Bayesian Nonparametric Classification for Incomplete Data With a High Missing Rate: An Application to Semiconductor Manufacturing Data

Sewon Park, Kyeongwon Lee, Da-Eun Jeong, Heung-Kook Ko, and Jaeyong Lee

Abstract—During the semiconductor manufacturing process, predicting the yield of the semiconductor is an important problem. Early detection of defective product production in the manufacturing process can save huge production cost. The data generated from the semiconductor manufacturing process have characteristics of non-normal distributions, random missing patterns and high missing rate, which complicate the prediction of the yield. We propose the Dirichlet Process - Naive Bayes model (DPNB) that can simultaneously impute missing values and address classification problems. Since the DPNB is based on the infinite Gaussian mixture model, it can estimate complex data distributions and make predictions for missing datasets with some missing patterns due to nice properties of the Gaussian distribution. The DPNB also performs well for high missing rates since it uses all information of observed components. Experiments on various real datasets including semiconductor manufacturing data show that the DPNB has better performance than state-of-the-art methods in terms of predicting missing values and target variables as percentage of missing values increases.

Index Terms—Missing data, imputation, classification, semiconductor defects, yield estimation.

I. INTRODUCTION

MISSING data occur widely in engineering problems and scientific research. Especially, handling missing values is often the first problem to consider in analyzing the data from the manufacturing process, social sciences, and biology fields. Prediction improvement of the yield of semiconductor manufacturing is the main problem for having stable and consistent manufacturing processes. Early detection of defective wafer production in manufacturing can save high production costs. For this purpose, a massive amount of data generated from many sequential processes are analyzed to find the relationship between various factors and the yield in wafer fabrication processes. However, managing the data is very cost-intensive, and about 90 percent of data is not recorded in practice. In semiconductor manufacturing analytics, one major challenge is handling the data with a very high missing rate for wafer yield prediction. Since the way missing values are handled has a significant effect on predicting the yield of the semiconductor, proper methods to deal with missing data should be considered.

References [1] and [2] established three types of missingness mechanisms: (a) missing completely at random (MCAR), (b) missing at random (MAR), and (c) missing not at random (MNAR). When missingness is unrelated to both observed and unobserved variables, the data are called MCAR. When missingness only depends on observed data, the data are MAR. MAR is a more plausible assumption than MCAR. When missingness depends on unobserved data or the missing value itself, the data are said to be MNAR. An example of the MNAR is censored missing values caused by the LOD [3]. In this paper, we focused on the MCAR and MAR. These missingness mechanisms only affected by randomness, which can be represented by a probabilistic model. We present empirical comparisons between the DPNB and other competitors under the MCAR and MAR.

The simplest approach to handle missing data is to remove incomplete observations or cases. But this method may lead to biased conclusions and lose some useful information. Another approach is the imputation, which substitutes missing data for plausible values generated from statistical learning. Imputing missing data may be a more reasonable way than discarding missing values but not necessarily give better results.

We review several popular imputation techniques for estimating missing values. The basic method is the mean imputation, which replaces missing values with the mean of the observed values in a certain variable. The mean imputation is simple to apply but underestimates the variance and produces biased results under both MAR and MNAR [2]. Another method is the multivariate imputation by chained equations (MICE) developed by [4]. The MICE as a type of multiple imputations [1] constructs separate conditional models for each incomplete variable and iteratively imputes the missing values. The Gaussian mixture model (GMM) is one of the most
widely used model-based imputation methods [5], [6], [7]. Model-based methods assume the joint distribution of all variables in the data and estimate the parameters of the distribution [8]. The GMM on incomplete data imputes missing values using the conditional distribution properties of a multivariate normal distribution.

Contrary to the model-based methods, there are imputation strategies based on machine learning algorithms that do not rely upon the distributional assumptions on the data. References [9] and [10] designed decision-tree-based imputation methods using classification and regression tree (CART) and random forest, respectively. Another non-parametric approach is K-Nearest Neighbors (KNN) based imputation [11]. Reference [12] improved the accuracy of estimated missing values using an iterative process. The aforementioned imputation methods from machine learning can deal with both numerical and categorical data as well as mixed-type data. Thus, these methods have been commonly used in various fields related to missing data problems.

Deep learning-based imputation methods have been proposed in different neural network architectures. References [13], [14], and [15] reconstructed missing values by using feed-forward neural networks (FNN). References [16] and [17] introduced recurrent neural networks to handle incomplete sequential data. References [18] and [19] designed deep generative models combining denoising autoencoders (DAE) to create the clean output from the noisy input considered as missing data. References [20] and [21] provided modified generative adversarial nets (GAN) for filling missing values.

Many methodologies have been proposed for the imputation and classification of semiconductor manufacturing data. [22], [23], [24] pointed out that the semiconductor manufacturing data are imbalanced and suggested using SMOTE algorithm [25] to overcome the class imbalance. In the data analysis, [22], [23] removed all missing values for classification problems. Reference [26] used the UCI SECOM (SEmi-CONductor Manufacturing) dataset to investigate the applicability and impact of machine learning techniques for semiconductor manufacturing processes. They applied Boruta [27] and MARS [28] to select the relevant variables and compared the Random Forest and Gradient Boosted Tree for the classification problem. They used the mean imputation and KNN-based imputation for missing values. References [29], [30] addressed the multi-colinearity problem for the semiconductor manufacturing data and proposed tree based classifier such as XGBoost [31]. Reference [32] presented an experimental result of various approaches, including data pruning, imputation, feature selection, and classification methods based on the SECOM dataset. They argued that the KNN-based imputation showed the best performance.

