IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 19, 2020, accepted September 27, 2020, date of publication October 9, 2020, date of current version October 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3029826

Assembly Line Anomaly Detection and Root Cause Analysis Using Machine Learning

OSAMA ABDELRAHMAN AND PANTEA KEIKHOSROKIANI
School of Computer Sciences, Universiti Sains Malaysia, Minden 11800, Malaysia

Corresponding author: Pantea Keikhosrokiani (pantea@usm.my)

The work of Pantea Keikhosrokiani was supported by the School of Computer Sciences, and Division of Research & Innovation, Universiti Sains Malaysia, Short Term Grant (304/PKOMP/6315435).

ABSTRACT Anomaly detection is becoming widely used in Manufacturing Industry to enhance product quality. At the same time, it plays a great role in several other domains due to the fact that anomaly may reveal rare but represent an important phenomenon. The objective of this paper is to detect anomalies and identify the possible variables that caused these anomalies on historical assembly data for two series of products. Multiple anomaly detection techniques were performed; HBOS, IForest, KNN, CBLOF, OCSVM, LOF, and ABOD. Moreover, we used AUROC and Rank Power as performance metrics, followed by Boosting ensemble learning method to ensure the best anomaly detectors robustness. The techniques that gave the highest performance are KNN, ABOD for both product series datasets with 0.95 and 0.99 AUROC respectively. Finally, we applied a statistical root cause analysis on the detected anomalies with the use of Pareto chart to visualize the frequency of the possible causes and its cumulative occurrence. The results showed that there are seven rejection causes for both product series, whereas the first three causes are responsible for 85% of the rejection rates. Besides, assembly machines engineers reported a significant reduction in the rejection rates in both assembly machines after tuning the specification limits of the rejection causes identified by this research results.

INDEX TERMS Anomaly detection, assembly lines, big data, machine learning, manufacturing industries, root cause analysis, unsupervised learning.

I. INTRODUCTION

For years, Manufacturing Industry has been adopting new quality measurement tools that led to an intensive-data environment, paving the way for using Machine Learning (ML) methods to extract information from the data as an endeavor to reduce the production cost and enhance the product quality [1].

In Manufacturing Industry, Assembly machines are considered as essential components and are widely used in the production lines. To ensure the final product quality, each assembly machine has an integrated inspection system that is supported by high-speed vision cameras to measure each assembled piece's dimensions. Additionally, each assembly machine has different specification limits depending on the design of the product to be assembled [2].

The dataset used in this paper includes information about dimensions measurements for two series of connectors, where both series have gold-coated pins that makes it very expensive to reject any assembled piece in the production line. The final product quality is ensured bypassing the product design specification limits to the inspection system in the assembly machine and after the inspection system measuring each assembled part dimensions. In case any part's measurements exceeded the design specification limits, the assembled part will be considered as an anomaly and automatically rejected and thrown to the trash [2], [3].

Anomaly is an unusual behavior in data that does not follow the expected behavior. Anomaly detection (AD), also synonymously termed as outlier detection, novelty detection and deviation detection, is the process of detecting patterns that do not follow the expected behavior in a given dataset. Although anomaly may reveal rare, it represents an important phenomenon. Thus, Anomaly detection has attracted considerable attention from the research community [4], [5].

Anomaly detection has been used in a wide range of application domains; for example, credit card fraud detection, insurance, health care, computer security intrusion detection, image processing, security-critical device failure detection and many more [4], [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

An important aspect of anomaly detection is the nature of the anomaly. An anomaly can be categorized in the following ways [7]:

A. POINT ANOMALY

Point anomaly is a single independent data instance that represents an irregularity or deviation which happens randomly and may have no particular interpretation compared to defined normal behavior in a data set. For example, after the assembly process of a connector is finished, it is followed by the inspection process, there is a point at which one of the assembled part dimensions is measured with a value far from the rest of the measurements and even exceeds the design specification limits. This abnormal behavior is known as point anomaly [6], [8]. Samples of point anomaly in 2-dimensional space are illustrated in Fig. 1 (O1, O2 data points).

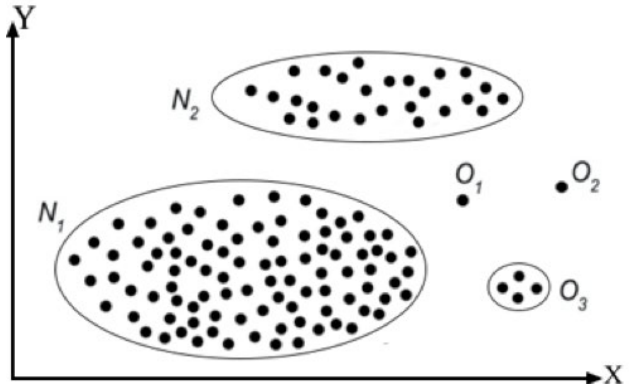


FIGURE 1. Samples of anomalies in 2-dimensional space.

B. CONTEXTUAL ANOMALY

Contextual anomaly also referred to as conditional anomaly, is a data instance that is considered as an anomaly in a specific context, but not otherwise [6]. This type of anomaly is common in time-series data streams, for example, after the assembly process of a connector is finished, the inspection system reports dimensions measurements for the assembled part with very high or very low values, but this happens only when the assembly machine starts working, or before it stops, or during high load hours, all these are considered as contextual anomalous behavior in the assembly machine. Sample of contextual anomaly is illustrated in Fig. 1 (N2 data points) with a condition that these data points hold different context from N1 data points [6], [8].

C. COLLECTIVE ANOMALY

Collective anomaly is a collection of individual data points, wherein each of the individual points in isolation appears as normal data instance while observed in a group that shows unusual characteristics with respect to an entire data set [6]. Collective anomaly in the assembly process appears when a sequence of assembled parts is reported with dimension's measurements that deviated from the normal behavior by the inspection system [6], [8]. Samples of collective anomaly is illustrated in Fig. 1 (O3 data points).

In machine learning manner, anomaly detection techniques have three main types according to the availability of the labels in the dataset in hand, as follows:

D. SUPERVISED TECHNIQUES

In supervised techniques, machine learning models are built for both anomalous data and normal data, where unseen data instance is classified as normal or anomaly by comparing which model it belongs to [7].

E. SEMI-SUPERVISED TECHNIQUES

In semi-supervised techniques, machine learning models are only built to fit the normal data, where unseen data instance is classified as normal if it fits the model sufficiently well; otherwise, the data instance is classified as an anomaly [7].

F. UNSUPERVISED TECHNIQUES

In unsupervised techniques, no any training dataset is needed. This is mainly because these approaches are based on the assumption that anomalies are much rarer than normal data in a given dataset [7].

Root cause analysis (RCA), refers to the process of identifying and delimiting the elements originating the anomaly [9]. RCA process aims at allocating the root cause by analyzing fault information with observed data [9]. In Manufacturing Industry, RCA is a highly effective technique for product design engineers and production managers to help in innovative problem-solving. Therefore, RCA has been used in numerous areas and is usually concerned with finding the root causes of events with safety, health, environmental, quality, reliability, production, and performance impacts [10], [11]. The process of RCA involves sorting the unstructured data and uncovering input, output relationships to identify the root causes and generally consists of the following four major steps [11]–[13]:

G. DATA COLLECTION

The Data collection stage is to gather the necessary data to be able to diagnose the root cause.

