Predicted Label	Business	Entertainment	Politics	Sports	Technology
Business	224 (0.67, 0.78)	19 (0.06, 0.04)	32 (0.10, 0.13)	12 (0.04, 0.04)	49 (0.15, 0.30)
Entertainment	3 (0.01, 0.01)	246 (0.90, 0.53)	7 (0.03, 0.03)	12 (0.04, 0.04)	5 (0.02, 0.03)
Politics	21 (0.08, 0.07)	34 (0.12, 0.07)	182 (0.66, 0.72)	15 (0.05, 0.05)	22 (0.08, 0.14)
Sports	1 (0.00, 0.00)	50 (0.14, 0.11)	18 (0.05, 0.07)	273 (0.79, 0.84)	4 (0.01, 0.02)
Technology	40 (0.15, 0.14)	115 (0.44, 0.25)	13 (0.05, 0.05)	12 (0.05, 0.04)	81 (0.31, 0.50)

KT Label Predictions

True Label \ Predicted Label	Business	Entertainment	Politics	Sports	Technology
Business	267 (0.79, 0.78)	7 (0.02, 0.02)	22 (0.07, 0.08)	6 (0.02, 0.02)	34 (0.10, 0.27)
Entertainment	4 (0.01, 0.01)	251 (0.92, 0.63)	7 (0.03, 0.03)	11 (0.04, 0.03)	0 (0.00, 0.00)
Politics	22 (0.08, 0.06)	18 (0.07, 0.04)	217 (0.79, 0.81)	10 (0.04, 0.03)	7 (0.03, 0.06)
Sports	1 (0.00, 0.00)	22 (0.06, 0.06)	10 (0.03, 0.04)	313 (0.90, 0.89)	0 (0.00, 0.00)
Technology	49 (0.19, 0.14)	102 (0.39, 0.26)	13 (0.05, 0.05)	11 (0.04, 0.03)	86 (0.33, 0.68)

(c) BBC News dataset

Figure 3-12: Confusion matrices for the Preliminary Labels and Key Terms (KT) label predictions. Each cell of the confusion matrix contains the number of documents in the first row, and the second row gives the fraction with respect to the row total and then the fraction with respect to the column total.

3.4.4 Manual Labeling Comparison

The natural alternative to using category descriptions to inform label predictions is a more manual approach. Specifically, a subset of the input documents is manually labeled and used to train a supervised classification algorithm, which can subsequently predict the labels for the remaining documents. To construct the supervised classifier, a subset of documents is sampled uniformly at random from the document corpus. The documents in this subset are assumed to be labeled correctly by an expert and are used to fit a supervised CNB classifier. In this section, the manual labeling burden necessary to generate a supervised classifier with superior performance to the DCRI method is quantified.

The performance of the supervised classifier is compared to the DCRI method in two settings. First, the accuracy of the supervised classifier and the DCRI method are evaluated on the full corpus of input documents, which is denoted the "fixed corpus" setting. This comparison is designed to determine the fraction of documents that must be manually labeled so that the accuracy of the supervised classifier on the full document corpus is statistically equivalent to the accuracy of the DCRI method. This setting is the appropriate comparison if the metric of interest is classification accuracy on the documents available at the time of training. Alternatively, a "dynamic corpus" setting is considered, where the metric of interest is classification accuracy on new documents not available when training the supervised classifier. In this setting, only the documents not chosen for expert labeling are relevant for the accuracy comparison.

Table 3.2 shows that, in the fixed corpus setting, an expert must label 25-30% of documents to train a classifier with equivalent performance to the DCRI method on the Pharma and NG datasets. This fraction is slightly lower for the BBC dataset at 15-20%, though this still translates to a nontrivial labeling effort of 223-298 documents. In the dynamic corpus setting, the subset of expert-labeled documents would need to be large for the Pharma and NG datasets, over 90% and 60-70% of the documents respectively, while 15-20% of the input documents remains sufficient for the BBC dataset.

	Fixed Corpus	Dynamic Corpus
Pharma	$25-30\%$	$>90\%$
NG	$25-30\%$	$60-70\%$
BBC	$15-20\%$	$15-20\%$

Table 3.2: Minimum required percentage of documents that would need to be manually labeled for a supervised classifier to outperform the unsupervised pipeline. Fixed corpus setting gives credit for a correct "prediction" to all manually labeled documents. Dynamic corpus setting only evaluates performance on the documents that are actually predicted, providing a sense of performance on unseen data.

3.5 Conclusions

The DCRI method proposed in this chapter offers a method for automatically categorizing documents in a setting where the categories are known but a labeled training set of documents is unavailable. The standard approach in this setting is to manually label some or all of these documents, training a supervised classification algorithm to predict the labels of the remaining documents. The DCRI method effectively replaces the training set of labeled documents, which is often difficult and expensive to obtain, with written reference information about the document categories that is assumed to be easily accessible. The Label Augmentation step in the DCRI method explicitly adjusts for the fact that the reference descriptions of the categories are not written by the same authors or for the same purpose as the unlabeled input documents.

Empirical results on three datasets demonstrate that a significant manual labeling effort, between 15-30% of the dataset, is required to match the accuracy of the DCRI method on a fixed corpus of documents. A larger set of labeled documents is generally required to create a supervised classifier that is competitive on new batches of unlabeled input documents, indicating that DCRI method is more scalable than manual labeling.

This chapter demonstrates that the DCRI method performs well in comparison to other benchmark approaches for integrating category description information into document classification algorithms. The Key Term Labeling optimization procedure imposes sparsity that effectively corrects for differences between the input documents and category descriptions, an adjustment that is not possible when existing clustering or topic modeling algorithms such as LDA. SSMNB is unable to consistently outperform the DCRI method, though SSMNB does offer superior performance for one of the three datasets.

The empirical results demonstrate that the DCRI method can effectively substitute existing reference documentation in place of a labeled training set in certain unsupervised document classification settings.

Chapter 4

A Two-Stage Machine Learning Model for Defect Detection in Optical Transceiver Manufacturing

4.1 Introduction

Quality control is a critical aspect of the management of manufacturing lines, particularly of high-tech products that require high reliability. To obtain the desired quality, the design and operations of manufacturing lines include a variety of tests throughout the production process with the goal of identifying and hopefully eliminating quality problems at early stages. Many of these tests are conducted using dedicated machines and equipment, but the final determination of whether the test results signal a quality problem is often performed by highly skilled human experts. The use of highly skilled personnel to conduct repetitive tasks is not only costly, but potentially leads to inconsistent outcomes that could depend on specific individuals and their respective knowledge, training and expertise. This motivates the need to develop machine learning-enabled automation of the review of quality tests results. However, methods for optimally designing the interactions between advanced machine learning algorithms and human experts remains an open question. Specifically, while machine learning algorithms can leverage massive amounts of historical data, in many cases, the subject matter expertise of human experts and their awareness of contextual factors is still critical.

This work is based on collaborative work with an industry partner, a manufacturer of optical communication equipment. One of the products manufactured by this partner is a optical transceiver called a Quad Small Form factor Pluggable (QSFP) that interfaces between fiber optic cables and network hardware, such as servers. The key role of the QSFP is to convert between electrical and optical signals using lasers and photo diodes to transmit and receive optical signals, respectively. This work focuses on a critical laser quality test that is run near the end of the manufacturing process to evaluate the reliability of the transceiver when operating at high temperatures. While standard supervised classification algorithms have been applied broadly to quality test review [43, 51, 57, 29], the machine learning approach developed in this work explicitly integrates the operator process knowledge into the underlying models and algorithms.

This chapter proposes a two-stage machine learning classification model, called *R-RF*, that is able to make automated pass/fail decisions for the laser quality test and drastically reduces the need for manual review of test logs. The proposed approach codifies the operator's qualitative observation that some modules are much easier to classify than others. Thus, in the first stage, a simple rules-based classifier with data-driven thresholds is applied to all modules to identify the clearly passing modules with very high accuracy. These thresholds are determined through a data-driven optimization model. Any module that cannot be passed in the first stage is advanced to a second stage random forest classifier [6] trained specifically to make pass/fail predictions on modules that cannot be classified in the first stage. Modules that are not passed by the second stage are routed to the operator for manual inspection.

Motivated by the specific manufacturing setting, the performance of the proposed model is measured by its ability to minimize the fraction of modules that require manual review subject to a maximum false omission rate (FOR) upper bound (maximal fraction of modules predicted to pass that actually fail). Assessment on out-of-sample real data suggests that the model is able to reduce the manual review burden on the operator by 75-99% while on average satisfying the FOR upper bound. As a benchmark, the performance of the newly proposed two-stage classifier is compared to existing state-of-the-art tree-based algorithms, and R-RF is superior in reducing manual review at the expense of slightly inferior false omission rate (FOR) control.

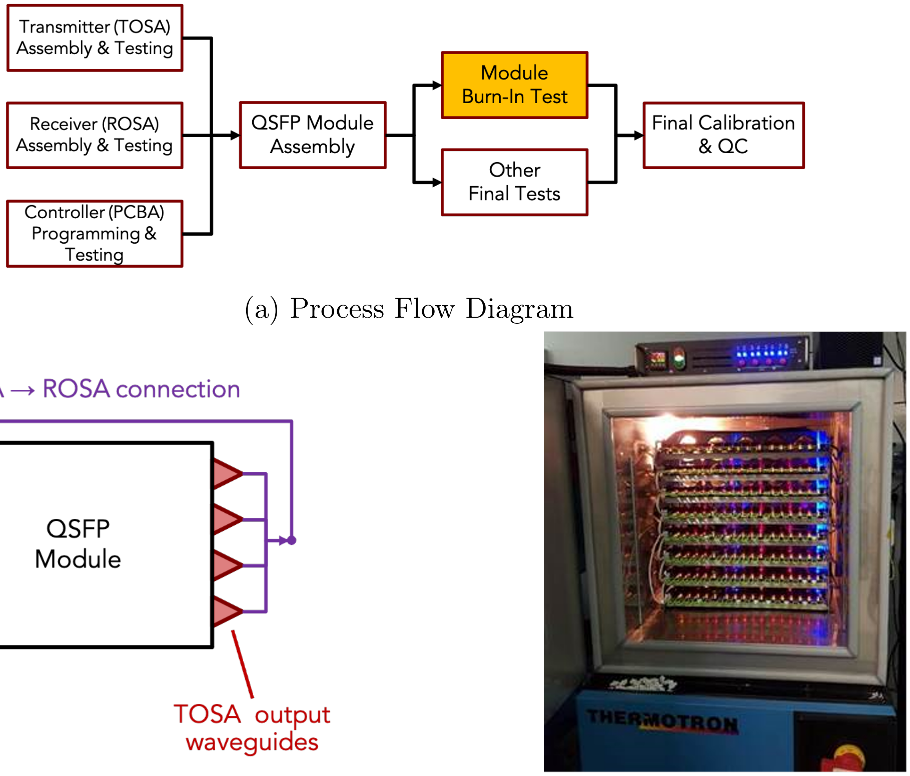
Section 4.2 provides background on the QSFP manufacturing process and the MBI test, and describes the data generated during a MBI test run. Section 4.3 details the structure and training of the two-stage R-RF model, including training and hyperparameter optimization procedures. Section 4.4 describes experimental results run on two datasets from our industrial partner and compares the R-RF model to several benchmark models, and Section 4.5 concludes.

4.2 Setting and Data Description

4.2.1 QSFP Manufacturing and Quality Testing

A QSFP transceiver, also known as a module, consists of three primary subassembly components, a transmitting component (TOSA) that outputs an optical signal via lasers, a receiving component (ROSA) capable of reading an input optical signal via photo diodes, and a programmable control unit (PCBA). The TOSA unit has four output channels, called waveguides, each with its own laser that outputs an optical signal at a specific wavelength when electrical current is applied. Similarly, the ROSA unit has four input waveguides, each with a distinct photo diode that can read an incoming laser signal.

Figure 4-1a illustrates the process flow diagram for the QSFP manufacturing process. In the first stage of the QSFP manufacturing process, the TOSA, ROSA, and PCBA are manufactured and tested independently. The second stage of the process involves assembly of these three components, followed by a series of final tests of the assembled module. This final functional testing sequence includes the module burn-in (MBI) test, the focus of our failure modeling, as well as a collection of other quality tests. After final functional tests, the QSFP is calibrated and receives a quality control inspection before shipment.



(a) Process Flow Diagram

(b) Module Configuration for Burn-In

(c) Module Burn-In Test Rack

Figure 4-1: QSFP Manufacturing and Testing Process

The MBI test is the key test of post-assembly laser quality. During the test, the TOSA output waveguides are connected to the ROSA input waveguides (Figure 4-1b). By running the lasers at a high temperature (70 degrees Celsius) for an extended period of time (24-44 hours), the test operator determines if both the lasers and receiving diodes are able to function reliably under stressful conditions. Figure 4-1c shows a MBI test oven, which can test up to 120 lasers in parallel during a single run.

During the MBI test, automated sensors record readings approximately once every six minutes for each module. These sensors measure the electrical current applied and the resulting optical power output at each of the four TOSA lasers, the power at each receiving ROSA photo diode, the voltage level at the TOSA and ROSA, and the module temperature. The objective of the MBI test is to ensure that all of the power, voltage, and current measurements are stable when the module is run for an extended period under high temperature conditions. A module fails the MBI test when one of the following occurs:

1. **High variability** in either the power output readings from the TOSA lasers or the power received readings from the ROSA photo diodes indicates that the module is not performing reliably under the stressed conditions. Variability in the voltage and current readings is also a concern, though this rarely leads to test failures.

2. **Divergence** between the power readings of four laser positions indicates a problem with a subset of the lasers.

3. **Very low power** readings near the beginning of the test run.

Currently, a skilled engineer manually reviews the logs of sensor data for each individual module test run ($\sim$60,000 per year) to determine if the module should pass the MBI test, a process that requires between 2-10 hours per week. The engineer employs a set of heuristically-derived thresholds on four features as well as visual inspection of the logs to determine which modules are defective. Specifically, the engineer passes modules that satisfy the thresholds on all four features with minimal review and then closely inspects the modules that do not satisfy these thresholds. The underlying assumption that motivates this approach is that some of the modules can be easily identified as passing via a small set of simple rules, while for the remaining modules the pass/fail decision is more difficult and requires additional inspection of the logs.

4.2.2 Module Burn-In Test Data

Two experimental datasets are used to test the proposed R-RF method, a dataset from September 2019 - May 2020 consisting of 21,577 test runs and a dataset from August 2020 - June 2021 consisting of 32,065 test runs. These datasets consist of all runs that exhibit no significant measurement errors. Runs with abnormally infrequent measurements, a high fraction of missing values, zero variability in TOSA optical power (indicating test setup failure), or unstable module temperature are excluded from the dataset. In an automated test review setting, these module runs could be identified and designated for retesting before applying the R-RF model using the same simple screening rules applied in this chapter.

Each test run has a record of the associated Pass or Fail label that was assigned manually by the operator. In practice, the operator may have determined that an MBI result was inconclusive and run the MBI test again for this module. In this chapter, the eventual Pass or Fail label is assigned to the initial MBI test run for each module, regardless of whether the module was retested. A machine learning classifier that is able to accurately determine the correct final label using only features from the first test adds additional value by reducing the need for module retests.

Both datasets exhibit a significant class imbalance due to low failure rates. The failure rate in the 2020-2021 dataset is even lower than the 2019-2020 dataset is due to the addition of a new quality control test at the laser supplier for lasers in the 2020-2021 modules. Since this change was known to our industrial partner and marked a clear regime switch, the two datasets were modeled separately.

The raw feature data for each module consists of the time series sensor readings, as mentioned in Section 4.2.1. Based on these sensor readings, $d = 20$ scalar features are created. These features represent summary statistics of the time series and serve as the input to our classification model. For convenience when training the R-RF model, these features are scaled such that a high value of the feature indicates an increased likelihood of MBI test failure.

The features are designed to capture the MBI test failure modes documented in the preceding section. For example, there are features that measure the range of TOSA power for each of the four lasers and use the maximum range as an indication of overall variability in TOSA power for the test. Most of the features, including the TOSA power range, ignore the beginning and end of the test period when the oven temperature is warming up and cooling down. The intervening steady temperature period is identified using a simple threshold on the change in module temperature. For more details on the steady temperature period calculation and summary statistics for the full list of features, please see Appendix C.

