Hindawi
Journal of Engineering
Volume 2023, Article ID 8557487, 10 pages
https://doi.org/10.1155/2023/8557487

Research Article

The Discriminants of Long and Short Duration Failures in Fulfillment Sortation Equipment: A Machine Learning Approach

Abed Mutemi and Fernando Bacao

NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, Lisboa 1070-312, Portugal

Correspondence should be addressed to Abed Mutemi; d20200455@novaims.unl.pt

Received 15 October 2022; Revised 2 March 2023; Accepted 24 March 2023; Published 15 April 2023

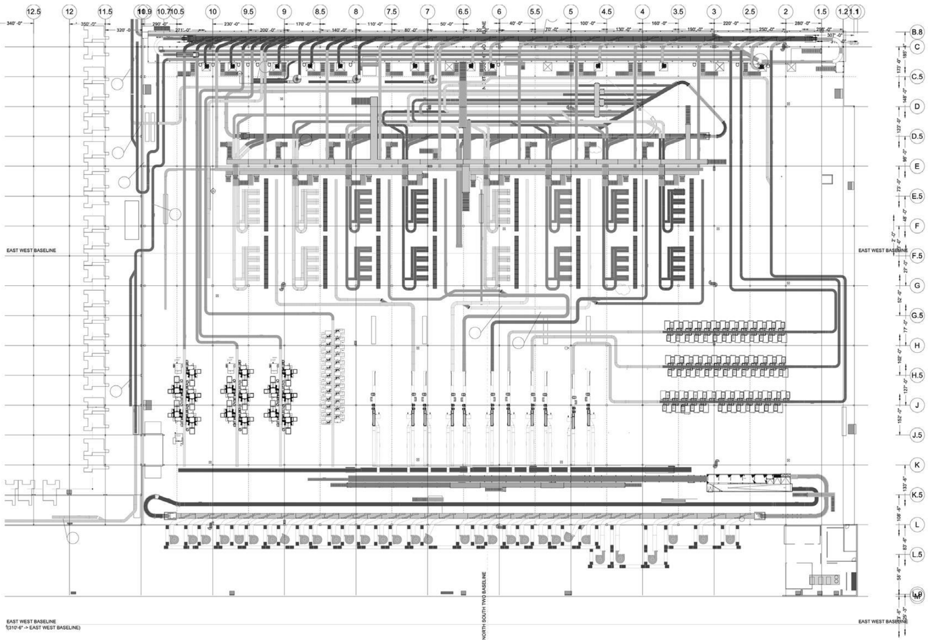
Academic Editor: Chong Leong Gan

Due to the difficulties inherent in diagnostics and prognostics, maintaining machine health remains a substantial issue in industrial production. Current approaches rely substantially on human engagement, making them costly and unsustainable, especially in high-volume industrial complexes like fulfillment centers. The length of time that fulfillment center equipment failures last is particularly important because it affects operational costs dramatically. A machine learning approach for identifying long and short equipment failures is presented using historical equipment failure and fault data. Under a variety of hyperparameter configurations, we test and compare the outcomes of eight different machine learning classification algorithms, seven individual classifiers, and a stacked ensemble. The gradient boosting classifier (GBC) produces state-of-the-art results in this setting, with precision of 0.76, recall of 0.82, and false positive rate (FPR) of 0.002. This model has since been applied successfully to automate the detection of long- and short-term defects, which has improved equipment maintenance schedules and personnel allocation towards fulfillment operations. Since its launch, this system has contributed to saving over $500 million in fulfillment expenses. It has also resulted in a better understanding of the flaws that cause long-term failures, which is now being used to build more sophisticated failure prediction and risk-mitigation systems for fulfillment equipment.

1. Introduction

Artificial intelligence advancements have resulted in smart devices that are now widely used in a variety of industries. These smart technologies, which range from robots to cameras to medical equipment to low-cost smart sensors, could help companies and industries achieve higher efficiency and effectiveness. AI is now being implemented outside of the data center, in various devices and machines, with processors designed to capture and process data at lightning rates while using minimal power and computing resources. Because of the growth of AI-powered smart gadgets that can detect and react to sights, sounds, and other patterns, pervasive intelligence is now being integrated into a wide range of practical applications. Machines are increasingly attaining high levels of performance through learning from their experiences, adjusting to changing settings, and forecasting events. While certain industries, such as aviation, have embraced these advancements, others are still catching up. Scaling fulfillment operations, for example, is a concern as the e-commerce business grows rapidly around the world. The overreliance on human input in decision-making is a major stumbling block to scale. The ability of humans to make rapid and efficient decisions is hampered by “information overload,” which includes too many tools to monitor and too many pages of best practices documentation to examine as input for maintenance decision support. Furthermore, the sheer magnitude of the equipment and structure that make up the fulfillment center complicates the implementation of these decisions. A modern Amazon warehouse, for example, is on average 800,000 square feet [1], with various forms of fulfillment equipment taking up more than half of that space.

A fulfillment sortation system (Figure1) is an automated warehouse sorting system meant to improve order picking, packing, consolidation, and shipping efficiency and



