이 학 석 사 학 위 논 문

Real-time Anomaly Detection Using RRCF+CAD: A Case Study of Sensor Data

RRCF+CAD를 활용한 실시간 이상탐지 : 센서 데이터 사례 연구

2023년 2월

서울대학교 대학원

통계학과

류 환 감

Real-time Anomaly Detection Using RRCF+CAD: A Case Study of Sensor Data

RRCF+CAD를 활용한 실시간 이상탐지 : 센서 데이터 사례 연구

지도교수 장 원 철

이 논문을 이학석사 학위논문으로 제출함
2022년 10월

서울대학교 대학원
통계학과
류 환 감

류환감의 이학석사 학위논문을 인준함
2023년 1월

위 원 장	오 희 석	(인)
부위원장	장 원 철	(인)
위 원	박 건 웅	(인)

ABSTRACT

Hwankam Ryu
The Department of Statistics
The Graduate School
Seoul National University

In this thesis, we investigate anomaly detection, which builds the stability of the system by separating data different from normal data. For a systematic and sustainable system, the task of continuously monitoring and classifying abnormal data plays an important role, and various methodologies using machine learning/deep learning as well as statistical methods are being used. In this paper, after briefly introducing the methodology used in previous studies, we propose the RRCF+CAD model, a real-time anomaly detection method that combines the Robust Random Cut Forest Model and Conformal Prediction. This method enables real-time updating of the model, and based on this, a statistical test method is executed by finding an anomaly score of the data.

Keywords: Outlier, Anomaly score, Semi-supervised learning, Robust Random Cut Forest, Conformal Prediction
Student Number: 2021 − 27025

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The goal of anomaly detection is to find abnormal data in multi-dimensional data. Anomalies can be identified for each variable using means, variance, quartiles, or the predicted residual error sum of squares(PRESS). This data refining method effectively screens data by removing anomalies. Additionally, in the context of machine learning for anomaly detection, defective goods or system flaws picked up by the detection model can aid the administrator in keeping a running system.

Anomaly detection has evolved into a crucial process. Several anomaly detection algorithms have been developed in accordance with these industries' specific requirements. Since there are fewer anomalous data points from systematic processes, approaches like oversampling, unsupervised learning, or semi-supervised learning have arisen to address the problem.

However, there are few studies on using real-time learning to detect anomalies in data. In this thesis, we examine properties of Robust Random Cut Forest(RRCF), which transforms Isolation

Forest among tree-based anomaly detection methods, and show the performance of RRCF models. We apply the RRCF method to a case study with sequentially generating data.

Current machine learning techniques arbitrarily define thresholds based on the outcomes of algorithms. As a result, it is difficult to update the threshold in real-time. To address this issue, we present a conformal prediction that method to update the threshold in real-time.

The issue of identifying pump system failure in real-time format will be covered in the case study. Sensor data on the pump was recorded every second and had a severe imbalance between labels for success and failure.

Chapter 2

Statistical Models

2.1 Keywords

2.1.1 Anomaly

Depending on the goal of the study and the type of information gathered, the definition of an anomaly can vary. For instance, it could refer to "outlier data" that significantly deviates from the original dataset or "novelty data" that falls under a brand-new category. Alternately, it could be "abnormal data" that occurred as a result of a data gathering mistake. In this study, our goal is to identify a suitable method for separating this differently defined anomaly from the normal, similar to creating an appropriate classification model when there are too many or too few data points on the specific label.

Anomaly detection is utilized because the data on each label is not balanced. The masking effect and the swamping effect should therefore be taken into consideration. The swamping effect relates

to a false positive in which normal data are mistakenly identified as anomalous, while the masking effect refers to a false negative in which anomalies are grouped and the model or test method cannot identify an true anomaly.

2.1.2 Imbalanced data

For the model to learn well, the balance between the label of data is important. The model can return all predictions with only one specific single class, especially when the imbalance between classes is quite severe. Yet even in this instance, the overall accuracy can be very high.

Therefore, an oversampling algorithm has emerged to amplify data to solve the problem of data imbalance. SMOTE(Synthetic Minority Oversampling Technique), introduced by Chawla et al. (2002), ADASYN(Adaptive Synthetic Sampling Approach for Imbalanced Learning), introduced by He et al. (2008), and Additional Generative Models, introduced by Xu et al. (2019), are three well-known algorithms that amplify structured data. However, there is a drawback in that the versatility of the data cannot be reproduced because these models generate data based on a limited quantity of previously gathered data.

On the other hand, there is a methodology that makes use of the current data's structure rather than data amplification. By focusing on the properties of the data itself or modeling only with normal data, an unsupervised learning or semi-supervised learning process can be used to identify the class. These techniques establish a suitable feature space and identify it as an anomaly when it

extends the feature space boundary.

2.2 Literature Review

2.2.1 Unsupervised Learning

Unsupervised learning collectively refers to a learning method that does not consider the response variable Y (Hastie et al., 2001). In other words, unsupervised learning is based on a marginal distribution $p(X)$ while supervised learning derives a conditional distribution $p(Y|X)$ for the independent variable X and the response variable Y. Popular unsupervised learning methods for anomaly detection are as follows.

1. DBSCAN(Density Based Spatial Clustering Application with Noise)

DBSCAN is based on a density-based clustering model (Ester et al., 1996). The three ideas of Core point, Noise point, and Border point are used in this method to find geometric grouping data. Using the ϵ value set by researchers, Points are categorized based on the quantity of data located in the ϵ radius of the data. The specific procedure for determining anomaly with DBSCAN is as follows.

Sets data X_i, radius ϵ, minimum number of data accepted as clusters m

1. Let the number of data present within the ϵ radius around

arbitrary data X_i be n_1.

2. If $n_1 \geq m$, assign the corresponding point X_i as a core point. If $n_1 < m$, assign the corresponding point X_i as a noise point.

3. Now repeat the following process. If one of the points existing within the ϵ radius of core point X_i is X_j, let's say the number of data in the ϵ radius around X_j is n_2 for this point.

4. If $n_2 \geq m$, assign the corresponding point X_j as a core point. If $n_1 < m$, assign the corresponding point X_i as a border point.

5. Repeat steps 1-4 to define the point characteristics for all data.

6. A cluster is created by connecting core point and border point, and for anomaly detection, it is considered normal data and the remaining noise points are classified as outliers.

2. Local Outlier Factor

Breunig et al. (2000) proposed a density-based classification model, the Local Outlier Factor model, which can be used to detect anomalies. This technique uses data from the neighborhood to determine the anomaly score. Using the hyperparameter k, the process of calculating the anomaly score of data using how close the K neighboring data exist around is summarized below.

- Data point X_i

- Number of neighborhood data you want to examine : k
- Distance between data : $d(X_i, X_j)$
- Average distance between k neighborhood data points and X_i : k-distance(X_i)
- neighborhood data closer than k-distance(X_i) for data X_i : $N_k(X_i)$

For $p \in N_k(X_i)$, the density of neighborhood data of the observation X_i may be expressed as follows.

local reachability density of data X_i (lrd)

$$lrd_k(X_i) = \frac{|N_k(X_i)|}{\sum_{p \in N_k(X_i)} \max\{\text{k-distance}(X_i), d(p, X_i)\}}$$

After comparing the density of the data with the density of its neighborhood data using the local reachability density value, the anomaly score of the data can be obtained as follows. The likelihood that anything is anomalous increases with the size of the anomaly score.

$$LOF_k(p) = \frac{\sum_{p \in N_k(X_i)} \frac{lrd_k(p)}{lrd_k(X_i)}}{|N_k(X_i)|}$$

3. Robust Covariance

Peña and Prieto (2001) proposed a distance-based anomaly detection method for multi-dimensional data. Researchers can classify normal and abnormal data using the Mahalanobis distance

while taking the covariance structure of the data into account and changing the border limits.