Recently, deep learning-based models for imputation and classification have been studied. Reference [33] suggested convolutional neural network [34] (CNN)-based classifier, and [35] suggested a GAN-based imputation method and applied it to the SECOM dataset. There are also studies [36], [37], [38], [39] that predict the defect using the wafer itself, not the observation from the semiconductor production process [40]. References [41], [42], [43], [44] proposed an end-to-end fault detection and classification model using CNN models such as VGGNet [45] and ResNet [46]. Reference [47] proposed a method based on deep convolutional generative adversarial network [48] to increase the data used for training and balance the classes at the same time. Note that in these studies, statistical methodologies for missing values are not necessary for the wafer image data do not have missing values.

In this paper, we propose a new combined method that performs both imputation and classification tasks. Our proposed method has four advantages over the existing methods. First, the proposed classifier is based on the class conditional distribution constructed by an infinite Gaussian mixture model instead of the single Gaussian used in a standard naïve Bayes classifier. Since mixture models can approximate arbitrary continuous distributions [49], [50], they can create flexible decision boundaries. Using the infinite Gaussian mixture model, we don’t need to consider the number of Gaussian distributions and can save computation time regarding choosing the optimal number of components via information criteria such as AIC, BIC and DIC. In the proposed model, when the number of classes is high, the automatic model selection is more useful in terms of computation time. It does not have to compare many candidate models.

Second, the proposed imputer is more accurate than other imputation methods for incomplete data with high missing rates. In other words, as the percentage of missing values increases, the imputation technique based on class conditional distribution outperforms state-of-the-art methods under both MCAR and MAR assumptions. The proposed method use all the data regardless of missing patterns to estimate the joint distribution of variables in the data set and use the conditional distribution from the estimated joint distribution to impute missing values. This is unlike other imputation methods which learn a conditional distribution based only on the data that match the missing pattern. Since other competing methods do not use all the information in the data, the proposed method performs better as the missing rate rises. Note that other imputation methods have a low reliability in rare patterns, but proposed method is less prone to this problem.

Third, the DPNB is a hybrid approach that handles both imputation and classification concurrently. Most existing methods do them separately. The proposed method has relatively lower classification errors because it models the joint distribution of observed data and missing data and builds the decision classifier.

Fourth, the DPNB is capable of predicting the labels of a set of new cases, which have any missing patterns from a predictive model. Most imputation techniques focus on estimating missing values and cannot conduct subsequent statistical analyses such as classification or regression. As a result, researchers used to transform missing data to complete cases by using imputation methods and build a regressor or classifier in the training phase. However, missing data may exist not only in the training set but also in the test set. Missing values in the test set are filled with the mean of columns in the transformed data or estimated values obtained by refitting the
imputers without regard to the training set. Our method can address the test set, which has missing values with any missing patterns. Experiments also show that the proposed method gives better classification accuracy on multiple datasets with high missing rates from the UCI Machine Learning Repository.

The remainder of this paper is structured as follows. Section II provides preliminary notions of imputation through finite Gaussian mixture models and the Dirichlet process prior as a key factor in our proposed method. In Section III, we propose an extension of the well-known naive Bayes classifier, which uses an infinite Gaussian mixture model on incomplete data. Section IV shows empirical results on the accuracy of imputation and classification in different settings. A real data example is presented in Section V. We end this paper in Section VI with conclusions.

II. PRELIMINARIES

In this section, we review the Gaussian mixture models for missing data and the Dirichlet process, which are core concepts of our proposed method.

A. Notations

The probability distribution $N_p(\mu_p, \Sigma_p)$ denotes the multivariate Gaussian (normal) distribution of a $p$ dimensional random vector with $p$ dimensional mean vector $\mu_p$ and $p \times p$ dimensional covariance matrix $\Sigma_p$. The probability distribution $\text{Dir}(\alpha_1, \ldots, \alpha_k)$ denotes the Dirichlet distribution of order $k$ with parameters $\alpha_1, \ldots, \alpha_k > 0$. The probability distribution $\mathcal{IW}(\mathbf{B}, \nu)$ denotes the inverse Wishart distribution with $p \times p$ positive definite matrix $\mathbf{B}$ and degrees of freedom $\nu > p - 1$. We use $F(x; \theta)$ as a probability density function of a random variable $X \sim F(\theta)$, where $\theta$ is a parameter of the probability distribution. For instance, $N_p(x; \mu_p, \Sigma_p)$ denotes a probability density function of the multivariate Gaussian distribution $N_p(\mu_p, \Sigma_p)$. We write $x_i \overset{\text{ind}}{\sim} F$ when random variables $x_i$'s are independently distributed as a probability distribution $F$. Similarly, we write $x_i \overset{\text{id}}{\sim} F$ when random variables $x_i$'s are independent and identically distributed. The cardinality $|A|$ of a set $A$ denotes the number of elements in $A$.

B. Gaussian Mixture Models for Missing Data

Algorithms for Gaussian mixture models on missing data have been studied in the last few decades. Reference [5] used the Expectation-Maximization (EM) algorithm to find parameter values and missing components, maximizing the likelihood of Gaussian mixture models. Reference [51] developed a Bayesian approach of mixture models using a Gibbs sampler. This method uses full conditional distributions to obtain the joint posterior distribution of parameters and missing values. Reference [7] introduced variational inference based on the mean-field approximation for Bayesian mixture models. Both missing values and parameters are iteratively updated until the evidence lower bound (ELBO) converges.

We focus on the estimation of a Gaussian mixture model, permitting simultaneous inference of missing values through Gibbs sampling is a Markov Chain Monte Carlo (MCMC).