H. CASUAL FACTOR CHARTING

Casual factor charting is a simple sequence diagram depicting the actions that led to an occurrence.

I. ROOT CAUSE IDENTIFICATION

When the sequence of events leading to an occurrence is determined, they can be used in turn to determine the actual root cause of the occurrence.

J. RECOMMENDATION GENERATION AND IMPLEMENTATION

After determining the root cause occurrence, a list of recommendations is made on how to minimize or completely remove the occurrence of the cause.

The process of detecting anomalies in assembly lines and analyzing its root causes is done based on the available datum. In contrast, the defects of assembled parts are classified into two types; surface defects and damage. The texture and color characteristics are used to identify surface defects. On the other hand, shape dimensions measures are used to identify part damages, and both tasks are to ensure high-quality shipping products by notifying the robot controller to throw anomalous parts in the tray [12], [13].

In this paper, we developed a machine learning model using several anomaly detection techniques such as; HBOS, IForest, KNN, CBLOF, OCSVM, LOF, and ABOD to detect the anomalies in the assembly line for 54104, 54132 product series and to identify the possible variables that caused these anomalies by performing a root cause analysis on the product series anomalies. The remaining part of this paper is organized as follows; Section II describes Related works in the Literature, Section III presents the methodology, Section IV discusses the result, and Section V concludes the research and discusses further future studies.

II. RELATED WORKS

Anomaly detection has been the major focus of several kinds of research and scientific papers over the years and according to Xu et al in [14], Anomaly detection is a hot topic in terms of machine learning and its increasing greatly, being applied in a wide range of fields while it plays a great role in several other domains. Therefore, in the existing literature, several techniques and approaches have been proposed to detect anomalies as well as to improve the performance of existing anomaly detection techniques. In this section, we will discuss the most relevant techniques in the state-of-the-art.

A. ANOMALY DETECTION

In [14], several techniques have been surveyed and the main focus was on comparing the techniques for unlabeled high-dimensional benchmark datasets. The authors faced a challenge in identifying the threshold between anomaly and non-anomaly data points and another challenge in choosing the best features in this high-dimensional space. They proposed an ensemble learning-based approach to detect the anomalies in a dataset with such challenges; they used multiple anomaly detection algorithms including Angle-based Outlier Detection (ABOD), k-Nearest Neighbors Detector (KNN), and Local Outlier Factor (LOF). Moreover, in order to find the best performance and to evaluate these models, the authors used Area Under the Receiver Operating Characteristics (AUROC), Precision, and Rank Power as an evaluation metric. The results showed that FastABOD and KNN achieved the best performance in the area under the curve of ROC (AUROC) reaching up to 75% in detecting the true positive anomalies in the dataset.

Similarly, the authors in [15] used a large scale data with a high-dimensional space. They also faced the same challenges as in [14]. However, they investigated different types of learning models for deep anomaly detection techniques. Finally, they adopted a deep hybrid model by using one-class SVM (OC-SVM) together with deep neural networks model called One class neural network (OC-NN) to detect the anomalies and conversely to [14], the authors here used Autoencoders as feature extractors and nonetheless they used the same techniques to evaluate the model while the results did not show clear improvement.

On the other hand, the authors in [16], [17] used Local Outlier Factor (LOF) and Isolation Forest (IForest) techniques to detect the anomaly in a large scale data. Moreover, they used F1-score, Precision, and Recall as performance metrics except for Galante in [17] who added One-Class Support Vector Machines (OCSVM) as an additional technique to detect anomalies and he added AUROC as performance metrics. Additionally, all of them faced the same challenge in handling high-dimensional space and principal component analysis (PCA) was used to overcome this challenge by extracting new features that better represent the data which ended up with improving the final model performance F1 score to 64%.

In [18], the authors compared multiple unsupervised techniques to detect the anomaly on ten benchmark datasets. They used KNN, LOF, OCSVM, Connectivity-Based Outlier Factor (COF), Cluster-Based Local Outlier Factor (CBLOF), and Histogram-based Outlier Score (HBOS). Moreover, they faced a challenge in setting the threshold between anomaly and non-anomaly data points. The authors compared the performance of these techniques for various types of datasets including large-scale data with a high dimensional space. In addition, they used AUROC together with the Precision score and Rank Power as performance metrics and to compare and evaluate the performance of all these techniques. As a result, they found that KNN and LOF algorithms performed best compared to the other algorithms with AUROC equal to 98% for low dimensional space datasets and 54% for high dimensional space datasets and they relied on the computation time to pick from these two algorithms.

Again in [19], the authors used anomaly detection cluster-based techniques such as NKICAD, K-means, CBLOF, and LDCOF to detect the anomalies in network traffic dataset. Despite their dataset being unlabeled, the authors relied on statistical methods to calculate labels to help in evaluating the used algorithms with the use of Accuracy and Sensitivity metrics. The authors did experimental analysis first on benchmark datasets to validate their approach and then used this approach for their network traffic dataset. Finally, the result shows these cluster-based techniques are on average, able to detect 87% of the anomalies in network traffic dataset.

In [20], the authors used OCSVM, LOF and Random Forests as anomaly detection techniques. Also, only AUROC and Computation time are used to evaluate the performance. Thus, they stated that IForest achieved AUROC 70% in high dimensional data while their dataset does not contain anomalies in the training samples. Meaning that they tested the models under semi-supervised anomaly detection type contrary to the authors in [16], [18].

Furthermore, the same techniques are used by the authors in [21]; however, F-score, Recall, and Precision were added as performance metrics to evaluate the techniques in their approach. They stated that the proposed approach achieved a high detection rate and low false-positive rate at the same time which is the target optimal solution.

B. ROOT CAUSE ANALYSIS

In [9], The authors introduced a procedure for an automated root cause analysis using machine learning algorithms. They named the anomaly points as parts outside the tolerance limits, and they proposed a supervised and unsupervised approach to detect root causes of these anomalies' parts. The proposed approach used a decision tree algorithm to detect root causes on a large scale with a large number of variables.

Furthermore, Sarkar in [11] has shown the usefulness of an empirical cluster technique analysis to classify different anomaly types into a smaller number of categories and then use engineering knowledge to identify the root causes associated with these clusters. This approach is a combination of machine learning techniques such as cluster analysis and engineering domain knowledge to detect the root cause analysis. Additionally, Sarkar used hierarchical clustering to groups based on combinations of variables to obtain a manageable number of clusters, and then he used the engineering knowledge to investigate each of these clusters more closely to find out the root cause associated with the cluster.

Conversely to the previous works, the authors in [22] used the quality tools to detect the root cause. The authors used Pareto analysis to help in identifying and classifying the defect according to percentage significance. In addition, they also plotted the Cause and effect diagram to identify the major causes of the anomaly.