- multidimensional data X
- mean vector of each feature on data : $\mu_{\mathbf{x}} = [\mu_{\mathbf{x1}}, \mu_{\mathbf{x2}}, \cdots, \mu_{\mathbf{xn}}]$
- data covariance matrix Σ
- normal data limit range η
- Determine data as anomaly if $\sqrt{(X-\mu)^T \Sigma^{-1}(x-\mu)} \geq \eta$

A distance-based anomaly detection method that takes into account the distribution of data, especially by using Mahalanobis distance rather than Euclidian distance, reflects on the correlation between variables.

4. Isolation Forest

Liu et al. (2008) proposed a tree-based anomaly detection method. Abnormal data will deviate from normal data. That is, when inspecting the data on a binary tree data structure, the abnormal data will be found at the top of the tree because the abnormal data is more likely to branch first. On the other hand, normal data will be similarly clustered and close in distance, so a binary tree will recursively partition data and require a relatively large number of branches. That means the normal data is more likely to be located at the bottom of the tree.

- We create Isolation Tree for a sampling without replacement S from data X. Based on this, an Isolation Forest, which is

an ensemble structure, may be constructed and a masking effect through this sampling may be prevented.

- When making the tree, the criterion of the branch is an arbitrary value within a range of the arbitrarily selected variable

  - Sets the number of features constituting data as M. Then, feature selected as the basis of the branch has a probability of $\frac{1}{M}$.
  - Selects a junction $X_i \sim Unif[\min x_i, \max x_i]$ for the selected feature X_i.

- Let's calculate the number of branches of data point X_i for each tree as $h(X_i)$, and set that the average number of branches of points X_i in Forest is $E(h(X_i))$.

- For the typical average number of spliting $c(N)$ for n data points, we define an anomaly score for the entire data as follows.

$$Score(X_i, N) = 2^{-\frac{E(h(X_i))}{c(N)}}$$

2.2.2 Semi-Supervised Learning

According to whether or not response variables (also known as data labels) were used for learning, supervised learning and unsupervised learning were segmented in the previous section. On the other hand, semi-supervised learning seeks to develop a model while simultaneously using data with and without response variables. In order to effectively use data without response variables in

semi-supervised learning, Van Engelen and Hoos (2020) presented three conditions.

1. smoothness: If the density of predictor $P(x)$ is similar, the response variable accordingly is also the same or very similar.

2. low-density: No boundary of classification occurs in the part where the value of the probability density function $P(x)$ is high.

3. manifold: It follows the manifold assumption that the structure of high-dimensional data is composed of several low-dimension. Therefore, even high-dimensional complex data can be accessed by projecting it into low-dimensional structures, and several existing response variables help to infer the non-existing class of response variables.

Based on these three conditions, semi-supervised learning is the process of improving performance on unlabeled data by using labeled data; in this thesis, we will apply the methodology of Song et al. (2017). This is to conduct machine learning with only normal data when the proportion of normal data is very high. After that, data outside the normal data space is counted as an anomaly. In a similar preceding study, deep learning researchers suggested a strategy for categorizing anomalies using data feature space (Lukas et al., 2018).

2.2.3 Real-time Learning

The advance of technology has facilitated the accumulation of a vast amount of data. The amount of data attainable is still in-

creasing, especially as it becomes possible to use sensors to acquire data in real-time. The interest in real-time learning to instantly incorporate real-time data into the model has increased due to its larger size and faster data collection, and various past studies have been carried out.

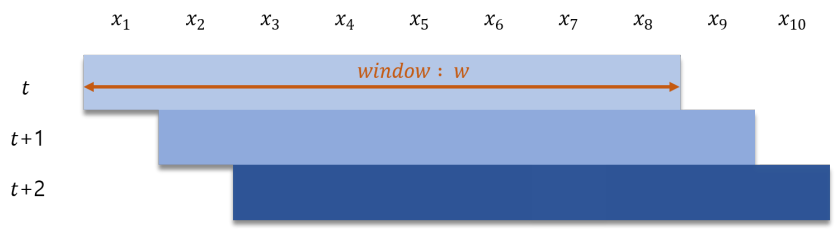
According to Cesa-Bianchi and Orabona (2021), real-time learning is defined as updating existing predictive models while processing data. Real-time learning techniques range from those that quickly update parameters in sequential data to those that use parameter-free real-time learning algorithms based on analysis and model properties. One well-known early real-time learning technique is the "online newton step" or "online linear regression."

According to Bahri et al. (2021), the characteristics of real-time learning and data can be summarized as follows.

- As the amount of information increases, high-dimensional data is generated, which affects memory issues and model learning time.
- In real-time learning, it is important to process data as quickly as possible.
- As the data itself evolves which concept drift means in AI or machine learning, the model required for prediction must also be changed in real time.
- Generating data labels is slower than the data itself. If data labeling is slow, it can be difficult to utilize in the model and the model performance is degraded in the concept drift situation.

- If the data structure is imbalanced, learning has no choice but to proceed with only a specific class in a streaming environment.

To deal with this real-time learning environment, numerous real-time learning procedures have been developed. Bahri et al. (2021) has outlined the approaches currently being employed. Among them, the sliding window model, which will be used for data analysis in Chapter 3.2, is well recognized. In a drift shift structure where the data distribution progressively changes, Ng and Dash (2010) referred to the idea of a continuous bundle of sequential data as a "window" to obtain the necessary information. In particular, sliding windows, which move windows sequentially over time, can be utilized if the purpose of the modeling is to predict the new data label by using some late data. This refers to sequentially extracting the most up-to-date information over time by utilizing the window size w to store information continuously in a data streaming environment. However, because only the most recent w bits of information are used according to the time stamp, the initial data in w size are not continuously taken into account over time. Below is a picture of the concept of window sliding.



2.3 Robust Random Cut Forest Model

Next, the Robust Random Cut Forest(RRCF) model is introduced.

2.3.1 Different Concepts of anomaly in models

The idea of anomaly has been freshly defined within the RRCF model, which is a tree-based model such as the Isolation Forest model previously discussed. RRCF defines anomaly based on how quickly the tree structure changes whether specific data is in or not, whereas earlier tree models defined anomaly based on data distant from a normal and specified cluster. Guha et al. (2016) defined this amount of depth changes as displacement. It serves as an anomaly score.

Definition 1. *Defines the displacement for the data point x. For tree T, let the probability of randomly selecting a tree T be P[T], and for the dataset Z, let's say the depth of the data point y is*

f(y,Z,T).

$$Disp(x, Z) = \sum_{T, y \in Z - \{x\}} p[T](f(y, Z, T) - f(y, Z - \{x\}, T'))$$

Additionally, even when data is clustered, abnormal data can be found because random sampling produces numerous tree structures in a forest. In other words, it avoids the masking effect and appropriately takes into account a cluster of data that hides an anomaly. Guha et al. (2016) called it colluder. In the same vein, it is necessary to calculate the amount of variations considering the colluder by appropriately capturing a cluster of data. The average value of the maximum displacement in a tree-changing colluder is what we refer to as "collaborative displacement" because we are unsure of how far to select the neighbors to value the colluder.