Color	TAGS	Description	Unit Count	Total Length
▨	1064	AFE Lower Rebin Fingers 49-64	0	0'
▨	1081	AFE Upper Rebin Fingers 01-16	0	0'
▨	1082	AFE Upper Rebin Fingers 17-32	0	0'
▨	1083	AFE Upper Rebin Fingers 33-48	0	0'
▨	1084	AFE Upper Rebin Fingers 49-64	0	0'
▨	1151	ARSP Discharge - Level 3	33	1012'-4"
▨	1161	ARSP Charge - Level 3	38	920'-9"
▨	1163	High Velocity Lvl 1 thru ARSP 2 Lvl 3	12	699'-1"
▨	1191	High Velocity Divert to Singles	21	726'
▨	1192	High Velocity Divert to Routing PreMerge	7	371'-5"
▨	1211	ARSP Level 1/2 - Routing PreMerge	0	0'
▨	1212	ARSP Level 1/2 - Outbound Merge Lines	0	0'
▨	1221	ARSP Level 3/4 - Routing PreMerge	3	591'-9"
▨	1222	ARSP Level 3/4 - Outbound Merge Lines	31	818'-5"
▨	1311	Routing Merge	30	1122'-4"
▨	1312	Routing Recirc.	4	314'-5"
▨	1321	Routing Sorter	3	534'-9"
▨	1322	Routing Induct	5	78'-5"
▨	1332	Routing T.A.s 01-17, Wrangle, J.P., Spir	28	301'-2"
▨	1342	Routing T.A.s 18-33, Wrangle, J.P., Spir	32	392'-3"
▨	1401	Singles Pack	31	1413'
▨	1402	Singles Pack Outbound	29	794'-9"
▨	1411	SmartPac Inbound	28	1188'-11"
▨	1412	SmartPac to Flats Collector	8	658'-4"
▨	1431	Gift Wrap	7	221'-6"
▨	1511	Shipping Sorter	0	0'
▨	1512	Shipping Induct	7	100'-1"
▨	1513	Shipping Takeway Junctions/Curves	0	0'
▨	1521	Shipping Recirc	11	601'-4"
▨	1531	Shipping Collector	4	335'-3"
▨	1561	Flats Collector and Intralox Sort	4	433'-6"
▨	1564	Crossbelt Area Infinity Conveyor	0	0'
▨	1801	AFE Upper Merge	1	76'-1"
▨	1802	AFE Upper Recirc.	11	403'-11"
▨	1811	AFE Upper Sorter	3	400'-8"
▨	1812	AFE Upper Sorter Induct	6	47'-2"
▨	1813	AFE Upper Takeways Junctions/Curves	17	211'
▨	1821	AFE Upper Inbound 51-58, MT Tote	34	836'-9"
▨	1822	AFE Upper PreMerge from Zippers 51-58	12	312'
▨	1823	AFE Upper Zipper Merge 51-52	20	352'-1"
▨	1824	AFE Upper Zipper Merge 53-54	18	299'
▨	1825	AFE Upper Zipper Merge 55-56	19	274'-5"
▨	1826	AFE Upper Zipper Merge 57-58	19	260'-1"
▨	1827	AFE Upper Jackpot	5	174'-5"
▨	1831	AFE Upper Inbound 59-66, MT Tote	43	1050'-11"
▨	1832	AFE Upper PreMerge from Zippers 59-66	12	312'
▨	1833	AFE Upper Zipper Merge 59-60	14	181'-2"
▨	1834	AFE Upper Zipper Merge 61-62	12	167'-3"
▨	1835	AFE Upper Zipper Merge 63-64	13	219'-7"
▨	1836	AFE Upper Zipper Merge 65-66	13	289'-10"
▨	1841	AFE Upper Rebin 51 & 52	102	762'-7"
▨	1851	AFE Upper Rebin 53 & 54	102	762'-6"
▨	1861	AFE Upper Rebin 55 & 56	102	763'-1"
▨	1871	AFE Upper Rebin 57 & 58	102	762'-6"
▨	1881	AFE Upper Rebin 59 & 60	102	762'-4"
▨	1891	AFE Upper Rebin 61 & 62	102	758'-3"
▨	1901	AFE Upper Rebin 63 & 64	0	0'
▨	1911	AFE Upper Rebin 65 & 66	208	1549'-7"
▨	1921	AFE Upper Outbound (Rebins 51-54)	24	476'-1"
▨	1922	AFE Upper Outbound (Rebins 55-58)	21	428'-6"
▨	1931	AFE Upper Outbound (Rebins 59-62)	28	540'-2"
▨	1932	AFE Upper Outbound (Rebins 63-66)	19	410'-11"
▨	1334	Routing Spirals	1	3'-4"

FIGURE 1: Material flow from upstream order picking processes to downstream order packing and shipping on a fulfillment sorter (source: dematic sorters [2]).

accuracy. This equipment set is distinguished by several components that aid in the performance of its functions. Even with advanced maintenance practices, the sortation equipment will fail due to ongoing degradation from continuous operation due to its complexity. In these cases, the ability to quickly restore equipment makes all the difference in meeting customer orders as promised. As a result, the ability to swiftly identify and pinpoint failure causes (faults) in this crucial piece of fulfillment equipment is critical for ensuring a steady flow of order units. Furthermore, the ability to predict failure lengths at various levels is important for risk mitigation and cost reduction because such signals are crucial inputs.

The goal of this research is to create a machine learning framework for making maintenance choices and resolving faults on fulfillment sortation equipment. We do this by combining past failure and fault data from condition-monitoring systems with AI-powered sensors put along the equipment's length. The sensors continuously monitor the functioning of numerous components that make up the equipment system by recording data such as vibration, current, order traffic, weight, acoustic signals, temperature, and moving component speeds, among other signals. The machine learning framework uses these data to find a classifier that can distinguish between long and short duration failures while also finding factors that are strongly linked to those failures. Long failures are defined as those that last longer than 15 minutes and are assumed to be more essential fulfillment operations in terms of cost of operations.

By incorporating machine learning into traditional condition-based or time-based maintenance decision systems, equipment maintenance and reliability teams may be able to optimize and scale maintenance routines by precisely targeting fault resolution. These activities have the potential to greatly reduce unplanned downtime, which has direct costs associated with maintenance procedures and indirect costs associated with missed output, shipping, and labor [3]. Maintenance costs have risen steadily over time; depending on the industry, maintenance costs account for 15–70 percent of total production costs [4]. It could also help with a better knowledge of the flaws associated with protracted failures, which could lead to better fulfillment sortation designs in the future.

When sortation equipment malfunctions in the existing state, it affects the flow of orders, resulting in a cascade of cost overruns in both upstream and downstream operations. Identifying and isolating defects linked with a failure is a time-consuming process. An operator must wander around the apparatus to discover and repair flaws that have caused or will cause a failure. When there are many problems present, the issue becomes much more complicated, and the operator must decide which defect is the most vital to correct first. In this case, the proposed approach wins on two counts: (i) the operator receives a prioritized list of faults to rectify thanks to automatic identification of the most critical defects related with the present failure. This essentially contributes to labor cost savings; (ii) with the directed workflow, the operator can quickly fix the fault and get the equipment back up and running with minimal latency, while maintaining the flow of both upstream and downstream processes to minimize the effects of unplanned downtime.

When long-term failures are unavoidable, the model predicts them, and the company can take risk-mitigation measures like rerouting orders to other equipment or facilities.

The envisioning takes on a basic maintenance decision support system and incorporates a machine learning application layer that creates the capability to simplify, standardize, direct, or automate maintenance decisions, while the integration of machine learning solutions takes place in a broader context outside the scope of this paper. We see the system as having three key components: (i) the environment, (ii) the brain, and (ii) the body.

The environment is a collection of infrastructure that includes a data historian and an event stream processor, with the goal of storing, normalizing, and providing machine operating data from sortation equipment.

The brain is the machine learning program that recognizes, learns, and predicts defects and failure patterns captured in the environment. The brain's purpose is to eliminate the operator's responsibility for identifying and prioritizing errors and failure problems.

The body is the user interface that allows the operator to interact with the system directly, obtain insights, and take maintenance actions. The body's purpose is to reduce clutter and labor strain by only displaying high-priority failures that have a major influence on sortation operations.