Let $x_i, i = 1, \ldots, n$ be $n$ independent $p$-dimensional observations from the mixture distribution consisting of $H$ Gaussian components. We partition an observation into two components $x_i = \{x_i^0, x_i^m\}$, where $\mathcal{I}_i \subset \{1, 2, \ldots, p\}$ is an index set of observed variables and $m_i \subset \{1, 2, \ldots, p\}$ is an index set of missing variables. That is, $x_i^0$ and $x_i^m$ indicate the observed components and missing components from the $i$th observation $x_i$, respectively. We can express the mixture distribution as follows:

$$f(x_i) = \sum_{h=1}^{H} w_h N_p(x_i; \mu_h, \Sigma_h)$$

$$= \sum_{h=1}^{H} w_h N_p\left(\mu_h, \Sigma_h, \Sigma_h^{\mu_h, \mathbf{1}_h}, \Sigma_h^{\mathbf{1}_h, \mu_h}\right)$$

(1)

Here, $w_h$ is the non-negative mixing proportion and sum up to one, i.e., $\sum_{h=1}^{H} w_h = 1$.

To implement the Gibbs sampling for mixture models with incomplete data, we need the full conditional posterior distributions of model parameters and missing values. In this paper, we only cover the full conditional posterior for missing values and skip other parameters of mixture models; see the paper by [52] for more details. If the data come from a multivariate normal distribution with mean $\mu$ and covariance $\Sigma$, a full conditional density for missing values based on the observed data can be derived easily the following equation (2) by using a special property of multivariate normal distribution.

$$f(x_i^0 | x_i^m, \text{others}) \sim N_{|\mathcal{I}_i|}(x_i^0; \mu_{|\mathcal{I}_i|}, \Sigma_{|\mathcal{I}_i|})$$

$$\mu_{|\mathcal{I}_i|} = \mu + \Sigma_{|\mathcal{I}_i|, \mathcal{I}_i}^{-1}(x_i^0 - \mu^0),$$

$$\Sigma_{|\mathcal{I}_i|} = \Sigma_{|\mathcal{I}_i|, \mathcal{I}_i} - \Sigma_{|\mathcal{I}_i|, \mathcal{I}_i}^{-1}(\Sigma_{|\mathcal{I}_i|, \mathcal{I}_i})^{-1}\Sigma_{\mathcal{I}_i, \mathcal{I}_i}. \quad (2)$$

Missing values are filled with samples drawn from the full conditional distribution and combined with the fixed observed values to subsequently update other parameters from the full conditionals. This updating process is called the multivariate normal imputation, first implemented by [53], which is one of the multiple imputation methods. In the case of the mixture models, we assume that all data points are generated from the mixture of multivariate normal distributions. An observation with the missing values belong to $h$th mixture component is imputed by sampling the full conditional distribution for $x_i^m$ given $\mu_h$ and $\Sigma_h$ associated with $h$th mixture component. Mixture components that indicate clusters are determined by latent auxiliary variables. See [51].

C. Dirichlet Process

Dirichlet process (DP) introduced by [54] is the most popular Bayesian nonparametric model and has been used in many applications for clustering in the last two decades. Let $\mathcal{X}$ be a measurable space and $B$ the Borel $\sigma$-field of subsets of $\mathcal{X}$. Then we say that the random probability measure $G$ on $(\mathcal{X}, B)$ follows a Dirichlet process with a concentration parameter $\alpha > 0$ and a baseline probability measure $G_0$, denoted by $G \sim DP(\alpha, G_0)$, if for every finite disjoint partition $A_1, \ldots, A_k$ of $\mathcal{X}$,

$$(G(A_1), \ldots, G(A_k)) \sim \text{Dir}(\alpha G_0(A_1), \ldots, \alpha G_0(A_k)).$$
Dirichlet process can be represented in three different ways: (1) pólya urn scheme [55] (2) Chinese restaurant process [56] (3) stick-breaking process [57].

Reference [57] also showed that its realizations are discrete almost surely, even if \( G_0 \) is a continuous distribution. The discreteness of the DP implies that it is unsuitable prior for data generated from continuous distributions. To eliminate this drawback, [58] adopted Dirichlet process mixture models (DPMM) having the following hierarchical model formulations: for \( i = 1, \ldots, n \),
\[
  x_i | \theta_i \overset{ind}{\sim} f(x_i | \theta_i),
\]
\[
  \theta_i \overset{ind}{\sim} G,
\]
\[
  G \sim DP(\alpha, G_0),
\]
where \( f \) is a parametric density function. The DPMM can be expressed as a limit of finite mixture models, where the number of mixture components is taken to infinity [59]. For example, if a base measure \( G_0 \) is a multivariate normal Inverse-Wishart conjugate prior and \( f(x_i | \theta_i) \) is multivariate normal, the DPMM can be an infinite normal mixture model. The advantage of the DPMM is that the number of mixture components is not fixed in advance and can be inferred automatically from the data, unlike finite normal mixture models. Various algorithms have been developed for posterior inference of the DPMM such as marginal Gibbs sampling [60], conditional Gibbs sampling [61], [62], split-merge MCMC sampling [63], and variational approximation [64].

III. PROPOSED MODEL

In this section, we propose a Bayesian approach for the imputation and classification on incomplete data. In this paper, we call the proposed method as Dirichlet Process-Naive Bayes model (DPNB).

A. Generative Model With DPMM

Generative models employ the Bayes theorem to build a classifier. Let the input or feature vector be \( X \) and the class label be \( Y \). We need the density of \( X \) conditioned on the class \( k \), \( P(X = x | Y = k) \) and the prior probability, \( P(Y = k) \) to compute the posterior probability:
\[
P(Y = k | X = x) = \frac{P(Y = k) \cdot P(X = x | Y = k)}{\sum_{l=1}^{K} P(Y = l) \cdot P(X = x | Y = l)}. \tag{3}
\]

Then, input \( x \) is assigned to the class having the highest posterior probability. Typical examples of generative models include linear discriminant analysis (LDA), quadratic discriminant, analysis (QDA), and Gaussian naive Bayes (GNB). The three models assume multivariate normal densities for \( P(X = x | Y = k) \). Another example is to assume that the class-conditional density of \( X \) is a finite mixture of normals. This method is called mixture discriminant analysis (MDA) [65], [66]. Instead of using a finite mixture of normals, we propose to use the Dirichlet process mixture model, as the number of Gaussians is automatically determined and not limited. Practically, it is time consuming to find the optimal number of components in finite mixture models via information criteria such as AIC, BIC, and DIC. In our cases, as the number of classes grows, the time spent searching the best model among many candidate models increases even more. Since real datasets are complicated, it is more efficient for the DPMM to adaptively fit the data when considering the total computation time.