Definition 2. *Defines the collusive displacement for the data point x. Let a group that hides anomaly due to clustered data be C, the size of group C be* $|C|$*, and the sample used for that tree be S because it makes several trees.* $CoDisp(x, Z, |S|)$ *is defined as follows:*

$$\mathbb{E}_{S \subset Z, T}[\max_{x \in C \subseteq S} \frac{1}{|C|} \sum_{y \in S - C} (f(y, S, T) - f(y, S - C, T')]$$

2.3.2 Algorithm

In the beginning, one can rapidly spot sudden changes in values. Collusive Displacement, an anomaly score, has a clear value since it can avoid useless axial branching, unlike Isolation Forest, which makes use of the tree structure. This is due to the fact that

RRCF avoids situations where anomalies cannot be discovered by an ineffective dimension by choosing a feature that turns into a branch criterion with varying probability based on the range of data. The branching proceeds in the following manner.

1. For $l_i = \max x_i - \min x_i$, choose a random dimension(feature) in proportional to $\frac{l_i}{\sum_j l_j}$.
2. Choose a split point $X_i \sim Unif[\min x_i, \max x_i]$ for the selected dimension.
3. Create a tree by repeating the process of selecting $S_1 = \{x | x \in S, x_i \leq X_i\}$ as the left child node, $S_2 = S \setminus S_1$ as the right child node.

Second, it can be suitable for real-time modeling. That means updating and removing data inside the model is acceptable because this model is founded on the characteristic that a range of data is taken into account when branching. Guha et al. (2016) propose the following lemma.

1. For any feature k, its data points set S and new datat point p, the probability of choosing any cut in RRCF is the same as the conditional probability of selecting a new cut for updating data set $S \bigcup \{p\}$.
2. Considering a random tree of RRCF($S \bigcup \{p\}$), based on the fact that first branch split the new point p from S, the remainder is the same as RRCF(S)

2.4 Conformal Prediction

Conformal Prediction (CP) is to determine how appropriate the expected observations are based on current observations. On the basis of the CP framework, possible prediction candidates can be found under a suitable level of confidence, whereas general regression or classification only produces one prediction. The framework is simply as follows.

1. Set a confidence level.

2. Determine the prediction region based on the confidence level. The predicted value $\hat{y}$ belonging to the area is as follows.

$$\left\{ \hat{y} : \frac{|\{i = 1, \ldots, n : \hat{y} \geq y_i\}| + 1}{n + 1} \geq \text{confidence level} \right\}$$

3. The more accurate the prediction region is, the smaller the prediction region is, and the true value is more likely to belong to the region.

The prediction region, according to the previously introduced, is determined by comparing the anticipated value with the prior true value data. And with the same mechanism, if the data does not fall into a predetermined prediction region, it is determined to be an anomaly. Therefore, for the next step, to utilize Conformal Prediction for anomaly detection, designing a score indicating how 'strange' the data is and creating criteria for the prediction region are required. This is referred to as a Non-Conformity Measure(NCM) in earlier studies. From the RRCF's point of view, an

algorithm can be created to determine anomalies by setting the anomaly score returned through RRCF as NCM.

The whole algorithm is as follows.

Algorithm 1 RRCF+CAD

1: For data X, make Robust Random Cut Tree (RRCT) T_n for sample without replacement S_n
2: Calculate the codisplacement value for each data point x from RRCT.
3: Codisplacement is specified as a non-conformity score in conformal prediction framework to determine how extreme that score is.
4: Consider domain characteristics or data, specify confidence level
5: Data whose codisplacement does not belong to the confidence region is determined as an anomaly.

By using the conformal prediction framework for anomaly detection, it is possible to indicate the statistical significance level for detection results. In other words, controlling the significance level is the same as controlling the threshold, which becomes an anomaly.

2.5 Real-time Learning model with RRCF+CAD

The CP framework can continuously determine whether the anticipated value satisfies the confidence level, making it applicable to real-time learning scenarios. Due to the algorithm's struc-

ture, which has already been discussed, RRCF is very simple to update and remove data without having to learn all of the data again. Therefore, we propose RRCF+CAD in this thesis, which combines RRCF and CP framework.
It has been suggested in previous studies to combine KNN, SVM, or other algorithms with the CP framework, but this approach has the drawback that it takes a while to acquire NCM for each technique and makes it challenging to update the model right away with new data.

2.6 Metrics

Anomaly detection is eventually a classification problem, specifically a binary classification problem to determine whether something is normal or not. There are several evaluation metrics that evaluate the performance of a model in a classification problem. ROC curve to compare the AUC value that examines the area at the bottom may be a good choice, or we could pay attention to Accuracy, Recall, Precision, F-beta-score, which are derived from contents in the Confusion Matrix. In particular, F-beta-score is a value created by a combination of harmonic means and particular weights for recall and precision. The evaluation criteria that researchers want to stress can be weighted differently to ensure proper evaluation. Here are the specifics:

F - beta - measure

- $\frac{(1+\beta^2)\cdot Precision\cdot Recall}{\beta^2\cdot Precision+Recall}$

- There are various F-measure depending on the β value of the evaluation.

    1. F1-measure : false-negative and false-positive are all important
    2. F2-measure : false-negative is important
       $\because$ maximize recall $\rightarrow$ minimize false negative
    3. F 0.5-measure : false-positive is important

Because the absolute value of evaluation metrics like Accuracy and AUC is too high or the difference in the score value is so small, it is challenging to view them as desired measures in the field of anomaly detection. That's because there are many fewer abnormal data points than normal data. Bifet et al. (2015) outlines the drawbacks of current evaluation techniques for real-time learning models and suggests evaluation metrics that help address these issues. The F-beta-score, which may assess the model's performance from imbalanced data, is used in this thesis to judge the analytical findings of real data. Because it is more crucial to limit the number of false negatives when the model can predict the failure of the pump sensor, we will utilize an F2-score in Chapter 3.3 of this study that prioritizes recall above precision. Particularly, we would like to utilize Macro-F2-score that is computed by arithmetic mean for all class F2-score to cover the extreme data imbalance.

Chapter 3

Case Study

3.1 Data Description

Pump sensor data collected from Kaggle (https://www.kaggle.com/datasets/nphan
sensor-data) were utilized to examine the efficiency of the RRCF+CAD algorithm described in Chapter 2. To prevent performance failures, we wish to verify the effectiveness of the RRCF+CAD algorithm, which identifies pump system faults in advance. The data that must be examined is a time series generated by the pump sensor's values.

Table 3.1 represents a portion of the raw data. It briefly depicts a cycle of normal operation, failure, repair, and return to normal.

Table 3.1: Pump sensor data

Timestamp	Sensor_00	Sensor_01	...	Sensor_51	Machine_status
2018-04-01 0:00	2.465394	47.09201	···	201.3889	NORMAL
2018-04-01 0:01	2.465394	47.09201	···	201.3889	NORMAL
2018-04-01 0:02	2.444734	2.444734	···	203.7037	NORMAL
:	:	:	:	:	:
2018-04-12 21:55	Null	53.34201	···	324.6528	BROKEN
2018-04-12 21:56	Null	53.55902	···	341.7245	RECOVERING
:	:	:	:	:	:
2018-04-13 13:39	0.305961	50.43402	···	38.77315	RECOVERING
2018-04-13 13:40	0.305961	50.43402	···	38.77315	NORMAL

Seven system failures occurred in total throughout the data collection period (April 2018 *sim* October 2018), with data being collected at one-second intervals. In other words, just seven data points out of more than 200,000 are anomalies, and they are noted in the "machine status" variable with the label *BROKEN*, which denotes a pump failure. After a system failure, not all sensors stop working; some continue to function, and the value is recorded. On the other hand, the sensor measurements of the parts directly related to the failure are not recorded.

This information is initially recorded in NORMAL condition, and when a pump failure is discovered, a process to repair the pump is necessary. Therefore, in this situation, *RECOVERING* is recorded in the variable "machine status." Null values appeared in the sensor during the recovery process, especially for sensors that failed. Once the recovery process is complete, the machine status is updated to *NORMAL*. That indicates that the regular process

can start.

Anonymity is a characteristic of this data. Because of this, it is challenging for us to interpret the features. Therefore, proper feature interpretation will enable us to execute more reliable preprocessing.

We start by examining the data plot across time. Axis X represents time, and axis Y represents sensor value. When a failure happens, it is indicated by the symbol "X" in blue, and when a recovery follows the failure, it is indicated in green. If a value is missing from any sensor, it is indicated in red. The plot was removed for sensors 15 and 50 which had a significant missing value ratio.

Figure 3.1: Time series plot :sensor_00 ∼ sensor_07

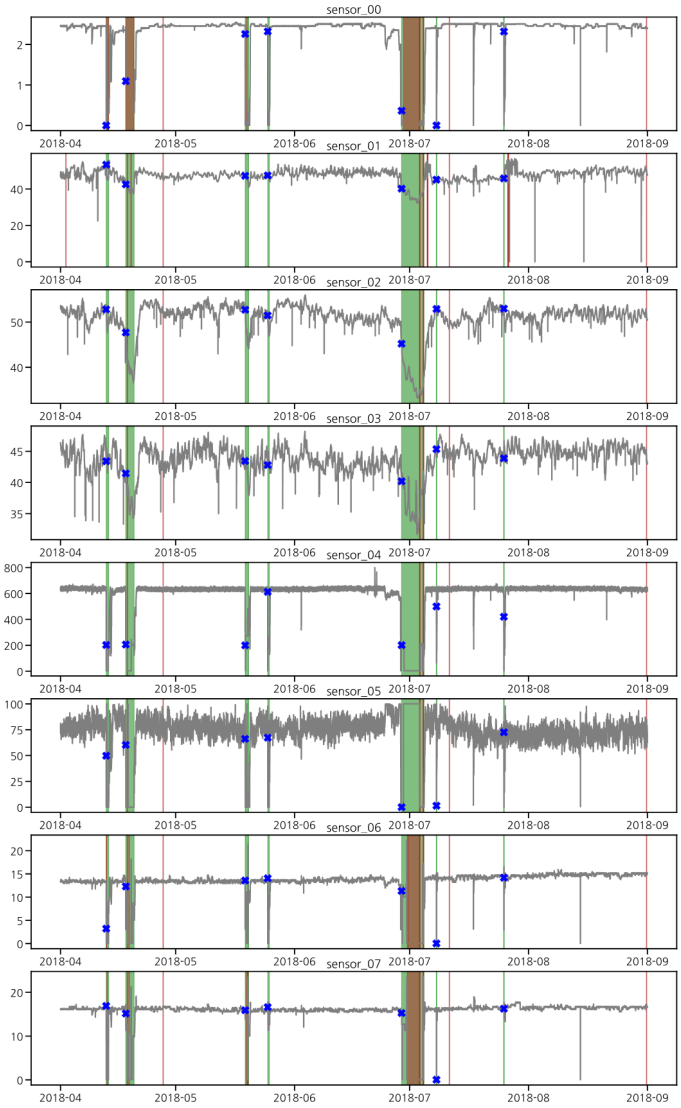


Figure 3.2: Time series plot : sensor_08 ∼ sensor_14, sensor_15

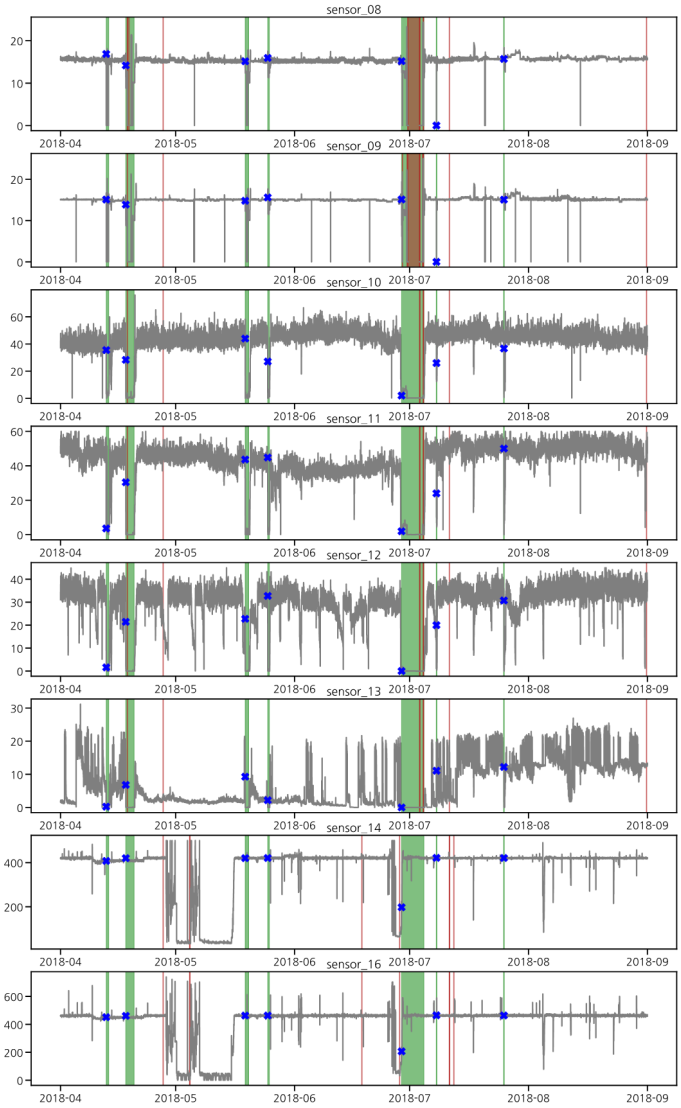


Figure 3.3: Time series plot : sensor_17 ∼ sensor_24

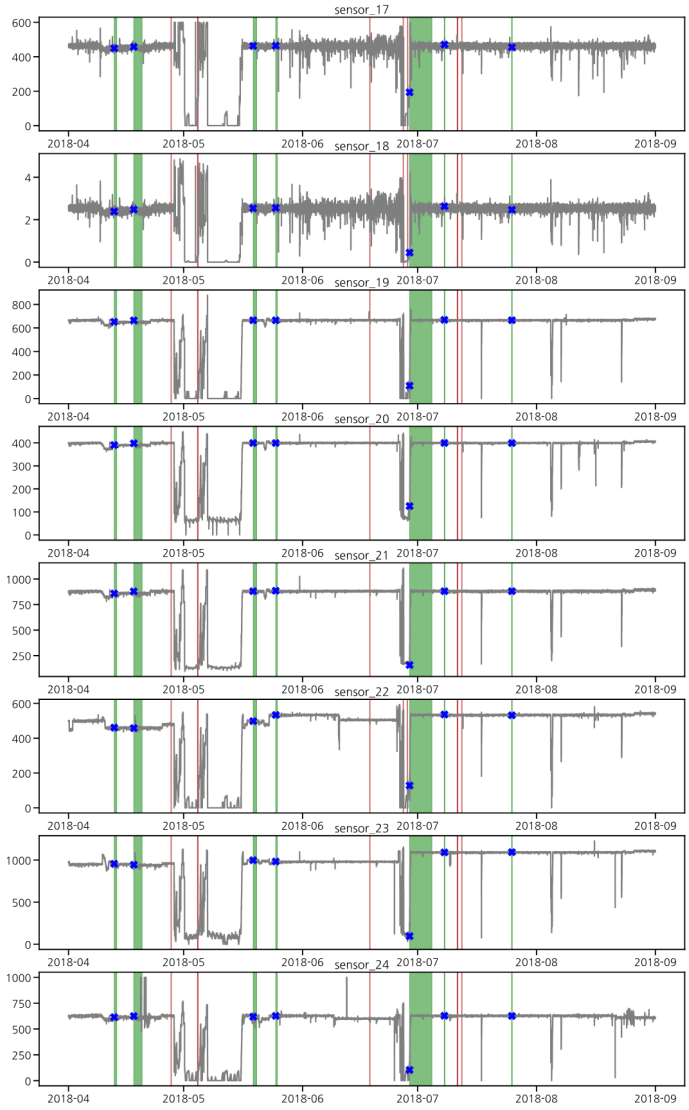


Figure 3.4: Time series plot : sensor_25 ∼ sensor_32

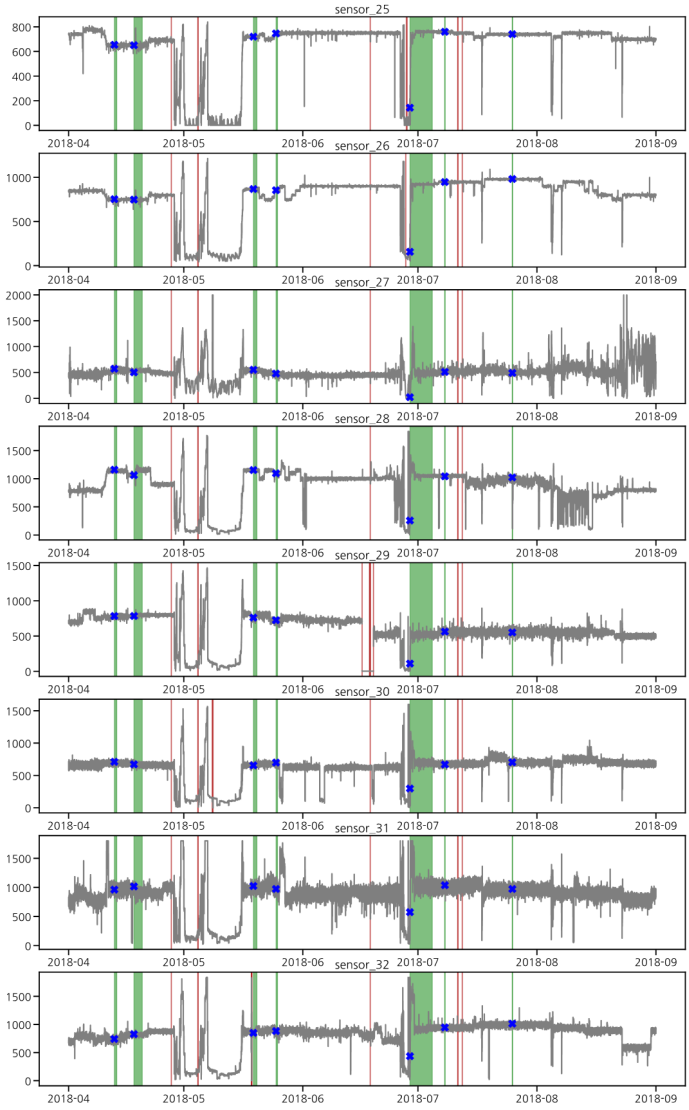


Figure 3.5: Time series plot : sensor_33 ∼ sensor_40

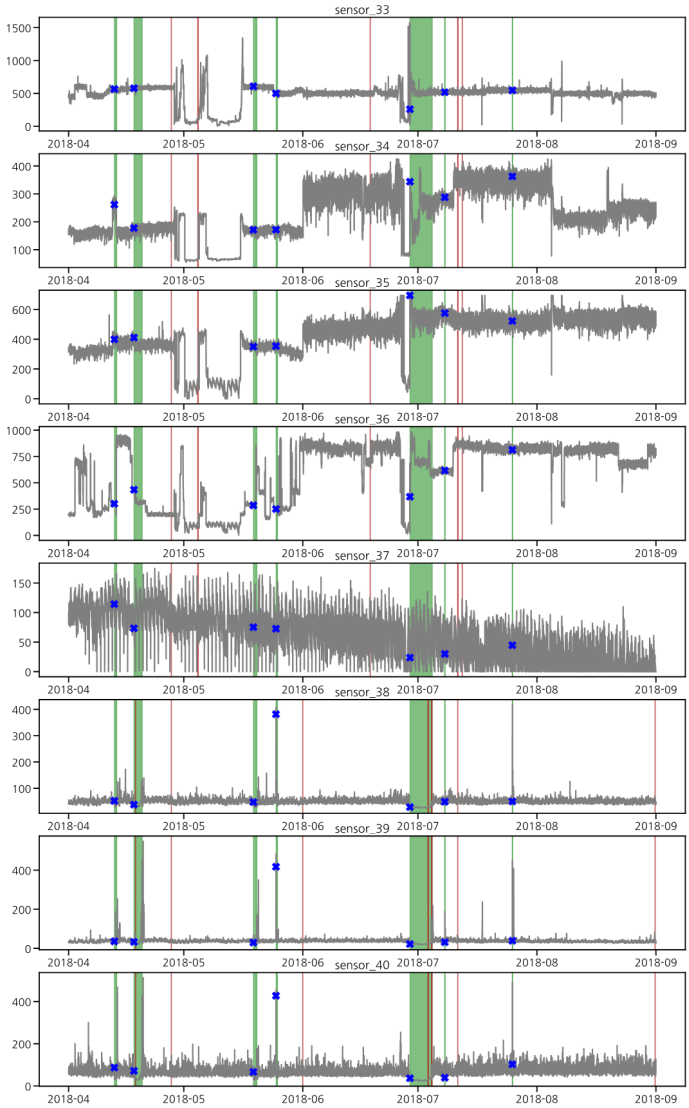


Figure 3.6: Time series plot : sensor_41 ∼ sensor_48

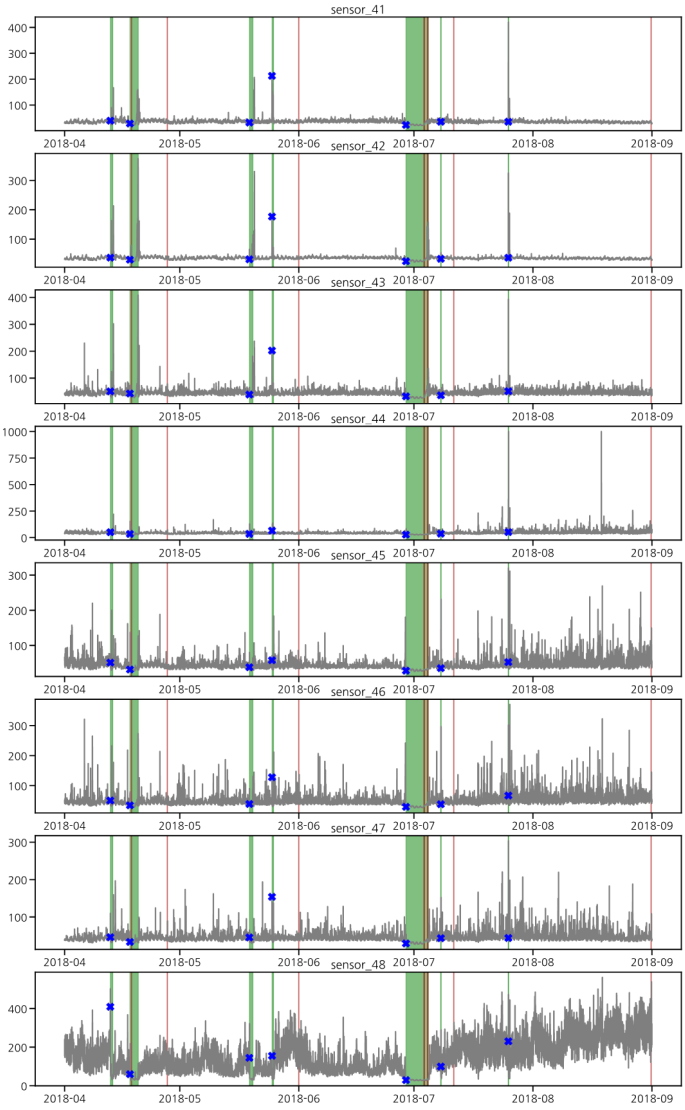
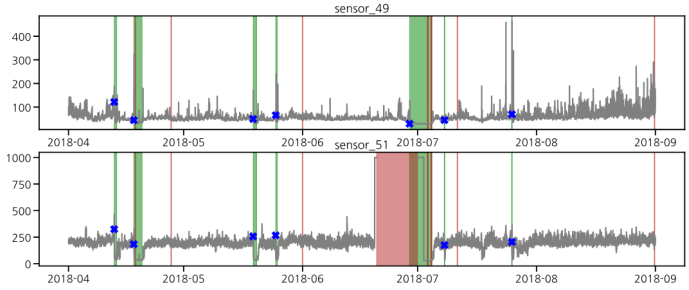


Figure 3.7: Time series plot : sensor_49, sensor_51



3.2 Data Analysis

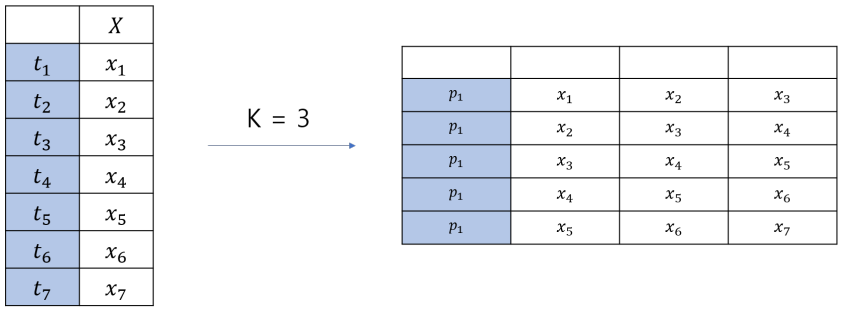
1. Data with missing values was initially processed for real-time modeling. The 51 total sensors are not used for analysis since sensors 15 and 50 have unusually high missing values of 1 and 0.35, respectively. For the remaining sensor values, a simple substitution through the previous value was used. This is for the most intuitive and fast missing imputation for real-time modeling. Additionally, when viewing the data as a time series per variable, the value of 0 can occasionally be seen; therefore, taking into account the time series data's autocorrelation, it was handled as a missing value and interpolated.