The experiments in this study support the environment and brain activity by drawing data for machine learning algorithms using developments in AI sensors and smart devices. Sensor readings from sensors throughout the length of the equipment are used to provide input signals for defect and failure instances to the historian and event stream processor, which are then employed in the machine learning context to anticipate failure outcomes. The data are fed into body applications, which I immediately inform the operator's decision to prioritize and repair the most critical faults and (ii) integrate with more complex equipment health decision support systems. A reliability maintenance operator eats the brain's output in the form of alerts in the grand scheme of things. Each alert comes with a step-by-step workflow for quickly mitigating the danger of a potential failure or resolving the problem if self-healing isn't possible.

The peculiarity of this technique is that it takes a labor-intensive equipment maintenance strategy like the CBM and reduces the strain on labor by using machine learning to produce efficiencies that can improve fulfillment operations. This issue has not been solved in fulfillment sortation equipment to the best of our knowledge. Similar methodologies have, however, been used in other practical domains, such as the aviation industry [5], to investigate the failure of aircraft components. Other researchers have investigated the use of machine learning algorithms for defect diagnosis, particularly for individual components like bearings ([6–8] and [9]).

The other portions of the paper are divided as follows: Section 2 provides an overview of background and relevant work, Section 3 provides the methodology, Section 4 provides the results and discussion, and Section 5 concludes the paper.

2. Associated Work

2.1. Maintenance Strategies for Equipment. The term maintenance is well defined in the literature; nevertheless, several maintenance-related phrases have ambiguous definitions, and thus, we use the following definitions:

Maintenance is defined as "the combination of technical and associated administrative actions intended to retain an item or a system in, or restore it to, a state in which it can perform its required function," according to ISO 14224, 2006 [10]. Another variation of this definition is "all actions appropriate for retaining an item/part/equipment in, or restoring it to, a given condition," according to literature [11].

In Figure 2, we present a list of the important terminology used in the equipment maintenance context, as well as a brief description of what each one includes.

Corrective and preventive maintenance are the two main types of equipment maintenance [12]. Corrective maintenance helps manage repair actions and unscheduled fault events, such as equipment and machine failures. When sortation equipment components fail in use, they are repaired or replaced. Preventive maintenance involves periodic maintenance protocols to avoid equipment failures or machinery breakdowns [13].

Condition-based maintenance (CBM) and time-based maintenance are two types of preventive maintenance (TBM). TBM is defined as a traditional preventive maintenance approach that involves time-based execution of maintenance protocols sourced from failure time analysis [14], whereas CBM is defined as a preventive maintenance approach based on performance and/or parameter monitoring and subsequent actions [15, 16].

CBM can be carried out at various levels of technology, with the goal of gathering condition data being to detect incipient failure so that maintenance operations can be scheduled or carried out continually. CBM can also be broken down into three basic steps: data collection, data processing, and maintenance decision-making as shown in Figure 3.

Diagnostics and prognostics are two further classifications within CBM, with the former dealing with fault detection, isolation, and identification after it occurs and the latter with fault prediction before it occurs. Because there is a large and diverse literature on machinery diagnostics and prognostics due to the wide variety of systems, components, and parts, rather than going through the litany of articles published in this area, we will limit our review to fault diagnostics because it is relevant to this work, but first we will highlight the three main steps mentioned above.

For the purposes of CBM, data acquisition is the act of gathering and storing relevant information from specified physical assets. Obtaining these data can be difficult; for example, obtaining failure data and labeling it in practice can be tough. Another issue is that the volume of data to be handled is frequently vast, necessitating specialized infrastructure, specialist knowledge, and, in certain circumstances, unique software [17]. A third issue is that equipment makers are very protective of their raw data and will only give postprocessed aggregates in limited quantities. As a result, researchers must deal with these difficulties. The primary goal of CBM data collecting is to gather information on events that occur around the equipment. These could involve a variety of repairs, breakdowns, health issues, and maintenance procedures. Sensors such as microsensors,

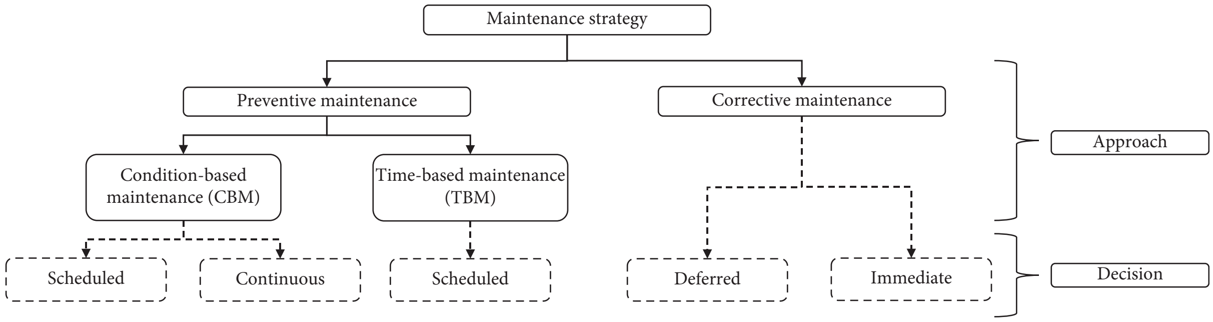


Figure 2: Common concepts and procedures in reliability maintenance.

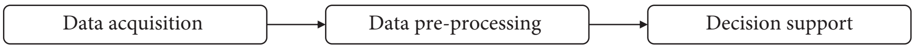


Figure 3: Condition based maintenance processes.

acoustic emission sensors, ultrasonic sensors, and others acquire condition monitoring data such as vibration data, motor temperature, acoustic data, pressure, moisture, humidity, and weather, while events data are collected in both manual and automatic mechanisms [16].

Because data acquired through any system is prone to errors, the first stage in this procedure is to cleanse the data to remove various forms of errors (for example, data entry errors for manual processes and sensor faults for automated processes). Several ways to cleaning sensor data are discussed in [18, 19]. These actions are referred to as data processing.

Processing sensor readings into conditions/states is also an important part of this stage. Typically, the data belong into the following types: (i) value type: single value data collected at a specific time epoch for a condition monitoring variable, e.g., temperature, pressure, (ii) waveform: time series data collected at a specific time epoch for a condition monitoring variable, e.g., vibration, acoustic, and (iii) multidimensional type: multidimensional data collected at a specific time epoch for a condition monitoring variable, e.g., image data such as thermographs, X-ray image, and visual images.