4.2.3 Examples of Easy and Complex Pass/Fail Decisions

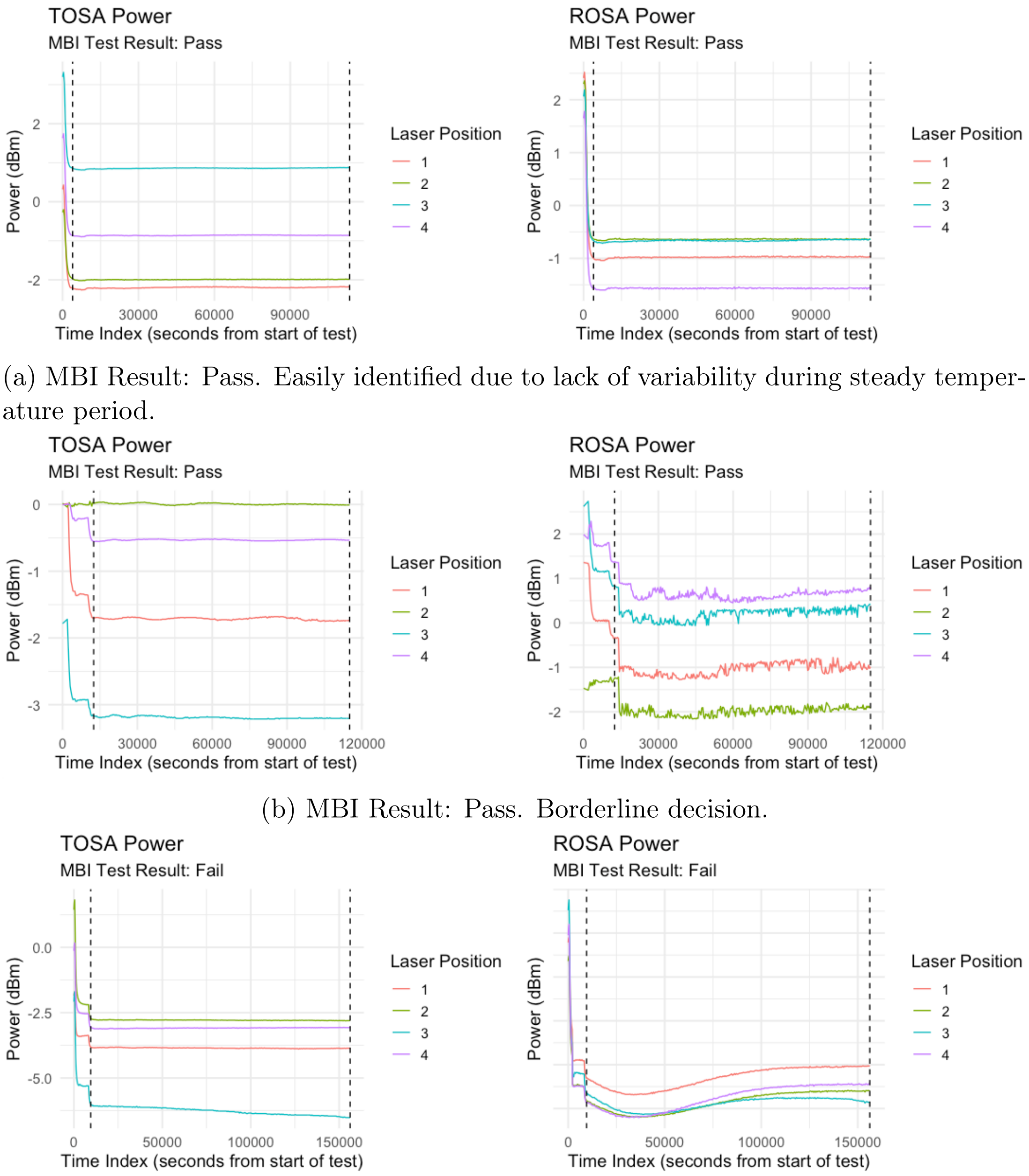
Figure 4-2 illustrates the motivation for a two-stage classification approach by considering three instances of MBI test runs. In Figure 4-2a, the TOSA and ROSA power readings show almost no variation through the steady temperature period, indicating no variability in operating performance. Module runs with this profile could be easily identified with a strict threshold on the maximum range of the power readings during the steady temperature period.

However, Figure 4-2b shows the power readings for another passing module where the decision is not as obvious. This module has some noise in the ROSA power readings and slight variation in TOSA power for laser 1, but the variation was deemed tolerable and the power readings for all lasers are tightly correlated, so the module was passed. The module in Figure 4-2c exhibits very little noise, but laser 3 deviates from the trend of the other lasers in both the TOSA and ROSA readings, so this laser is deemed defective and the module fails the MBI test. The example in Figure 4-2c is illustrative because subtle variation in laser 3's power readings in either the TOSA or the ROSA plot might be attributed to measurement noise, but the presence of the same signal in both plots yields the failure result. The failure decision in this case involves the interaction between multiple features, which is better captured by a tree-based machine learning model than a set of simple rules.

These examples motivate the use of a simple, rules-based classifier to separate modules with little to no variability from the rest of the modules, and a more complex classifier to distinguish borderline passing modules from the ones that should fail.

4.3 R-RF Model Description

The R-RF model is a two-stage classification model that takes as input a module's raw log data from the MBI test and designates a subset of modules to be automatically passed. The modules marked for Automatic Pass are considered to have passed the



(a) MBI Result: Pass. Easily identified due to lack of variability during steady temperature period.

(b) MBI Result: Pass. Borderline decision.

(c) MBI Result: Fail. Borderline decision.

Figure 4-2: Examples of Easy and Complex MBI Decisions. Dotted black lines indicate the boundaries of the steady temperature period.

MBI test and do not receive further manual review, while the remaining modules are directed to the operator for a final Pass or Fail decision. The model also takes as input a FOR target, which specifies an acceptable fraction of the Automatic Pass modules that would have been marked Fail by the operator. The objective of the R-RF model is to satisfy the FOR target while minimizing the number of manually reviewed modules that the operator marks as Pass.

Figure 4-3 demonstrates how the trained R-RF model makes a prediction for a single MBI test run once the raw log data has been converted to the feature vector introduced in Section 4.2.2. The first step is to apply the Stage 1 classifier, which consists of a threshold on each of the features. If every feature value is below its associated threshold, then the Stage 1 classifier labels the test run as an Automatic Pass. However, if any of the thresholds is violated, then the Stage 1 classifier designates the test run for review by the Stage 2 classifier.

The Stage 2 classifier is a random forest model consisting of 125 decision trees, each with access to a random selection of four input features. For every test run that is not automatically passed by the Stage 1 classifier, the Stage 2 classifier assigns a probability that the test run should be automatically passed. A decision threshold for this probability, calibrated during training to the prespecified FOR target, is used to determine which of the remaining modules should be marked as Automatic Pass and which should be retained for manual review.

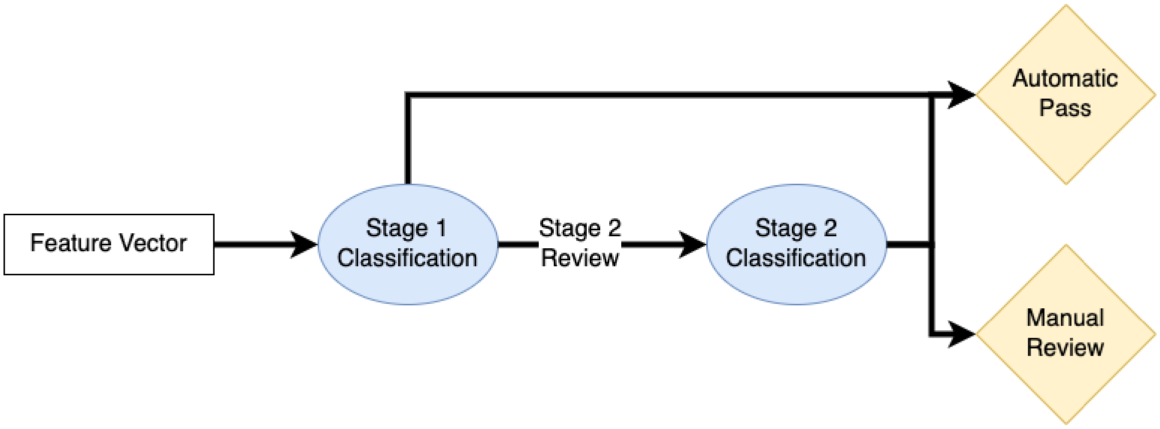


Figure 4-3: R-RV Model Prediction

4.3.1 Model Training

The R-RF model is trained using a nested cross validation procedure [44] consisting of random outer cross validation splits to estimate the accuracy of the predictions and an inner k-fold cross validation procedure to optimize model hyperparameters. For each outer split, 75% of the module runs are sampled without replacement and used for training and validation of the classifiers. This dataset is denoted $\mathcal{D}_{trainval}$. The remaining 25% of the module runs are reserved as a held-out test set, denoted $\mathcal{D}_{test}$. The test set is used to evaluate the performance of the R-RF model according to the realized FOR rate, manual review rate (fraction of modules that are not marked as Automatic Pass), and precision (fraction of predicted Manual Review modules where the true operator label is Fail). Additionally, 100 outer splits are performed to analyze the variability of these metrics due to sampling noise.

Figure 4-4 illustrates the training procedure for the R-RF model on a single outer cross validation split. First, a k-fold inner cross validation split is performed on $\mathcal{D}_{trainval}$. Each inner split consists of a training set (80% of the training and validation set) denoted $\mathcal{D}_{train}$ and a validation set (20%) denoted $\mathcal{D}_{val}$. The purpose of the inner cross validation is to perform hyperparameter optimization via grid search on the maximum tree depth and decision threshold for the Stage 2 random forest, a step that is detailed in the next section.

Once the optimal hyperparameter values have been chosen, the Stage 1 and 2 classifiers are trained on $\mathcal{D}_{trainval}$. The parameters of the Stage 1 classifier, the feature thresholds, are trained first using a greedy optimization algorithm detailed in Section 4.3.3. The algorithm sets the thresholds to maximize the number of Automatic Pass predictions under the strict constraint that no run can be labeled as an Automatic Pass if the true operator label is Fail. The modules that are labeled as Automatic Pass using the trained thresholds are removed from $\mathcal{D}_{trainval}$, and the remaining modules are used to build the decision trees that compose the Stage 2 random forest.

Finally, the trained Stage 1 and 2 classifiers are used to make final Automatic Pass predictions for runs in $\mathcal{D}_{test}$. These test set predictions are used to calculate the

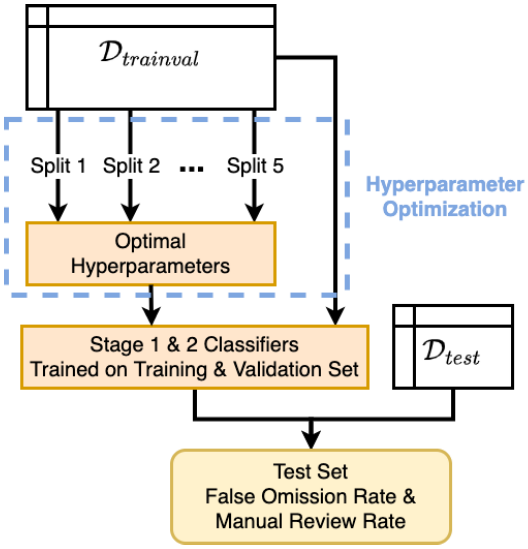

$\mathcal{D}_{trainval}$
Split 1
Split 2
...
Split 5
Hyperparameter Optimization
Optimal Hyperparameters
Stage 1 & 2 Classifiers Trained on Training & Validation Set
$\mathcal{D}_{test}$
Test Set False Omission Rate & Manual Review Rate

Figure 4-4: R-RV Model Nested Cross Validation

FOR rate, manual review rate, and precision for the outer split.

4.3.2 Hyperparameter Optimization

Figure 4-5 illustrates the hyperparameter optimization procedure for a single inner split with training dataset $\mathcal{D}_{train}$ and validation dataset $\mathcal{D}_{val}$. The hyperparameter optimization uses a grid search to optimize the maximal tree depth and decision threshold for the Stage 2 random forest by maximizing precision subject to a constraint on the FOR rate. In addition to the datasets, hyperparameter tuning takes in the FOR upper bound and a grid of feasible hyperparameter values defined by a range on each hyperparameter. The ranges are set such that the optimal hyperparameter values lie in the interior of the ranges.

Using $\mathcal{D}_{train}$, the Stage 1 feature thresholds are obtained using the greedy optimization procedure described in Section 4.3.3. Then, the modules labeled for Stage 2 review are used to train a Stage 2 random forest model for each each combination of hyperparameters in the input hyperparameter grid. The resulting collection of Stage 2 classifiers, along with the Stage 1 thresholds, is evaluated on the $\mathcal{D}_{val}$ using a metric denoted in this chapter as *conditional precision*. Conditional precision is equal to precision if the FOR is below the target rate and $-k$ if the FOR is not below the target. A classifier only gets credit for its achieved precision if the FOR is well-controlled. Additionally, the condition precision for each hyperparameter combination is calculated for each inner split and the hyperparameter selection step chooses the combination of hyperparameters that maximizes the mean conditional precision across the k inner splits. Therefore, a hyperparameter combination will have a negative mean conditional precision score across all k folds and should never be chosen if any of the folds misses the FOR target.

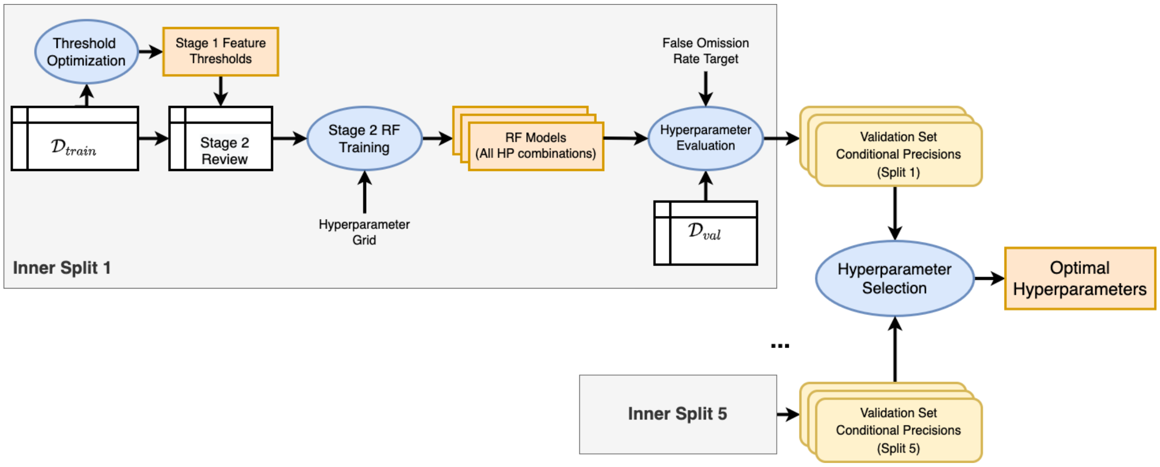

Threshold Optimization
Stage 1 Feature Thresholds
$\mathcal{D}_{train}$
Stage 2 Review
Stage 2 RF Training
Hyperparameter Grid
RF Models (All HP combinations)
False Omission Rate Target
Hyperparameter Evaluation
$\mathcal{D}_{val}$
Inner Split 1
Validation Set Conditional Precisions (Split 1)
Hyperparameter Selection
Optimal Hyperparameters
...
Inner Split 5
Validation Set Conditional Precisions (Split 5)

Figure 4-5: R-RV Model Hyperparameter Optimization

4.3.3 Stage 1 Greedy Optimization

The choice of the Stage 1 classifier thresholds can be formulated as a constrained optimization problem with the threshold vector, denoted $\tau \in \mathbb{R}^d$, as the decision variables. The objective is to maximize the number of modules that are labeled as Automatic Pass, that is the number of modules with all feature values strictly below their respective thresholds. The choice of τ is constrained such that the FOR falls below a target value, and the R-RF model sets this FOR target to zero. The input data for this optimization problem include a training dataset of n module runs, denoted $X \in \mathbb{R}^{n \times d}$ and the operator labels for these runs, denoted $y \in \{\text{Pass}, \text{Fail}\}^n$. X_j denotes the column vector of X associated with feature $j \in [d]$, where $[d] = \{1, 2, \ldots, d\}$. While this optimization problem could be formulated as a nonconvex nonlinear program or as a mixed integer linear program, the proposed greedy approximation is significantly more tractable at the problem sizes in our examples (15,000 module runs or more).