We consider a binary classification problem and assume class labels \( Y \in \{0, 1\} \) has a binomial distribution with parameters \( n \) and \( q \in [0, 1] \). If we have \( K (\geq 2) \) classes, \( Y \in \{1, \ldots, K\} \) is assumed to have a multinomial distribution with \( n \) and \( q = (q_1, \ldots, q_K) \). The distribution of \( X \) conditioned on the class \( k \) is modeled by the following hierarchical formulation for DPMM:
\[
  x_i | y_i = k \sim \mathcal{N}_p(x_i^k, \mu_i^k, \Sigma_i^k),
\]
\[
  \left( \mu_i^k, \Sigma_i^k \right) \overset{iid}{\sim} G_k, \quad i = 1, 2, \ldots, n_k,
\]
\[
  G_k \sim DP(\alpha, M_k), \quad k = 0, 1, \tag{4}
\]
where \( x_i^k \) is the \( i \)th feature vector and \( \mu_i^k \) and \( \Sigma_i^k \) are parameters of a normal distribution which belongs to only class \( k \). Here, the base measure for class \( k \), \( M_k \) is the conjugate multivariate normal–inverse Wishart distribution, i.e., \( M_k = \mathcal{N}_p(\mu_i^k; \mu_0^k, \Sigma_i^k/\tau_0^k) \cdot \mathcal{IW}(\Sigma_i^k; B_0^k, v_0^k) \), where \( \mu_0^k \) and \( \Sigma_0^k/\tau_0^k \) are the prior mean and covariance of \( \mu_i^k \), respectively. \( B_0^k \) is a positive definite scale matrix and \( v_0^k \) is a positive scalar degrees of freedom parameter. Then the class-conditional density of \( X \) is given by
\[
P(X = x | Y = k) = \sum_{h=1}^{\infty} w_h^k \mathcal{N}_p(x; \mu_h^k, \Sigma_h^k), \quad k \in \{0, 1\} \tag{5}
\]
using posterior samples for mixing proportion and cluster specific parameters. We use the improved slice sampler suggested by [67] to generate samples from the posterior distribution. We describe the detailed MCMC algorithm for the DPMM on incomplete data in the Supplement. Since \( Y \) has a binomial distribution, marginal probabilities over classes are estimated by
\[
P(Y = k) = \frac{n_k}{n}, \quad k \in \{0, 1\}, \tag{6}
\]
where \( n_k \) denotes the number of observations belonging to the class \( k \). Putting (5) and (6) together in the equation (3), we can produce the following posterior probabilities for classes:
\[
P(Y = k | X = x) = \frac{n_k \times \sum_{h=1}^{\infty} w_h^k \mathcal{N}_p(x; \mu_h^k, \Sigma_h^k)}{\sum_{l=0}^{1} \left( n_l \times \sum_{h=1}^{\infty} w_h^l \mathcal{N}_p(x; \mu_h^l, \Sigma_h^l) \right)}. \tag{7}
\]

We compute posterior probabilities for all \( k \) and choose only one class associated with the highest probability. Both LDA and QDA assume multivariate normal densities for \( P(X = x | Y = k) \) and GNB assumes that the features, \( x_1, x_2, \ldots, x_p \) are conditional independent on \( Y \) and \( P(x_i | Y = k) \) is univariate normal density:
\[
P(X = x | Y = k) = \prod_{j=1}^{p} P(x_j | Y = k).
\]

Since three generative models are designed for different covariance structures of multivariate normal distributions, they
applied in the mixture models with infinite components. For
described in the previous Section II-B is capable of being
target variables and missing values. The imputation approach
B. Imputation and Prediction Strategy
Machine (SVM) and Random Forest (RF).
The GNB is not necessarily a linear classification according
the LDA are linear, whereas those for the QDA are quadratic.
and GNB. In other words, the DPNB is a much more flex-
approximates most decision boundaries than the LDA, QDA,
approximation techniques in various settings. Second, we quantitatively
compare the prediction accuracy of the DPNB model and compe-
tors using multiple datasets. First, we assess the imputation
accuracy of our proposed methods with state-of-the-art impu-
have their respective classifiers. The decision boundaries for
the LDA are linear, whereas those for the QDA are quadratic.
The GNB is not necessarily a linear classification according
to the data. However, MDA allows close approximation of
not only linear but also nonlinear decision boundaries since
mixture models can approximate arbitrary continuous distribu-
tions [66], [68]. Fig. 1 shows that DPNB more accurately
approximates most decision boundaries than the LDA, QDA,
and GNB. In other words, the DPNB is a much more flex-
able classifier than them. Furthermore, the classifier of the
DPNB provides comparable results with that of Support Vector
Machine (SVM) and Random Forest (RF).

B. Imputation and Prediction Strategy

The DPNB can make predictions simultaneously for both
target variables and missing values. The imputation approach
described in the previous Section II-B is capable of being
applied in the mixture models with infinite components. For
estimating (5), the DPNB should divide the data into subsets
where each subset belongs to only one class according to the
inference process. Then, it substitutes separately missing val-
ues with plausible values generated from samplers (2) based on
respective subsets. We expect that imputation based on sub-
sets is more accurate than that based on the full data. This
is because it is easy to find easily probable values estimated
from homogeneous input data, which belong to only one class.
In practice, the empirical results on four UCI datasets indic-
ate that this split and merge imputer has better performance
than state-of-the-art imputation algorithms as the missing rate
increases.