Signal processing is the technical term for waveform and multidimensional data processing, as well as the numerous approaches that analyze and interpret these data types to extract usable features for diagnostic and prognostic purposes. Image processing techniques are an important part of data preprocessing since photographs of equipment components, as well as images from some waveform processing, such as time-frequency analysis, can be a significant source of condition monitoring data. Wang and McFadden [20], Utsumi et al. [21], Heger and Pandit [22], and Ellwein et al. [23] are some examples of using image processing techniques in condition monitoring. Nixon and Aguado [19], as well as Xu and Kwan [18], have applied complicated imaging algorithms to extract characteristics in this area. There are numerous signal processing algorithms for mechanical systems in the literature, and the best one depends on the application area [16].

2.2. Decision-Making Assistance. Any equipment maintenance strategy must include maintenance decision assistance. In this step, the data have been collected, and preprocessed is transformed into insights that guide decision-making and maintenance. Diagnostics and prognostics are the two primary types of decision assistance in the CBM maintenance strategy [11].

2.2.1. Diagnostics. Diagnostics is the process of converting data from the measurement space into machine faults in the fault space [16]. This mapping procedure, also known as pattern recognition, is usually done manually using data analysis techniques such as power spectrum, phase spectrum graph, cepstrum graph, AR spectrum graph, spectrogram, wavelet scalogram, and wavelet phase graph, according to these authors. They also advocate for automating the process due to the increased demand for highly skilled workers. According to Williams et al. [24] and Korbicz et al. [25], automatic pattern recognition can be accomplished through classification of data using statistical learning techniques (e.g., machine learning) or artificial intelligence (AI) methodologies.

(1) Statistical Methods. Fault diagnostics entails determining whether a certain fault exists based on condition monitoring data. The first statistical approach is to formulate the detection problem as a hypothesis test. The null hypothesis may be written as "fault is not present," and it would be evaluated against the alternative hypothesis of "fault is present." Test statistics are used to summarize condition monitoring data to determine whether the null hypothesis should be rejected or not. This method is covered in depth by Ma and Li [26], Sohn et al. [27], and Kim et al. [28].

Statistical process control (SPC) is a second statistical method that originated in control theory and has since become extensively accepted and used in fault detection and diagnostics. Its principle is to evaluate whether a current signal is within control limits by measuring its deviation from a reference signal indicating the normal situation [29].

Cluster analysis, for example, is a univariate or multivariate method that clusters signals into distinct fault categories based on their similarity of feature properties. It finds natural clusters of signals that can be utilized to infer a malfunction using distance or similarity measurements. Euclidean distance, Mahalanobis distance, Kullback–Leibler distance, and Bayesian distance are all common distance approaches. Artes et al. [30] and Schurmann [31] investigated the use of cluster analysis methodologies. Goumas et al. [32] and Lou and Loparo [33] are two papers that investigate the usage of distance measures for fault diagnosis.

(2) Approaches to Machine Learning. SVMs (support vector machine) are supervised learning models that analyze data for classification or regression analysis [34]. The algorithm's goal is to find a hyperplane in an n-dimensional space that categorizes the data points clearly. Support vectors are data points that are close to the hyperplane and have an influence on the hyperplane's orientation and location. These support vectors are utilized to increase the classifier's margin. The use of SVM in this field has been discussed in [35, 36]. Machine learning algorithms have been used in fault diagnostics to determine whether an item should be repaired or replaced [17]. By learning patterns from torque sensors, Elliot et al. [37] employ a random forest classifier to identify the failures of a robotic arm. A random forest, also known as random choice forests, is a classification and regression ensemble learning method that works by generating many decision trees during training. For classification tasks, the random forest's output is the class chosen by most trees. Individual trees' mean predictions are returned for regression [38].

Another often used approach in this field for fault classification and diagnostics is the hidden Markov model (HMM) [37, 39]. In the HMM's early applications, the real machine defective states and normal machine states were treated as hidden states. In hidden states of the HMM, other versions have no physical meaning. The trained HMMs are then utilized to decode an observation with an unknown machine condition to classify faults.

(3) Approaches to AI. Artificial intelligence is referred to by the abbreviation AI. It refers to the realm of "intelligent agents," which includes any system that detects its surroundings and takes activities to increase its chances of achieving its objectives [40]. Artificial neural networks (ANNs) and expert systems (ESs) are two common AI approaches for machine diagnosis [16]. Fuzzy logic systems, fuzzy neural networks (FNNs), neural-fuzzy systems, and evolutionary algorithms are some of the other AI methodologies used (EAs). Siddique et al. [41] provide an overview of recent breakthroughs in AI applications.

2.2.2. Prognostics. The goal of prognostics is to forecast when a system or component will no longer perform its intended function [42]. The most common prognostics are used to estimate how much time is left before a failure (one or more defects) occurs, based on the present machine condition and previous operating profile, also known as remaining usable life (RUL). Prognostics, according to some academics, are superior to diagnostics since it prevents defects or failures from developing. However, not all failures are preventable or anticipated in practice, therefore, prognostics cannot fully replace diagnostics. A diagnostics tool is useful in the event of a faulty prediction, in addition to giving maintenance decision support. Furthermore, diagnostic data can be used to provide feedback for system redesign. We do not go into detail about prognostics because it is not important to this paper.

3. Methodology

The data, evaluation metrics, machine learning techniques, experimental procedure, and software implementation are all described in this part. The flow of these processes is depicted in Figure 4.

3.1. Data from the Fulfillment Sorter. The sortation data are a collection of failure events and current sorter component fault states from the past. The failure events are stated in terms of the duration of the failure, and that is, the amount of time the sorter was down. The defects data are preprocessed sensor readings that show the state of several sorter subcomponents that are critical to the sortation equipment's proper operation.

Data are collected from several sensor devices mounted along the primary sortation equipment system by a data collection system. The sensors provide real-time information on the state of the sorter system and its components. A preprocessor is applied to the raw sensor readings in each case to generate fault features, with the process' output being a binary feature indicating whether a specific component of the sorter is faulty. Another system collects data on all primary sorter failure events, and a preprocess builds a binary feature for short vs. long downtime occurrences. The two streams are then matched, resulting in a final data set that is pushed to the simple storage service (s3) for machine learning applications.

As mentioned in [43], we define an interval P-F as the time interval between a possible failure (P) recognized by a fault condition indicator and a functional failure (F) of sortation equipment. For the investigation, this procedure generated roughly 22,300 failure events with failure lengths ranging from 1 minute to 2000 minutes and 59 sorter subcomponent fault states (features). The complete data set contains 211 minority class cases and little over 22,000 majority class cases, with our interest in forecasting failure events lasting more than 15 minutes.