The proposed Algorithm 1 below takes a greedy approach to obtain a tractable optimization procedure to set τ. To begin, note that the the FOR constraint implies a minimum target number of true positives, modules failed by the operator that do not pass the thresholds, that must be delivered by any feasible choice of τ. For example, if the maximum allowable FOR is zero, then the target number of true positives, denoted TP_{target}, is equal to the number of modules labeled Fail in y. Additionally, note that it is possible to limit the choice of τ to a discrete set of values $\mathcal{S}$, which consists of the values that appear in X as well as a value that is slightly higher than the maximum feature value in X. To see this, observe that for any threshold vector $\tilde{\tau}$ containing values that do not appear in S, a vector $\tilde{\tau}'$ can be constructed using only values in S with the same the number of Automatic Pass modules and the same FOR. Based on these observations, Algorithm 1 takes as input (1) TP_{target}, and (2) a matrix of potential thresholds $T \in \mathbb{R}^{(n+1)\times d}$ with elements t_{ij}. Each column T_j sets $t_{1j} = \max(X_j) + 1$, and the rest of the column contains the values in X_j sorted in descending order.

The procedure starts by setting $\tau_j = t_{1j}$, such that all runs in X are initially predicted as Automatic Pass. Clearly, the number of true positives under this initialization is zero, so the thresholds must be adjusted. The procedure progresses toward a feasible set of thresholds that achieves TP_{target} by iteratively reducing τ_j for some feature j such that at least one incremental true positive is generated. The notation $\Delta\mathrm{TP}(\tau, j, a)$ indicates the number of incremental true positives introduced by changing the threshold of feature j from τ_j to a while leaving all other elements of τ unchanged. Of course, a threshold reduction may also induce incremental false positives, modules that do not pass the thresholds but are labeled Pass by the operator, similarly denoted as $\Delta\mathrm{FP}(\tau, j, a)$.

During each iteration, a threshold reduction is considered for each feature independently holding τ constant for the other features. The threshold reduction considered is the minimum reduction necessary to induce one or more incremental true positives.

The threshold reduction that induces the smallest number of incremental false positives per true positive is implemented. If the number of true positives has not reached TP_{target}, then the algorithm proceeds to the next iteration.

Algorithm 1: Greedy Rules Threshold Optimization

```
Input  : $\text{TP}_{target}$; matrix $T$ of potential thresholds with elements $t_{ij}$
Output: Feature thresholds $\tau \in \mathbb{R}^d$
Initialize row index of $T$ for feature $j$, $i_j = 1 \quad \forall j \in [d]$;
Initialize thresholds $\tau_j = t_{1,j} \quad \forall j \in [d]$;
Initialize current true positives $TP = 0$;
while $TP < TP_{target}$ do
    $i_j^{'} = \max_i \{i \in [n] : \Delta\text{TP}(\tau, j, t_{ij}) > 0\} \quad \forall j \in [d]$;
    $\mathcal{Z} = \left\{j \in [d] : \Delta\text{FP}(\tau, j, t_{i_j^{'},j}) = 0\right\}$;
    if $|\mathcal{Z}| > 0$ then
        $j^* = \text{argmax}_{j \in \mathcal{Z}} \left\{\Delta\text{TP}(\tau, j, t_{i_j^{'},j})\right\}$;
    else
        $j^* = \text{argmin}_{j \in [d]} \left\{\Delta\text{FP}(\tau, j, t_{i_j^{'},j}) / \Delta\text{TP}(\tau, j, t_{i_j^{'},j})\right\}$;
    end
    $\text{TP} = \text{TP} + \Delta\text{TP}(\tau, j^*, t_{i_{j^*}^{'},j^*})$;
    $\tau_j = t_{i_{j^*}^{'},j^*}$;
end
return $\tau$
```

4.4 Results

This section compares the R-RF models to several standard classification model benchmarks as well as alternative two-stage model formulations. The R-RF model is compared to the benchmarks by measuring test set FOR, manual review rate, and precision on 100 outer cross validation splits on the 2019-2020 and 2020-2021 datasets.

4.4.1 Benchmark Models

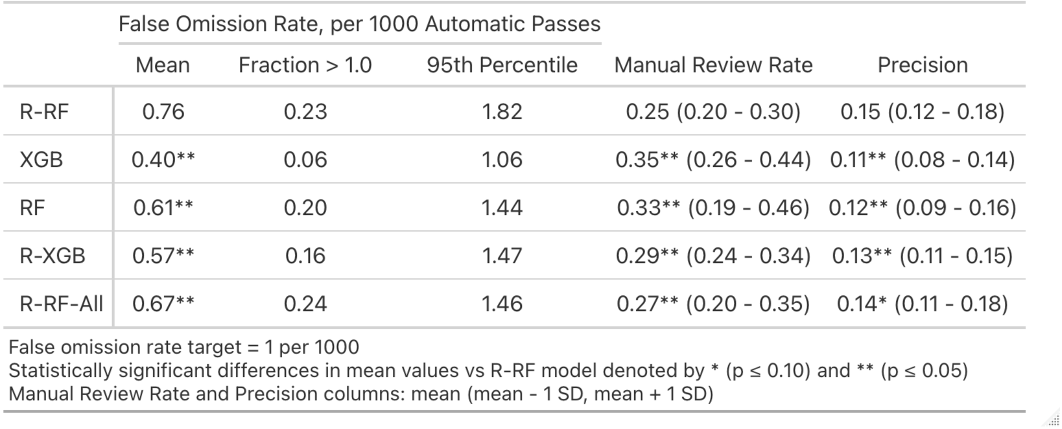
The performance of the R-RF model is compared to two state-of-the-art existing tree-based classification algorithms, a gradient boosted decision tree [8] (denoted XGB in the results) and a random forest (denoted RF). These benchmarks are used to determine whether the custom rules-based classifier provides better performance than existing classifiers on the easy pass predictions, and whether existing single stage classifiers are able to learn effective decision rules for both easy and difficult passing modules.

Both of these algorithms were trained using the same nested cross validation procedure described in Section 4.3.1. In addition to tuning the decision threshold as a hyperparameter, the number of estimators was optimized as well as the maximum delta step for the gradient boosted tree and the maximum tree depth for the random forest. The grid search over these additional parameters is coarse, with a choice of 5 or 7 for the maximum tree depth, a choice of 100 or 125 for the number of estimators, and a choice of 0 or 1 for the maximum delta step.

The benchmark analysis also included two alternative two-stage classifier models. The first is a version of the R-RF that replaces the random forest with a gradient boosted tree, denoted as R-XGB. This benchmark enables comparison of the performance of the Stage 2 random forest against an alternative Stage 2 algorithm. The second benchmark, denoted R-RF-All, uses the same Stage 1 and Stage 2 classifier models as R-RF, but R-RF-All changes the training procedure for the random forest to include all training data, rather than just the training runs not passed by Stage 1. The hyperparameter grids for the maximum delta step, number of estimators, and maximum tree depth are the same as the grids described above for the RF and XGB models. The difference between R-RF-All and R-RF isolates the impact of the decision to remove the obvious passing modules from the RF training set.

4.4.2 Prediction Performance

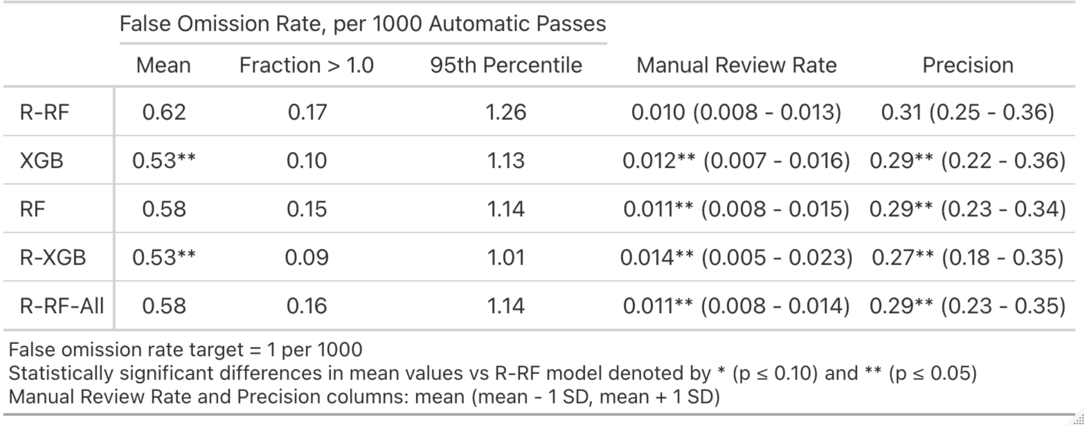
Figure 4-6 shows the predictive performance for the R-RF model and the four benchmarks described above for the 2019-2020 and 2020-2021 datasets using a FOR target of 1 per 1000. Overall, the R-RF model is able to reduce the manual review burden of the operator by, on average, 75% in the 2019-2020 data and 99% in the 2020-2021 data while achieving the FOR target in the average case. When comparing the R-RF model to the benchmarks, the results show a tradeoff between the ability of a particular model to control the FOR below the target and the fraction of modules that require manual review. In both datasets, the R-RF model offers a statistically significant ($p < 0.05$) lower manual review rate compared to the benchmark models by providing higher precision, but the R-RF model carries greater risk of a violation of the target FOR rate. The benchmark models, in particular the XGB model, are able to offer better control over the FOR in exchange for the higher manual review rate (lower precision).



	False Omission Rate, per 1000 Automatic Passes				
	Mean	Fraction > 1.0	95th Percentile	Manual Review Rate	Precision
R-RF	0.76	0.23	1.82	0.25 (0.20 - 0.30)	0.15 (0.12 - 0.18)
XGB	0.40**	0.06	1.06	0.35** (0.26 - 0.44)	0.11** (0.08 - 0.14)
RF	0.61**	0.20	1.44	0.33** (0.19 - 0.46)	0.12** (0.09 - 0.16)
R-XGB	0.57**	0.16	1.47	0.29** (0.24 - 0.34)	0.13** (0.11 - 0.15)
R-RF-All	0.67**	0.24	1.46	0.27** (0.20 - 0.35)	0.14* (0.11 - 0.18)

False omission rate target = 1 per 1000
Statistically significant differences in mean values vs R-RF model denoted by * (p ≤ 0.10) and ** (p ≤ 0.05)
Manual Review Rate and Precision columns: mean (mean - 1 SD, mean + 1 SD)

(a) 2019-2020 Dataset



	False Omission Rate, per 1000 Automatic Passes				
	Mean	Fraction > 1.0	95th Percentile	Manual Review Rate	Precision
R-RF	0.62	0.17	1.26	0.010 (0.008 - 0.013)	0.31 (0.25 - 0.36)
XGB	0.53**	0.10	1.13	0.012** (0.007 - 0.016)	0.29** (0.22 - 0.36)
RF	0.58	0.15	1.14	0.011** (0.008 - 0.015)	0.29** (0.23 - 0.34)
R-XGB	0.53**	0.09	1.01	0.014** (0.005 - 0.023)	0.27** (0.18 - 0.35)
R-RF-All	0.58	0.16	1.14	0.011** (0.008 - 0.014)	0.29** (0.23 - 0.35)

False omission rate target = 1 per 1000
Statistically significant differences in mean values vs R-RF model denoted by * (p ≤ 0.10) and ** (p ≤ 0.05)
Manual Review Rate and Precision columns: mean (mean - 1 SD, mean + 1 SD)

(b) 2020-2021 Dataset

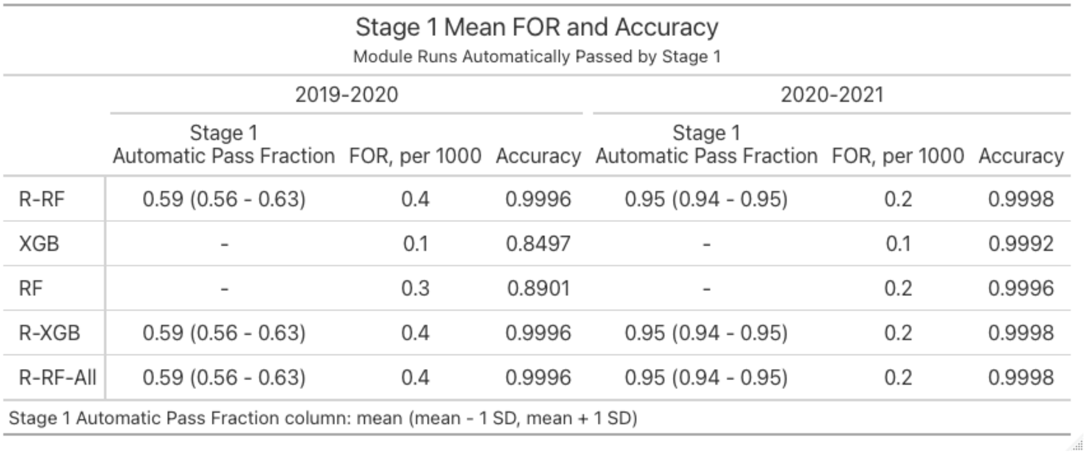
Figure 4-6: Prediction Results

The tradeoff between FOR control and the manual review rate is particularly relevant in the 2019-2020 dataset, where the fraction of modules labeled Fail by the operator is higher by a factor of ten. The R-RF model flags 25% of the module runs for manual review versus 35% for the XGB model, but the 95th percentile of the FOR rate is 1.82 modules per 1000 for the R-RF model and only 1.06 for XGB. Both of the alternative two-stage models, R-XGB and R-RF-All, offer manual review rates and FORs between R-RF and XGB. The RF model also falls between R-RF and XGB, but the R-XGB model has a lower FOR and lower manual review rate.

In the 2020-2021 dataset, the tradeoff between the R-RF and XGB models still exists, but the practical difference in the performance of these models is very small. All of the models are able to automatically label 98% of the dataset or more on average, and the difference in average manual review rates across models is less than 0.5%, though the R-RF model still exhibits the lowest manual review rate. The difference in average FOR between the models is also smaller, though the 95th percentile FOR is highest for the R-RF model. The results suggest that the pass/fail decision for nearly all modules in the 2020-2021 dataset can be made accurately using a simple set of thresholds, since 95% of 2020-2021 modules are passed automatically in Stage 1 of the R-RF model compared to 59% of 2019-2020 modules.

Figure 4-7 illustrates why the R-RF model is able to achieve lower manual review rates while sacrificing some FOR control. In Figure 4-7a, model performance is shown only for the module runs marked Automatic Pass by the Stage 1 rules classifier. The rules classifier used by R-RF, R-RF-All, and R-XGB has a slightly higher FOR rate on these modules than XGB or RF, but the overall accuracy of the rules classifier is much higher on the 2019-2020 dataset. This means that the rules created by the Stage 1 classifier are able to automatically pass many modules that are flagged for manual review according to the XGB and RF models. This result is directionally the same for the 2020-2021 dataset, but the accuracy difference is minuscule in comparison (0.02 - 0.04%).

Figure 4-7b shows the FOR and precision for the modules classified by Stage 2 of the R-RF model. Overall, the same tradeoff exists between FOR and precision for this subset of modules. In particular, note that the R-RF practice of training the Stage 2 random forest on a restricted dataset does not offer strictly better performance than training on the full dataset (R-RF-All). The Stage 2 precision of R-RF is higher than R-RF-All, but this higher precision comes at the cost of an increase in the FOR.