Similarly, the authors in [23] conducted a study on three months' data to observe the process going on in the production line, to reduce the rejection rate by tracking the root causes. Moreover, to achieve the objectives, the authors used Pareto analysis to identify and classify the defects according to percentage significance and Cause-Effect diagram was used to determine the major causes. The outcome of the study led to reducing the overall rejection rate in the production line from 10% to 7 %.

To conclude, the reviewed papers study anomaly detection techniques and aim to achieve good detection rates, whereas the majority of these papers adopted unsupervised machine learning due to the nature of the available data. Moreover, most of the authors of the reviewed papers faced a challenge in picking the right features to reduce the dimensionality. Another challenge was faced while identifying the anomaly and non-anomaly threshold for large scale data. Although the authors used different combinations of detection algorithms and evaluation metrics, most of them managed to overcome these challenges to different degrees. Finally, the authors of the reviewed works in root cause analysis agreed on using statistical quality control tools like Pareto analysis and Cause-Effect diagram to determine the major causes of the detected anomalous data points.

III. METHODOLOGY

This section describes data science project lifecycle, including the proposed approach to achieve the objectives with the description and justification of the used techniques.

Furthermore, in an ideal environment, every successful project must go through a data science lifecycle. Starting with data collection, model development, model evaluation, results presentation, and model deployment. Moreover, adopting this life cycle could determine the project's success and having useful outcomes [24].

A. DATA COLLECTION

Assembly machines for the product series under the scope of this study have a vision system used for the inspection process, where each machine consists of four inspection cameras dedicated to measuring the assembled piece dimensions and then store these measurements as a CSV format in the company server. Therefore, for the purpose of this study, the dataset files for both product series were obtained from the server as the only available source.

B. DATA DESCRIPTION

The data are stored by Assembly machines in a specific format which is difficult for a normal machine learning algorithm to read and process. Fig. 2 below, illustrates how the original CSV files are saved in the server.

	A	B	C	D	E	F	G	H	I
1	2	50							.....
2	1	14	14	0					.....
3	Tail Alignment	0.046	0.036	0.037	0.036	0.035	0.034	0.032	.....
4	Tail HiLo	0.044							.....
5	Tail Alignment	0.05	0.037	0.039	0.037	0.038	0.033	0.03	.....
6	Nail Out	0.058	0.042						.....
7	Nail In	0.041	0.049						.....
8	Nail Left	1.324							.....
9	Nail Right	1.379							.....
10	2	14	14	0					.....
11	Non Contact	0.047	0.047	0.05	0.05	0.057	0.05	0.045	.....
12	Contact Position	0.666	0.66	0.663	0.667	0.671	0.659	0.667	.....
13	3	15	14	1					.....
14	S/Tail Lenght	0.562	0.556	0.56	0.557	0.561	0.564	0.567	.....
15	Nail Position	2.258	2.225						.....
16	Position Deviation:P	0.032	0.031	0.024	0.024	0.016	0.018	0.023	.....

FIGURE 2. The original dataset for one assembled pieces.

The columns and rows are swapped. In addition, for each instance in the data, there are specific numbers of rows where each row represents a dimension measurement for its respective number of pins, whereas the number of columns depends on the number of pins in each series. Table 1 below demonstrates the number of instances, measurements, columns, and rows for each series.

For each attribute in the dataset, Table 2 below displays the name, type, and description.

After each instance in the dataset, there is a blank line and there are also empty cells. This is due to the nature of

TABLE 1. Instances Details per Product Series.

Details / Series	54104	54132
Total number of instances	31732	57652
Measurements per instance	309	158
Rows per instance	16	18
Number of columns (pins)	50	30
Data duration	5 Days	5 Days

TABLE 2. Dataset Attributes Description.

Feature	Type	Description
Tail Alignment	Numeric	Distance between solder tail and housing as datum
Tail HiLo	Numeric	Difference between highest and lowest solder tail
Nail Out & In	Numeric	Left & Right Fitting Nail alignment
Nail Left & Right	Numeric	Left and Right Nail length
NonContact	Numeric	Distance between housing, contact pin at noncontact side
Contact & Nail Position	Numeric	Numeric Contact pin and Fitting nail position
Solder Tail Length	Numeric	Solder tail length
Position Deviation	Numeric	Distance between solder tail and housing as datum

how the data files are formatted, whereas there are no missing values in both series datasets. However, there are many special characters in the dataset, such as (#, !, O, and #1). Furthermore, some data points have very high values compared to the specification limits (such as 9.999). Regarding this, assembly machines engineers reported that this happens when the inspection camera misses any dimension measurement. Thus, it uses these high values and immediately rejects this assembled piece, therefore, any instance has a high value or any of the special characters is considered as an anomaly. Finally, TailAlignment attribute is duplicated, and the engineers described this as a confirmatory measurement. Thus, both TailAlignment attributes reflect the same type of measurement performed using the same camera.

C. PROPOSED SOLUTION

The proposed approach to achieve the objectives of this study consists of four main stages, as illustrated in Fig. 3 below with the proposed mains tasks in each stage.

D. DATA PRE-PROCESSING

The dataset files require many preparations to make it suitable for use by machine learning algorithms. These preparations include reformatting the dataset, handing duplicated measures, instances with high values and special characters, feature scaling, feature selection, and the dimensionality reduction as follows:

1) REFORMATTING DATASETS

The data are stored as a CSV file in a specific format that makes it nonreadable by machine learning algorithms.

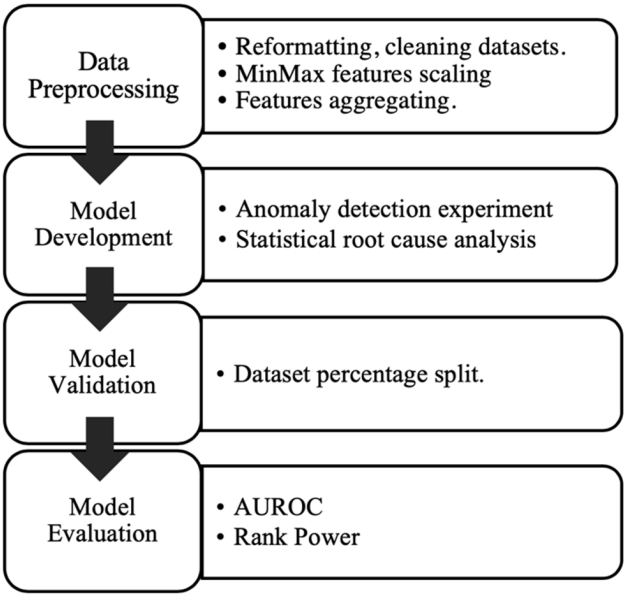


FIGURE 3. Proposed solution flowchart.

Therefore, due to the dataset size and the high number of columns and rows in the data, we used Python to reformat the dataset to a readable format. Moreover, part of the reformatting process was changing the names of the attributes, where each attribute in the dataset is renamed to have a meaningful name that reflects the real measure done by the inspection cameras.