In reality, missing data may occur in both training and test
dataset. DPNB can make predictions for classes even if all
elements of some features in the test set are absent. Let a new
ly observed input vector denote $\mathbf{x}_o^*$ and a new predicted
class label $Y^*$. The prediction rule for DPNB is given by
\[
P(Y^* = k | \mathbf{x}_o^*) = \frac{P(Y^* = k) \cdot P(\mathbf{x}_o^* | Y^* = k)}{\sum_{l \in \{0,1\}} P(Y^* = l) \cdot P(\mathbf{x}_o^* | Y^* = l)},
\]
for $k \in \{0, 1\}$. We need to compute the predictive density of
$\mathbf{x}_o^*$ conditioned on the class $k$ to complete the equation (7), the
posterior probability of the new class label. It can be approxi-
mated by using the Monte Carlo Integration from the posterior
samples in the training process, as shown in (8).
\[
P(\mathbf{x}_o^* | Y^* = k) = \int P(\mathbf{x}_o^*, \mathbf{x}_m^* | Y^* = k) \, d\mathbf{x}_m^*
= \int P(\mathbf{x}_o^*, \mathbf{x}_m^* | \mu, \Sigma, Y^* = k) \pi(\mu, \Sigma | Y^* = k) \, d\mu \, d\Sigma \, d\mathbf{x}_m^*
= \int \sum_h \pi_h^k P(\mathbf{x}_o^*, \mathbf{x}_m^* | \mu_h, \Sigma_h, Y^* = k)
\times \pi(\mu_h, \Sigma_h | Y^* = k) \, d\mu_h \, d\Sigma_h \, d\mathbf{x}_m^*
\approx \frac{1}{Q} \sum_{j=1}^Q \sum_{h_j} \pi_h^k \int P(\mathbf{x}_o^*, \mathbf{x}_m^* | \mu_{h_j}, \Sigma_{h_j}, Y^* = k) \, d\mathbf{x}_m^*
= \frac{1}{Q} \sum_{j=1}^Q \sum_{h_j} \pi_h^k P(\mathbf{x}_o^* | \mu_{h_j}, \Sigma_{h_j}, Y^* = k)
= \frac{1}{Q} \sum_{j=1}^Q \sum_{h_j} \pi_h^k \cdot \mathcal{N}_{[\mu_{h_j}], \Sigma_{h_j}}(\mathbf{x}_o^*)^{\alpha_o}, \mathbb{R}^{\alpha_o}, Y = k
\]
where $Q$ is the number of posterior samples generated by
MCMC. Therefore, DPNB can build classifiers regardless of
missingness and missing rate. Experiments also support that it
predicts classes more accurately than do competitive models.

IV. EXPERIMENTS

In this section, we evaluate both the imputation and
prediction performance of the DPNB model and competi-
tors using multiple datasets. First, we assess the imputation
accuracy of our proposed methods with state-of-the-art impu-
tation techniques in various settings. Second, we quantitatively
compare the prediction accuracy of the DPNB model and a
benchmark classification algorithm based on multiple incom-
plete datasets under different conditions. In all experiments,
we apply various missing rates for the covariates (from 10%
to 60%) and two missingness scenarios: missing completely
at random (MCAR), missing at random (MAR). We also assume
that the target (or class) variables have no missing values.
We conducted experiments on four real-life datasets from
the UCI Machine Learning Repository [69]: Ecoli, Wine,
Breast Cancer Wisconsin (Diagnostic), and Wine Quality
datasets. Specifically, the Ecoli dataset contains 336 observ-
vations with 8 features and multiple classes. We transform

multi-class labels into binary labels, which have positive (type im) and negative (the rest). Two discrete variables, Lip and Chg are also removed. The red wine quality dataset contains 1599 observations on 11 attributes of wine and 6 wine quality classes. We divide them into 3 groups for quality: Excellent ($\geq 7$), Good (6), and Poor ($\leq 5$). The rest of the multiple datasets were originally used in this experiment. Those datasets have only continuous input variables. A summary of the UCI datasets is given in Table I. Here, the imbalanced ratio (IR) is defined by

$$\text{IR} = \frac{\max_{C \in \mathcal{A}} |C|}{\min_{C \in \mathcal{A}} |C|},$$

where $\mathcal{A}$ is the set of all classes. The higher the imbalance ratio is, the bigger disproportion exists between the majority class and minority class. This paper does not consider the UCI SECOM dataset from the experiments because we handle a real semiconductor manufacturing dataset in Section V.

### A. Imputation Performance

For each UCI dataset, we generate 100 different incomplete datasets, removing 10% to 60% of the complete values based on the MCAR or MAR missing assumptions. We use the normalized root mean squared errors (NRMSE) as the imputation accuracy measure along with its standard deviation across the 100 replicated datasets. The NRMSE is defined as

$$\text{NRMSE} = \sqrt{\frac{\text{mean}(X_{\text{true}} - X_{\text{imp}})^2}{\text{Var}(X_{\text{true}})}},$$

where $X_{\text{true}}$ is the original data and $X_{\text{imp}}$ the imputed data.

We compare our proposed methods with several popular imputation methods such as multivariate imputation by chained equations using the predictive mean matching; [4] (denoted by MICE), random forest-based imputation; [10] (denoted by RF), KNN-based imputation with the optimal number of neighbors, which minimizes cross-validation errors; [11] (denoted by KNN), imputer using deep denoising autoencoders; [19] (denoted by MIDA), and imputation technique by adapting generative adversarial nets; [21] (denoted by GAIN).