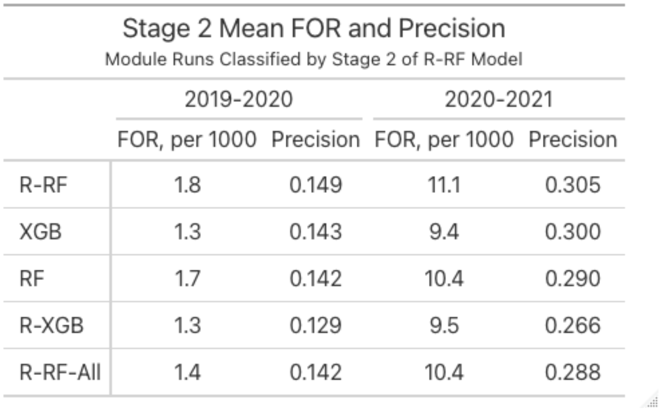
Stage 1 Mean FOR and Accuracy

Module Runs Automatically Passed by Stage 1

	2019-2020			2020-2021		
	Stage 1 Automatic Pass Fraction	FOR, per 1000	Accuracy	Stage 1 Automatic Pass Fraction	FOR, per 1000	Accuracy
R-RF	0.59 (0.56 - 0.63)	0.4	0.9996	0.95 (0.94 - 0.95)	0.2	0.9998
XGB	-	0.1	0.8497	-	0.1	0.9992
RF	-	0.3	0.8901	-	0.2	0.9996
R-XGB	0.59 (0.56 - 0.63)	0.4	0.9996	0.95 (0.94 - 0.95)	0.2	0.9998
R-RF-All	0.59 (0.56 - 0.63)	0.4	0.9996	0.95 (0.94 - 0.95)	0.2	0.9998

Stage 1 Automatic Pass Fraction column: mean (mean - 1 SD, mean + 1 SD)

(a) Prediction Results on Modules Classified by R-RF Stage 1



Stage 2 Mean FOR and Precision

Module Runs Classified by Stage 2 of R-RF Model

	2019-2020		2020-2021	
	FOR, per 1000	Precision	FOR, per 1000	Precision
R-RF	1.8	0.149	11.1	0.305
XGB	1.3	0.143	9.4	0.300
RF	1.7	0.142	10.4	0.290
R-XGB	1.3	0.129	9.5	0.266
R-RF-All	1.4	0.142	10.4	0.288

(b) Prediction Results on Modules Classified by R-RF Stage 2

Figure 4-7: Prediction Results

These results demonstrate that the R-RF model is successful at meaningfully reducing the manual review burden for the MBI test at the cost of introducing a small degree of additional error in its automated pass decisions. In relative terms, the R-RF model may be preferred to the benchmark models in binary classification scenarios when two criteria are met. Namely, (1) a subset of the data distribution defined by a single threshold on each feature can be classified into one category with high accuracy, and (2) the remainder of the data distribution requires a more complex decision rule to accurately distinguish between the classes. If condition 1 is not met, then the Stage 1 rules classifier would not perform well, and if condition 2 is not met, then, as is evident in the 2020-2021 dataset, a single standard tree-based classifier is able to learn an accurate decision rule for the entire dataset. If both conditions are met, then the R-RF model can offer substantial performance benefits for a user who places a high cost on manual review relative to the cost of a mistakenly passed module.

4.5 Conclusions

This chapter describes a model to automate review of sensor data from the MBI test, a key quality control test in an optical transceiver manufacturing process where the pass/fail decision is currently made via manual review of the sensor data. Specifically, the chapter describes a two-stage classification model, R-RF, that reduces the fraction of manually reviewed test results by 75% on one test dataset and by 99% on a second while automatically passing approximately 1 defective module per 1000 automatic passes. The experimental results illustrate that the R-RF model achieves a superior reduction in the need for manual test review by taking on a slightly increased risk of passing a defective module relative to existing benchmarks and alternative two-stage model formulations.

The R-RF model is a demonstration of a quality test automation method that integrates state-of-the-art general machine learning algorithms with subject matter expertise. The success of the two-stage model illustrates that despite the power of general purpose machine learning tools, informing an automation approach with the guidance of human experts can lead to superior problem-specific algorithms.

Appendix A

Appendix for Chapter 2

A.1 Initialization for HMM Parameter Estimation

While EM estimation approaches are widely used for parameter estimation in mixture and HMM models, these approaches are sensitive to parameter initialization because they are only guaranteed to find estimates that locally maximize the likelihood function. This issue is addressed by employing a standard technique of running the EM algorithms with multiple random initializations.

In the use case results, the binomial parameters for the state-specific MB distributions are initialized uniformly at random in the range $[0, 0.02]$. The transition probabilities are initialized with each element on the diagonal of the transition matrix equal to 0.6 and the rest of the probability mass for each row distributed evenly.

A.2 Description of the BIC

The form of the BIC is motivated by the notion that finding the best-fitting HMM structure for the observed data is equivalent to maximizing the likelihood of the data given the HMM hyperparameters S and C, $P(\mathbf{a}|S, C)$, over all possible combinations of S and C. The BIC approximates this likelihood, which is not observable, using the maximum likelihood parameter estimates. The approximation consists of two terms, a negative term that depends on the likelihood of the model evaluated at the maximum likelihood parameter values, and a positive complexity term that penalizes the number of estimated parameters (i.e., hidden states and mixture components) in the model. Therefore, a lower BIC value indicates increased plausibility of the model after accounting for model complexity.

A.3 Method Validation Simulation Results

A.3.1 One-State, Two-Component Generating HMMs

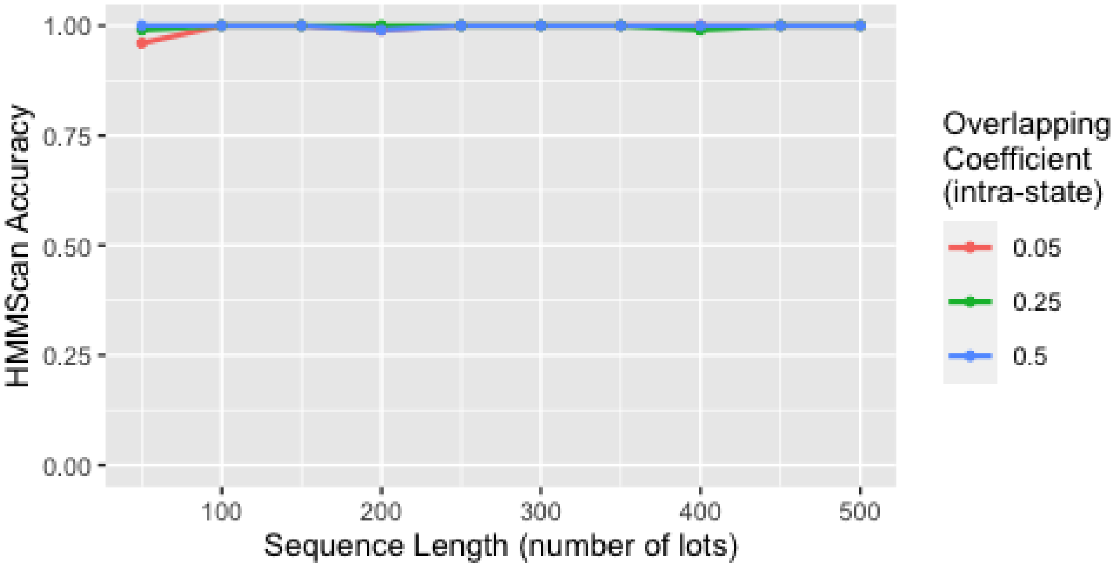


Figure A-1: Method validation simulation results for HMMs with one state and two mixture components. Each point represents the HMMScan detection accuracy calculated on 100 sample sequences with the same length as denoted on the x-axis. The probability for the first binomial mixture component is set to 0.01, and the probability for the second component is set to the value greater than 0.01 that induces the desired OVL value. The two mixture components are equally weighted for every instance.

	Binomial Probability	
OVL	Component 1	Component 2
0.05	0.01	0.026
0.25	0.01	0.019
0.50	0.01	0.015

Table A.1: One-state generating HMM state-specific binomial probabilities by OVL value.

A.3.2 Two-State, One-Component Generating HMMs

Table A.2 provides the transition matrices used to generate the model validation simulation results found in the body of the paper. The binomial probabilities for the single component state-specific distributions are the same as the probabilities presented in Table A.1.

		State 1 Stat. Prob. = 0.90			**State 1 Stat. Prob. = 0.75**			**State 1 Stat. Prob. = 0.50**		
		Transition Probabilities			**Transition Probabilities**			**Transition Probabilities**		
State 2 Mean Sojourn Time	**Hidden State**	**To: State 1**	**To: State 2**	**Stat. Probs**	**To: State 1**	**To: State 2**	**Stat. Probs**	**To: State 1**	**To: State 2**	**Stat. Probs.**
1.25	1	0.91	0.09	0.90	0.73	0.27	0.75	0.20	0.80	0.50
	2	0.80	0.20	0.10	0.80	0.20	0.25	0.80	0.20	0.50
2.00	1	0.94	0.06	0.90	0.83	0.17	0.75	0.50	0.50	0.50
	2	0.50	0.50	0.10	0.50	0.50	0.25	0.50	0.50	0.50
4.00	1	0.97	0.03	0.90	0.92	0.08	0.75	0.75	0.25	0.50
	2	0.25	0.75	0.10	0.25	0.75	0.25	0.25	0.75	0.50
10.00	1	0.99	0.01	0.90	0.97	0.03	0.75	0.90	0.10	0.50
	2	0.10	0.90	0.10	0.10	0.90	0.25	0.10	0.90	0.50
25.00	1	1.00	0.00	0.90	0.99	0.01	0.75	0.96	0.04	0.50
	2	0.04	0.96	0.10	0.04	0.96	0.25	0.04	0.96	0.50

Table A.2: Two-state, one-component generating HMM transition matrices.

Figure A-2 shows the state prediction accuracy results for the two-state, one-component simulations.

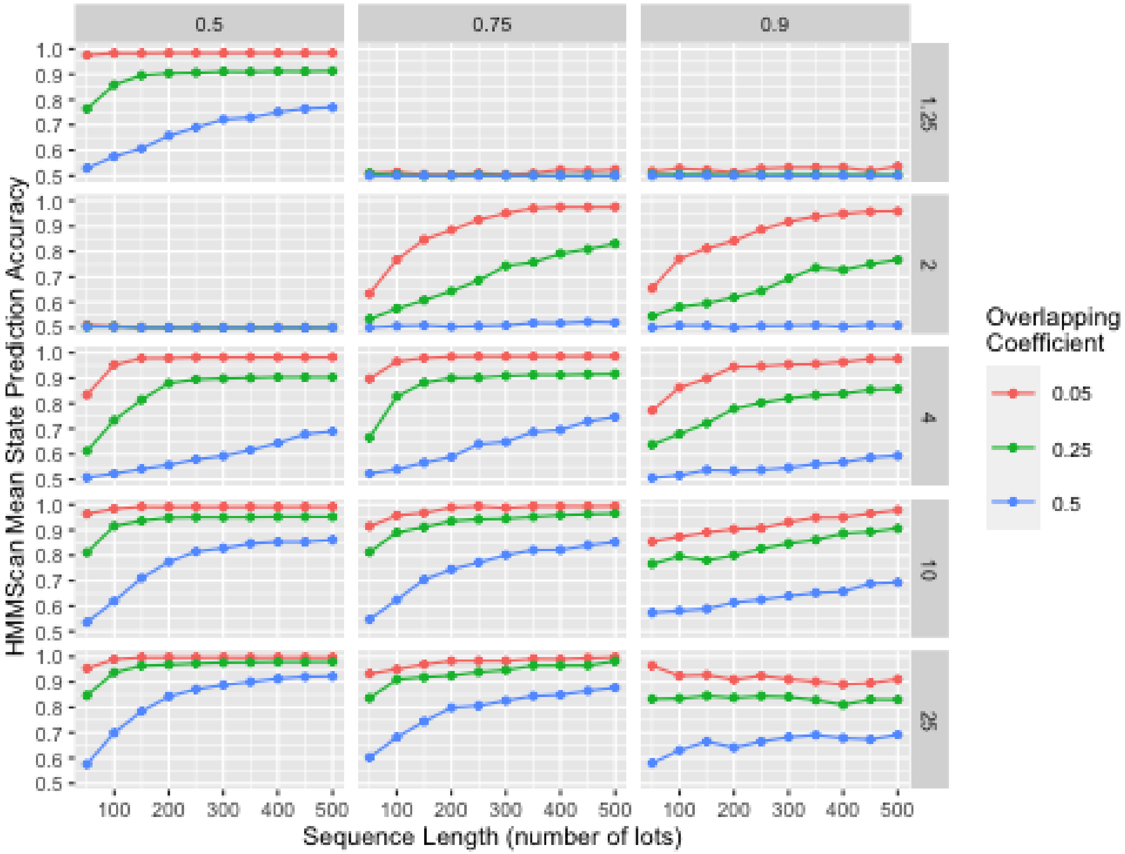


Figure A-2: Method validation simulation results for HMMs with two states and one mixture component. Each point represents the HMMScan mean state prediction accuracy calculated on 100 sample sequences with the same length as denoted on the x-axis. The panels are organized in columns based on the low-risk state stationary probability and in rows by the mean high-risk state sojourn length.

A.3.3 Two-State, Two-Component Generating HMMs

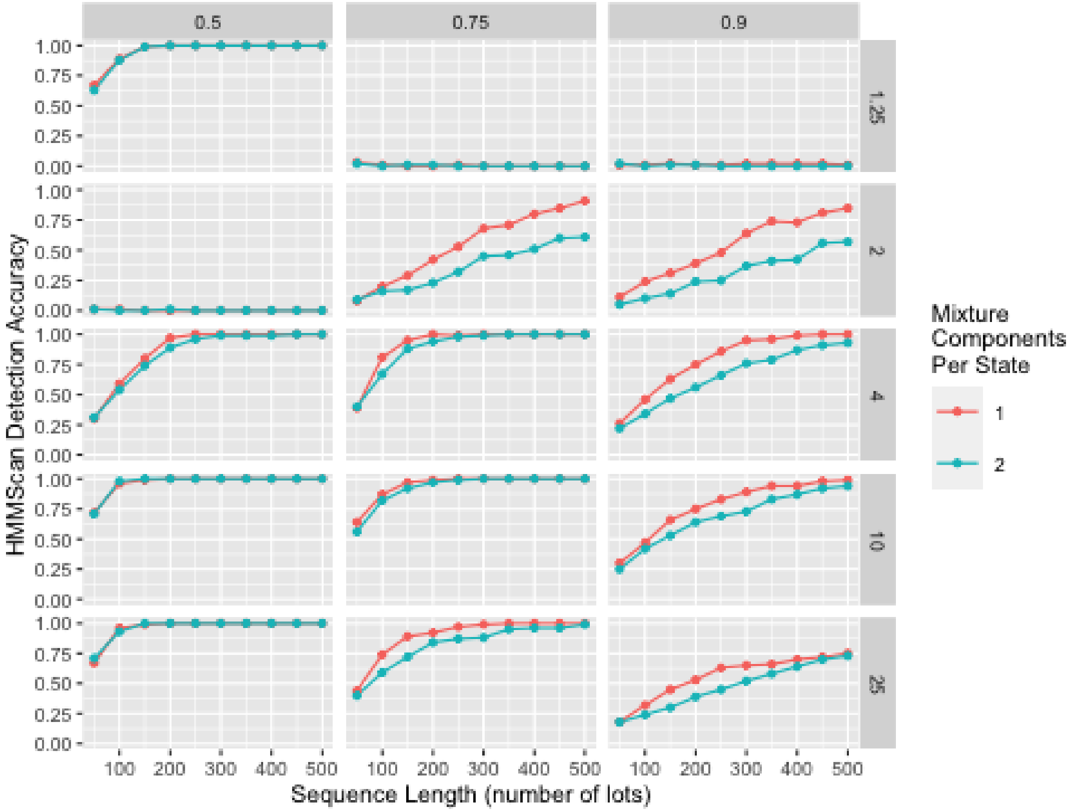


Figure A-3: Simulation results comparing the two-state, two-component generating HMMs with two-state, one-component HMMs. One-component results are identical to the results presenting in the body of the paper with OVL set to 0.25. The binomial probabilities for the two-component HMMs are set such that the OVL value between the states is 0.25 and the OVL value between the mixture components within each state is set to 0.50. The panels are organized in columns based on the low-risk state stationary probability and in rows by the mean high-risk state sojourn length. The results show that when the stationary probabilities are evenly balanced, the one-component and two-component results are very similar. However, when the high-risk state is observed less frequently, the additional variance in the state-specific distributions makes detecting the multiple-state nature of the generating distribution more difficult.