2) DUPLICATES

There is one attribute duplicated "TailAlignment" and since it is a confirmatory measure, we solved this problem through merging both attributes by taking the minimum and maximum values. This is because the rejection process is basically based on the minimum and maximum specification limits, thus it will not lead to losing the variance and anomalies in the data.

3) SPECIAL CHARACTERS AND MIS-MEASURES

Assembly machines engineers reported that the assembly machines use these special characters when the inspection cameras mismeasure any pin. Therefore, these instances are dropped and later these instances will be discussed since they are considered anomalies.

4) FEATURES SCALING

As reported by the author in [25], machine learning algorithms perform well when the features in the data are on the same scale since it will be used in dimensional space. For this, we used MinMax scaling technique to rescale the features to be in (0, 1) range. The use of MinMax was mainly because this technique does not affect the variance in the data since it is sensitive to the presence of the anomalies during the scaling process.

5) FEATURES EXTRACTION

The data are in high dimensional space with a number of attributes reaching to 309 attributes for 54104 series and 158 attributes for 54132 series. To overcome this problem, first, we looked at the correlations between the datasets attributes to understand their relationships. This is because anomaly detection techniques assume that the attributes are not highly correlated to better learn from the data. After that, the features were aggregated by taking the minimum and maximum values for each measurement type, and this mainly because we do not want to miss the variance in the data. Moreover, the aggregation process led to having a new 19 attributes for 54104 series dataset and 20 attributes for the 54132 series dataset. Nevertheless, we compared the aggregation technique to PCA. The result indicated that the aggregation technique improved the baseline model performance significantly compared to PCA. Therefore, features aggregation better represents the data and holds more variance.

E. MODEL DEVELOPMENT

The model development stage is divided into two parts; the first part is to achieve the first objective of this study by developing a machine learning model using anomaly detection techniques to detect the anomalies in the assembly line for 54104, 54132 product series. The second part proposed to achieve the second objective by performing root cause analysis on 54104, 54132 series anomalies to identify the possible variables that caused these anomalies.

1) ANOMALY DETECTION EXPERIMENT

We built a machine learning model based on the prepared data using multiple anomaly detection algorithms and compare their performance to pick the best one. The selected algorithms are HBOS, IForest, KNN, CBLOF, OCSVM, LOF, and ABOD. We ran each of these selected algorithms multiple times with many parameters' tunings and then we used Boosting ensemble learning method on the best models. The idea behind this is to have stable performance, as stated by the authors in [26] that anomaly detection models often suffer from instability due to its unsupervised nature. Moreover, as in the review of the related works in Section II, unsupervised anomaly detection techniques are used widely to detect the anomalies in the data and as highlighted by most authors. This is because the underlying data are unlabeled similarly to our data in this study, which means it is a better choice to use unsupervised learning techniques to discover any abnormal behavior in the data. Specifically, the authors in [27] reported that this type of learning is suitable to solve such a problem. Furthermore, each selected algorithm is either cluster-based, angle-based, statistical, neighbor-based, density-based, or classification-based. Therefore, each detection algorithm has a different work mechanism when detecting point, contextual, or collective anomalies. As an advantage, the combination of these algorithms will help in detecting any type of anomaly that may be present in the data.

2) ROOT CAUSE ANALYSIS

Root cause analysis is mainly used to find the pins measures that exceeded the specification limits. To achieve this, we worked on the detected anomalous instances as a result of the best anomaly detectors in the first objective by conducting a statistical analysis on these detected instances through studying the minimum and maximum values of each attribute in the data. In addition, we plot the statistical analysis results in Pareto chart to see the occurrence frequency and the cumulative contribution of each attribute in causing the anomalies. This approach is used widely in the related works, and although most of the authors agreed on performing a statistical analysis first, they used various types of visualization. In this study, Pareto chart is used due to its several advantages such as; ease in plotting the frequency of each cause, showing the cumulative frequencies for the cause, helping in deciding which cause to fix first and easier interpretation as compared to other visualization techniques.

F. MODEL VALIDATION

The datasets contain quite enough samples; therefore, we used one main validation approach which is Percentage split. This procedure evaluates the models on a percent data sample by splitting it into two; The first portion is used for training purposes and the second portion is used for testing.

G. MODEL EVALUATION

The In this study, the ideal model must be able to distinguish the anomaly data points from those normal data points, and since the final goal is to ensure the quality and deliver only products without any measurements issues, therefore, the best model must be able to detect any true positive instances. For this reason, we used two performance metrics in each training and testing cycle to measure the model's performance. These two metrics help in determining the best model in terms of detecting the true positive instances and both metrics are commonly used in the related works to evaluate anomaly detection models. The first metric is AUROC (Area Under the Receiver Operating Characteristics), this metric to helps in evaluating the model based on its ability to distinguish between anomaly and non-anomaly data points by diagramming a curve of the true-positive rate against the false-positive rate of the model. Thus, the more underlying area reflects better detection for the anomalies. The second metric is Rank power (RP) and we used this as an additional metric to rank the models based on the ability of each model to detect and rank the real anomalies at the top and before all other detected data points. Finally, to have a better idea on how each model performed, it is important to mention that we calculated the labels for the test set for both product series with the help of the specification limits and we confirmed these label with the machines engineers and this was compared with the rejection rate for each assembly machine.

H. IMPLEMENTATION

Python and Jupyter notebooks are the tools used for the model development and root cause analysis. Moreover, the main libraries used are Sklearn, PyOD, Pandas. These libraries offers the required preprocessing methods and anomaly detection techniques.

IV. RESULTS AND DISCUSSION

Preprocessing tasks resulted in a ready and clean dataset while it affected the number of the instances since some instances were dropped, Table 3 below shows the instances after the preprocessing tasks.

TABLE 3. Dataset after preprocessing tasks.

Series	54104	54132
Original instances	31732	57652
Original columns	309	158
Mis measures	457	243
Duplicated measures	50 pins	0 pins
Final number of instances	31275	57409
Final number of columns	259	158

On the other hand, during the dimensionality reduction process, we used KNN algorithm as a baseline to see how the selected features in each approach affect the model performance. Table 4 and Table 5 below state the number of selected attributes using Filter and Wrapper approach with its effect on the performance compared to the baseline for both series.

TABLE 4. 54132 selected features performance compared to the baseline.

Approach	Method	Selected Features	AUCROC
Baseline	None	All Features	0.83
Filter	Variance threshold	6 Features	**0.85**
	Correlation coefficient	8 Features	0.53
Wrapper	Recursive Feature Elimination	8 Features	0.55

TABLE 5. 54104 selected features performance compared to the baseline.

Approach	Method	Selected Features	AUCROC
Baseline	None	All Features	0.68
Filter	Variance threshold	21 Features	0.72
	Correlation coefficient	9 Features	**0.73**
Wrapper	Recursive Feature Elimination	10 Features	0.72

Features selection approaches did not improve the model performance compared to the baseline. Therefore, we used PCA features extraction approach with various numbers of components to see if the extracted components improve the performance. The results for both product series are illustrated in Table 6 below.

From the comparison table above, we observe that PCA improved the model performance. Although there is a huge reduction in dimensionality, the extracted components do not still hold much information about the data and even when increasing the number of components, and it reduces the model performance. Moreover, as an additional improvement, we aggregated the dataset features based on the measurement's types and this is because the assembly machines' rejection mechanism uses the minimum and maximum values for each type of measurement. The best performances for all these techniques are compared to MinMax aggregated features as shown in Table 7 and Table 8 below for both product series.

TABLE 6. Comparison of PCA results for both series to the baseline.

Series	54132		54104	
	Baseline	PCA	Baseline	PCA
Number of Features	158	25	259	8
Information represented	100%	80%	100%	84%
ROC score	0.83	**0.90**	0.68	**0.79**

TABLE 7. Comparison of reduction techniques for 54132 datasets.

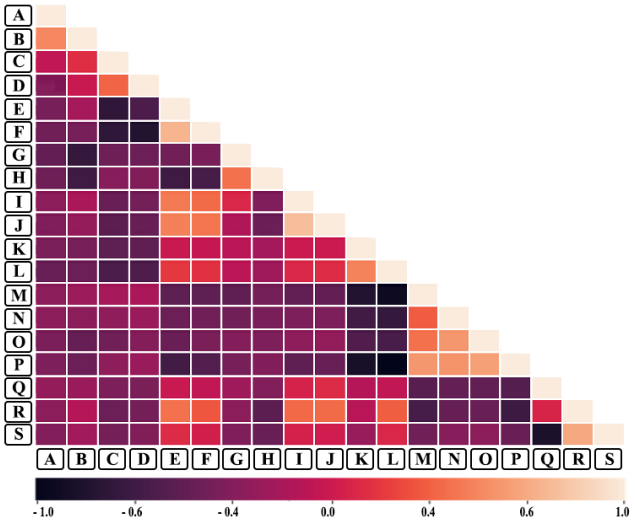
	Selected / Extracted Features	AUCROC
Baseline	All Features	0.80
Variance threshold	6 Features	0.85
Correlation coefficient	8 Features	0.53
Recursive Feature Elimination	10 Features	0.55
Principal Component Analysis	25 Components	0.90
Aggregated MinMax Features	20 Derived	**0.97**

TABLE 8. Comparison of reduction techniques for 54104 datasets.

	Selected / Extracted Features	AUCROC
Baseline	All Features	0.68
Variance threshold	21 Features	0.72
Correlation coefficient	9 Features	0.73
Recursive Feature Elimination	10 Features	0.72
Principal Component Analysis	5 Components	0.72
Aggregated MinMax Features	22 Derived	**0.94**

By comparing dimensionality reduction techniques to the baseline, it shows that MinMax features have the highest performance. Moreover, for a better understanding of the relationships of the derived features, Fig. 4 below displays the heat map of the correlation matrix with scale from 1 to −1. Where 1 (lighter color) represents a strong positive correlation, −1 (darker) represents a strong negative correlation whereas a value near to zero represents weak or no correlation.

From the correlation heat map above, we can confirm that the derived features do not have that much strong positive or negative correlation to each other, this is what makes anomaly detection algorithms consider all features instead



A	TailAlignmentsMin	B	TailAlignmentsMax
C	TailHilo	D	NailOutMin
E	NailOutMax	F	NaillnMin
G	NaillnMax	H	NailLeft
I	NailRight	J	NonContactsMin
K	NonContactsMax	L	ContactPositionsMin
M	ContactPositionsMax	N	SolderTaillengthsMin
O	SolderTailLengthsMax	P	NailPositionsMin
Q	NailPositionsMax	R	PositionDeviationsMin
S	PositionDeviationsMax		

FIGURE 4. Correlation matrix for MinMax derived features.

of being influenced by the strong correlations between the features. Therefore, we conclude that this feature representation holds enough information to help the model learn from the data.

A. ANOMALY DETECTION EXPERIMENTS

The proposed approach in the model development stage involves using a combination of multiple anomaly detection algorithms and as stated earlier in the Methodology section, the selected algorithms are HBOS, IForest, KNN, CBLOF, OCSVM, LOF, and ABOD. The development process is initiated by using the best feature representation. Table 9 below shows the best performance for each model on test data in both product series.

TABLE 9. Best performance for each model on test data for both series.

Test set	54132		54104	
	AUROC	Rank Power	AUROC	Rank Power
HBOS	0.89	0.11	0.91	0.07
IForest	0.82	0.11	0.87	0.03
KNN	0.99	0.50	**0.95**	**0.24**
CBLOF	0.97	0.33	0.91	0.08
OCSVM	0.86	0.33	0.92	0.13
LOF	0.99	0.56	0.91	0.14
ABOD	**0.99**	**0.61**	0.92	0.28

From the comparison of the results above, ABOD algorithm achieved the highest result by scoring 0.99 AUROC in detecting anomalies in 54132 series data with 0.61 as the ranking of the actual anomalies are detected first. For 54104 series, the best performance accomplished by KNN algorithm by scoring 0.95 AUROC in detecting the anomalies in 54104 series data with 0.24 for ranking the actual anomalies are detected first. Table 10 below comparing the performance of the detection algorithms to the calculated anomalies and machine rates.

TABLE 10. Algorithms detection VS calculated and machine anomalies.

	54132		54104	
	Count	Percent	Count	Percent
Machine rate	168	0.0030%	602	0.0189%
Calculated	35	0.0006%	203	0.0065%
HBOS	35	0.0006%	376	0.0120%
IForest	42	0.0007%	523	0.0167%
KNN	88	0.0015%	343	0.0109%
CBLOF	34	0.0006%	214	0.0068%
OCSVM	31	0.0005%	225	0.0072%
LOF	60	0.0010%	431	0.0138%
ABOD	62	0.0011%	432	0.0138%

As mentioned previously in Table 9, ABOD algorithm achieved 0.99 of the anomalies are detected and by comparing this percent to the number of the detected ones in Table 10 above as 62 data points are predicted as anomalies, whereas the calculated anomalies are only 35, this means that there are additional 27 instances wrongly predicted as anomalies and by comparing the 62 anomalies to the machine rates which is 168 anomalies instances. Actually, this number is for all instance in the train and test set data, and we know that there are mismeasures that dropped previously during dataset cleaning tasks. After subtracting the dropped instances we found that there are only 75 anomaly instances in the test set, meaning that ABOD algorithm has only 13 wrong predictions. Therefore, we can say that the assembly machine for 54132 does not have over reject, while we should not forget the data are for five days.

On the other hand, KNN algorithm achieved 0.95 AUROC when detecting the anomalies from Table 9. By comparing this to the number of the detected ones in Table 10, there are 343 data points predicted as anomalies while the calculated anomalies are 203. This means that there are additional 140 wrong predictions, and by comparing these 343 anomalies to machine rates which are 602 anomalies and after subtracting the dropped ones, we end up having only 145 anomaly instances in the test set. Therefore, KNN algorithm has only 5 wrong predictions, which give a sign that the assembly machine for 54104 does not have high over reject. However, we should not forget that the data are only for five days.

For a better understanding of the AUROC performance for the detection process done by all anomaly detection techniques used, Fig. 5 and Fig. 6 illustrate the area under the curve of ROC (AUROC) as a comparison between the detection techniques used for both series datasets.

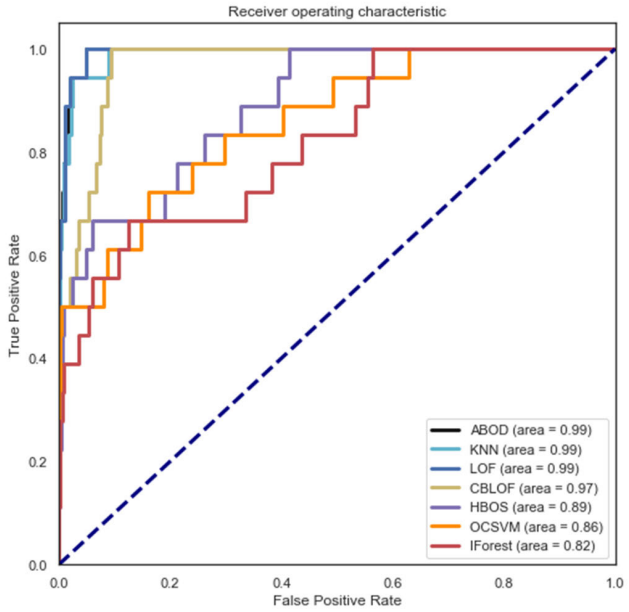


FIGURE 5. Comparison of the detection techniques for 54132 series.

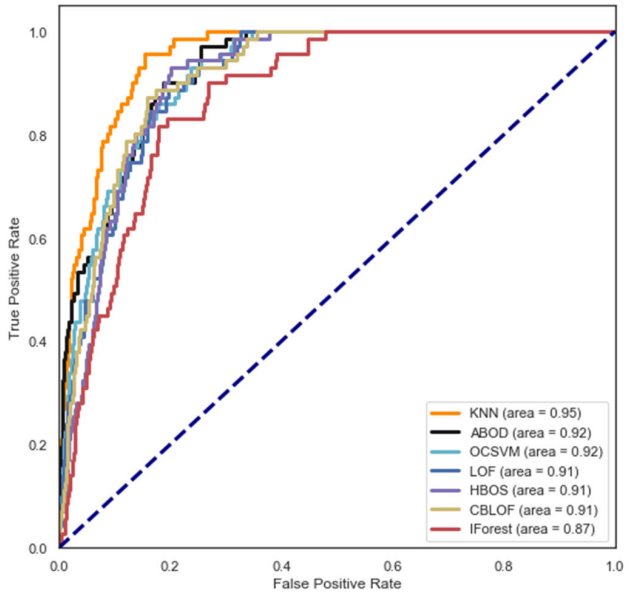


FIGURE 6. Comparison of the detection techniques for 54104 series.

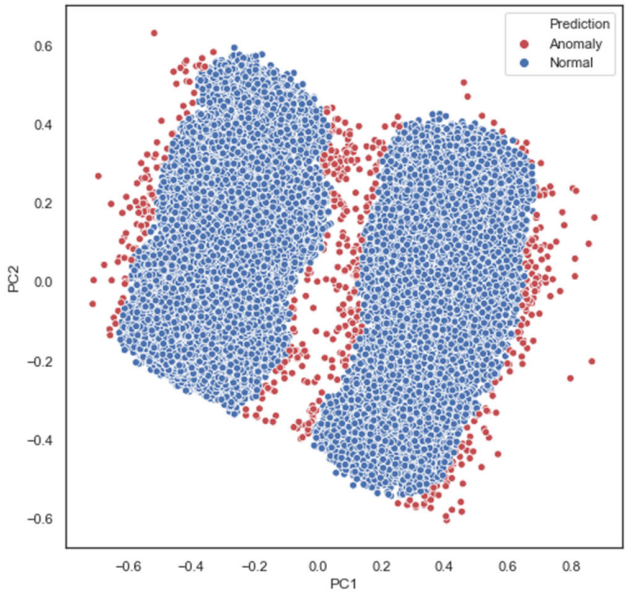


FIGURE 7. Scatter plot for ABOD algorithm predictions on 54132.

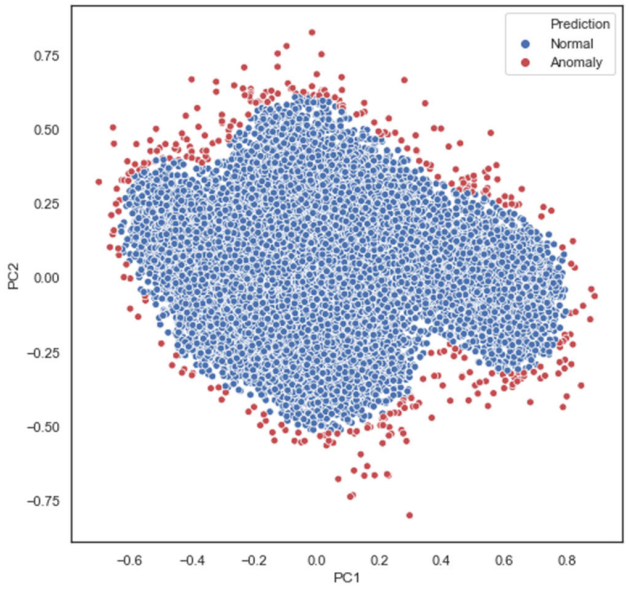


FIGURE 8. Scatter plot for KNN algorithm predictions on 54104.

From Fig. 5 above, for 54132 series, we can confirm that ABOD algorithm (black color line) covered 99% of the area under the curve while also KNN covered the same area with 99%. Though, we picked ABOD because it scored higher in Rank Power metric. Also, Fig. 6 for AUROC 54104 series displays that most of the area under the curve is covered by KNN (dark orange line) with 95% whereas the least area is covered by IForest algorithm. Furthermore, the scatter plots below in Fig. 7 and Fig. 8 show which instances are predicted to be an anomaly using the best-selected detection algorithms.

From the scatter plot for ABOD algorithm predictions on 54132 series dataset, anomaly data points in red color while the blue color for the normal ones, we can see that this algorithm did well in detecting the point anomalies whereas no clear presence of another type of anomalies in the data. Similarly, the scatter plot in Fig. 8 below, KNN algorithm predictions on 54104 series datasets, we can see that KNN algorithm was able to detect the point anomalies whereas no clear presence of another type of anomalies.

B. ROOT CAUSE ANALYSIS

The second objective is mainly statistical-based analysis to find the possible variables that cause the detected anomalies. In addition, since assembly machines reject the parts that exceed the limits, therefore, by checking Boxplots for both

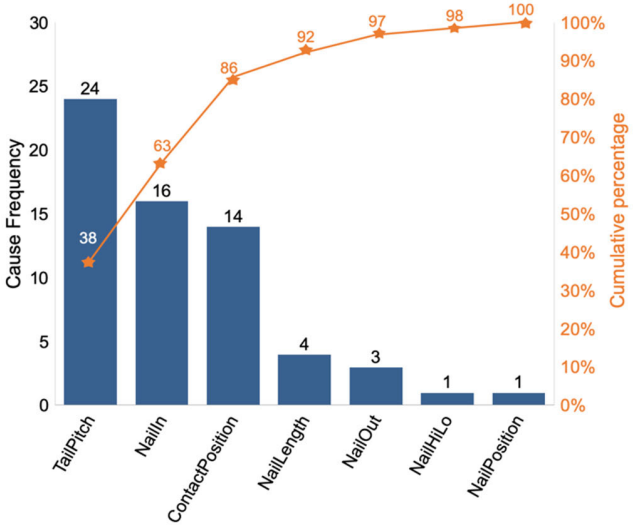


FIGURE 9. Pareto Chart for 54132 series rejection causes.

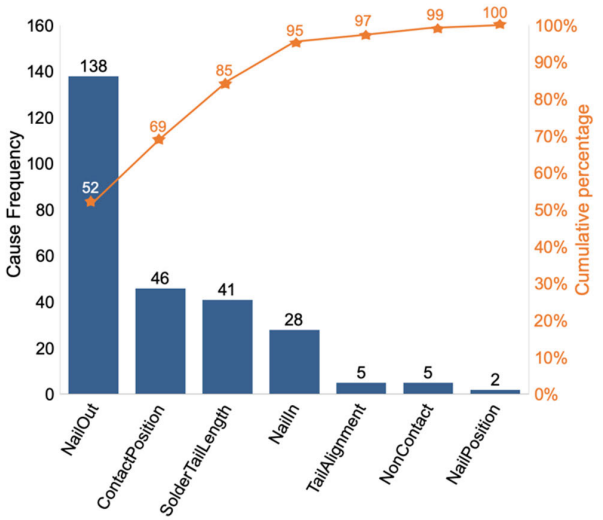


FIGURE 10. Pareto Chart for 54104 series rejection causes.

series to see the data point in terms of their first quartiles, mean, median and the third quartiles. The results are concluded in Pareto chart as shown in Fig. 9 for 54132 series.

From Pareto chart above, TailPitch measurements exceeded the limits 24 times followed by NailIn which exceeded the limits 16 times, then ContactPosition exceeded the limits 14 times. These three measurements are the major causes of the rejection based on their cumulative occurrence, these three measurements cause 86% of the rejection in this series. Therefore, it shows that tuning and fixing the specification limits for these three measures will result in 86% reduction in the rejection rate for these 54132 series which is a significant reduction. Moreover, Table 11 below concise all pins measurements that exceeded the limits with identifying each pin and either if it exceeded the minimum limits or the maximum limits, and the total occurrence for each measurement type in the 54132 series' dataset.

TABLE 11. Possible variable caused 54132 rejection.

Cause	Exceed	Pins number	Freq.	Total
TailPitch	Min	4,6,9,10,13,15,19,21,25,27	12	24
	Max	2,5,8,12,16,17,24,26,28,29	12	
NailIn	Min	1	14	16
	Max	2	2	
ContactPosition	Min	3,5,7,15,18,20,25	9	14
	Max	1,4,11,13,21	5	
NailLength	Max	2	4	4
NailOut	Max	2	3	3
NailHiLo	Max	2	1	1
NailPosition	Min	1	1	1

Moreover, 54104 series boxplots results are concluded in Pareto chart below in Fig. 10.

Pareto chart above shows that NailOut measurements exceeded the specification limits 138 times followed by ContactPosition which exceeded the limits 46 times and then SolderTailLength exceeded the limits 41 times. Moreover, these three measurements are the top three causes for the rejection based on their cumulative occurrence, these three measurements causing 85% of the rejection for this series, meaning that tuning the specification limits for these three pins measures will result in 85% reduction in the rejection rate for this 54104 series which is a significant reduction. Moreover, Table 12 below concise all pins measurements that exceeded the limits with identifying each pin and it either exceeded the minimum limits or the maximum limits, and the total occurrence for each measurement type in the 54104 series' dataset.

TABLE 12. Possible variable caused 54104 rejection.

Cause	Exceed	Pins number	Freq.	Total
NailOut	Max	1,2	138	138
ContactPosition	Min	2, 6, 11, 12, 17, 18, 20, 21, 22, 27, 28, 32, 37	40	46
	Max	39, 41, 45, 47	6	
SolderTail Length	Min	3, 5, 7, 15, 18, 20, 25	41	41
NailIn	Max	1, 2	28	28
TailAlignment	Min	1, 28, 29, 30, 31	5	5
NonContact	Max	2, 19, 27, 40, 45	5	5
NailPosition	Min	2	2	2

In conclusion, the first objective is achieved using anomaly detection techniques whereas the best algorithm for 54132 series is ABOD with 0.99 AUROC. On the other hand, the best algorithm for 54104 series is KNN with 0.95 AUROC. By comparing the first objective results to the related works in the literature, 54132 series model achieved 0.99 AUROC which yields better results than the related works whereas 54104 series using KNN scored 0.95 AUROC. However, there is an approach in the related works which scored 0.98, which is thus better than our approach using

ABOD model. For the second objective, the root causes diagnosed for 54123 anomalies are TailPitch, NailIn, ContactPosition, NailLength, NailOut, NailHiLo, and NailPosition, whereas the first three causes are responsible for 86%. Moreover, the root causes for 54104 anomalies are NailOut, ContactPosition, SolderTailLength, NailIn, TailAlignment, NonContact, and NailPosition. The first three causes for this series are responsible for 85% of the anomalies in both series.

V. CONCLUSION AND FUTURE WORKS

In this paper, we studied assembly data for two product series to detect anomalous data points and to diagnose the possible causes of these anomalies. The results showed that there are 62 anomalous data points for 54132 using ABOD algorithm and 343 anomalous data points for 54104 using KNN algorithm with no clear presence of over reject in assembly machines for both series. In addition, the results showed that there are seven rejections causes for each series, whereas the first three causes are responsible for 86% and 85% of the rejection rates in 54132 and 54104 product series respectively. Ultimately, the results of this research are expected to lead to a significant reduction in the rejection rates in both assembly machines to different degrees. Nevertheless, this reduction depends on performing an appropriate tuning for the specification limits of the identified rejection causes for each series.

A. FUTURE WORKS

This paper studied historical data for two assembly line machines to detect the anomalies and its root causes, future work can focus on predicting when these anomalies will happen instead of just focusing on when it happened which could lead to more quick and effective results when making decisions. Another possible future work is to study the dimensions' measurements as time-series data which might lead to putting more focus on each measurement type separately with time. Furthermore, it is well-known in the literature that anomaly detection techniques are facing instability issues due to the data and the unsupervised learning nature, and since we trained the best-selected models using five days' data for both product series, a recommended future work may train the models using data for a longer period, this is more likely to have stable and robust models and thus it can generalize well for unseen data.

ACKNOWLEDGMENT

The authors are thankful to the School of Computer Sciences, Universiti Sains Malaysia for the unlimited support. The authors are also grateful to the production lines managers and assembly machines engineers at Molex (Malaysia) Sdn. Bhd. for their support and the great domain knowledge they shared.

REFERENCES

[1] F. Pittino, M. Puggl, T. Moldaschl, and C. Hirschl, "Automatic anomaly detection on in-production manufacturing machines using statistical learning methods," *Sensors*, vol. 20, no. 8, p. 2344, Apr. 2020, doi: 10.3390/s20082344.

[2] A. Graß, C. Beecks, and J. A. C. Soto, *Unsupervised Anomaly Detection in Production Lines*. Berlin, Germany: Springer, 2019.

[3] M. Raut and D. D. S. Verma, "To improve quality and reduce rejection level through quality control," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 5, no. July, pp. 764–768, 2017.

[4] X. Liu and P. S. Nielsen, "Regression-based online anomaly detection for smart grid data," 2016, *arXiv:1606.05781*. [Online]. Available: https://arxiv.org/abs/1606.05781

[5] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer, "Detection of anomalies in large scale accounting data using deep autoencoder networks," Aug. 2018, *arXiv:1709.05254*. [Online]. Available: http://arxiv.org/abs/1709.05254

[6] C. Chahla, H. Snoussi, L. Merghem, and M. Esseghir, "A novel approach for anomaly detection in power consumption data," in *Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, May 2019, pp. 483–490, doi: 10.5220/0007361704830490.

[7] W. Cui and H. Wang, "A new anomaly detection system for school electricity consumption data," *Information*, vol. 8, no. 4, p. 151, Nov. 2017, doi: 10.3390/info8040151.

[8] M. Fahim and A. Sillitti, "Anomaly detection, analysis and prediction techniques in IoT environment: A systematic literature review," *IEEE Access*, vol. 7, pp. 81664–81681, 2019, doi: 10.1109/ACCESS.2019.2921912.

[9] T. Mueller, J. Greipel, T. Weber, and R. H. Schmitt, "Automated root cause analysis of non-conformities with machine learning algorithms," *J. Mach. Eng.*, vol. 18, no. 4, pp. 60–72, Nov. 2018, doi: 10.5604/01.3001.0012.7633.

[10] B. Vo, E. Kongar, and M. F. Suarez-Barraza, "Root-cause problem solving in an industry 4.0 context," *IEEE Eng. Manag. Rev.*, vol. 48, no. 1, pp. 48–56, Mar. 2020, doi: 10.1109/EMR.2020.2966980.

[11] P. Sarkar, "Clustering of event sequences for failure root cause analysis," *Qual. Eng.*, vol. 16, no. 3, pp. 451–460, Jan. 2004, doi: 10.1081/QEN-120027946.

[12] T. Josefsson, "Root-cause analysis through machine learning in the cloud," Uppsala Univ., Uppsala, Sweden, Tech. Rep., 2017.

[13] P. Arjun and T. T. Mirnalinee, "Machine parts recognition and defect detection in automated assembly systems using computer vision techniques," *Rev. Tec. la Fac. Ing. Univ. del Zulia*, vol. 39, no. 1, pp. 71–80, 2016, doi: 10.21311/001.39.1.08.

[14] X. Xu, H. Liu, L. Li, and M. Yao, "A comparison of outlier detection techniques for high-dimensional data," *Int. J. Comput. Intell. Syst.*, vol. 11, no. 1, pp. 652–662, 2018, doi: 10.2991/ijcis.11.1.50.

[15] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: https://arxiv.org/abs/1901.03407

[16] H. John and S. Naaz, "Credit card fraud detection using local outlier factor and isolation forest," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 4, pp. 1060–1064, Apr. 2019, doi: 10.26438/ijcse/v7i4.10601064.

[17] L. Galante, "A comparative evaluation of anomaly detection techniques on multivariate time series data," *Int. J. Comput. Sci. Eng.*, vol. 18, pp. 17–29, Jan. 2019, doi: 10.13140/RG.2.2.18638.72001.

[18] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS ONE*, vol. 11, no. 4, pp. 1–31, 2016, doi: 10.1371/journal.pone.0152173.

[19] M. Ahmed and A. N. Mahmood, "Novel approach for network traffic pattern analysis using clustering-based collective anomaly detection," *Ann. Data Sci.*, vol. 2, no. 1, pp. 111–130, Mar. 2015, doi: 10.1007/s40745-015-0035-y.

[20] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 1–44, 2012, doi: 10.1145/2133360.2133363.

[21] A. Karami and M. Guerrero-Zapata, "A fuzzy anomaly detection system based on hybrid PSO-kmeans algorithm in content-centric networks," *Neurocomputing*, vol. 149, pp. 1253–1269, Feb. 2015, doi: 10.1016/j.neucom.2014.08.070.

[22] K. Singh and A. N. Tiwari, "Defects reduction using root cause analysis approach in gloves manufacturing unit," *Int. Res. J. Eng. Technol.*, vol. 3, no. 7, pp. 173–183, 2016.

[23] A. Ashwini and K. S. Avinash, "Rejection analysis in piston manufacturing unit," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 4, no. 3, pp. 1157–1163, 2015. [Online]. Available: http://www.ijirset.com/upload/2015/march/72_REJECTION.pdf

[24] Z. Nina and J. Mount, *Practical Data Science With R*. New York, NY, USA: Manning Publications Co., 2014.

[25] E. Alpaydın, *Introduction to Machine Learning*, vol. 1107, 2nd ed. Cambridge, MA, USA: MIT Press, 2014.
[26] D. Eyl, "Anomaly detection in wireless sensor networks data by using histogram based outlier score method," in *Proc. 2nd Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, 2018, pp. 1–6, doi: 10.1109/ISMSIT.2018.8567262.
[27] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognit.*, vol. 36, no. 2, pp. 451–461, Feb. 2003, doi: 10.1016/s0031-3203(02)00060-2.

OSAMA ABDELRAHMAN received the B.S. degree in computer science from Omdurman Islamic University (OIU), Sudan, in 2012, and the M.S. degree in data science and analytics from the School of Computer Science, Universiti Sains Malaysia (USM), Malaysia, in 2020. His research interests include computer vision, artificial intelligence, and deep learning.

PANTEA KEIKHOSROKIANI received the Bachelor of Science degree in electrical and electronics engineering, the master's degree in information technology from the School of Computer Sciences, Universiti Sains Malaysia (USM), Malaysia, and the Ph.D. degree in service system engineering, information system. She was a Teaching Fellow with the National Advanced IPv6 Centre of Excellence (Nav6), USM, where she is currently a Senior Lecturer with the School of Computer Sciences. Her recent book was published entitled Perspectives in the *Development of Mobile Medical Information Systems: Life Cycle, Management, Methodological Approach and Application,* in 2019. Her articles was published in distinguished edited books and journals, including *Telematics and Informatics* (Elsevier), *Cognition, Technology, and Work* (Springer), Taylors and Francis, and IGI Global. She was indexed by ISI, Scopus, and PubMed. Her research and teaching interests include information systems development, database systems, health and medical informatics, business informatics, location-based mobile applications, big data, and technopreneurship.