Fig. 2 shows that the imputer of the DPNB model performs well in most datasets with either the lowest or the second lowest average NRMSE values across 100 replicates, irrespective of the missingness mechanism (see Supplementary Table S1 for details). In particular, we discovered from the Breast Cancer and Wine Quality datasets that the proposed model is more accurate than other imputation algorithms as missing rates increase. The DPNB model can fill missing values via different covariance structures constructed by all available observations. However, methods to form conditional distributions using RF and MICE are less accurate than the DPNB because they use partial observations and variables instead of full information. Deep learning-based imputation methods such as GAIN and MIDA have poor performance due to model complexity relative to the number of observed data points. Among all models, the worst-performing method is the KNN based imputation.

### B. Predictive Performance

We performed 10-fold cross-validation to obtain an estimate of the classification accuracy using simulated missing datasets generated from the previous Section IV-A. The cross-validation process was repeated 10 times for fair comparisons. Since four UCI datasets have various class distributions and binary or multi-class classification problems, we use three types of classification performance metrics to measure the performance of classifiers: (a) accuracy rate, (b) area under the ROC curve (AUC), and (c) F1-score defined as

$$\text{F1-score} = \frac{2}{\text{precision}^{-1} + \text{recall}^{-1}}.$$
where the precision is the rate of true positives among the predicted positives and the recall is the rate of true positives among positives. The classification metrics corresponding to datasets are also shown in Table II.

For prediction performance, we compared the DPNB model with benchmark classifiers on the incomplete dataset. They predict test cases after the imputation stage. Some details of the procedures regarding post-imputation prediction are as follows. First, we divided the dataset with missing values into training and test sets. Imputation algorithms utilized in Section IV-A fill missing values of training sets. Second, we built the support vector machines (SVM) with radial basis function (RBF) kernels as a benchmark classifier on the imputed training datasets. Third, missing values in test sets were replaced with the mean of the features of the imputed training set. Finally, we made predictions on the imputed test set using the trained model. We call all competing methods types of imputation algorithms used in both training and test phases.

Fig. 3 provides that the DPNB model comes up with the best performance except for the Breast Cancer Wisconsin (Diagnostic) dataset, where it is competitive. It also shows that the DPNB has a better classification performance than other strategies on all datasets for higher missing rates. As missing rates increase, the prediction accuracy of the DPNB is not greatly reduced, unlike other competitive approaches. This shows that the DPNB is less affected by both missing data mechanisms and proportions missing proportions than other methods. See Table S2 in the Supplement for further details on these results.

V. Applications

The main goal of our study is to impute and predict the class of given data with high missing rate. We now apply the DP-Naive Bayes model to a semiconductor manufacturing dataset provided by Samsung Electronics’ DS division, which consists of manufacturing operation variables and the semiconductor quality variable related to defect rates. This dataset also includes many missing values and aims to reduce defective wafers made up of semiconductor materials. These challenges are expected to further illustrate the features of the DP-Naive Bayes model.

For our analysis, we used the data of 2218 wafer records and 60 manufacturing process variables, discarding features with missing rates larger than 97.5%. Fig. 4(a) depicts the observed and missing values of the pre-processing data in colored tiles and white tiles, respectively. It is made up of mostly white tiles. Missing values in the dataset account for about 95% of all values and most manufacturing operation variables have more than 90% missing rate, as can be seen in the Fig. 4(b). We can say that the data is MCAR since missing values seem to be pretty randomly scattered and actually occur at random in the manufacturing processes. Furthermore, the target variable related to defect rates was all measured and had no missing values. It is highly imbalanced, with most records falling in the “No defect” class. The imbalance ratio of this dataset is roughly 9.4.

A. Imputation Performance

To assess the imputation accuracy based on the real world dataset, we consider three scenarios by deleting artificially 50, 100, and 500 complete values in the semiconductor dataset under the MCAR assumption. In every case, we made 100 replicates, respectively. For performance comparisons, we then compute the average of normalized root mean squared errors obtained by each method in all cases.

The results of the DPNB and competing models are provided in Table III. It demonstrates that the DPNB model provides more accurate values and consistent performance through both the minimum mean and the lowest standard deviation values over 100 replications of NRMSEs. As shown in
the Breast Cancer example, both RF and MICE yield similar imputation performance with the DPNB model. Imputation methods based on deep architectures, including MIDA and GAIN have difficulty selecting appropriate parameters to prevent the problems of overfitting. These methods have then lower imputation accuracy than the DPNB, RF and MICE.

B. Predictive Performance

In this experiment, we randomly partitioned the real dataset into training and test sets with different ratios. The training set was set from 50% to 90% of the overall dataset and the test set was accordingly set from 50% to 10%. For example, if 90% of the data is used as the training set, then the remaining 10% is used as the test set. Likewise, we created 100 replicate datasets in all cases. Furthermore, the bootstrapping-based oversampling technique that replicates observations from the minority class was applied to training sets in order to balance the two classes. Note that the same indices of observations selected from the minority class were used in all methods given percentages of the training sets.

We add the SVM with a linear kernel as another base classifier. We will name competitive methods by combining imputation techniques and kernels of support vector machine, e.g., “RF+L” means that the RF imputes missing values of training sets and the SVM with a linear kernel is fitted on the imputed training sets and predicts the test set values. We are interested in detecting defective wafers, the minority class in the data set. Since the class distributions of the real-world dataset are highly unbalanced, the classification accuracy is not a appropriate measure for classifier performance. Instead, we use the F1-score to compare performance of the classification methods. The F1-score is more suitable classifier performance measure for unbalanced data sets than the accuracy.

The results of this experiment are given in Table IV. The columns of competing methods represent their average F1-score over 100 replicate datasets, normalized by the average F1-score of the DPNB model. The values higher than one indicate that the methods provide better performance than the DPNB and values lower than one indicate worse performance in imbalanced settings.