A.3.4 Three-State, One-Component Generating HMMs

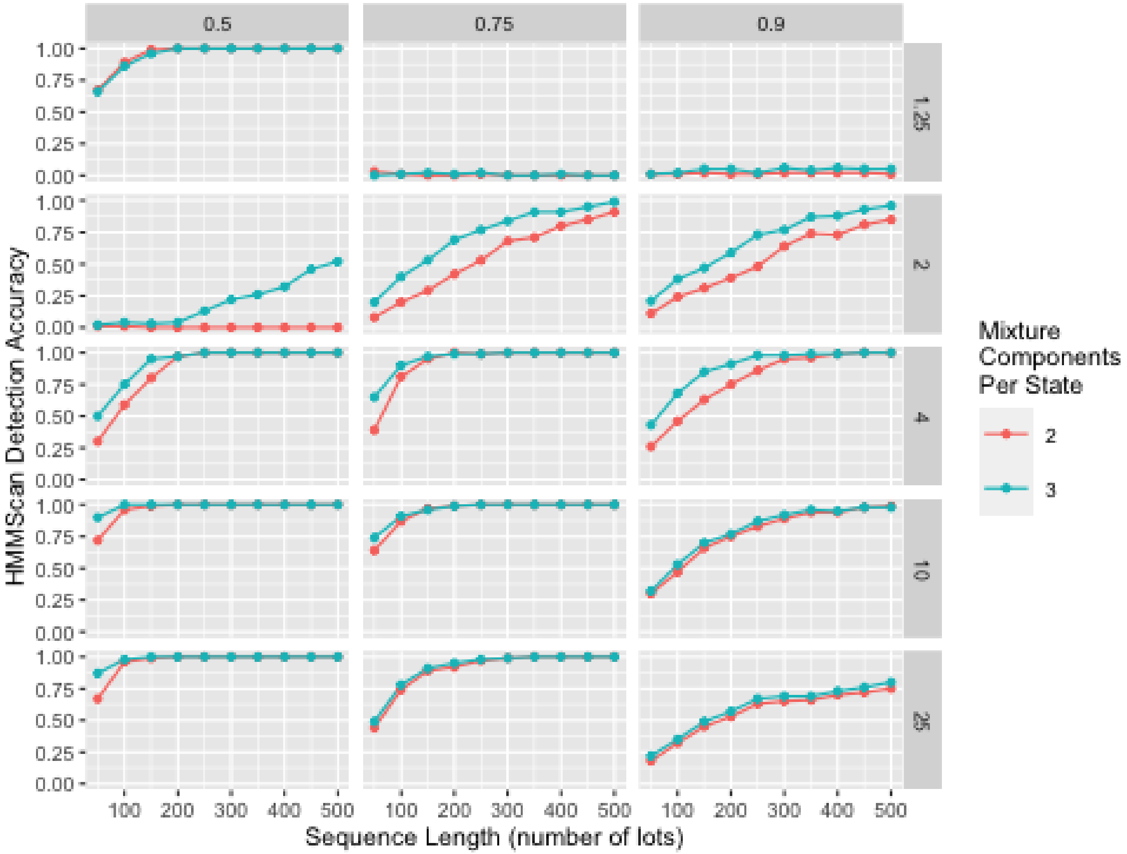


Figure A-4: Simulation results comparing three-state, one-component HMMs and two-state, one-component HMMs. The three-state transition matrices are created by splitting the high-risk state into two states with similar properties. The OVL value between state-specific distributions is 0.25 for all models shown. The panels are organized in columns based on the low-risk state stationary probability and in rows by the mean high-risk state sojourn length. For the three-state models, both the medium- and high-risk states are assigned the same mean sojourn length. The results indicate that at lower sequence lengths and shorter high-risk sojourns, the HMMScan accuracy is higher for the three-state models because there is more separation between the highest-risk state and the lowest-risk state.

		State 1 Stat. Prob. = 0.90				**State 1 Stat. Prob. = 0.75**				**State 1 Stat. Prob. = 0.50**			
		Transition Probabilities				**Transition Probabilities**				**Transition Probabilities**			
States 2+3 Mean Sojourn Time	**Hidden State**	**To: State 1**	**To: State 2**	**To: State 3**	**Stat. Probs**	**To: State 1**	**To: State 2**	**To: State 3**	**Stat. Probs**	**To: State 1**	**To: State 2**	**To: State 3**	**Stat. Probs.**
1.25	1	0.91	0.05	0.05	0.90	0.73	0.14	0.14	0.75	0.20	0.40	0.40	0.50
	2	0.80	0.19	0.01	0.05	0.80	0.19	0.01	0.125	0.80	0.19	0.01	0.25
	3	0.80	0.01	0.19	0.05	0.80	0.01	0.19	0.125	0.80	0.01	0.19	0.25
2.00	1	0.94	0.03	0.03	0.90	0.83	0.08	0.08	0.75	0.50	0.25	0.25	0.50
	2	0.50	0.48	0.03	0.05	0.50	0.48	0.03	0.125	0.50	0.48	0.03	0.25
	3	0.50	0.03	0.48	0.05	0.50	0.03	0.48	0.125	0.50	0.03	0.48	0.25
4.00	1	0.97	0.01	0.01	0.90	0.92	0.04	0.04	0.75	0.75	0.13	0.13	0.50
	2	0.25	0.71	0.04	0.05	0.25	0.71	0.04	0.125	0.25	0.71	0.04	0.25
	3	0.25	0.04	0.71	0.05	0.25	0.04	0.71	0.125	0.25	0.04	0.71	0.25
10.00	1	0.99	0.01	0.01	0.90	0.97	0.02	0.02	0.75	0.90	0.05	0.05	0.50
	2	0.10	0.86	0.05	0.05	0.10	0.86	0.05	0.125	0.10	0.86	0.05	0.25
	3	0.10	0.05	0.86	0.05	0.10	0.05	0.86	0.125	0.10	0.05	0.86	0.25
25.00	1	0.996	0.002	0.002	0.90	0.987	0.007	0.007	0.75	0.96	0.02	0.02	0.50
	2	0.04	0.91	0.05	0.05	0.04	0.91	0.05	0.125	0.04	0.91	0.05	0.25
	3	0.04	0.05	0.91	0.05	0.04	0.05	0.91	0.125	0.04	0.05	0.91	0.25

Table A.3: Three-state, one-component generating HMM transition matrices for the "split high-risk state" scenario.

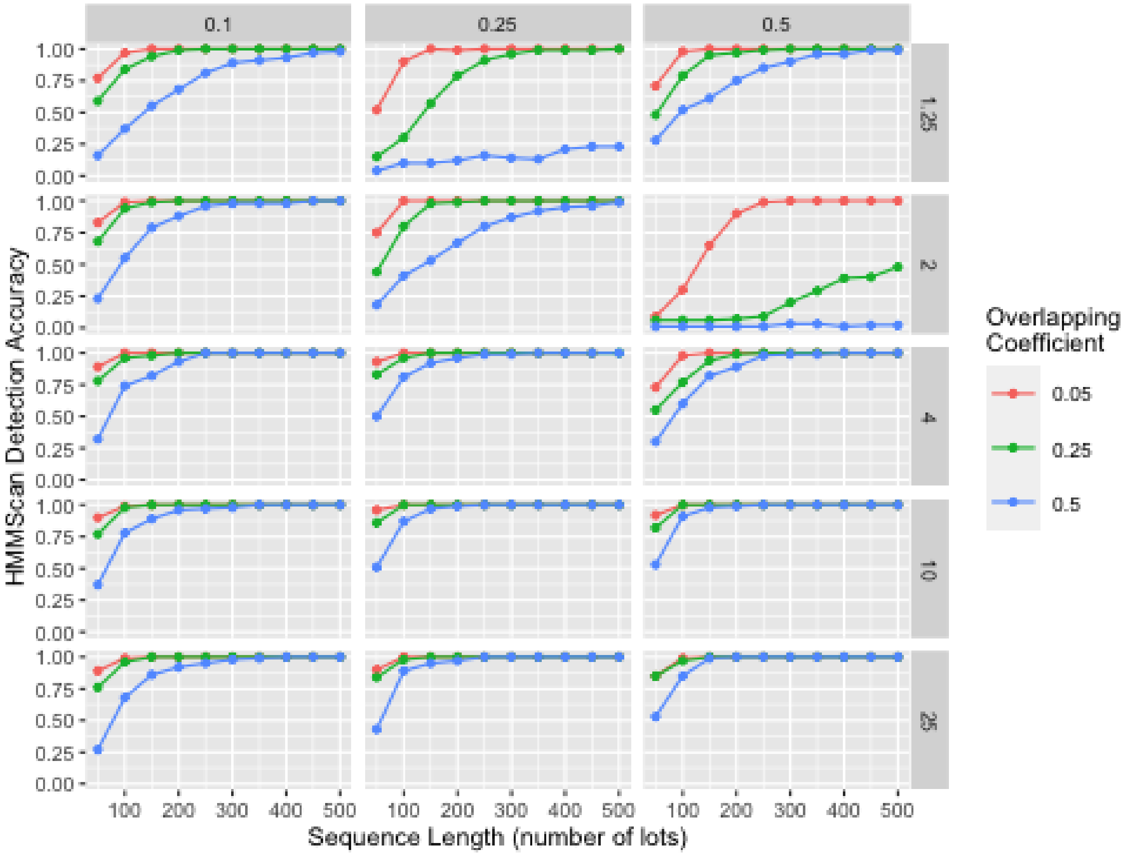


Figure A-5: Simulation results for three-state, one-component generating HMMs corresponding to the "split low-risk state" scenario. The transition matrices for these HMMs were generated by splitting the low-risk state from the two-state generating HMMs found in Table E2. The panels are organized in columns based on the low-risk state stationary probability and in rows by the mean highest-risk state sojourn length. In this case, the stationary probability for states 1 and 2 sum to the inputted low-risk stationary probability. The equal stationary probabilities between states 1 and 2 contributes to the high detection probabilities seen in this figure and reinforces the conclusion that HMMScan has high accuracy on multiple-state generating models when at least two of these states have reasonably high long-term frequencies. Note that for the (2, 0.25) and (2, 0.5) panels all three states have similar probabilities of returning to the highest-risk state. The similarity in the transition matrix rows makes the detection problem generally more difficult.

		State 3 Stat. Prob. = 0.10				State 3 Stat. Prob. = 0.25				State 3 Stat. Prob. = 0.50			
		Transition Probabilities				Transition Probabilities				Transition Probabilities			
State 3 Mean Sojourn Time	Hidden State	To: State 1	To: State 2	To: State 3	Stat. Probs	To: State 1	To: State 2	To: State 3	Stat. Probs	To: State 1	To: State 2	To: State 3	Stat. Probs.
1.25	1	0.87	0.05	0.09	0.45	0.70	0.04	0.27	0.375	0.19	0.01	0.80	0.25
	2	0.05	0.87	0.09	0.45	0.04	0.70	0.27	0.375	0.01	0.19	0.80	0.25
	3	0.40	0.40	0.20	0.10	0.40	0.40	0.20	0.250	0.40	0.40	0.20	0.50
2.00	1	0.90	0.05	0.06	0.45	0.79	0.04	0.17	0.375	0.48	0.03	0.50	0.25
	2	0.05	0.90	0.06	0.45	0.04	0.79	0.17	0.375	0.03	0.48	0.50	0.25
	3	0.25	0.25	0.50	0.10	0.25	0.25	0.50	0.250	0.25	0.25	0.50	0.50
4.00	1	0.92	0.05	0.03	0.45	0.87	0.05	0.08	0.375	0.71	0.04	0.25	0.25
	2	0.05	0.92	0.03	0.45	0.05	0.87	0.08	0.375	0.04	0.71	0.25	0.25
	3	0.13	0.13	0.75	0.10	0.13	0.13	0.75	0.250	0.13	0.13	0.75	0.50
10.00	1	0.94	0.05	0.01	0.45	0.92	0.05	0.03	0.375	0.86	0.05	0.10	0.25
	2	0.05	0.94	0.01	0.45	0.05	0.92	0.03	0.375	0.05	0.86	0.10	0.25
	3	0.05	0.05	0.90	0.10	0.05	0.05	0.90	0.250	0.05	0.05	0.90	0.50
25.00	1	0.95	0.05	0.004	0.45	0.94	0.05	0.01	0.375	0.91	0.05	0.04	0.25
	2	0.05	0.95	0.004	0.45	0.05	0.94	0.01	0.375	0.05	0.91	0.04	0.25
	3	0.02	0.02	0.96	0.10	0.02	0.02	0.96	0.250	0.02	0.02	0.96	0.50

Table A.4: Three-state, one-component generating HMM transition matrices for the "split low-risk state" scenario.

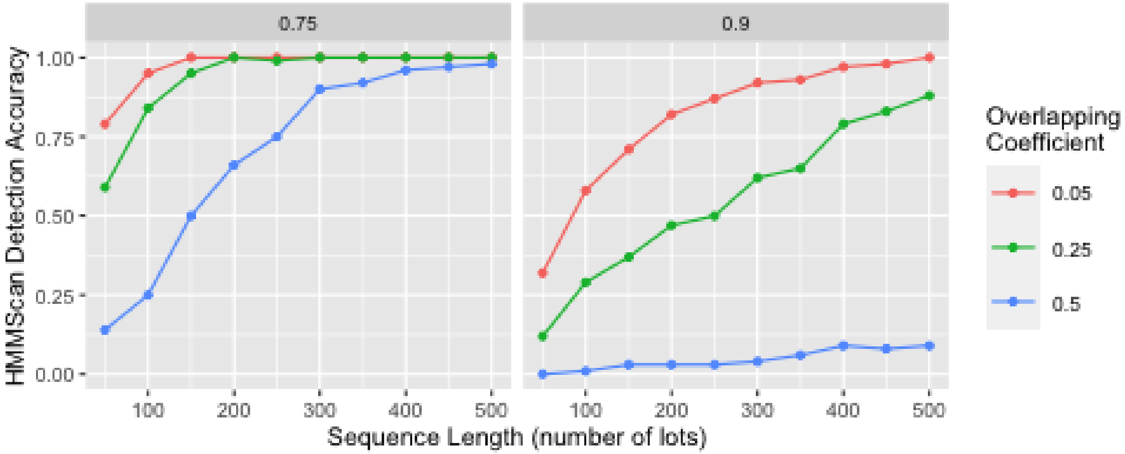


Figure A-6: Simulation results for three-state, one-component generating HMMs corresponding to the "graduated response" scenario. In this scenario, the highest-risk state is infrequent (low stationary probability) and not very persistent, and it has a high probability of returning to the lowest-risk state. The panels are organized in columns based on the lowest-risk state stationary probability. This transition matrix structure corresponds to a practical scenario where the lowest-risk state is most common and there is an increasing urgency of remedying problems as the number of AEs increases. The results show that HMMScan has a very high probability of accuracy multiple-state detection (>96% for OVL = 0.25) at lot sizes greater than 150 lots if the highest-risk state is prevalent enough.

	State 1 Stat. Prob. = 0.90				**State 1 Stat. Prob. = 0.75**			
	Transition Probabilities				**Transition Probabilities**			
Hidden State	**To State: 1**	**To State: 2**	**To State: 3**	**Stat. Prob.**	**To State: 1**	**To State: 2**	**To State: 3**	**Stat. Prob.**
1	0.95	0.05	0.01	0.90	0.85	0.14	0.02	0.750
2	0.40	0.55	0.05	0.09	0.41	0.55	0.04	0.225
3	0.90	0.01	0.10	0.01	0.80	0.01	0.19	0.025

Table A.5: Three-state, one-component generating HMM transition matrices for the "graduated response" scenario.