3.2. Selection of Features. Given the large number of categorical variables in this data collection, the "curse of dimensionality" problem is likely. As a result, we use a filter strategy to find the optimum combination of features that gives us the best classifier results. This reduces the effects of overfitting and long training cycles. This set was chosen using the relief method. It is a feature selection approach that calculates feature weights using a random selection of

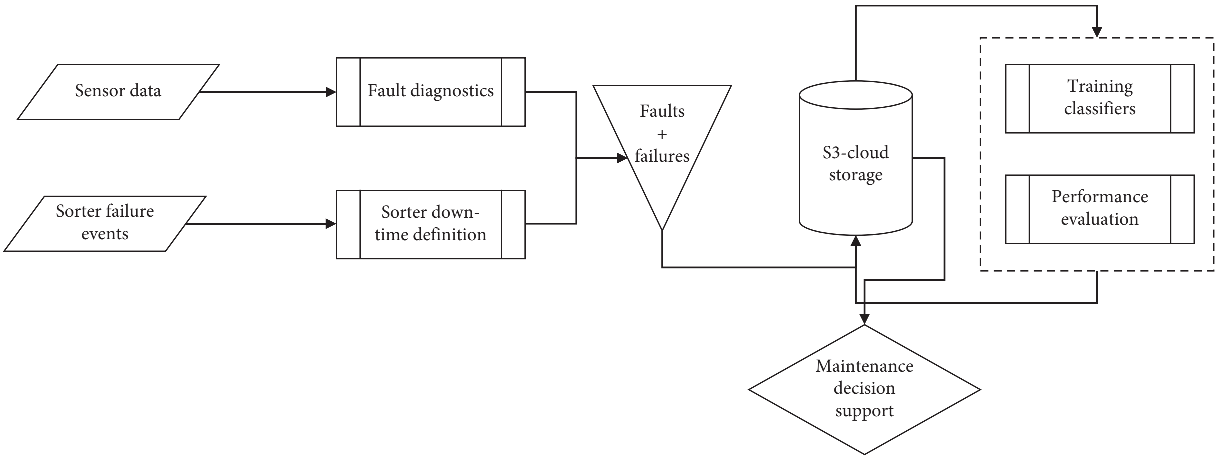


FIGURE 4: Machine learning architecture for fulfillment sorter equipment showing the data flow from acquisition to decision support predictions.

examples [44, 45]. It iteratively calculates feature weights based on their ability to distinguish between neighboring models.

3.3. Algorithms for Machine Learning. We chose 7 machine learning methods for our application from similar work in this area. On our data set, we want to tune each algorithm for the optimum hyperparameter combination. We also use ensemble techniques to see if they boost performance beyond what individual classifiers can achieve. Evaluation of each classifier's performance using the evaluation metrics is listed as follows.

We use logistic regression [46], k-nearest neighbors [47], support vector machines [34], decision trees [48], random forest [49], Naive bayes [50], and gradient boosting classifier [51] to train our classifiers. These methods are all suitable for binary classification and are extensively used in machine diagnosis and prognosis.

3.4. Measures of Evaluation. In this area, we apply standard evaluation metrics [52]. Accuracy, precision, recall, and the F-score are among them. A sample of metrics and their equations are shown in Table 1.

We also present the false positive rate (FPR), often known as the fall out rate, which is an important indicator in this context. It shows the likelihood of a false alarm, which is crucial in the commercial world. Because a false alert has a cost, if our classifier creates many false alarms, it would be inappropriate for use in the production system.

3.5. Setups for Experiments. Our experiment is divided into two halves. The first section employs a grid search strategy and a repeated stratified k-fold cross-validation design, as well as a hyperparameter tweaking approach. The data are divided into k-groups for each of the seven classifiers listed in Table 2, $k = 5$ in our example, with the value of k determined by the literature in the field.

$k - 1$ sets of data are used for training in each training iteration, while the rest is used for validation. For our binary problem setting, the groups are created while maintaining the composition of the classes, and each classifier is trained k times.

We have a 5-fold cross-validation with $k = 5$. Set 1, set 2, set 3, set 4, and set 5 are the five groups of data (see Figure 5): set 1, set 2, set 3, set 4, and set 5. The algorithm is trained a total of five times. Sets 1 through 4 are used as the training set, and set 5 is used as the validation set in the first iteration, whereas sets 1, 2, 3, and 5 are used as the training set, and set 5 is used as the test set in the second iteration. This approach is repeated until all the training and testing sets have been used. To reduce sample selection error, the data are shuffled at random before each split. A voting method is used to total up the skill of each algorithm across all iterations, as evaluated by their separate validation scores on the validation set.

The test set is then used to evaluate the trained classifier's performance in a production-like environment, as seen as follows.

The experiment's second phase employs an ensemble technique, stacking the seven classifiers mentioned previously. We use layered generalization, which includes merging predictions from different machine learning models on the same data set, such as bagging and boosting. We do this to answer the question of how to choose from various machine learning models that are skilled at solving a problem in different ways. A stacking model's architecture consists of two or more base models, also known as level-0 models, and a meta-model that combines the predictions of the base models, also known as level-1 models. In our case, the meta-model is trained using predictions from the basis models on the hold out data set. The input and out pairs of training data set used to fit the meta-model are the predictions and expected outputs. We use k-fold cross-validation of the basic models, with the out-of-fold predictions serving as the foundation for the training data set. Figure 6 illustrates this architecture.

Table 1: Common classification model evaluation metrics in the equipment reliability context.

Metric	Formula	Description
Accuracy (acc)	acc = (tp + tn/tp + tn + fp + fn)	The ratio of correct predictions by all predictions made
Precision (p)	p = (tp/tp + fp)	The ratio of correct positive predictions by all positively predicted classes
Recall (r)	r = (tp/tp + fn)	The ratio of correct positive predictions by all true positive classes
f1-score (f1)	$f = 2 * (p * r/p + r)$	The harmonic mean between precision and recall
Fall out rate (fpr)	fpr = (tp/tp + tn)	The probability of a false alarm
Error rate (err)	err = (fp + fn/tp + fp + tn + fn)	The ratio of incorrect predictions by all predictions made

Table 2: Common classifiers in machines and equipment failure modelling.

Classifiers	Hyperparameters	Values
Logistic regression (LR)	max_iter	500
	Classifier penalty	[None, l1, l2, “elastic net”]
	Classifier c	[100, 10, 1.0, 0.1, 0.01]
	Classifier solver	[“Liblinear,” “newton_cg,” “libfgs”]
k-Nearest neighbor (KNN)	Number of neighbors	[1, 22]
	Metric	[“Euclidean,” “manhattan,” “minkowski”]
	Weights	[“Uniform,” “distance”]
Support vector machines (SVM)	Kernels	[“Linear,” “poly,” “rbf,” “sigmoid”]
	Classifier	[0.05, 0.1, 0.5, 0.7, 1]
	Gamma	[0.05, 0.1, 0.5, 0.7, 1]
Decision trees (cart)	Criterion	[“gini”]
	max_depth	[2, 3, 4, 5]
Random forest (RF)	max_features	[1 to 20]
	n_estimators	[10, 100, 1000]
Naïve bayes (GNB)	Cv	[n_splits = 5]
Gradient boosting (GBC)	n_estimators	[1, 2, 4, 8, 16, 32, 64, 100, 200, 300, 500,1000,10000]
	max_depth	[1, 40]
	learning_rate	[1, 0.5, 0.25, 0.1, 0.05, 0.01]

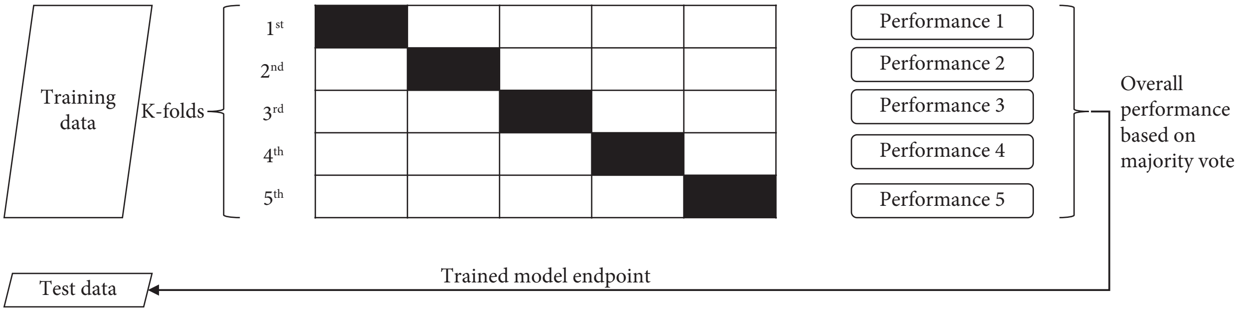


Figure 5: Five fold cross-validation process and a holdout set.

3.6. Implementation of Software. We use Scikit–Learn [53] in conjunction with other common Python libraries such as NumPy, Pandas, matplotlib, seaborn, and SciPy to build the experimental approach in Python.

4. Discussions and Findings

The performance of the various classifiers is presented in this section. First, we will go over the first half of the experiment's findings. Table 3 summarizes the results, which show that all of the classifiers achieve good accuracy, precision, recall, and F1 values. In conclusion, precision scores vary between 0.72 and 0.76, recall scores between 0.77 and 0.82, and accuracy scores between 0.77 and 0.82. The cross-validation ratings for each classifier are summarized in Table 3.

We employ stacking in the second half of the experiment because all of our base classifiers have skill on our data set, but they make distinct assumptions about how to handle the predictive modelling assignment in various ways. We choose a basic meta-model (logistic regression) to aggregate the predictions of the other models, even though some of the basis models are quite sophisticated. The accuracies of the

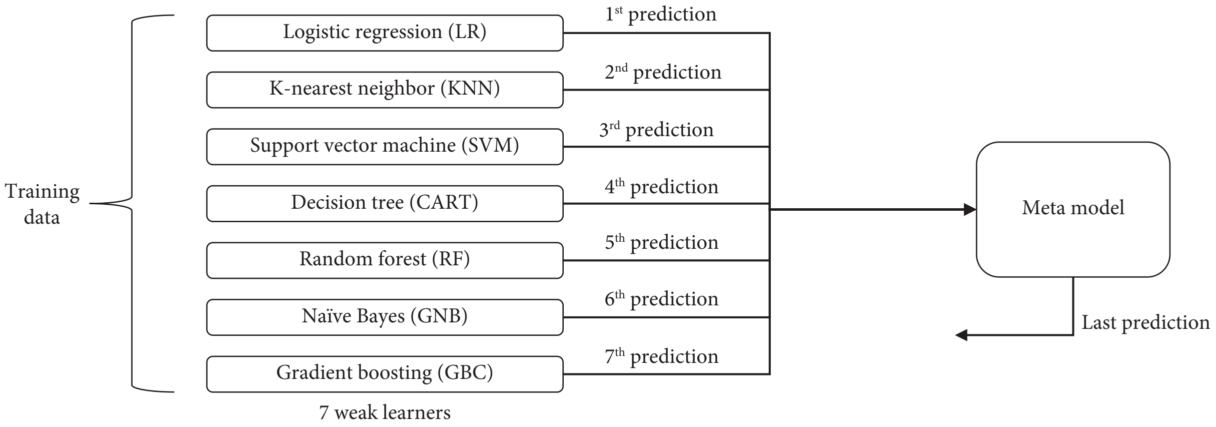


Figure 6: Stacked generalization architecture with 7 weak learners and one final metal-model.

Table 3: Overall cross-validation scores from the seven classifiers.

Classifier	Precision	Recall	Fpr
Logistic regression	0.73	0.82	0.003
K-Nearest neighbor	0.75	0.81	0.021
Support vector machine	0.74	0.82	0.0008
Decision tree (CART)	0.76	0.82	0.012
Random forest (RF)	0.76	0.82	0.0009
Naïve bayes (GNB)	0.72	0.77	0.086
Gradient boosting	0.76	0.82	0.002

seven classifiers are plotted in a Box–Whisker plot in Figure 7. The outcomes are consistent with what we have seen thus far.

After that, we use a stacked ensemble to see whether it can increase learning on our data beyond what individual classifiers can do. In Figure 8, we compare the ensemble's performance to that of the other classifiers. Stacked generalization's accuracy is equivalent to that of the gradient boosting classifier (GBC), which is not surprising given that GBC is another ensemble that uses a boosting method. All the other classifiers continue to work as expected. There is a notable gap in performance between the Naïve bayes classifier and the rest of the classifiers. We argue that this gap is as a result of several factors that distinctly affect this classifier compared to the rest in the group. First, Naïve bayes assumes that features are independent of each other, which might be violated given the nature of the condition monitoring. Second, failure events last between 1 minute and 2000 minutes creating an opportunity for outliers to adversely affect the performance of the classifier, and finally, the class imbalance within our data may also be affecting its performance. The Naïve bayes classifier is more sensitive to imbalanced data sets compared to the other classifier in this group.

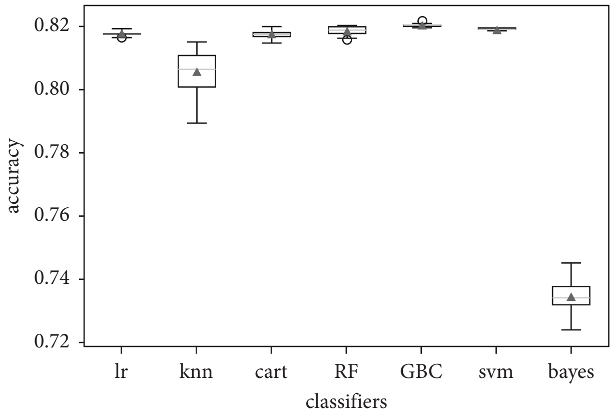


Figure 7: Box–Whisker plots comparing the performance of the classifiers.

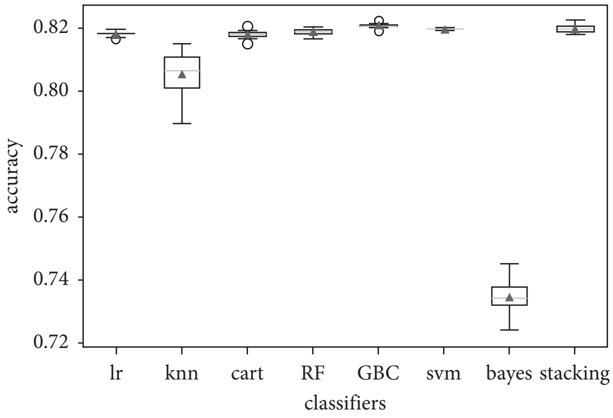


Figure 8: Comparing the performance of the seven classifiers and the stacked generalizer.

This experiment's potential to guide practical business decisions is a significant outcome. The false positive rate is one statistic that aids in achieving this goal. In machine operations and the business at large, this statistic has a cost meaning. It offers us an idea of how often our machine learning solution generates false alarms. The implication is that the system warns that the equipment will be down for more than 15 minutes, even when this is not the case. If a response is taken in response to such an alert, it merely means that the company is spending an unjustified expenditure. In resource-constrained contexts, such judgments can be extremely costly, especially if the rate is high.

The FPR rates produced by all of our classifiers are modest, ranging between 0.0008 and 0.08, though no false alarms are desirable.

In conclusion, tree-based algorithms and stacked ensemble outperform their equivalents. Given the tight competition for first place in terms of performance among six of the eight classifiers (lr, cart, RF, GBC, SVM, and Stacking), production decisions must be made on more than just accuracy, precision, and recall. Other factors, such as compute resource demand and prediction latency, must be considered when determining which classifier to advance to production in our machine operations management decision support system.

Finally, SVM stood out as a regularly used classifier in this context in the literature [35], and perhaps, the reason behind its preference in this context lies in its ability to model failure data effectively. It has an extremely low rate of false positives. Its precision and recall numbers compare favorably to those of the other high-performing algorithms in this group.

5. Conclusion and Work in the Future

We developed a data-driven machine learning approach for recognizing, isolating, and alerting on sortation equipment failures caused by malfunctioning sorter subcomponents in this paper. On the data set, our experiment examines the performance of seven classifiers, various hyperparameter combinations, and ensembles. We draw some key inferences from the findings which have theoretical and practical commercial implications.

The first discovery has far-reaching business consequences. Six of the eight trained classifiers achieve good overall scores, and any of them might be put into production to generate alerts that help with equipment maintenance choices. These findings are extremely beneficial to the company since they have a high potential for distinguishing between long- and short-term failures, which translates to making the right judgments most of the time. As a result, significant cost savings in terms of labor, material, parts, and other costs connected with equipment failure (about $500 million in savings) are realized.

Second, we find that six out of the eight classifiers achieve good performance. The six include the linear regression, cart (decision tree), random forest, gradient boosted trees, support vector machine and the stacked ensemble. In literature, SVM is touted as a good performer in this context and our results confirm that finding.

Finally, we find that different fault combinations result in diverse failure durations, with long duration failures being less common than short duration failures, posing a class asymmetry concern. Based on this discovery, we believe that the asymmetry issue is likely affecting the performance of some of our current classifiers, particularly the naïve Bayes classifier which starkly performs lower than the rest of the classifiers in our research. As such, future research and implementation should incorporate imbalanced learning techniques.

Data Availability

The data supporting the results in this manuscript are available and can be made available upon request and signing of an NDA.

Disclosure

The authors' opinions in this paper are their own and do not represent the institutions they represent in any way.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] BigRentz, "Mapping Amazon Warehouses: How Much Square Footage Does Amazon Own?," 2022, https://www.bigrentz.com/blog/amazon-warehouses-locations.

[2] D. Dematic, "Smart Task Activation Processing Maximizes Order Fulfillment Efficiency," 2020, https://www.dematic.com/en-us/insights/articles/smart-task-activation-processing-maximizes-order-fulfillment-efficiency/.

[3] ManagerPlus, "The True Cost of Equipment Failure Will Shock You," 2020, https://managerplus.iofficecorp.com/blog/equipment-failure-cost.

[4] R. S. Velmurugan and T. Dhingra, "Maintenance strategy selection and its impact in maintenance function," *International Journal of Operations & Production Management*, vol. 35, no. 12, pp. 1622–1661, 2015.

[5] K. Celikmih, O. Inan, and H. Uguz, "Failure Prediction of Aircraft Equipment Using Machine Learning with a Hybrid Data Preparation Method," *Scientific Programming*, vol. 2020, Article ID 8616039, 10 pages, 2020.

[6] J. McBain and M. Timusk, "Feature Extraction for novelty Detection as Applied to Fault Detection in Machinery," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 1054–1061, 2011.

[7] X. Zhang, B. Wang, and X. Chen, "Intelligent Fault Diagnosis of Roller Bearings with Multivariable Ensemble-Based Incremental Support Vector Machine," *Knowledge-Based Systems*, vol. 89, pp. 56–85, 2015.

[8] Y.-X. Cao, X.-J. Li, and L.-L. Jiang, "Fault Diagnosis of Motor Rotor Based on Fuzzy C-Means Clustering Analysis," *Applied Mechanics and Materials*, vol. 273, pp. 409–413, 2013.

[9] C. S. Chen and J. S. Chen, "Rotor Fault Diagnosis System Based on sGA-Based Individual Neural Networks," *Expert Systems with Applications*, vol. 38, no. 9, pp. 10822–10830, 2011.

[10] B. S. Dhillon, *Engineering Maintenance: Amodern Approach*, CRC Press, Boca Raton, FL, USA, 2002.

[11] P. Gackowiec, "General overview of maintenance strategies – concepts and approaches," *Multidisciplinary Aspects of Production Engineering*, vol. 2, no. 1, pp. 126–139, 2019.

[12] Q. Fan and H. Fan, "Reliability analysis and failure prediction of construction equipment with time series models," *Journal of Advanced Management Science*, vol. 3, pp. 203–210, 2015.

[13] P. Bastos, I. Lopes, and L. Pires, "A maintenance prediction system using data mining," *World Congress on Engineering*, vol. 3, no. 3, 2012.

[14] R. Ahmad and S. Kamaruddin, "An overview of time-based and condition-based maintenance in industrial application," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 135–149, 2012.

[15] P. Kumar and S. Rk, "Development of a condition based maintenance architecture for optimal maintainability of mine excavators," *IOSR Journal of Mechanical and Civil Engineering*, vol. 11, no. 3, pp. 18–22, 2014.

[16] A. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[17] Z. Kang, C. Catal, and B. Tekinerdogan, "Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks," *Sensors*, vol. 21, no. 3, p. 932, 2021.

[18] R. Xu and C. Kwan, "Robust isolation of sensor failures," *Asian Journal of Control*, vol. 5, no. 1, pp. 12–23, 2003.

[19] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, 2002.

[20] W. J. Wang and P. D. McFadden, "Early detection of gear failure by vibration analysis II. Interpretation of the time––frequency distribution using image processing techniques," *Mechanical Systems and Signal Processing*, vol. 7, no. 3, pp. 205–215, 1993.

[21] S. Utsumi, Z. Kawasaki, K. Matsu-Ura, and M. Kawada, "Use of wavelet transform and fuzzy system theory to distinguish wear particles in lubricating oil for bearing diagnosis," *Electrical Engineering in Japan*, vol. 134, no. 1, pp. 36–44, 2001.

[22] T. Heger and M. Pandit, "Optical wear assessment system for grinding tools," *Journal of Electronic Imaging*, vol. 13, no. 3, pp. 450–461, 2004.

[23] C. Ellwein, S. Danaher, U. Jager, and Jager, "Identifying regions of interest in spectra for classification purposes," *Mechanical Systems and Signal Processing*, vol. 16, no. 2-3, pp. 211–222, 2002.

[24] J. H. Williams, A. Davies, and P. R. Drake, *Condition-Based Maintenance and Machine Diagnostics*, Chapman Hall, London, UK, 1994.

[25] J. Korbicz, J. M. Koscielny, and Z. Kowalczuk, *Fault Diagnosis: Models, Artificial Intelligence, Applications*, Springer, Berlin, Germany, 2004.

[26] J. Ma and J. C. Li, "Detection of localised defects in rolling element bearings via composite hypothesis test," *Mechanical Systems and Signal Processing*, vol. 9, no. 1, pp. 63–75, 1995.

[27] H. Sohn, K. Worden, and C. R. Farrar, "Statistical damage classification under changing environmental and operational conditions," *Journal of Intelligent Material Systems and Structures*, vol. 13, no. 9, pp. 561–574, 2002.

[28] Y. W. Kim, G. Rizzoni, and V. I. Utkin, "Developing a fault tolerant power-train control system by integrating design of control and diagnostics," *International Journal of Robust and Nonlinear Control*, vol. 11, pp. 1095–1114, 2001.

[29] M. L. Fugate, H. Sohn, and C. R. Farrar, "Vibration-based damage detection using statistical process control," *Mechanical Systems and Signal Processing*, vol. 15, no. 4, pp. 707–721, 2001.

[30] M. Artes, L. Del Castillo, and J. Perez, "Failure prevention and diagnosis in machine elements using cluster," in *Proceedings of the Tenth International Congress on Sound and Vibration*, pp. 1197–1203, Stockholm, Sweden, 2003.

[31] J. Schurmann, *Pattern Recognition: A Unified View of Statistical and Neural Approaches*, Wiley, Hoboken, NJ, USA, 1996.

[32] S. K. Goumas, M. E. Zervakis, and G. S. Stavrakakis, "Classification of washing machines vibration signals using discrete wavelet analysis for feature extraction," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 3, pp. 497–508, 2002.

[33] X. Lou and K. A. Loparo, "Bearing fault diagnosis based on wavelet transform and fuzzy inference," *Mechanical Systems and Signal Processing*, vol. 18, no. 5, pp. 1077–1095, 2004.

[34] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[35] S. Poyhonen, P. Jover, and H. Hyotyniemi, "Signal processing of vibrations for condition monitoring of an induction motor," in *Proceedings of the 2004 First International Symposium on Control, Communications and Signal*, Hammamet, Tunisia, March 2004.

[36] M. Guo, L. Xie, S. Q. Wang, and J. M. Zhang, "Research on an integrated ICA-SVM based framework for fault diagnosis," in *Proceedings of the 2003 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, Washington, DC, USA, October 2003.

[37] R. J. Elliot, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*, Springer, Berlin, Germany, 1995.

[38] T. K. Ho, "Random decision forests," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 278–282, Montreal, QC, Canda, August 1995.

[39] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[40] D. Poole and A. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*, Cambridge University Press, Cambridge UK, 2017.

[41] A. Siddique, G. S. Yadava, and B. Singh, "Applications of artificial intelligence techniques for induction machine stator fault diagnostics: review," in *Proceedings of the 4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, pp. 29–34, Atlanta, GA,USA, August 2003.

[42] G. Vachtsevanos, F. L. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, Wiley Hoboken, Hoboken, NJ, USA, 2006.

[43] J. Moubray, "Reliability-Centred Maintenance," *Fuel and Energy Abstracts*, vol. 4, no. 36, p. 304, 1997.

[44] K. Kira and L. A. Rendell, "A Practical Approach to Feature Selection," in *Proceedings of the Machine Learning 1992*, D. Sleeman and P. Edwards, Eds., pp. 249–256, 1992.

[45] I. Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF," *European Conference on Machine Learning*, vol. 94, pp. 171–182, 1994.

[46] P. McCullagh and J. Nelder, *Generalized Linear Models*, Routledge, England, UK, 1989.

[47] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[48] S. L. Salzberg, "Programs for Machine Learning by J. Ross Quinlan, Morgan Kaufmann Publisher," *Machine Learning*, vol. 16, pp. 235–240, 1993.

[49] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[50] G. H. John and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers," 1995, https://arxiv.org/abs/1302.4964.

[51] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[52] J. Akosa, "Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data," *Proceedings of the SAS global forum*, vol. 12, pp. 1–4, 2017.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, and B. Thirion, "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.