As shown in Table IV, the DPNB method performs well in practice since it ranks high in almost all cases. In particular, it yields the best prediction performance on two of the five ratios. We can say that the DPNB model provides stable performance by obtaining the highest average normalized F1-score and average rank and stable inference in the incomplete data with high missing rates. However, since the semiconductor dataset has very high missing rates, the DPNB does not provide remarkable performance against other methods, as shown in the previous section. The methods using the KNN imputer have poor performance irrespective of the type of classifier. We also present the actual average F1-score in the Supplement.

VI. CONCLUSION

We suggested a Bayesian approach for imputation and classification on incomplete data based on the Dirichlet process mixture model and naive Bayes classifier. The proposed method is free from the distribution assumption and can construct a flexible imputer and classifier simultaneously. The

| Ratio         | RF+L | RF+R | MICE+L | MICE+R | KNN+L | KNN+R | MIDA+L | MIDA+R | GAIN+L | GAIN+R | DPNB |
|---------------|------|------|--------|--------|-------|-------|--------|--------|--------|--------|------|
| 50% / 50%     | 0.9516 (3) | 0.6638 (9) | 0.8618 (6) | 0.0349 (11) | 0.8359 (7) | 0.7016 (8) | 0.6508 (10) | 0.9813 (2) | 0.8891 (5) | 0.9461 (4) | 1 (1) |
| 60% / 40%     | 0.9829 (3) | 0.7192 (8) | 0.8833 (6) | 0.0282 (11) | 0.8559 (7) | 0.7067 (9) | 0.6434 (10) | 0.9786 (4) | 0.9262 (5) | 1.0096 (1) | 1 (2) |
| 70% / 30%     | 0.9545 (5) | 0.7248 (9) | 0.8516 (6) | 0.0427 (11) | 0.8782 (7) | 0.7432 (8) | 0.7174 (10) | 1.0431 (3) | 1.0031 (2) | 0.9931 (4) | 1 (3) |
| 80% / 20%     | 1.0184 (1) | 0.6591 (10) | 0.9351 (6) | 0.0453 (11) | 0.9019 (7) | 0.7314 (9) | 0.7814 (8) | 0.9965 (4) | 1.0124 (2) | 0.9715 (3) | 1 (3) |
| 90% / 10%     | 0.9982 (2) | 0.6281 (10) | 0.9621 (6) | 0.0298 (11) | 0.8186 (9) | 0.7394 (8) | 0.7329 (9) | 0.9841 (3) | 0.9662 (5) | 0.9825 (4) | 1 (1) |

Average normalized F1-score 0.987 0.679 0.913 0.036 0.857 0.724 0.705 0.996 0.960 0.979 2.0

Average rank 2.8 9.2 6.0 11.0 7.0 8.4 9.4 2.8 3.8 3.6 2.0
flexibility and effect of the DPNB model are verified by various experiments. Moreover, the DPNB model suffers less from the overfitting problem, which frequently occurs using a flexible model, by considering proper priors. The DPNB model shows stable and better performance than other methods on experiments even missing rate is high. All methodologies, including statistical methods and machine learning models, can’t avoid large uncertainty in estimation when the data have high missing rates. However, we found that the DPNB model was less vulnerable to high missing rates than other methods through empirical experiments. As an improvement to the DPNB model, we would like to address the problem with computational time. Due to the limit of the MCMC algorithm, the DPNB model takes a longer time compared to other models by increasing the data size (see Table S4 and Table S5 in the Supplement for details). In future studies, we would like to propose ways to reduce computational time for the DPNB model such as variational methods.

REFERENCES

[1] D. B. Rubin, “Inference and missing data,” Biometrika, vol. 63, no. 3, pp. 581–592, 1976.
[2] R. J. Little and D. B. Rubin, Statistical Analysis With Missing Data, vol. 793, Hoboken, NJ, USA: Wiley, 2019.
[3] R. Wei et al., “Missing value imputation approach for mass spectrometry-based metabolomics data,” Sci. Rep., vol. 8, no. 1, pp. 1–10, 2018.
[4] S. V. Buuren and K. Groothuis-Oudshoorn, “Mice: Multivariate imputation by chained equations in R,” J. Stat. Softw., vol. 45, no. 3, pp. 1–68, 2010.
[5] Z. Ghahramani and M. I. Jordan, “Supervised learning from incomplete data via an EM approach,” in Proc. Adv. Neural Inf. Process. Syst., 1994, pp. 120–127.
[6] T. I. Lin, J. C. Lee, and H. J. Ho, “On fast supervised learning for normal mixture models with missing information,” Pattern Recognit., vol. 39, no. 6, pp. 1177–1187, 2006.
[7] D. Williams, X. Liao, Y. Yue, L. Carin, and B. Krishnapuram, “On classification with incomplete data,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 3, pp. 427–436, Mar. 2007.
[8] S. Das, S. Datta, and B. B. Chaudhuri, “Handling data irregularities in classification: Foundations, trends, and future challenges,” Pattern Recognit., vol. 81, pp. 674–693, Sep. 2018.
[9] L. F. Burgette and J. P. Reiter, “Multiple imputation for missing data via sequential regression trees,” Amer. J. Epidemiol., vol. 172, no. 9, pp. 1070–1076, 2010.
[10] D. J. Stekhonov and P. Buhlmann, “MissForest—Non-parametric missing value imputation for mixed-type data,” Bioinformatics, vol. 28, no. 1, pp. 112–118, 2012.
[11] O. Troyanskaya et al., “Missing value estimation methods for DNA microarrays,” Bioinformatics, vol. 17, no. 6, pp. 520–525, 2001.
[12] L. P. Brás and J. C. Menezes, “Improving cluster-based missing value estimation methods of DNA microarray data,” Biomolecular Eng., vol. 24, no. 2, pp. 273–282, 2007.
[13] P. K. Sharpe and R. Solly, “Dealing with missing values in neural network-based diagnostic systems,” Neural Comput. Appl., vol. 3, no. 2, pp. 73–77, 1995.
[14] A. Gupta and M. S. Lam, “Estimating missing values using neural networks,” J. Oper. Res. Soc., vol. 47, no. 2, pp. 229–238, 1996.
[15] E. L. Silva-Ramírez, R. Pino-Mejías, M. López-Ceolio, and M.-D. Cabiles-de-la Vega, “Missing value imputation on missing completely at random data using multilayer perceptrons,” Neural Netw., vol. 24, no. 1, pp. 121–129, 2011.
[16] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” Sci. Rep., vol. 8, no. 1, 2018.
[17] Y.-J. Kim and M. Chi, “Temporal belief memory: Imputing missing data during RNN training,” in Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI), 2018, pp. 1–7.
S. Liu and W. Deng, “Very deep convolutional neural network based classification and image retrieval using convolutional neural network,” IEEE Trans. Semicond. Manuf., vol. 31, no. 2, pp. 309–314, May 2018.