2. All available features were included in the model. When analyzing the correlation between variables in the data, some variables have a very high correlation. Nevertheless, because all data aspects are masked, it is challenging to determine the precise meaning of each variable, and if the correlation is large, it might be much more challenging to identify a good variable to omit. It is fair to include all variables because, in the event of pump failure, if strongly correlated variables change swiftly at the same time, employing an RRCF model that detects changes in variable values will be more sensitive. In fact, the performance was better.

3. In the analysis using time series data, the data value was shifted using the K-Shingling technique to detect abnormal errors in advance. Anand and Jeffrey (2010) suggested the shingling technique to combine short-term pieces of consecutively generated time series data into a single dataset. By appropriately modifying the k value and batching and processing the data created within a

given time, it has the advantage of lowering the similarity of time series data.

Figure 3.8: shingling example : k=3



4. There are only seven failures to predict through modeling, and if a failure occurs, the process of repairing the pump machine continues. As a result, total data was reconstructed into 7 data sets by grouping them into cycles beginning with the repair and ending with the failure.

5. For semi-supervised learning, 3000 initial normal data points are used to model an initial RRCF before real-time data are added to the model. Using the sliding window model proposed by Ng and Dash (2010), we take the oldest data out of 3000 from the model while adding the latest data to it, and then calculate and return the codisplacement value of the incoming data from the model. With the assistance of the domain specialist, the researcher may have set the initial 3,000 in a more appropriate manner. In the event of a failure, the machine starts the recovery process. After the failure and recovery procedure is complete, the model instantly begins real-time learning and prediction utilizing the data created

in the subsequent cycle, based on RRCF with data collected one minute prior to the failure.

6. The model must sound an alarm prior to the failure time because the challenge at hand is to foresee the failure in advance and prevent the failure of the pump machine. Therefore, even when assessing a model's success, the timing of the failure will be based on whether the failure is classified well in advance, not on whether the point of failure is well classified. To this end, the performance of the model will also be evaluated by considering the six points immediately before the failure as failures.

7. According to Guha et al. (2016), the RRCF model calculates the data's codisplacement value concurrently with learning the new data. The p-value value is calculated by comparing this with the pre-computed 3000 codisplacement values. Conformal prediction determines significance level based on domain characteristics or the researcher's settings, and this level can be chosen from a variety of factors.

3.3 Analysis Results

3.3.1 RRCF and RRCF+CAD

We will examine the model's performance when the RRCF+CAD algorithm's significance level, which is established by normal data, is set to 0.003. The reason for setting the lower limit as above is that of the 3000 data points used for the real-time learning model,

the value of 1/3000 comes out when the data of interest is the most extreme codisplacement value. Time series graphs depict the detection of pump failure using RRCF in black and RRCF+CAD in red for each cycle from the normal state right after repair to the point of each failure for 7 pump failures.

We discuss the rationale behind the significance level control used to determine how extreme the current value is in comparison to the non-conformity score (in RRCF, codsiplacement) of the previously produced data for RRCF+CAD. As can be seen in the graph below, using the RRCF+CAD model, we demonstrate that red dots appear one after another and indicate an impending machine failure. When the values observed by the pump sensors change quickly, a failure is anticipated, and an alert signal is released. On the other hand, it's crucial to figure out how large the codisplacement value is while utilizing solely the RRCF model. That is, the RRCF method by itself does not constitute a probabilistic control (significance level). It is challenging to estimate the model's codisplacement value since it depends on the distribution of the data and the data cycle, and the standards for determining this value are similarly empirical. As a result, in this thesis, the choice was based on the number 350, which optimizes the macro-F2-score among the model's performances.

Figure 3.9: failure prediction alerts when 1,2,3,4th failure

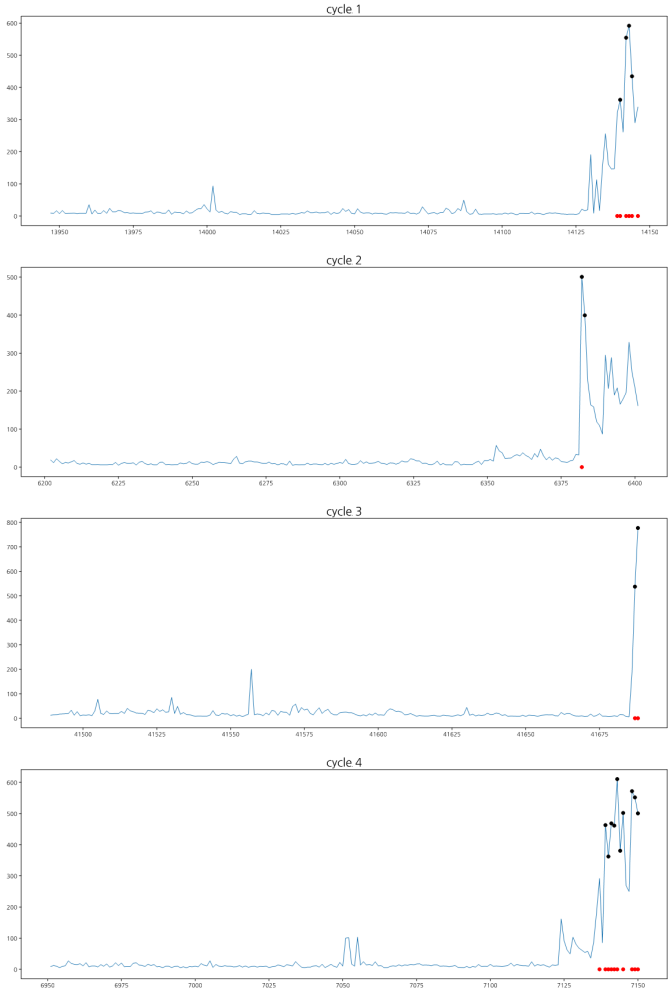
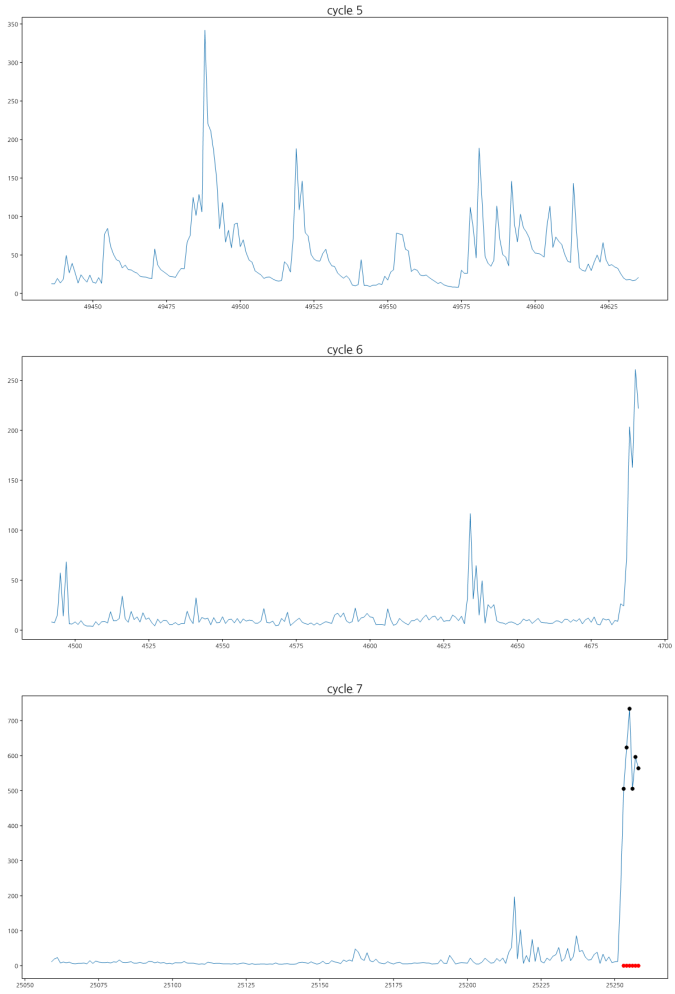


Figure 3.10: failure prediction alerts when 5,6,7th failure



Let's contrast the RRCF model with the threshold set experimentally with the RRCF+CAD model with the threshold set on a reasonable basis. Both models failed to forecast the fifth and sixth failures at all, although it is clear that the failures were similarly identified beforehand for the other failures. Following is a compar-

ison of how well the two models performed on all of the data up to the seventh failure. As described in Chapter 2, evaluation metrics will use Accuracy, Recall, Precision, F1-score, and F-beta-score. Since the data is so severely unbalanced, we will mainly compare Macro-F2-score among the six metrics to check if the model just predicts failures.

It is confirmed that RRCF+CAD is showing better performance on all metrics.

Table 3.2: Evaluation metircs for total data : RRCF vs RRCF+CAD

	F2-score (marco)	F1-score (marco)	Recall	Precision	Accuracy
RRCF+CAD	***0.4720***	***0.5291***	***0.3469***	***0.0337***	***0.9965***
RRCF	0.4709	0.5289	***0.3469***	0.0317	0.9963

3.3.2 RRCF+CAD and Other Machine Learning Models

Previously, we compared the performance differences between the existing RRCF model and the RRCF+CAD model proposed in the paper using the whole dataset with 7 failures. In this part, the evaluation metrics for seven failure cycles for RRCF+CAD, Isolation Forest, Local Outlier Factor, and Robust Covariance are compared.

Isolation Forest is an unsupervised learning method, as mentioned in Chapter 2.2.1. In the Python package for data analysis, the ratio, which is the number of anomalies to the total data, should be set as a hyperparameter. In order to do this, the anomaly

ratio of 0.003 was applied, the same as in RRCF+CAD. Similarly, for the Local Outlier Factor model, the number of neighborhood data to be explored was used equally as the 3000 data that were retained within the model in RRCF+CAD. By adjusting the Mahalanobis distance while taking the covariance structure of the data into consideration, the Robust Covariance technique identified anomalous data. The data with the greatest distance from an anomaly that met the requirement that the ratio of the anomaly is 0.003 as in the RRCF+CAD model was used for the determination. The results are listed below, organized by cycle.

Table 3.3: Evaluation metrics for 1th cycle : RRCF+CAD vs others

	F2-score (marco)	F1-score (marco)	Recall	Precision	Accuracy
RRCF+CAD	***0.6700***	***0.5954***	***0.7142***	***0.1111***	***0.9970***
IF	0.5926	0.5501	0.4286	0.0577	0.9969
LOF	0.4835	0.4878	0.4286	0.0030	0.9412
RC	0.5926	0.5501	0.4286	0.0577	0.9969

Table 3.4: Evaluation metrics for 2th cycle : RRCF+CAD vs others

	F2-score (marco)	F1-score (marco)	Recall	Precision	Accuracy
RRCF+CAD	***0.4990***	***0.4992***	0	0	***0.9967***
IF	0.4958	0.4972	0	0	0.9888
LOF	0.4123	0.4377	***1***	***0.0046***	0.7643
RC	0.4986	0.4989	0	0	0.9958

Table 3.5: Evaluation metrics for 3th cycle : RRCF+CAD vs others

	F2-score (marco)	F1-score (marco)	Recall	Precision	Accuracy
RRCF+CAD	**0.5218**	***0.5093***	***0.2857***	***0.0109***	0.9955
IF	0.4988	0.4992	0	0	***0.9968***
LOF	0.4291	0.4531	0.1429	0.0001	0.8279
RC	0.4988	0.4992	0	0	***0.9968***

Table 3.6: Evaluation metrics for 4th cycle : RRCF+CAD vs others

	F2-score (marco)	F1-score (marco)	Recall	Precision	Accuracy
RRCF+CAD	***0.7165***	***0.6594***	0.5714	***0.2222***	***0.9976***
IF	0.4987	0.4989	0	0	0.9959
LOF	0.5436	0.5158	***1***	0.0258	0.9631
RC	0.5487	0.5335	0.1429	0.0454	0.9962

Table 3.7: Evaluation metrics for 5th cycle : RRCF+CAD vs others

	F2-score (marco)	F1-score (marco)	Recall	Precision	Accuracy
RRCF+CAD	0.4986	0.4991	0	0	0.9963
IF	***0.5977***	***0.5442***	***1***	***0.0470***	***0.9971***
LOF	0.4147	0.4430	0	0	0.7953
RC	0.4988	0.4992	0	0	0.9968

Table 3.8: Evaluation metrics for 6th cycle : RRCF+CAD vs others

	F2-score (marco)	F1-score (marco)	Recall	Precision	Accuracy
RRCF+CAD	0.4998	0.4995	0	0	0.9982
IF	0.4986	0.4988	0	0	0.9953
LOF	0.4442	0.4633	0	0	0.8634
RC	***0.6151***	***0.5899***	***0.2857***	***0.1333***	***0.9962***

In light of the findings, it cannot be said that RRCF+CAD consistently outperforms other methods. In particular, the IF model performed exceptionally well in cycle 5 when compared to other models, while the robust covariance model performed quite well

Table 3.9: Evaluation metrics for 7th cycle : RRCF+CAD vs others

	F2-score (marco)	F1-score (marco)	Recall	Precision	Accuracy
RRCF+CAD	***0.6603***	***0.5827***	0.8571	***0.0923***	***0.9976***
IF	0.4988	0.4992	0	0	0.9967
LOF	0.4945	0.4933	***1***	0.0057	0.9518
RC	0.4988	0.4992	0	0	0.9967

in cycle 6. However, the RRCF+CAD model outperformed other models based on the macro-F2-score in other cycles (1, 2, 3, 4, and 7). In particular, the RRCF+CAD model updates the data in real-time over time, achieving results comparable to or better than popular algorithms by taking into account its predicted performance and updating the model. In contrast, other models measured the performance of the model with full data for each cycle.

Chapter 4

Conclusion

Anomaly detection using machine learning is used in various fields. Signals can be found to stop failures by total score using the RRCF+CAD method presented in this work. This allows a typical system to function in any situation where a sensor is used. Additionally, when it comes to the world of finance, RRCF+CAD algorithms might be utilized to detect credit fraud and anticipate fraudulent card transactions.

Additionally, the volume of data increases, and the generated data loads into the computer quickly. The reality or trend is changing faster than ever, so it is important to choose wisely based on the most recent information. Given these changes, efforts to handle anomalies and their real-time detection will contribute to discovering new values. In this thesis, I examine prior research and use real-time learning models, the RRCF+CAD model, and a case study using them.

The RRCF+CAD model processes data in real-time, maintaining only the appropriate data within the model while demon-

strating superior performance. Of course, more investigation will be required to determine how much data is kept in the model. In addition, it would be a more valuable study if other real-time models could be compared together. Furthermore, it is expected in future work that a confidence interval using bootstrapping can be set while utilizing Conformal Prediction, or that research using a multiple testing method can be conducted using the codisplacement value of a specific interval.

Bibliography

Shafer, G. and Vovk, V. (2008). A Tutorial on Conformal Prediction. *Journal of Machine Learning Research*, **9**, 371-421.

Guha, S., Mishra, N., Roy, G. and Schrijvers, O. (2016). Robust Random Cut Forest Based Anomaly Detection On Streams. *Proceedings of the 33 rd International Conference on Machine-Learning.*, **48**.

Liu, F.T., Ting, K.M. and Zhou, Z.-H. (2008). Isolation forest. *In 2008 Eighth IEEE International Conference on Data Mining.*, 413-422.

Breunig, M.M., Kriegel, H.-P., Ng, R.T. and Sander, J.(2000). LOF: Identifying Density-Based Local Outliers. *SIGMOD Conference, ACM.*, 93-104.

Chandola, V., Banerjee, A. and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computational Survey 41*, **15**, 1-58.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, **41**, 226–231.

Van Engelen, J.E. and Hoos, H.H.(2020). A survey on semi-supervised learning. *Machine Learning*, **109**, 373-440.

Bahri, M., Bifet, A., Gama, J., Gomes, H.G. and Maniu, S. (2021). Data stream analysis: Foundations, major tasks and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.*

Cesa-Bianchi, N. and Orabona, F. (2021). Online learning algorithms. *Annual reviews*

Anand, R. and Jeffrey, U. (2010). Mining of Massive Datasets. *Cambridge University Press*, **3**.

Ng, W. and Dash, M. (2010). Discovery of Frequent Patterns in Transactional Data Streams. *Transaction on large-scale data and knowledge-centered systems II.*

Bifet, A., Morales, G.d F., Read, J and Pfahringer, B. (2015). Efficient online evaluation of big data stream classifiers *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 59-68.

Song, H., Jiang, Z., Men, A. and Yang, B. (2017). A Hybrid Semi-Supervised Anomaly Detection Model for High-Dimensional Data. *Computational Intelligence and Neuroscience*, **17**, 1-9.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). The Elements of Statistical Learning. *Springer Series in Statistics Springer New York Inc.*, **2**.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal Of Artificial Intelligence Research*, **16**, 321-357.

He, H., Y. Bai, E. A. Garcia, and S. Li. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 1322-1328.

Xu, L. Skoularidou, M., Cuesta-Infante, A. and Veeramachaneni, K. (2019). Modeling Tabular Data using Conditional GAN. *Proceeding of the 33th Conference on Neural Information Processing Systems*.

Peña, D. and Prieto, F.-J. (2001). Multivariate Outlier Detection and Robust Covariance Matrix Estimation. *Technometrics*, **43**, 286-310.

Lukas, R., Robert, V., Nico, G., Lucas, D., Shoaib, A.S., Alexander, B., Emmanuel, M. and Marius, K. (2018). Deep One-Class Classification. *Proceedings of the 35th International Conference on Machine Learning*, **80**, 4393-4402.

국문초록

류환감
통계학과
대학원
서울대학교

본 논문에서는 정상 데이터와는 다른 데이터를 분리해내어 시스템의 안정을 구축하는 이상치 탐지에 대해 알아본다. 체계적이고 지속가능한 시스템을 위해서 이상 데이터를 지속적으로 감시하고 분류하는 작업은 중요한 역할을 하며, 통계적 방법 뿐만 아니라 머신러닝/딥러닝을 활용한 다양한 방법론이 사용되고 있다. 본 논문에서는 선행 연구에서 사용된 방법론을 간략히 소개한 뒤, Robust Random Cut Forest Model과 Conformal Prediction을 결합한 실시간 이상탐지 방법인 RRCF+CAD 모델을 제안한다. 이 방법은 모델의 실시간 업데이트가 가능하며 이를 기반으로 데이터의 이상 score를 찾아 통계적 검정 방법을 실행한다.

주요어 : 이상치, 이상 스코어, 준지도학습, 로버스트 랜덤 컷 포레스트, 적합 예측
학 번 : 2021 – 27025