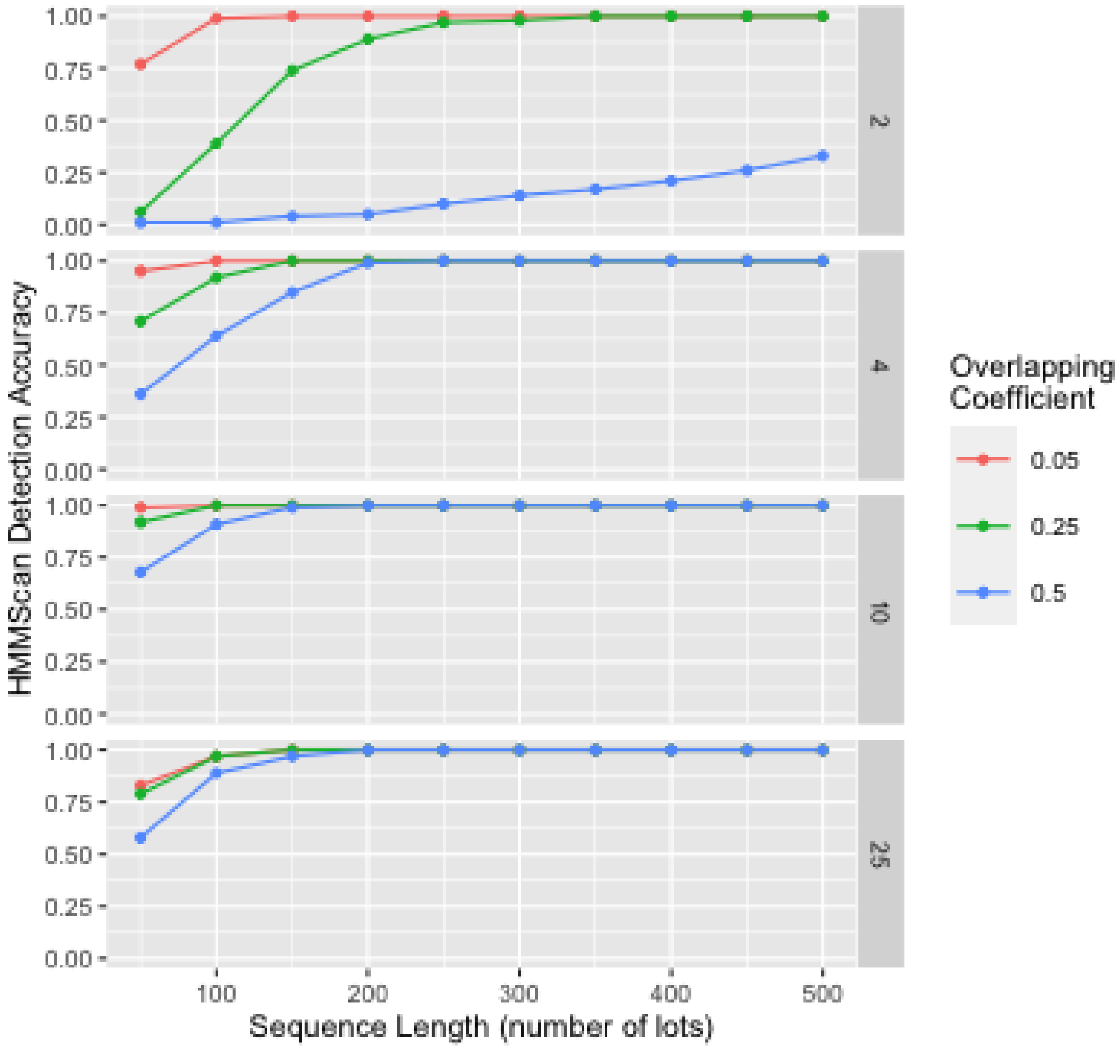


Figure A-7: Simulation results for three-state, one-component generating HMMs corresponding to the "equal stationary probabilities" scenario. In this scenario, the stationary probabilities and mean sojourn times of the states are all equal, and the transition probabilities are set such that the following transition path is most likely: low-risk, medium-risk, high-risk, low-risk. The panels are organized in rows based on the mean hidden state sojourn length. This transition matrix structure corresponds to a practical scenario where the lowest-risk state is most common and the urgency to remedy manufacturing issues increases as the number of observed AEs increases. The results show that HMMScan has a very high probability of accuracy multiple-state detection (>96% for OVL =0.25) at lot sizes greater than 150 lots if the mean high-risk sojourn is not too short.

		Transition Probabilities			
Mean Sojourn Time	**Hidden State**	**To State: 1**	**To State: 2**	**To State: 3**	**Stat. Prob.**
2.00	1	0.50	0.48	0.03	0.33
	2	0.02	0.50	0.48	0.33
	3	0.48	0.03	0.50	0.33
4.00	1	0.75	0.24	0.01	0.33
	2	0.01	0.75	0.24	0.33
	3	0.24	0.01	0.75	0.33
10.00	1	0.90	0.10	0.01	0.33
	2	0.005	0.900	0.095	0.33
	3	0.095	0.005	0.900	0.33
25.00	1	0.96	0.04	0.002	0.33
	2	0.002	0.960	0.038	0.33
	3	0.038	0.002	0.960	0.33

Table A.6: Three-state, one-component generating HMM transition matrices for the "equal stationary probabilities" scenario.

A.3.5 Four-State, One-Component

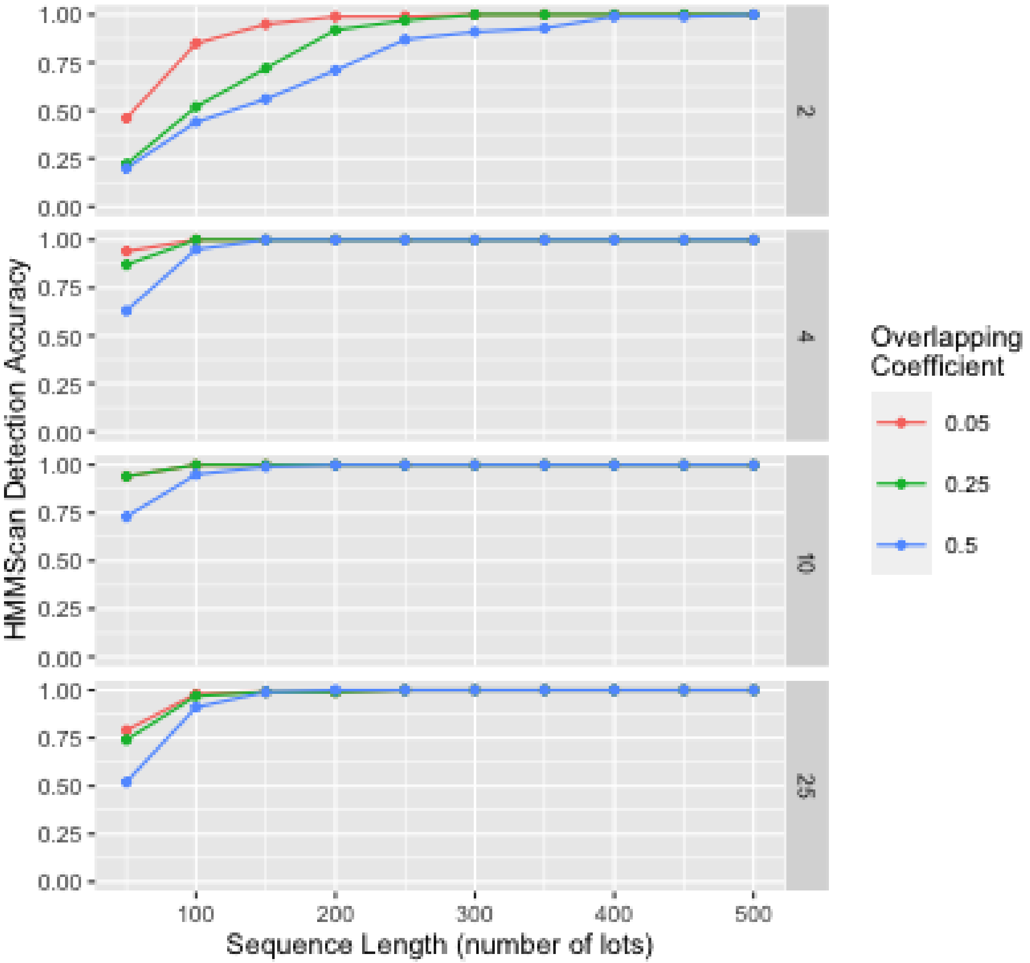
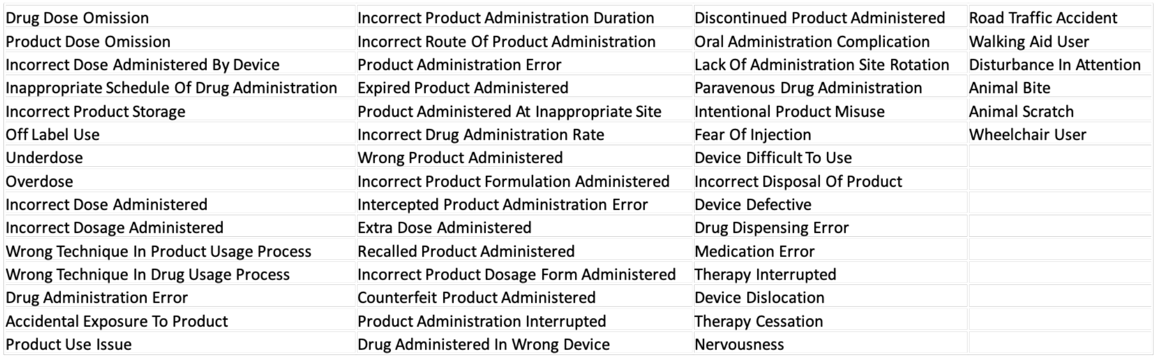


Figure A-8: Simulation results for four-state, one-component generating HMMs corresponding to the “equal stationary probabilities” scenario. In this scenario, the stationary probabilities and mean sojourn times of the states are all equal. Besides self-transitions, transitions to the states with the next lowest AE risk and the next highest risk are most likely. The panels are organized in rows based on the mean hidden state sojourn length. The results show that the detection accuracy is very high with a long enough sojourn period even at high overlap with sequences of 100 lots or longer, particularly at medium high-risk sojourn lengths.

		Transition Probabilities				
Mean Sojourn Time	**Hidden State**	**To State: 1**	**To State: 2**	**To State: 3**	**To State: 4**	**Stat. Prob.**
2.00	1	0.50	0.23	0.05	0.23	0.25
	2	0.23	0.50	0.23	0.05	0.25
	3	0.05	0.23	0.50	0.23	0.25
	4	0.23	0.05	0.23	0.50	0.25
4.00	1	0.75	0.11	0.02	0.11	0.25
	2	0.11	0.75	0.11	0.02	0.25
	3	0.02	0.11	0.75	0.11	0.25
	4	0.11	0.02	0.11	0.75	0.25
10.00	1	0.90	0.05	0.01	0.05	0.25
	2	0.05	0.90	0.05	0.01	0.25
	3	0.01	0.05	0.90	0.05	0.25
	4	0.05	0.01	0.05	0.90	0.25
25.00	1	0.96	0.02	0.004	0.02	0.25
	2	0.02	0.96	0.02	0.004	0.25
	3	0.004	0.02	0.96	0.02	0.25
	4	0.02	0.004	0.02	0.96	0.25

Table A.7: Four-state, one-component generating HMM transition matrices for the "equal stationary probabilities" scenario.

A.4 Administrative and Unrelated Reactions



- Drug Dose Omission
- Product Dose Omission
- Incorrect Dose Administered By Device
- Inappropriate Schedule Of Drug Administration
- Incorrect Product Storage
- Off Label Use
- Underdose
- Overdose
- Incorrect Dose Administered
- Incorrect Dosage Administered
- Wrong Technique In Product Usage Process
- Wrong Technique In Drug Usage Process
- Drug Administration Error
- Accidental Exposure To Product
- Product Use Issue
- Incorrect Product Administration Duration
- Incorrect Route Of Product Administration
- Product Administration Error
- Expired Product Administered
- Product Administered At Inappropriate Site
- Incorrect Drug Administration Rate
- Wrong Product Administered
- Incorrect Product Formulation Administered
- Intercepted Product Administration Error
- Extra Dose Administered
- Recalled Product Administered
- Incorrect Product Dosage Form Administered
- Counterfeit Product Administered
- Product Administration Interrupted
- Drug Administered In Wrong Device
- Discontinued Product Administered
- Oral Administration Complication
- Lack Of Administration Site Rotation
- Paravenous Drug Administration
- Intentional Product Misuse
- Fear Of Injection
- Device Difficult To Use
- Incorrect Disposal Of Product
- Device Defective
- Drug Dispensing Error
- Medication Error
- Therapy Interrupted
- Device Dislocation
- Therapy Cessation
- Nervousness
- Road Traffic Accident
- Walking Aid User
- Disturbance In Attention
- Animal Bite
- Animal Scratch
- Wheelchair User

Figure A-9: Administrative and unrelated reactions

A.5 Use Case Dose Form C Best-Fitting HMM Model Parameters (Two-State, Three-Component)

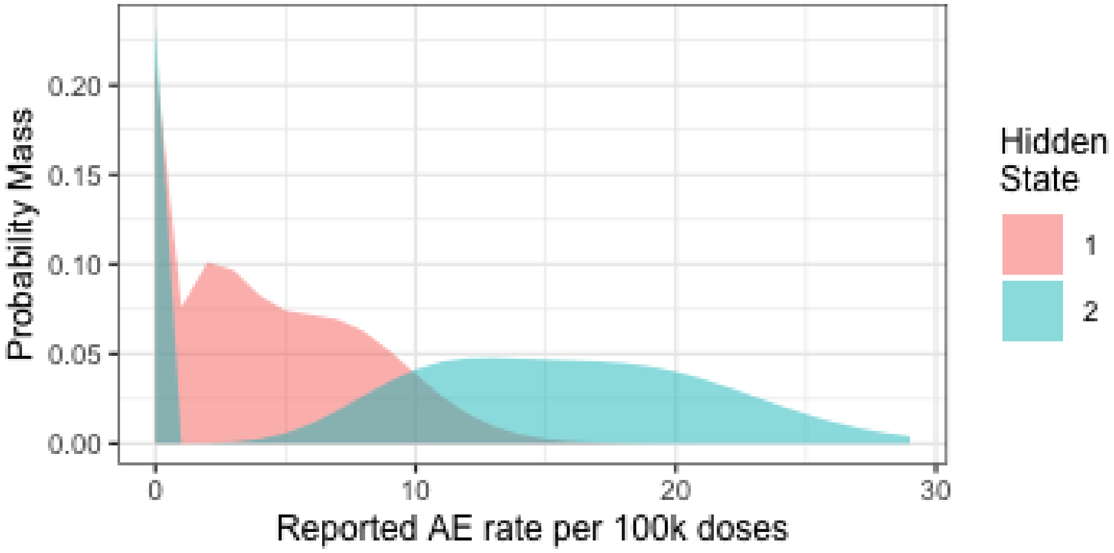


Figure A-10: Fitted state-specific binomial mixture distributions for the best-fitting HMMs for dose form C. Each panel shows the distribution for the state-specific distribution associated with each hidden state.

	Dose Form C			
	Transition Probabilities (from row state to column state)			
Hidden State	To State 1	To State 2	Mean AE Rate	Stat. Prob.
1	0.94	0.06	4.3	0.17
2	0.01	0.99	12.2	0.83

Table A.8: Estimated transition matrix and state-specific mean AE rates for dose form C best-fitting HMM.

A.6 Application of HMMScan to Vaccines

To illustrate the applicability of the HMMScan method to products beyond the use case described in Chapter 2, the method is applied to nine lot sequences across eight vaccine products. The raw AE counts and lot numbers for these applications are taken from the U.S. Vaccine Adverse Events Reporting System (VAERS) database [47]. However, without access to the lot metadata from the manufacturers of these products, crucial information is missing, such as the exhaustive list of all lots packaged during a particular time period and the sizes and packaging dates of these lots. The remainder of this section describes the additional assumptions made to construct sequences of per lot AE rates for each product without these metadata and summarizes the HMMScan results.

A.6.1 Obtaining and Ordering the Lot Sequences

For each product considered in the VAERS analysis, the set of alphanumeric lot numbers that appear in the VAERS dataset between January 1, 2000 and May 14, 2020 were compiled. Within each set of lot numbers, a subset of lot numbers was retained where the lexicographical order of the lot numbers appeared to be strongly related to date when the first AE report associated with each lot was filed. The third column of Figure A-11 shows the Spearman rank correlation between the first AE report dates and the lexicographical order of the lot numbers for each lot sequence. These correlations are very high (0.74 - 0.97), suggesting that lexicographical ordering is a reasonable proxy for packaging date ordering in these sequences. Note that two subsets of lots are compiled for the Prevnar vaccine, one spanning lots with first AE reports between 2000 and 2005, and a second spanning 2008-2010. These two sequences are treated separately because the structure of the lot numbers changes in the interim period, and the intervening lots do not have a strong positive Spearman correlation between lot order and first AE report date.

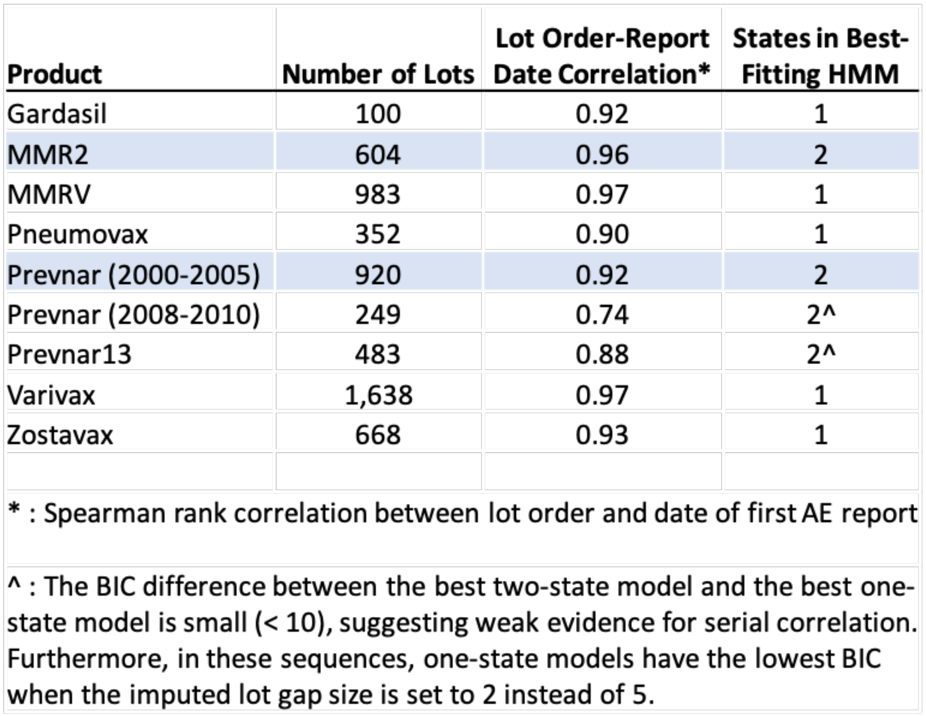
A.6.2 Unobserved Lot Numbers

Once the sequence of observed lot numbers has been established and ordered, the problem of potentially unobserved lots remains. The set of observed lot numbers from the VAERS data represents a censored sample of the full list of all lot numbers that were produced during the time period of interest for each product. An assumption is required to determine which of the lot numbers in each sequence should be associated with valid lots that were packaged and distributed, and which of the lot numbers are invalid, i.e., did not correspond to packaged and distributed lots. The assumption is made that lot numbers contained in small gaps (after lexicographical ordering) between observed lots were valid lot numbers, and therefore should be assigned zero AEs. All lot numbers that are contained in large gaps are assumed to be invalid.

Once the small gaps between observed lots are filled, the result is a collection of ordered lot sequences for each product. This collection of sequences is modeled as a set of independent samples from an HMM, an assumption that enables straightforward parameter estimation via the Baum-Welch algorithm as described in Section 2.3.3.

The HMMScan method is run on each product, using the same set of candidate models as described in Section 2.5.2 ($S_{max} = 4, C_{max} = 9$). The fourth column of Figure A-11 shows the number of hidden states in the best-fitting HMM model for each product. In these results, a gap size of 5 lot numbers or fewer is considered small, though a gap size of 2 lots or fewer is also tested. The gap size assumption does not change the fundamental conclusions of the analysis, except in two cases that are noted in the figure.

Two of the nine lot sequences (MMR2 and the earlier Prevnar sequence) exhibit strong evidence in favor of temporal correlation in AE rates, where the difference in the BIC score between the best-fitting two-state model is greater than 10. Two other lot sequences (the later Prevnar sequence and the Prevnar13 sequence) exhibit weak evidence of temporal correlation in AE rates. In the latter two cases, the BIC difference is less than 10 between the best one-state and the best two-state models. Furthermore, when the gap size assumption is changed from 5 to 2, one-state models provide the best fit for these products.



Product	Number of Lots	Lot Order-Report Date Correlation*	States in Best-Fitting HMM
Gardasil	100	0.92	1
MMR2	604	0.96	2
MMRV	983	0.97	1
Pneumovax	352	0.90	1
Prevnar (2000-2005)	920	0.92	2
Prevnar (2008-2010)	249	0.74	2^
Prevnar13	483	0.88	2^
Varivax	1,638	0.97	1
Zostavax	668	0.93	1

* : Spearman rank correlation between lot order and date of first AE report

^ : The BIC difference between the best two-state model and the best one-state model is small (< 10), suggesting weak evidence for serial correlation. Furthermore, in these sequences, one-state models have the lowest BIC when the imputed lot gap size is set to 2 instead of 5.

Figure A-11: HMMScan results on VAERS data. The lot sequences where the HMMScan method indicates strong evidence in favor of serial correlation in the AE rates are highlighted in blue.

Appendix B

Appendix for Chapter 3

B.1 Pharma Dataset Deviation Descriptions

Each deviation description is the concatenation of four component text fields, the *short description*, the *long description*, the *root cause analysis*, and the *corrective and preventative action description*. These text fields are generated during the life-cycle of a deviation, which can be described as having three primary stages: *Reporting*, *Investigation*, and *Action Determination*.

During the Reporting stage, the deviation is first documented by an operator who notices a problem. The deviation is documented in the short description field, which contains a short (1-2 sentence) summary of the problem that occurred. A more detailed description (1-4 paragraphs) is included in the long description field. The deviation is then turned over to an investigator.

During the Investigation stage, the investigator attempts to understand the severity of the deviation and performs a root cause analysis for serious deviations with potential product impact. During the course of the investigation process, the investigator may add details to the long description. For deviations deemed to have potential product impact, the investigator documents the determined root causes of the deviation in the root cause analysis field.

Finally, during the Action Determination stage, the investigator collaborates with other colleagues to determine any corrective and preventative actions that must be taken in response to the deviation and root cause analysis. These actions are documented in the corrective and preventative action description.

B.2 Pharma Dataset Categories

The Pharma dataset categories are described below with examples of typical associated deviations.

1. **Bioreactor Additions**: The bioreactor process is initiated when inoculum and growth media are added to the bioreactor tank. Additions of feed media and other inputs also occur while the process is running. A departure from procedures regarding the amount or timing of additions is an example of a potential deviation in this category.

2. **Process Monitoring**: While the bioreactor is running and producing the desired product, the internal state of the vessel is monitored in real time by various probes and sensors. These sensors measure conditions such as temperature and pH. Sensor measurements that flag conditions outside of expected ranges constitute the majority of process monitoring deviations.

3. **Sample Processing**: In addition to real time monitoring, samples are taken from the bioreactor at regular intervals. These samples are prepared and analyzed using separate equipment to provide additional detailed information about the composition of the bioreactor contents at a particular time. Any deviations from sample handling protocols or standard operating procedures for the sample processing equipment would be included in this category.

4. **Maintenance**: This category consists of all cleaning and equipment maintenance tasks that are performed on the bioreactor before, during and after production. Incorrect or missed cleaning steps and broken equipment are deviations that would be associated with the maintenance category.

5. **Filter Integrity Testing**: Filter integrity is critical to the proper function of the production bioreactor, so testing occurs regularly before, during and after production. Failure to complete filter integrity testing at the appropriate time is an example of a deviation that would be classified in this category.

B.3 Term Frequency Adjustment

Some relevant terms in the document datasets, such as "user interface", consist of pairs of words (bigrams) with specific meanings that are different from the meanings of the component unigrams. Therefore, both bigrams and unigrams are incorporating into the vocabulary, and a term frequency adjustment procedure is applied such that the term frequencies for the unigrams consist only of the appearances that occur outside of valid bigrams. Consider an illustrative example of the term frequency adjustment with a vocabulary that consists of the following terms: “lives”, “united”, “states”, “united states” and a document that reads "He states that he lives in the United States". In this document, the word “states” appears twice, but one of these appearances is in the context of a bigram. Therefore, our term frequency adjustment procedure reduces the raw count of the unigram “states” from 2 to 1.

Note that this counting procedure is not useful if the vocabulary contains every bigram and unigram that appears in the corpus, since no unigram would have a positive frequency count. Therefore, only the 30% of bigrams that have the highest category-term importance scores appear in the DCRI method vocabulary.

B.4 Category-Term Importance Score Calculation

The category-term importance score calculation procedure takes as input the term frequency f_{cv}, which is the count of term v in the category description for category c after making the bigram adjustment described in Appendix B.3. An additional input based on f_{cv} is δ_{cv}, which is equal to 1 if $f_{cv} > 0$ for term v and category c and 0 otherwise.

As an intermediate step, the category-term importance score calculation presented below outputs the category-term importance score matrix $\mathbf{S} \in \mathbb{R}^{C \times V}$ that is introduced in Section 3.3.1. These scores can be used directly in the Preliminary Labeling and Label Augmentation steps of the DCRI method if the vocabulary is restricted to unigrams only. To integrate bigrams into the vocabulary, several additional normalization steps are required. The final output of these normalization steps yields a nonnegative transformation of $\log(\mathbf{S})$, which is denoted here as $\tilde{\mathbf{S}} \in \mathbb{R}_{\geq 0}^{C \times V}$ and has elements $\tilde{s}_{cv}$ for each term v and each category c. The implementations of the DCRI method for the use cases described in Chapter 3 substitute $\tilde{\mathbf{S}}$ directly in place of $\log(\mathbf{S})$.

The category-term importance scores are calculated separately for the set of unigrams in the vocabulary, denoted $\mathcal{U}$, and the set of bigrams, denoted $\mathcal{B}$. The calculation procedure for the unigrams is described below. Steps 1-3 and 5 are adopted from steps 1-3 and 5 in Table 4 of the Transformed Weight-Normalized Complement Naive Bayes (TWCNB) importance scores from [37]. Step 4 replaces the complement calculation in step 4 of the TWCNB procedure with a parameter estimation calculation that is analogous to Multinomial Naive Bayes (MNB). Steps 6-8 are additional normalization procedures that scale the bigram and unigram importance scores separately. The bigram calculation procedure is identical to the unigram calculation after substituting $\mathcal{B}$ for $\mathcal{U}$, with a few exceptions that are noted.

1. $s_{cv} = \log(f_{cv} + 1)$ (term frequency transformation)
2. $s_{cv} = s_{cv} \cdot \log\left(\frac{C+1}{\sum_{c \in [C]} \delta_{cv}}\right)$ (inverse document frequency transformation)
3. $s_{cv} = \frac{s_{cv}}{\sqrt{\sum_{v \in \mathcal{U}} (s_{cv})^2}}$
4. $s_{cv} = \frac{s_{cv}+1}{\sum_{v \in \mathcal{U}} (s_{cv}+1)}$

The output of step 4 is the matrix $\mathbf{S}$, which can be used directly in the DCRI method if the vocabulary is limited to unigrams as described above. The following steps involve additional normalization necessary for incorporating bigrams into the vocabulary:

5. $\tilde{s}_{cv} = \log(s_{cv})$

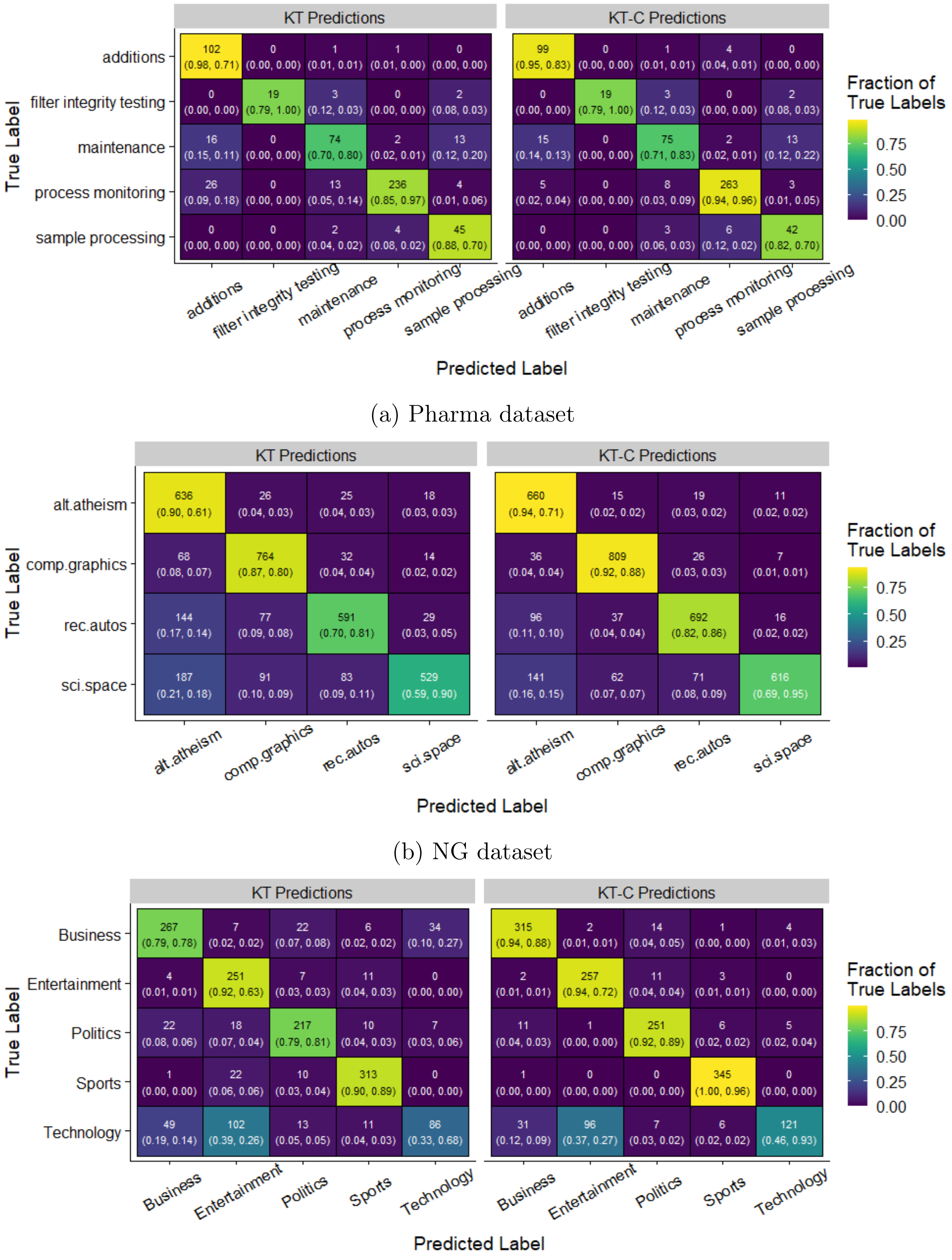
6. $\tilde{s}_{cv} = \tilde{s}_{cv} + \min_{c\in[C],v\in\mathcal{U}} \{\tilde{s}_{cv}\}$ (nonnegativity transformation)

7. $\tilde{s}_{cv} = \frac{\tilde{s}_{cv}}{\max_{c\in[C],v\in\mathcal{U}}\{\tilde{s}_{cv}\}}$ (normalizes importance scores the range $[0, 1]$)

8. $\tilde{s}_{cv} = b \cdot \tilde{s}_{cv}$ (b is a scaling term equal to 1 for unigrams and 2 for bigrams)

The final scaling step is performed to increase the value of each bigram appearance in the input documents relative to each unigram appearance. This scaling reflects the hypothesis that bigrams retained in the vocabulary are more likely to be useful features for classification than the unigrams.

B.5 KT vs KT-C Classification Performance



(a) Pharma dataset

(b) NG dataset

(c) BBC dataset

Figure B-1: Confusion matrices comparing the category predictions during the Label Augmentation step before (KT) and after (KT-C) the CNB supervised classification layer is applied.

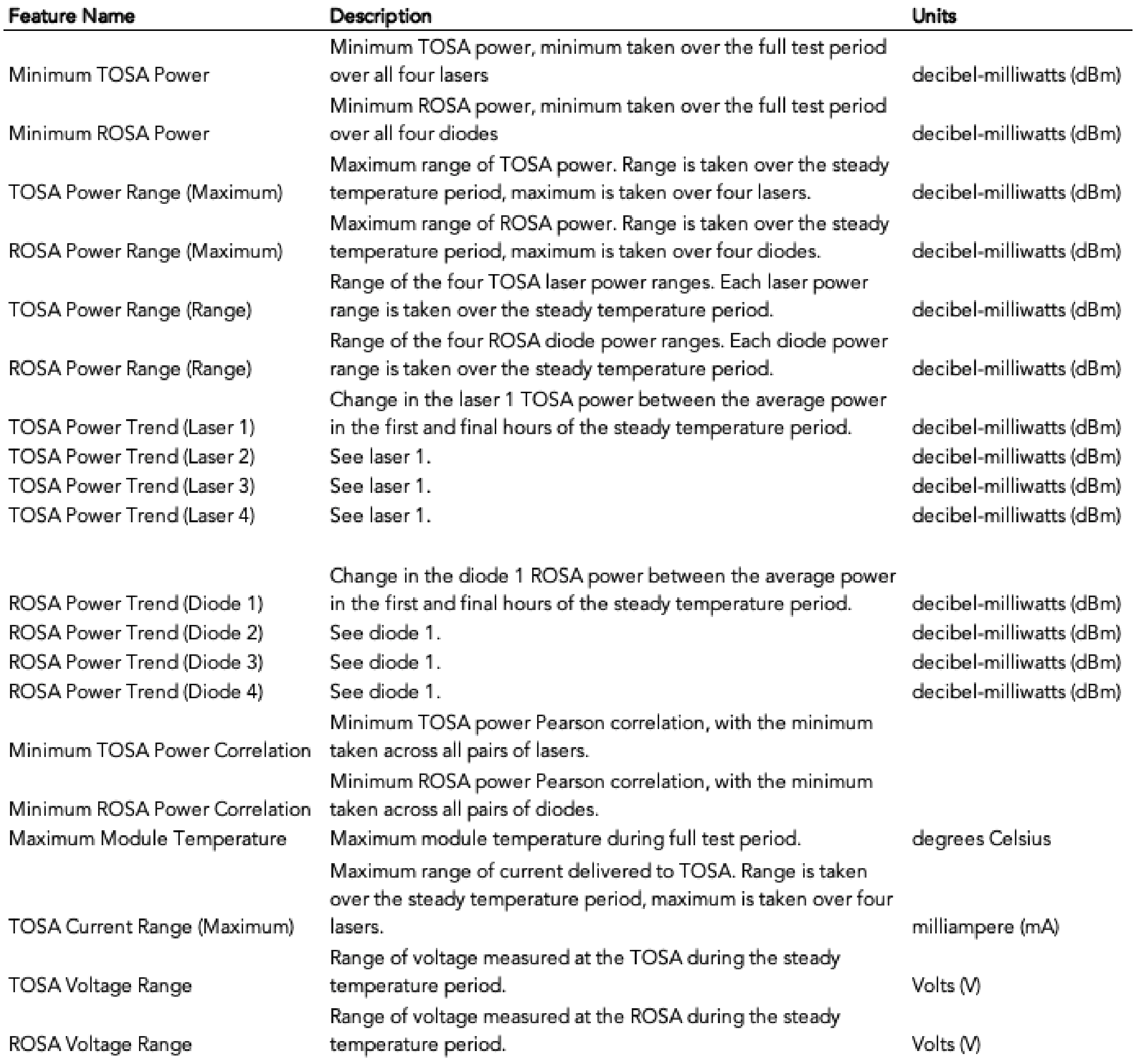
Appendix C

Appendix for Chapter 4

C.1 MBI Test Features Summary Statistics

C.2 Steady Temperature Period Calculation

The start of the steady temperature period comes at the final time point in the first half of the module run where the module temperature changes by more than 0.75 degrees Celsius. The end of the steady temperature period comes at the first time point in the second half of the module run where the module temperature changes by more than 0.75 degrees Celsius.



Feature Name	Description	Units
Minimum TOSA Power	Minimum TOSA power, minimum taken over the full test period over all four lasers	decibel-milliwatts (dBm)
Minimum ROSA Power	Minimum ROSA power, minimum taken over the full test period over all four diodes	decibel-milliwatts (dBm)
TOSA Power Range (Maximum)	Maximum range of TOSA power. Range is taken over the steady temperature period, maximum is taken over four lasers.	decibel-milliwatts (dBm)
ROSA Power Range (Maximum)	Maximum range of ROSA power. Range is taken over the steady temperature period, maximum is taken over four diodes.	decibel-milliwatts (dBm)
TOSA Power Range (Range)	Range of the four TOSA laser power ranges. Each laser power range is taken over the steady temperature period.	decibel-milliwatts (dBm)
ROSA Power Range (Range)	Range of the four ROSA diode power ranges. Each diode power range is taken over the steady temperature period.	decibel-milliwatts (dBm)
TOSA Power Trend (Laser 1)	Change in the laser 1 TOSA power between the average power in the first and final hours of the steady temperature period.	decibel-milliwatts (dBm)
TOSA Power Trend (Laser 2)	See laser 1.	decibel-milliwatts (dBm)
TOSA Power Trend (Laser 3)	See laser 1.	decibel-milliwatts (dBm)
TOSA Power Trend (Laser 4)	See laser 1.	decibel-milliwatts (dBm)
ROSA Power Trend (Diode 1)	Change in the diode 1 ROSA power between the average power in the first and final hours of the steady temperature period.	decibel-milliwatts (dBm)
ROSA Power Trend (Diode 2)	See diode 1.	decibel-milliwatts (dBm)
ROSA Power Trend (Diode 3)	See diode 1.	decibel-milliwatts (dBm)
ROSA Power Trend (Diode 4)	See diode 1.	decibel-milliwatts (dBm)
Minimum TOSA Power Correlation	Minimum TOSA power Pearson correlation, with the minimum taken across all pairs of lasers.	
Minimum ROSA Power Correlation	Minimum ROSA power Pearson correlation, with the minimum taken across all pairs of diodes.	
Maximum Module Temperature	Maximum module temperature during full test period.	degrees Celsius
TOSA Current Range (Maximum)	Maximum range of current delivered to TOSA. Range is taken over the steady temperature period, maximum is taken over four lasers.	milliampere (mA)
TOSA Voltage Range	Range of voltage measured at the TOSA during the steady temperature period.	Volts (V)
ROSA Voltage Range	Range of voltage measured at the ROSA during the steady temperature period.	Volts (V)

Figure C-1: List of Engineered MBI Test Features

Bibliography

[1] Yasser M. Alatawi and Richard A. Hansen. Empirical estimation of underreporting in the U.S. Food and Drug Administration Adverse Event Reporting System (FAERS). *Expert Opinion on Drug Safety*, 16(7):761–767, 7 2017.

[2] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 2 1970.

[3] Paul Beninger. Opportunities for Collaboration at the Interface of Pharmacovigilance and Manufacturing. *Clinical Therapeutics*, 39(4):702–712, 4 2017.

[4] Francisco Betti, Felipe Bezamat, Stephan Bloempott, Memia Fendri, and Daniel Küpper. Data Excellence: Transforming manufacturing and supply systems. Technical report, World Economic Forum, 1 2021.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] Leo Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.

[7] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3(1):1–27, 1974.

[8] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 13-17-August-2016, pages 785–794, New York, NY, USA, 8 2016. ACM.

[9] Traci E Clemons and Edwin L Bradley. A nonparametric measure of the overlapping coefficient. *Computational Statistics & Data Analysis*, 34(1):51–61, 7 2000.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 9 1977.

[11] Thierry Dumont. Context Tree Estimation in Variable Length Hidden Markov Models. *IEEE Transactions on Information Theory*, 60(6):3196–3208, 6 2014.

[12] William Dumouchel, Nancy Yuen, Nassrin Payvandi, Wendy Booth, Andrew Rut, and David Fram. Automated method for detecting increases in frequency of spontaneous adverse event reports over time. *Journal of Biopharmaceutical Statistics*, 23(1):161–177, 2013.

[13] Bradley Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 5 1994.

[14] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 377–384, New York, New York, USA, 2006. ACM Press.

[15] R Harpaz, W DuMouchel, N H Shah, D Madigan, P Ryan, and C Friedman. Novel Data-Mining Methodologies for Adverse Drug Event Discovery and Analysis. *Clinical Pharmacology & Therapeutics*, 91(6):1010–1021, 6 2012.

[16] Lorna Hazell and Saad A W Shakir. Under-Reporting of Adverse Drug Reactions. *Drug Safety*, 29(5):385–396, 2006.

[17] Günter Heimann, Rossella Belleli, Jouni Kerman, Roland Fisch, Joseph Kahn, Sigrid Behr, and Conny Berlin. A nonparametric method to detect increased frequencies of adverse drug reactions over time. *Statistics in Medicine*, 37(9):1491–1514, 2018.

[18] Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, 2010.

[19] Charles Khouri, Thuy Nguyen, Bruno Revol, Marion Lepelley, Antoine Pariente, Matthieu Roustit, and Jean Luc Cracowski. Leveraging the Variability of Pharmacovigilance Disproportionality Analyses to Improve Signal Detection Performances. *Frontiers in Pharmacology*, 12(May):1–7, 2021.

[20] Ioannis Kontoyiannis, Lambros Mertzanis, Athina Panotopoulou, Ioannis Papageorgiou, and Maria Skoularidou. Bayesian context trees: Modelling and exact inference for discrete time series. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 4 2022.

[21] S. Krubasik, V. Dirlea, Kidambi R., and Sachsenedr C. Quality 4.0: Preventive, Holistic, Future-Proof. Technical report, ATKearney, 2015.

[22] Martin Kulldorff, Inna Dashevsky, Taliser R. Avery, Arnold K. Chan, Robert L. Davis, David Graham, Richard Platt, Susan E Andrade, Denise Boudreau, Margaret J. Gunter, Lisa J. Herrinton, Pamala A. Pawloski, Marsha A. Raebel, Douglas Roblin, and Jeffrey S. Brown. Drug safety data mining with a tree-based scan statistic. *Pharmacoepidemiology and Drug Safety*, 22(5):517–523, 5 2013.

[23] David D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Machine Learning: ECML-98*, volume 1398, pages 4–15. Springer, Berlin, Heidelberg, 1998.

[24] Olivia Mahaux, Vincent Bauchau, Ziad Zeinoun, and Lionel Van Holle. Tree-based scan statistic – Application in manufacturing-related safety signal detection. *Vaccine*, 37(1):49–55, 2018.

[25] Timothy Miller, Dmitriy Dligach, and Guergana Savova. Unsupervised Document Classification with Informed Topic Models. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 83–91, Stroudsburg, PA, USA, 2016. Association for Computational Linguistics.

[26] Frederick Mosteller and David L. Wallace. Inference in an Authorship Problem. *Journal of the American Statistical Association*, 58(302):275, 6 1963.

[27] Radford M. Neal and Geoffrey E. Hinton. A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants. In *Learning in Graphical Models*, pages 355–368. Springer Netherlands, Dordrecht, 1998.

[28] David F. Nettleton, Albert Orriols-Puig, and Albert Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306, 4 2010.

[29] Dhiraj Neupane and Jongwon Seok. Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset With Deep Learning Approaches: A Review. *IEEE Access*, 8:93155–93178, 2020.

[30] Kamal Nigam, Andrew McCallum, and Tom Mitchell. Semi-Supervised Text Classification Using EM. In *Semi-Supervised Learning*, pages 32–55. The MIT Press, 9 2006.

[31] Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun, and Tom Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. *Machine learning*, 39:103–134, 2000.

[32] Alexander A. Popov, Tatyana A. Gultyaeva, and Vadim E. Uvarov. Training hidden Markov models on incomplete sequences. In *2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, volume 2, pages 317–320. IEEE, 10 2016.

[33] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[34] Adrian E Raftery. Bayesian Model Selection in Social Research. *Sociological Methodology*, 25(1995):111, 1995.

[35] Roger Ratcliff and Francis Tuerlinckx. Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. *Psychonomic Bulletin & Review*, 9(3):438–481, 9 2002.

[36] Jason D.M. Rennie. Home Page for 20 Newsgroups Data Set.

[37] Jason D.M. Rennie, Lawrence Shih, Jaime Teevan, and David Karger. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceedings, Twentieth International Conference on Machine Learning*, volume 2, 2003.

[38] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1 1988.

[39] Lovisa Sandberg, Henric Taavola, Yasunori Aoki, Rebecca Chandler, and G. Niklas Norén. Risk Factor Considerations in Statistical Signal Detection: Using Subgroup Disproportionality to Uncover Risk Groups for Adverse Drug Reactions in VigiBase. *Drug Safety*, 43(10):999–1009, 2020.

[40] Jacob Schreiber and Paul G Allen. pomegranate: Fast and Flexible Probabilistic Modeling in Python. *Journal of Machine Learning Research*, 18:1–6, 2018.

[41] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2), 3 1978.

[42] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 3 2002.

[43] Chao Shang and Fengqi You. Data Analytics and Machine Learning for Smart Process Manufacturing: Recent Advances and Perspectives in the Big Data Era. *Engineering*, 5(6):1010–1016, 12 2019.

[44] M. Stone. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1 1974.

[45] United States FDA Center for Biologics Evaluation and Research. Best Practices in Drug and Biological Product Postmarket Safety Surveillance for FDA Staff. Technical report, United States Department of Health and Human Services, 2019.

[46] U.S. Department of Health and Human Services. 21 CFR 600.14 Reporting of biological product deviations by licensed manufacturers.

[47] U.S. Department of Health and Human Services. Vaccine Adverse Event Reporting System (VAERS), 2022.

[48] U.S. Food and Drug Administration. FDA Adverse Event Reporting System (FAERS), 2022.

[49] Eric Jan Wagenmakers, Roger Ratcliff, Pablo Gomez, and Geoffrey J. Iverson. Assessing model mimicry using the parametric bootstrap. *Journal of Mathematical Psychology*, 48(1):28–50, 2004.

[50] Steven Walfish. A Review of Statistical Outlier Methods. *Pharmaceutical Technology*, 2006.

[51] Jinjiang Wang, Peilun Fu, and Robert X. Gao. Machine vision intelligence for product defect inspection based on deep learning and Hough transform. *Journal of Manufacturing Systems*, 51:52–60, 4 2019.

[52] Murray S. Weitzman. Measures of Overlap of Income Distributions of White and Negro Families in the United States. Technical report, U.S. Bureau of the Census, Washington, D.C., 1970.

[53] Joshua Wilde. HMMScan: Surveillance of Adverse Event Variability across Manufacturing Lots in Biologics, 10 2022.

[54] Joshua Wilde and Retsef Levi. HMMScan Data Repository, 10 2022.

[55] Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and Regularization of Support Vector Machines. *Journal of Machine Learning Research*, 10:1485–1510, 2009.

[56] Huan Xu, Constantine Caramanis, and Shie Mannor. Robust Regression and Lasso. *IEEE Transactions on Information Theory*, 56(7):3561–3574, 7 2010.

[57] Jing Yang, Shaobo Li, Zheng Wang, Hao Dong, Jun Wang, and Shihao Tang. Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges. *Materials*, 13(24):5755, 12 2020.

[58] Avigdor Zonnenshain and Ron S. Kenett. Quality 4.0—the challenging future of quality engineering. *Quality Engineering*, 32(4):614–626, 10 2020.