S. Cheon, H. Lee, C. O. Kim and S. H. Lee, “Convolutional neural network for wafer surface defect classification and the detection of unknown defect class,” IEEE Trans. Semicond. Manuf., vol. 32, no. 2, pp. 163–170, May 2019.

M. Saqlain, Q. Abbas, and J. Y. Lee, “A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes,” IEEE Trans. Semicond. Manuf., vol. 33, no. 3, pp. 436–444, Aug. 2020.

B.-S. Lin, J.-S. Cheng, H.-C. Liao, L.-W. Yang, T. Yang, and K.-C. Chen, “Improvement of multi-lines bridge defect classification by hierarchical architecture in artificial intelligence automatic defect classification,” IEEE Trans. Semicond. Manuf., vol. 34, no. 3, pp. 346–351, Aug. 2021.

S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” in Proc. IEEE 3rd IAPR Asian Conf. Pattern Recognit. (ACPR), 2015, pp. 730–734.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

M. Abd Al Rahman, S. Danishvar, and A. Mousavi, “An improved capsule network (WaferCaps) for wafer bin map classification based on DCGAN data Upsampling,” IEEE Trans. Semicond. Manuf., vol. 35, no. 1, pp. 50–59, Feb. 2022.

A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2015, arXiv:1511.06434.

T. T. Nguyen, H. D. Nguyen, F. Chamroukhi, and G. J. McLachlan, “Approximation by finite mixtures of continuous density functions that vanish at infinity,” Cogent Math. Stat., vol. 7, no. 1, 2020, Art. no. 1750861.

I. J. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

J. Zhang and R. Everson, “Bayesian estimation and classification with incomplete data using mixture models,” in Proc. IEEE Int. Conf. Mach. Learn. Appl., 2004, pp. 296–303.

J. Franzén, Bayesian Inference for a Mixture Model Using the Gibbs Sampler, Dept. Stat., Univ. Stockholm, Stockholm, Sweden, 2006.

J. L. Schafer, Analysis of Incomplete Multivariate Data. Boca Raton, FL, USA: CRC Press, 1997.

T. S. Ferguson, “A Bayesian analysis of some nonparametric problems,” Ann. Stat., vol. 1, no. 2, pp. 209–230, Mar. 1973.

D. Blackwell et al., “Ferguson distributions via Polya urn schemes,” Ann. Stat., vol. 1, no. 2, pp. 353–355, 1973.

D. J. Aldous, “Exchangeability and related topics,” in École d’Été de Probabilités de Saint-Flour XIII—1983 (Lecture Notes in Mathematics, 1117). Berlin, Germany: Springer, 1985, pp. 1–198.

J. Sethuraman, “A constructive definition of Dirichlet priors,” Statistica Sinica, vol. 4, no. 2, pp. 639–650, 1994.

C. E. Antoniak, “Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems,” Ann. Stat., vol. 2, no. 6, pp. 1152–1174, Nov. 1974.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical Dirichlet processes,” J. Amer. Stat. Assoc., vol. 101, no. 476, pp. 1566–1581, 2006.

R. M. Neal, “Markov Chain sampling methods for Dirichlet process mixture models,” J. Comput. Graph. Stat., vol. 9, no. 2, pp. 249–265, 2000.

H. Ishwaran and L. F. James, “Gibbs sampling methods for stick-breaking priors,” J. Amer. Stat. Assoc., vol. 96, no. 453, pp. 161–173, 2001.

S. G. Walker, “Sampling the Dirichlet mixture model with slices,” Commun. Stat. Simulat. Comput., vol. 36, no. 1, pp. 45–54, 2007.

S. Jain and R. M. Neal, “A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model,” J. Comput. Graph. Stat., vol. 13, no. 1, pp. 158–182, 2004.

D. M. Blei and M. I. Jordan, “Variational inference for Dirichlet process mixtures,” Bayesian Anal., vol. 1, no. 1, pp. 121–143, 2006.

T. Hastie and R. Tibshirani, “Discriminant analysis by Gaussian mixtures,” J. Roy. Stat. Soc. B Methodol., vol. 58, no. 1, pp. 155–176, 1996.

C. Fraley and A. E. Raftery, “Model-based clustering, discriminant analysis, and density estimation,” J. Amer. Stat. Assoc., vol. 97, no. 458, pp. 611–631, 2002.

H. Ge, Y. Chen, M. Wan, and Z. Ghahramani, “Distributed inference for Dirichlet process mixture models,” in Proc. Int. Conf. Mach. Learn., 2015, pp. 2276–2284.

C. Wang, X. Liao, L. Carin, D. B. Dunson, and D. Blei, “Classification with incomplete data using Dirichlet process priors,” J. Mach. Learn. Res., vol. 11, no. 12, pp. 3269–3311, 2010.

D. Dua and C. Graff, “UCI Machine Learning Repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml