future internet

MDPI

Article

Utilizing Random Forest with iForest-Based Outlier Detection and SMOTE to Detect Movement and Direction of RFID Tags

Ganjar Alfian [1,*], Muhammad Syafrudin [2,*], Norma Latif Fitriyani [3], Sahirul Alam [1], Dinar Nugroho Pratomo [1], Lukman Subekti [1], Muhammad Qois Huzyan Octava [1], Ninis Dyah Yulianingsih [1], Fransiskus Tatas Dwi Atmaji [4] and Filip Benes [5]

[1] Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia
[2] Department of Artificial Intelligence, Sejong University, Seoul 05006, Republic of Korea
[3] Department of Data Science, Sejong University, Seoul 05006, Republic of Korea
[4] Industrial and System Engineering School, Telkom University, Bandung 40257, Indonesia
[5] Department of Economics and Control Systems, Faculty of Mining and Geology, VSB—Technical University of Ostrava, 70800 Ostrava, Czech Republic
* Correspondence: ganjar.alfian@ugm.ac.id (G.A.); udin@sejong.ac.kr (M.S.)

Abstract: In recent years, radio frequency identification (RFID) technology has been utilized to monitor product movements within a supply chain in real time. By utilizing RFID technology, the products can be tracked automatically in real-time. However, the RFID cannot detect the movement and direction of the tag. This study investigates the performance of machine learning (ML) algorithms to detect the movement and direction of passive RFID tags. The dataset utilized in this study was created by considering a variety of conceivable tag motions and directions that may occur in actual warehouse settings, such as going inside and out of the gate, moving close to the gate, turning around, and static tags. The statistical features are derived from the received signal strength (RSS) and the timestamp of tags. Our proposed model combined Isolation Forest (iForest) outlier detection, Synthetic Minority Over Sampling Technique (SMOTE) and Random Forest (RF) has shown the highest accuracy up to 94.251% as compared to other ML models in detecting the movement and direction of RFID tags. In addition, we demonstrated the proposed classification model could be applied to a web-based monitoring system, so that tagged products that move in or out through a gate can be correctly identified. This study is expected to improve the RFID gate on detecting the status of products (being received or delivered) automatically.

Keywords: RFID; IoT; machine learning; tag direction; outlier detection; data balancing

check for updates

Citation: Alfian, G.; Syafrudin, M.; Fitriyani, N.L.; Alam, S.; Pratomo, D.N.; Subekti, L.; Octava, M.Q.H.; Yulianingsih, N.D.; Atmaji, F.T.D.; Benes, F. Utilizing Random Forest with iForest-Based Outlier Detection and SMOTE to Detect Movement and Direction of RFID Tags. *Future Internet* **2023**, *15*, 103. https://doi.org/10.3390/fi15030103

Academic Editors: Leonilde Varela, Goran D. Putnik and Claude Chaudet

Received: 17 January 2023
Revised: 5 March 2023
Accepted: 6 March 2023
Published: 8 March 2023

1. Introduction

Industry 4.0 allows for the digitalization of industrial machinery, operations, and assets, which offers new understandings and effectively addresses current challenges [1]. IoT (Internet of Things) is a key technology driving the fourth industrial revolution, also known as Industry 4.0. Industry 4.0 is the incorporation of cutting-edge technologies, such as IoT, Artificial Intelligence, big data, and cloud computing, into the manufacturing and industrial sectors. By connecting and communicating physical objects to the internet using IoT, large amounts of data can be gathered and analyzed, resulting in enhanced automation, efficiency, and decision-making in industrial procedures. For example, IoT-enabled sensors can be used for monitoring the environmental temperature of server rooms [2,3], network monitoring systems [4,5], monitoring automotive manufacturing [6,7], and healthcare monitoring systems [8,9].

RFID, or Radio-Frequency Identification, is a technology that uses radio waves to communicate between a reader and a tag attached to an object. RFID is considered a part of the Internet of Things (IoT) because it allows for the identification and tracking of physical objects over the internet, enabling them to connect and communicate with other devices and systems. This enables real-time monitoring, automation, and remote control of the objects. RFID tags can be placed on products and used to monitor their movement in and out of a warehouse by installing RFID readers at the gates. However, the reader cannot determine the direction of the tags, whether they are entering or exiting the warehouse. Previous studies have shown that using the received signal strength (RSS), timestamps, and machine-learning algorithms can be used to determine false positive tags [10–14]. However, these studies have not evaluated different types of tag movement that may occur in real-world scenarios. Therefore, it is important to use machine-learning models to identify the direction of tags by considering different types of tag movement to improve the efficiency of the RFID gate [15–18].

Random Forest is an ensemble machine learning method that combines multiple decision trees to improve classification and regression predictions by reducing overfitting and providing an estimate of feature importance. RF models have been effectively used for classification problems and have improved system performance [19–24]. However, machine learning algorithms often face difficulties such as outliers and imbalanced datasets, which can decrease accuracy. Studies have shown that by removing outliers using the Isolation Forest (iForest) method [25–28] and balancing imbalanced data with Synthetic Minority Over Sampling Technique (SMOTE) [29–32], the performance of the prediction system can be improved.

Despite this, there is no study on integrating the iForest-based outlier detection, SMOTE, and RF classifiers to improve the performance of RFID gates. Therefore, this study proposes a prediction model that combines iForest-based outlier detection, SMOTE, and RF to predict the movement and direction of RFID tags based on RSS and timestamps. Furthermore, implementing the proposed predicted model into a web-based monitoring system could detect the direction of the tagged product automatically. The contributions of the present study can be summarized as follows:

- We proposed a combined method of iForest outlier detection, SMOTE data balancing and Random Forest to classify movements and directions of RFID tags, which has never been done before.
- We evaluated the performance of the proposed prediction model on our dataset by considering more complex movements and directions of RFID tags that can happen in real warehouse environments.
- We improved the performance of the proposed model by removing the outlier and balancing the training-set.
- We conducted extensive comparative experiments on the proposed model with other prediction models and previous study results.
- We provided the impact analysis of outlier detection and data balancing method with or without iForest and SMOTE toward model's accuracy performance.
- Finally, we demonstrated the practicability of our proposed model by designing and developing the web-based RFID monitoring system.
- In addition, implementing the proposed predicted model into a web-based RFID monitoring system could be applied in warehouse to detect the direction of the tagged product automatically.

The rest of this study is organized as follows: In Section 2, we present machine learning models, including iForest, SMOTE, and RF. In Section 3, we explain the proposed prediction model for tag movement and direction. In Section 4, we report the results of our experiments and the implementation of our model. Lastly, in Section 5, we provide the conclusion, including any limitations of the study and potential areas for future research.

2. Literature Review

2.1. Tag Movement and Direction

RFID technology can be applied in a warehouse to identify items with RFID tags within the reader's range. While RFID readers can be used to track products as they are loaded onto trucks for transportation to other supply chain partners, the reader cannot differentiate between tags that are intentionally moving through the gate and those that happen to be in the reading range by chance. Several techniques have been suggested to tackle this problem, including using RSS and timestamps.

Previous studies have examined the use of RSS information to detect false positive tags. Keller et al. [10] looked at the characteristics of both static and moved tags from low-level reader data, including RSS. They explored the optimal threshold for separating static (false positive) and moved (true positive) tags. They developed a classifier based on a single RSS feature and achieved an accuracy of 95.69%. Ma et al. [11] recently addressed false positive detection based on RSS. They used several machine learning algorithms (DT, support vector machines (SVMs), and LR) to classify RFID readings using features related to RSS and phase shift. The SVM-based approach outperformed all other models, achieving the highest accuracy (up to 95.5%).

Zhu et al. [12] propose an algorithm to address the problem of false positive readings from supply chain radio frequency identification (RFID) systems. They extend the scenario to more complex access control systems, where detecting false-positive and false-negative data is critical to improving security and user satisfaction. The RFID data was divided into 70% training and 30% test data. Four training methods were used: ASOINN, SVM, LR, and DT, and the results show that all methods performed well. To improve prediction accuracy, Alfian et al. [13] proposed false positive detection based on RF and Inter Quartile Range (IQR) outlier detection. By removing the outlier, the prediction accuracy can be improved. The result shows that the integrated model successfully detects moving tags up to 97.496. Furthermore, Motroni et al. [14] proposed a model to classify the tags that move statically and pass-through portals. Sample data collected from an RFID reader and an array antenna are classified using SVM and LSTM. The results showed that SVM has an accuracy of more than 99%, while LSTM has an accuracy of around 95%. This result can be applied to RFID readers at the entrance of fashion stores as anti-theft systems.

Previous studies on false positive detection have not fully evaluated various types of tag movement that may occur in real-world scenarios. To improve the efficiency of RFID gates, it is necessary to use machine-learning models to identify tag direction while taking into account different types of tag movement. Motroni et al. [15] presented a classification method for monitoring forklifts in Industry 4.0 scenarios. The proposed method uses machine learning techniques to analyze sensor data from forklifts and accurately classify their actions, such as moving, loading, and unloading. The implementation uses a UHF-RFID Smart Gate with a single reader antenna and asymmetrical deployment, thus allowing the correct action classification with reduced infrastructure complexity and cost. The results show that the proposed method improves efficiency and safety in warehouse operations by enabling real-time monitoring and intervention when necessary. In addition, movement and direction detection for objects attached with RFID tags using machine learning was also presented in [16]. The detection of the position and orientation of moving objects was modeled as a classification problem and solved using a Light Gradient Boosting Machine. The proposed method was verified to have high accuracy in detecting the position and orientation of moving target boxes on a conveyor belt. Further experiments were conducted utilizing the robot to grip the moving object with a high success rate as the results. Alfian et al. [17] utilized XGBoost to classify the movement and direction of tagged products. Several movement types such as move in, move out, move close, static and turn back were considered in this study. The statistical features were extracted from RSS and timestamps of the single reader with two antennas. The result showed that the proposed model outperformed other machine learning models with an accuracy of up to 93.5%. Finally, Mizuno et al. [18] propose the state detection of multiple RFID tags using a single antenna placed at a certain angle. The states detected are stay still, forward, and backward movement. The purpose of the angled placement of the antenna is to obtain different values of received signal strength indication (RSSI) and phase angle according to the state of RFID tags. Those values are then evaluated using a random forest algorithm to define the state. The proposed method was evaluated through the experiment with 70 RFID tags and the better performance was shown by the angled antenna compared to the normal placement (without angle).

2.2. iForest Outlier Detection

Previous studies mostly concentrate on enhancing the precision of models, rather than the significance of preparing data. Identifying inconsistencies or outliers in data by using outlier detection techniques during the pre-processing stage can lead to the creation of better classifiers, resulting in better decision-making. Removing outliers from the training data will increase classification accuracy. Isolation Forest (iForest) is an algorithm for detecting outliers or anomalies in a dataset [33]. The algorithm uses the concept of isolation to separate outliers from normal observations. iForest is efficient and easy to use and can be applied to high-dimensional datasets.

Many studies have been conducted and have demonstrated that iForest outlier detection can significantly improve classification accuracy. Heigl et al. [25] introduce PCB-iForest, a new framework for outlier detection in streaming data. Based on F1 scores and trade-off with average runtime, PCB-iForest clearly outperformed nine competitors on multi-disciplinary ODDS in about 50% of the data set and achieved comparable results in 80%. Chang et al. [26] proposed three steps for training a classification model: signal segmentation, LOF-based anomaly score, and then isolation forest modeling. Signal sensors in the diffusion process of semiconductor manufacture capture both normal and abnormal data, but it is challenging to classify the anomalies since they only exist in a particular region. According to the experiment results, the proposed method outperforms other algorithms. Hu et al. [27] proposed a preprocessing method to improve the accuracy of hourly water demand forecasting models. This method reduced the RMSE of the SVR, ANN, and GRU models by 57.5%, 27.8%, and 30.0%, respectively. The local outlier detection and correction method effectively identifies global outliers and corrects them. GRU-based models perform better than ANN and SVR-based models, with the IF-CEEMDAN-GRU model being the most accurate. The proposed method also improves the accuracy of conventional models like SVR with a lower computational load. Finally, Chen et al. [28] compared five anomaly detection algorithms and found BS-iForest to be the best performing. Its AUCs on the BreastW dataset and campus CRS dataset were 0.9947 and 0.989, respectively, higher than the traditional forest method. The accuracy rates were also higher, at 0.9653 and 0.9896, respectively. BS-iForest screens sub-sampling sets before training and uses sets that are more likely to have outliers. However, compared to the standard iForest algorithm, iForest has a higher AUC index regardless of the proportion of outliers.

2.3. SMOTE

Dealing with imbalanced datasets, where one class is significantly more represented than the other, is a challenging task in supervised learning as traditional classification algorithms are designed to work with balanced class distributions. One solution is oversampling, which artificially increases the number of samples in the minority class to balance the class distribution. One popular oversampling technique is SMOTE, which generates new synthetic samples of the minority class by interpolating the feature space between existing minority class samples and their closest neighbors [29]. This allows for a more balanced dataset and can improve the performance of machine learning models.

There have been numerous studies that have proven that using SMOTE can significantly enhance the precision of classification. Sun et al. [30] proposed a model that utilizes SMOTE to tackle the problem of imbalanced datasets, to be used as a tool by banks to evaluate enterprise credit. When tested using financial data from 552 Chinese listed companies, the proposed model outperformed traditional models. Le et al. [31] used various oversampling techniques to address imbalance issues in a financial dataset collected from Korean organizations between 2016 and 2017. The results showed that a combination of SMOTE and Edited Nearest Neighbor (SMOTE + ENN), as well as RF, achieved the highest accuracy in predicting bankruptcy. In another study, a method that combines SMOTE with SVM was proposed to enhance the prediction accuracy for identifying old banknotes [32]. The findings revealed that the proposed method could improve performance by as much as 20% compared to the standard SVM algorithm.

2.4. Random Forest

The ensemble approach, which combines different classifiers, has been proposed to improve the performance and accuracy of diabetes analysis and prediction. Random Forest is one such ensemble technique that combines the findings of individual decision trees [34]. It is made up of an ensemble of classifiers, each of which is a decision tree with nodes representing attributes. Random Forest typically works by using the bagging method to generate subsets of training data. Previous studies have shown that Random Forest has been used in many areas and has produced significant results.

Mani et al. [19] utilized machine learning models such as Gaussian Naive Bayes, Logistic Regression, K-nearest neighbor, Classification and Regression Trees (CART), and Random Forest to predict type 2 diabetes for the next six months to one year based on electronic medical records (EMR) data. The results showed that RF outperformed other models in predicting diabetes. Lopez et al. [20] used Random Forest to identify the most important attributes related to diabetes. The proposed model was compared to other machine learning models such as Support Vector Machines and Logistic Regression (LR) and produced significant results. RF outperformed other models in terms of prediction accuracy and estimated relevance of the attributes. Alam et al. [22] used a Random Forest classifier for disease classification and tested it on 10 medical benchmark datasets. A feature ranking-based approach was developed and implemented for medical data classification and the proposed RF model produced promising results. Finally, Sun et al. [21] used a random forest (RF) classifier to model changes in hypertension control in 1294 patients at Vanderbilt University Medical Center. They found that the RF model was able to accurately predict changes in hypertension control status. This research could be used to develop personalized hypertension management plans.

Previous studies have shown that using a random forest (RF) model can be applied to prediction models in many areas. However, these studies have also revealed that the presence of outliers and imbalanced data in the training set can negatively impact the performance of the RF model. By removing outliers and balancing the training set, the RF model is expected to have improved classification accuracy.

3. Methodology

The proposed model in Figure 1 uses RSS and timestamps to predict tag movement and direction. Data pre-processing was done to remove inconsistent data and missing values were replaced with the mean. Outlier detection based on iForest has been applied to the RFID readings dataset so that the outliers can be eliminated. Statistical features were extracted from RSS and timestamps and the SMOTE technique was applied to generate new instances of the minority class. The RF algorithm was used for prediction and performance was evaluated by comparing the proposed model to other machine-learning models. The trained model was then integrated into a web-based application for easy use by end users. The evaluation of the model was done using stratified 10-fold cross-validation, a variation of k-fold cross-validation where each subset contains the same proportion of class labels as the original dataset.

3.1. Dataset

We proposed a prediction model based on RF to improve the prediction performance for tag movement and direction. We used a dataset from a previous study [17,35] which consists of five (5) types of tag movements, they are moving in, moving out, moving close, static, and turning back. There are 180 unique data generated for each move in, move out, move close and turn back, while for static tag 310 unique data were collected. In total, our study uses 1030 unique RFID movement data.

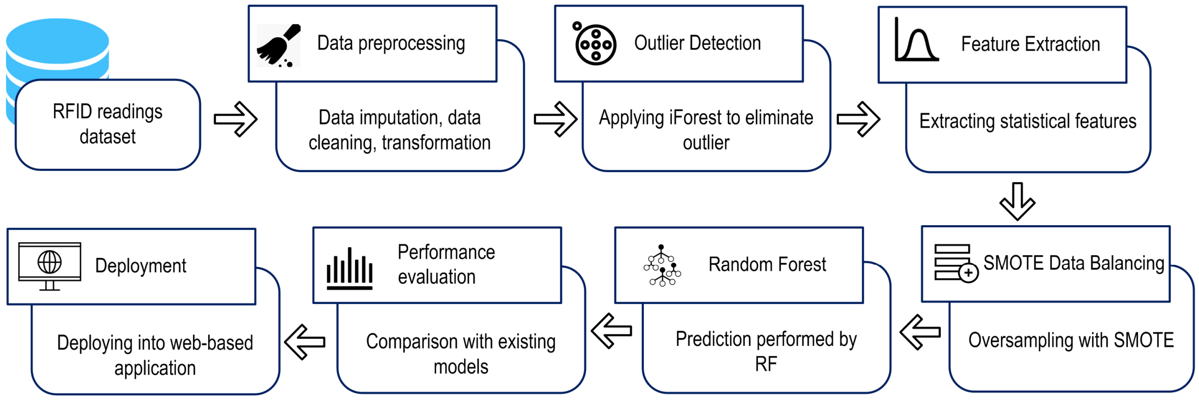

RFID readings dataset
Data preprocessing
Data imputation, data cleaning, transformation
Outlier Detection
Applying iForest to eliminate outlier
Feature Extraction
Extracting statistical features
SMOTE Data Balancing
Oversampling with SMOTE
Random Forest
Prediction performed by RF
Performance evaluation
Comparison with existing models
Deployment
Deploying into web-based application

Figure 1. Proposed ML model to detect movement and direction of tagged products.

The dataset was collected using the RFID reader ALR-9900+ from Alien Technology and two linear antennas, ALR-9610-AL with 5.90 dbi Gain. The reader operates at a frequency of 902–928 MHz and supports EPC Class1 Gen2 (18000-6C). Passive RFID tags were attached to one side of each product box, using the model 9662 with a frequency range of 860–960 MHz, IC type Alien H3 and EPC Class1 Gen2 (ISO 18000-6C) protocol.

The dataset in [17,35] was gathered by considering various possible movements and directions of tags that can happen in real warehouse environments such as moving in and out through the gate, moving near the gate, turning back, and static tags (Figure 2). The setup used for this scenario was a single reader with two antennas, one installed outside and the other inside the warehouse gate. This allows for the recording of tagged products moving in and out of the warehouse. However, the RFID reader may also pick up false positive readings like when the product is moved near the gate, during a turn-back movement, or from static tags. These static tags occur when tags are located within the nominal read range or when the range is extended accidentally by metal objects within the field. The prediction model in this study is utilized to record the products that move in or out through the gate and filter out false positive readings. Figure 3 illustrates an example of tagged products moving in through the gate in a real warehouse.

3.2. iForest Outlier Detection

iForest works by constructing an ensemble of isolation trees (*iTrees*) for each dataset, where outliers are defined as instances with short average lengths in the *iTrees* [33]. The *iTrees* are recursively built by dividing the dataset until all instances are isolated or a specific tree height is reached.

Algorithm 1 provides pseudocode for *iForest*. Given *D* (input data), *MaxSample* (subsample size), and *NumTree* (number of trees to build), the algorithm generates many *iTrees* and returns a *Forest*. The *iTree* is generated by using sample data from *D*.

Algorithm 1 Isolation forest -

```
Input      $D$, NumTree, MaxSample
Output     Set of iTrees
Initialize Forest
set height limit h = ceiling($\log_2(MaxSample)$)
for i = 1 to NumTree do
        $D' \longleftarrow$ sample($D$, MaxSample)
        Forest $\longleftarrow$ Forest $\cup$ iTree ($D'$, 0, h)
end for
return Forest
```

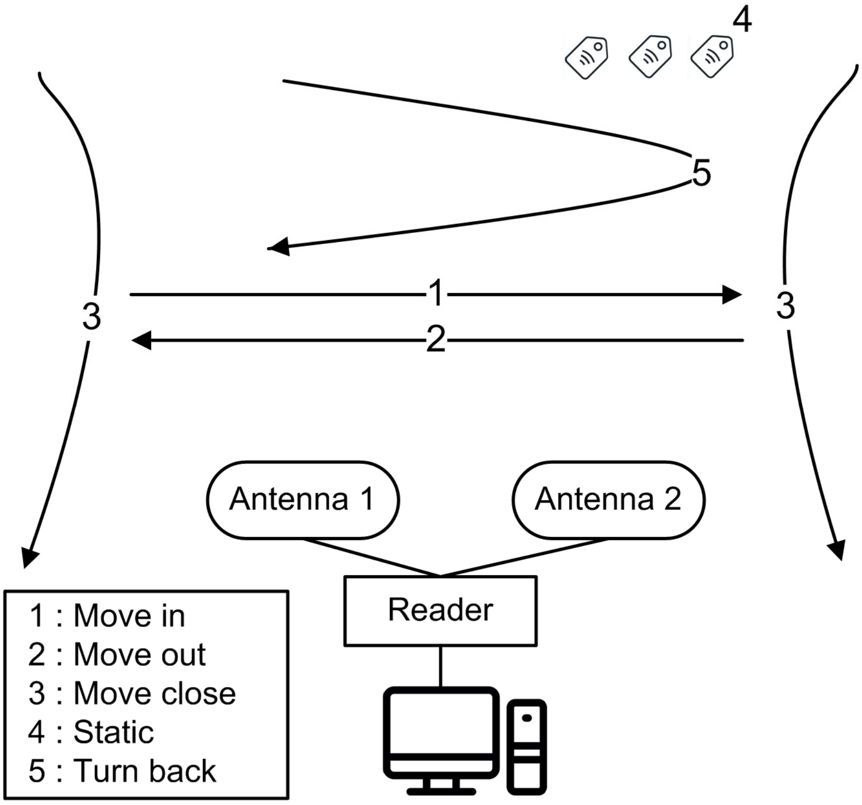


Figure 2. Possible tag movement types.

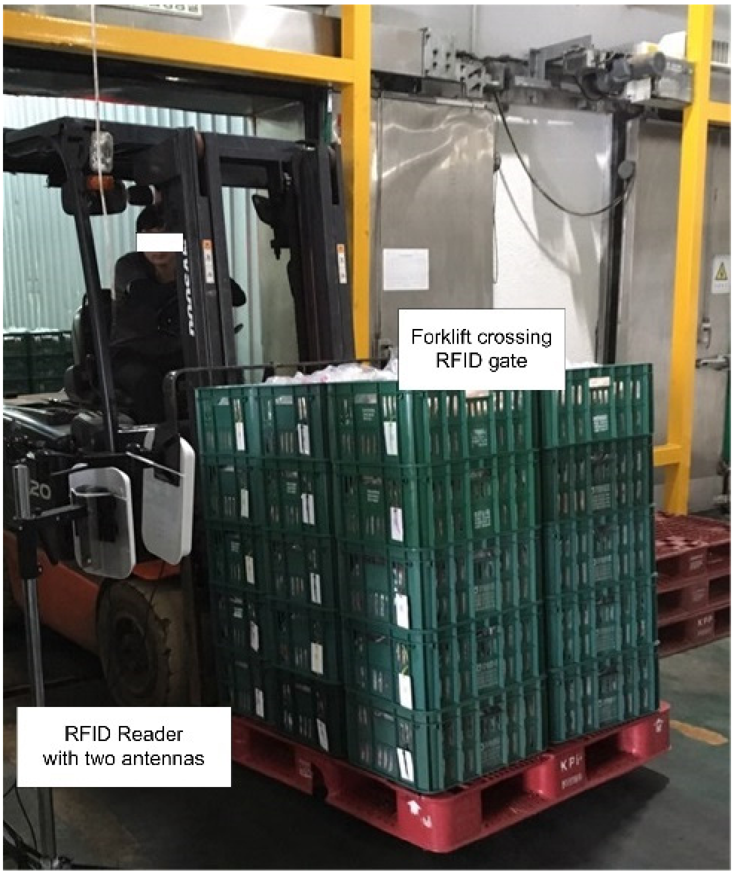


Figure 3. Application of RFID to track the products.

During *iTrees* generation, the next process is randomly select a feature q from each *iTree* and select a random value p within the range. The data is then split into two parts, they are left branch (data points $q < p$) and right branch (data points $q \geq p$). This process is iterated until there is only one data point in the branch or the *iTree* has reached the maximum depth. This process is repeated many times for each *iTree* and finally, a Forest is produced. The last step is finding data points from each *iTree* which has a shorter path length and is labeled as an outlier. We used the Scikit-learn Python library to implement iForest. We found 120 outliers' data and eliminated them from the RFID readings dataset and retained the remaining data for further analysis.

3.3. Feature Extraction

RSS attributes, which are used to distinguish between moving and stationary tags, have been used in previous studies [10,11,13,17]. The strength of the RSS signal is determined by the distance between the antenna and the tag, with closer tags producing stronger signals. The timestamp information, which is used to determine the direction of the tags, is also an important parameter [10,13,17]. Table 1 shows our statistical features extracted from a single antenna. Since this study utilizes two antennas, 36 attributes (including RSS and Timestamps) are extracted in total from both antennas. The statistical attributes were generated from the RFID readings dataset after outlier data were removed by iForest. We used all statistical features in [17] and add other features such as *median, kurtosis, skewness*, and *count* for both RSS and timestamps.

Table 1. Attributes extracted from RSS and TimeStamp.

Feature Type	Attribute Name	Description
RSS	RSS_Min	Minimum signal strength
	RSS_Max	Maximum signal strength
	RSS_Mean	The average signal strength
	RSS_Std	Standard deviation signal strength
	RSS_Med	The median signal strength
	RSS_Diff	Difference between the highest and lowest signal strengths
	RSS_Kurt	Indication of whether the RSS distribution is heavy- or light-tailed relative to normal
	RSS_Skew	Distribution asymmetry of signal strengths
	RSS_Count	Total number of reads for the tag
Timestamp	Time_Min	Timeframe (seconds) of tag read at the first time.
	Time_Max	Timeframe (seconds) of tag read at the last time.
	Time_Mean	The average value for the timeframe (seconds)
	Time_Std	Standard deviation value for the timeframe (seconds)
	Time_Med	The median value for the timeframe (seconds)
	Time_Diff	Total period (seconds) for a tag between the first and last read time
	Time_Kurt	Indication of whether the timeframe distribution is heavy- or light-tailed relative to normal
	Time_Skew	Distribution asymmetry of timeframe
	Time_Count	Total number of seconds for the tag

Finally, the extracted features from the dataset were used as input X, while the label of tag movement was used as output y. Machine-learning algorithms are utilized to learn from this pair of X and y in the training set, thus generating predictions. In our study, we use RF to detect the direction of RFID tags.

3.4. SMOTE

The SMOTE (Synthetic Minority Over-sampling Technique) is a method used to increase the number of minority class instances in a dataset by generating new synthetic instances of the minority class [29]. This is done by randomly selecting the nearest neighbors of each minority class sample and creating new synthetic samples between them (see Figure 4). In this study, SMOTE was applied to our training set during Stratified 10-Fold CV in order to balance the training set, so that could improve prediction accuracy.

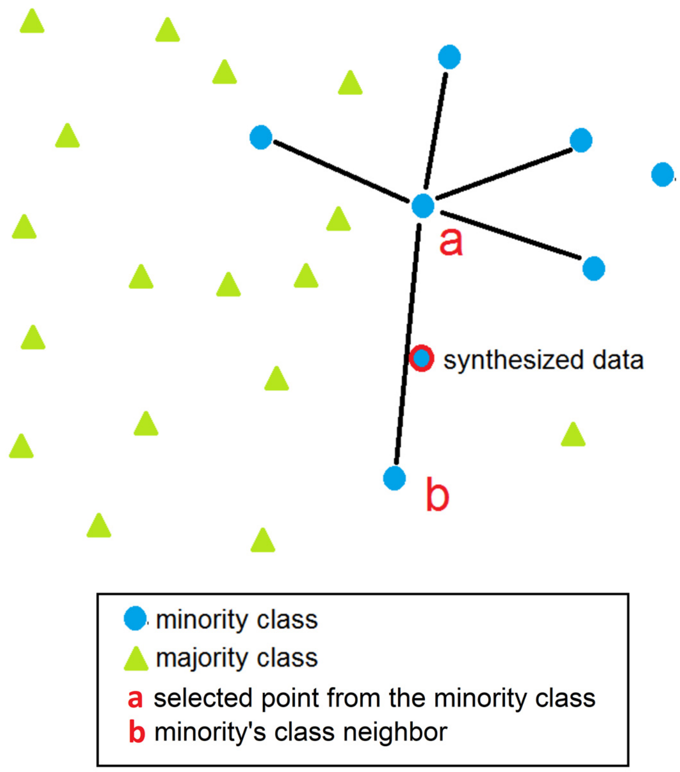

a
synthesized data
b
minority class
majority class
a selected point from the minority class
b minority's class neighbor

Figure 4. Synthesized data generated by SMOTE.

3.5. Random Forest

The RF (random forest) algorithm is a classification method that combines decision trees [34]. Previous research has shown that using a randomization approach, such as bagging or the random space method, can improve the performance of RF. This randomization is achieved by using bootstrapped sampling of the original data and randomly selecting a subset of features at each node to determine the best split. The process of generating each tree in an RF model is described in Algorithm 2.

Algorithm 2 Random forest

```
Input     : training dataset $D$, ensemble size $T$, subspace dimension $d$
Output    : majority votes from tree models
for $t$ = 1 to $T$ do
    Build a bootstrap sample $D_t$ from $D$
    Select $d$ features randomly and reduce the dimensionality of $D_t$ accordingly
    Train a tree model $M_t$ on $D_t$
    Split on the best feature in $d$
    Let $M_t$ grow without pruning
end
```

In the random forest method, individual decision trees are generated by randomly selecting a subset of features at each node to determine the split. The algorithm works as follows: given a training dataset (D), the number of trees (T) in the model, a subspace dimension (d), and the available features (F), then a bootstrapped sample (D_t) of the original dataset (D) is taken. This sample includes some records from the original dataset multiple times and excludes others. Then, a subset (d) of features is randomly selected from the bootstrapped sample (D_t) to use as candidates for the split at each node. The decision tree classifier is trained on the bootstrapped sample (D_t) and the selected features (d) and grown to their maximum size without pruning. This process is repeated for all the trees in the forest. During the classification phase, each tree votes, and the most popular class is chosen as the prediction result.

Random forests (RFs) can address several issues that affect decision trees, such as avoiding overfitting and generating low variance. In this study, data from RFID tags that were identified by iForest as outliers were removed. The SMOTE method was then used to balance the training set. Finally, a random forest model was trained to learn from five classes (moving in and out through the gate, moving near the gate, turning back, and static tags) on the prepared training set and its prediction results were compared to the testing set to evaluate the model's accuracy. In this study, the input features are the statistical information from RSS and timestamps of RFID tags and the output is the movement types. The random forest model has 100 trees (T), and 36 features (F), and uses the Gini index for splitting the reduced number of features (d).

Machine learning models were used to classify the different types and directions of movement for tagged products. The classification models were implemented in Python, XGBoost, and Scikit-learn and used default parameters provided by Scikit-learn [36]. The models were evaluated using stratified 10-fold cross-validation and their performance was measured using true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates [37]. TP and TN represent correctly classified instances, while FP and FN represent incorrectly classified instances (see Table 2).

Table 2. Measures for multi-class classification. tp_i, fp_i, fn_i, and tn_i are true positive, false positive, false negative, and true negative for class C_i, respectively. M indices represent macro-averaging.

Metric	Formula
$Average\ accuracy$	$\frac{\sum_{i=1}^{l} \frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$
$Precision_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fp_i}}{l}$
$Specificity_M$	$\frac{\sum_{i=1}^{l} \frac{tn_i}{tn_i+fp_i}}{l}$
$Recall_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fn_i}}{l}$
$Fscore_M$	$\frac{(\beta^2+1)Precision_M Recall_M}{\beta^2 Precision_M+Recall_M}$

4. Results and Discussion

This section discusses the performance of the proposed model and how outlier detection and the oversampling impact it. We also demonstrate the usefulness of the model by applying it to a web-based monitoring system.

4.1. Performance of Machine Learning Models

We examined the performance of machine learning models and the effect of iForest outlier detection and SMOTE data balancing on the model's accuracy. We compared a model based on a random forest with iForest and SMOTE to other classification models for predicting five types of movement and direction of tags. We used several Machine Learning algorithms, including multi-layer perceptron (MLP), logistic regression (LR), K-nearest neighbor (KNN), decision tree (DT), naïve bayes (NB), eXtreme Gradient Boosting (XGBoost), and adaptive boosting (AdaBoost), as models for predicting tag movement and direction. The results in Table 3 show the performance of different models in terms of accuracy, precision, recall, and f-score, using both RSS and Timestamp features. Our findings showed that the proposed model outperformed the other models by up to 94.251%, 93.751%, 98.612%, 93.502%, and 93.332% for accuracy, precision, specificity, recall, and f-score, respectively.

Table 3. Performance evaluation results.

Method	Performance Evaluation (%)				
	Accuracy	Precision	Specificity	Recall	F-Score
MLP	71.165 ± 5.745	70.001 ± 7.665	92.923 ± 1.388	67.699 ± 6.106	65.728 ± 6.537
LR	63.301 ± 3.998	60.646 ± 4.426	90.711 ± 0.959	58.978 ± 4.241	58.402 ± 4.106
KNN	66.699 ± 3.336	64.346 ± 3.521	91.844 ± 0.827	62.355 ± 3.368	62.316 ± 2.960
DT	86.699 ± 4.343	86.477 ± 3.881	96.713 ± 1.075	85.570 ± 3.887	85.519 ± 4.084
NB	81.262 ± 3.475	80.395 ± 7.628	95.200 ± 0.914	79.022 ± 4.134	78.467 ± 5.869
XGBoost	93.592 ± 2.983	93.363 ± 3.028	98.434 ± 0.730	92.900 ± 3.207	92.726 ± 3.322
AdaBoost	93.107 ± 3.174	92.963 ± 3.161	98.312 ± 0.772	92.624 ± 3.139	92.418 ± 3.289
RF + iForest + SMOTE	94.251 ± 3.267	93.751 ± 3.547	98.612 ± 0.786	93.502 ± 3.510	93.332 ± 3.617

The proposed model was found to be highly accurate in detecting the movement and direction of tags. If tags moving in through the gate are wrongly classified as moving out, the management will think the products have been shipped to other supply chain partners. On the other hand, if tags moving out the gate are wrongly classified as moving in, the management will believe the products have been stored in cold storage. This shows that using the RF with iForest and SMOTE model will significantly improve warehouse management accuracy.

4.2. Impact of Outlier Detection and Data Balancing Method

In this study, we examined the effect of iForest outlier detection and SMOTE on the accuracy of classification models. We found that the implementation of these techniques slightly improved the accuracy of the models. After removing noisy or outlier data and balancing the training set, the average accuracy of the classification models increased by about 0.13% compared to conventional machine learning models. Figure 5 shows the detailed impact of outlier detection and oversampling on classification accuracy. In our dataset, removing outliers and increasing the distribution of minority cases generally improved classification model accuracy, except for logistic regression. In conclusion, integrating iForest-based outlier detection and SMOTE for balancing the dataset into classification models can improve model accuracy.

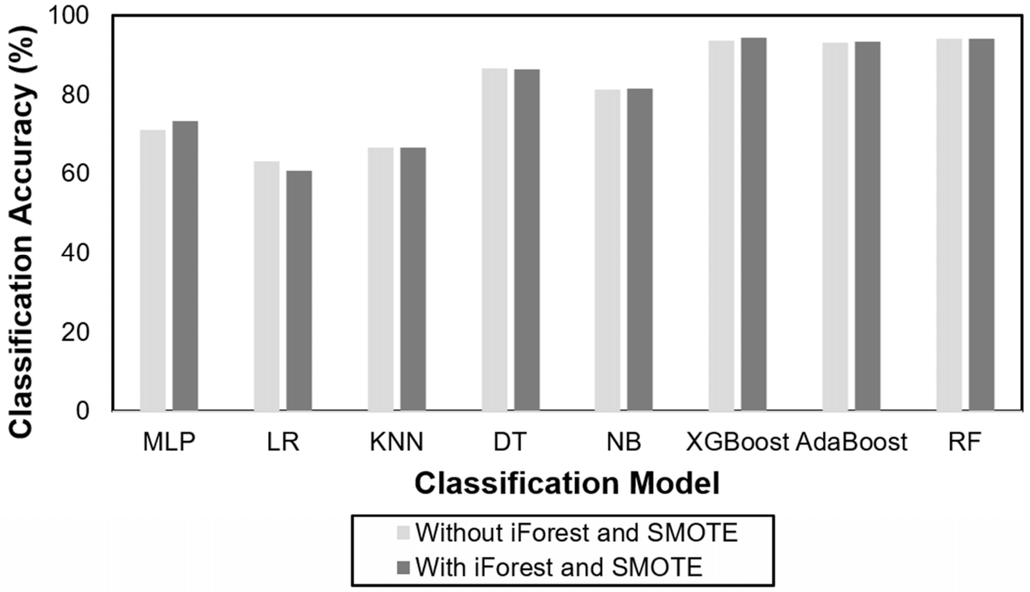


Figure 5. Impact of iForest and SMOTE on model prediction accuracy.

4.3. Comparison with Previous Studies

In this study, we compared our study with previous studies related to the prediction model for detecting the movement and direction of RFID tags. Table 4 presents a comparison of the findings between our study and earlier research.

Table 4. Comparison of our study with previous work.

Author	Purpose	Architecture	Feature	Method	Accuracy (%)	Practical Application
[10]	Detecting movement	One reader 4 antennas	Statistical features from RSS and timestamps	Information Gain	95.69	Not reported
[11]	Detecting movement	Two readers	Statistical Features from RSS and phase readings	SVM	95.3	Not reported
[13]	Detecting movement	One reader, one antenna	Statistical Features from RSS	RF with IQR outlier detection	97.496	Yes
[17]	Detecting movement and direction	One reader, two antennas	Statistical Features from RSS and Timestamp	XGBoost	93.5	Yes
Our study	Detecting movement and direction	One reader, two antennas	Statistical Features from RSS and Timestamp	RF with iForest Outlier Detection and SMOTE	94.251	Yes

Keller et al. [10] applied Information Gain to find possible split points for moved and static tags. By utilizing RSS and timestamps as features, the proposed model could achieve an accuracy of up to 95.69%. Ma et al. [11] utilized two readers to detect the movement of tags. The SVM model was utilized as a classifier to detect the moving tags. Alfian et al. [13] proposed false positive detection based on RF and Inter Quartile Range (IQR) outlier detection. The integrated model successfully detects moving tags up to 97.496.

In addition, a recent study considered not only tag movement but also tag direction [17]. The statistical features were extracted from the RSS and timestamps, while XGBoost was utilized as a classifier. Our study utilized a dataset from [17] and showed improvement in prediction accuracy. We proposed a prediction model based on RF with iForest outlier detection and SMOTE data balancing. Furthermore, our study offered practical application by developing a web-based monitoring system. By embedding a trained prediction model into a web-based system, the direction of the tagged product can be presented easily to management.

Table 4 should not be considered as strong evidence regarding model performance, but it provides a general comparison and allows discussions regarding the proposed model and previous approaches. We used dataset from [17] for the current study, which considered more complex scenarios (tag movement and direction) as compared to dataset from [10,11,13], where they considered tag movement only. Dataset used in [10,11,13] considered two classes (binary), they are moving and static tag, while in [17] and our study considered five classes (multiclass) such as going inside and out of the gate, moving close to the gate, turning around, and static tags. Multi-class classification is more complex than binary classification, as the result our prediction model generated lower accuracy than the models for binary classification in [10,11,13]. However, for same dataset our model generated higher classification accuracy compared to previous study in [17].

In addition, the number of reader and antennas used in experiment could influence the model performance as presented in [38]. However, in our study to detect the direction of the tags at least required single reader with two antennas. The first antenna could be installed outside and the other inside the warehouse gate. The purpose of our study was to try to minimize amount of hardware without reducing the system performance.

4.4. Practical Application

The goal of this study is to create a web-based system that uses a machine learning model to accurately track tagged products and assist management in decision-making.

Previous research has shown that this type of system can be useful in web-based traceability [13,17] and disease prediction [39,40]. By employing a prediction model based on machine learning, RFID tags crossing the gate and static or other types of tags movement can be detected accurately.

The web-based monitoring system was built using the PHP programming language and a MySQL database on the server side, Java for the capturing application, and Python for the tag movement and direction module on the client side (Client PC). The prediction model was implemented using the Flask web framework and Scikit-learn library on the client side and was used to filter out noise (false positive RFID tags) and detect the direction of the RFID tags as they moved through the gate. Figure 6 illustrates how the tag products moved by forklift through the RFID gate and the sensor data was sent to the Client PC where the prediction model was employed. The system received sensor data from an RFID reader and used the last six seconds of data as input to extract statistical features, which were then sent to an API based on Flask. The trained model was used to predict the direction of the tagged product, whether it was moving in or moving out, and the result was presented to management through a web-based interface. Figure 7 showed the interface displayed a history of the tagged products, allowing management to monitor their location in the warehouse in real time.

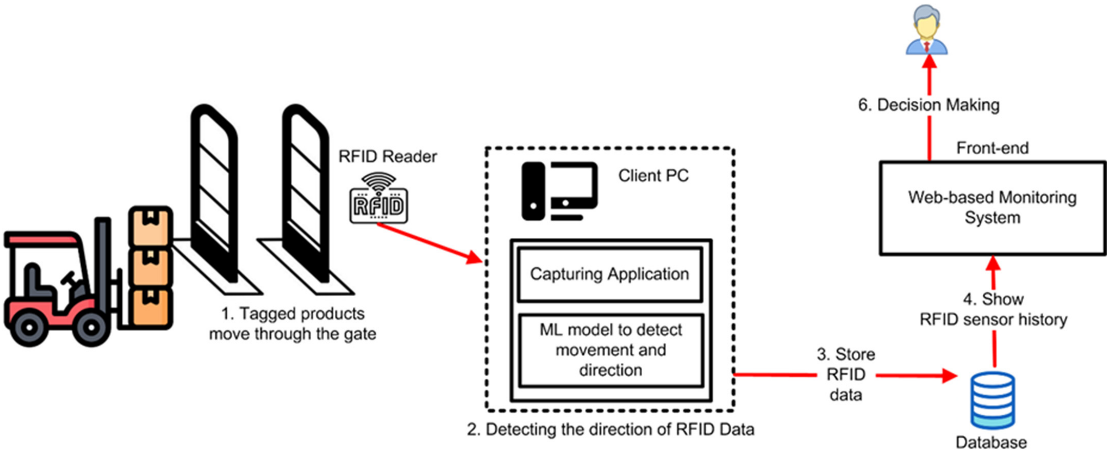
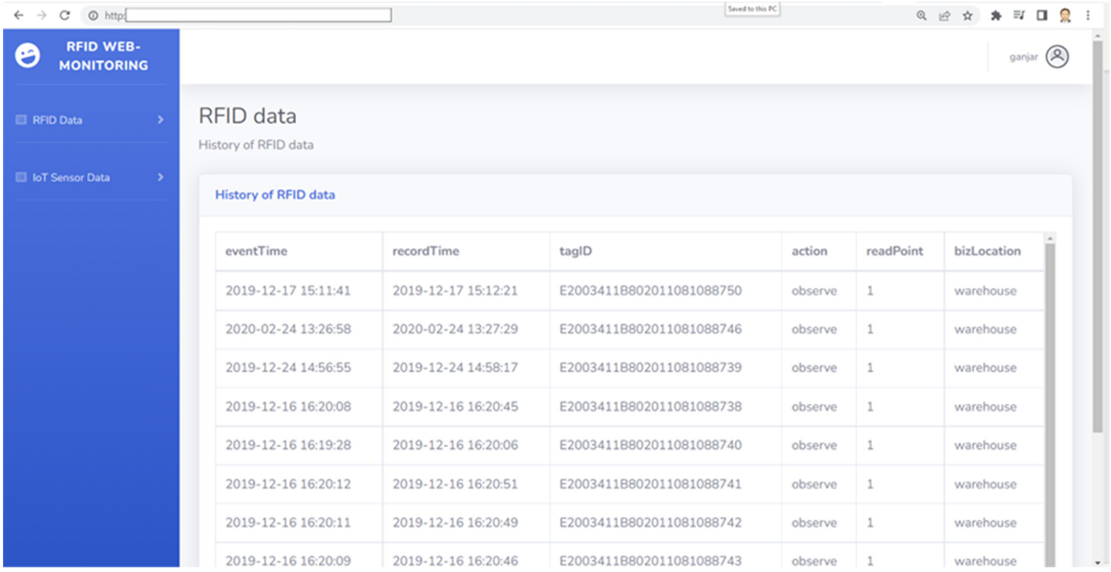

6. Decision Making
Front-end
RFID Reader
RFID
Client PC
Web-based Monitoring System
Capturing Application
ML model to detect movement and direction
1. Tagged products move through the gate
4. Show RFID sensor history
3. Store RFID data
2. Detecting the direction of RFID Data
Database

Figure 6. Framework design of web-based monitoring system.



RFID data

History of RFID data

History of RFID data

eventTime	recordTime	tagID	action	readPoint	bizLocation
2019-12-17 15:11:41	2019-12-17 15:12:21	E2003411B802011081088750	observe	1	warehouse
2020-02-24 13:26:58	2020-02-24 13:27:29	E2003411B802011081088746	observe	1	warehouse
2019-12-24 14:56:55	2019-12-24 14:58:17	E2003411B802011081088739	observe	1	warehouse
2019-12-16 16:20:08	2019-12-16 16:20:45	E2003411B802011081088738	observe	1	warehouse
2019-12-16 16:19:28	2019-12-16 16:20:06	E2003411B802011081088740	observe	1	warehouse
2019-12-16 16:20:12	2019-12-16 16:20:51	E2003411B802011081088741	observe	1	warehouse
2019-12-16 16:20:11	2019-12-16 16:20:49	E2003411B802011081088742	observe	1	warehouse
2019-12-16 16:20:09	2019-12-16 16:20:46	E2003411B802011081088743	observe	1	warehouse

Figure 7. The history of RFID sensor data presented in a web-based system.

5. Conclusions and Future Works

The proposed RF model with iForest outlier detection and SMOTE was able to identify the movement and direction of tags using RFID technology. It was tested on various types of tag movement, including entering and exiting the gate, moving close to the gate, turning back, and remaining stationary. The results showed that the proposed model outperformed other models, such as MLP, LR, KNN, DT, NB, RF, XGBoost, and Adaboost, by up to 94.251%, 93.751%, 98.612%, 93.502%, and 93.332% in terms of accuracy, precision, specificity, recall, and f-score, respectively. The trained model can be used in an RFID gate to detect whether a product is entering or leaving and can also filter out false positive readings like static tags, tags turning back, and tags moving close to the gate. This information can be combined with product information and stored in a database.

In the future, the performance of readers, tags, and IoT sensors will be evaluated under various conditions. It is also possible that the comparison with other classification models and the use of machine learning to identify miss-reads will be presented in the near future.

Author Contributions: Conceptualization, M.S. and N.L.F.; methodology, G.A.; software, G.A.; validation, M.Q.H.O., N.D.Y. and F.T.D.A.; formal analysis, M.S.; investigation, N.L.F.; resources, D.N.P.; data curation, S.A.; writing—original draft preparation, G.A.; writing—review and editing, S.A. and M.S.; visualization, D.N.P.; supervision, F.B.; project administration, F.B.; funding acquisition, L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study is publicly available at https://github.com/ganjar87/RFID_reading_dataset (accessed on 16 January 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, S.; M. R., A.R.; Kaminski, J.; Pepin, H. Opportunities for Industry 4.0 to Support Remanufacturing. *Appl. Sci.* **2018**, *8*, 1177. [CrossRef]
2. Senthilkumar, R.; Venkatakrishnan, P.; Balaji, N. Intelligent Based Novel Embedded System Based IoT Enabled Air Pollution Monitoring System. *Microprocess. Microsyst.* **2020**, *77*, 103172. [CrossRef]
3. Effendi, S.Z.; Oktiawati, U.Y. Implementation and Performance Analysis of Temperature and Humidity Monitoring System for Server Room Conditions on Lora-Based Networks. *J. Internet Softw. Eng.* **2022**, *3*, 20–25. [CrossRef]
4. Guevara, N.E.; Bolaños, Y.H.; Diago, J.P.; Segura, J.M. Development of a Low-Cost IoT System Based on LoRaWAN for Monitoring Variables Related to Electrical Energy Consumption in Low Voltage Networks. *HardwareX* **2022**, *12*, e00330. [CrossRef]
5. Subardono, A.; Hariri, I.K. Monitoring and Analysis of Honeypot System Performance Using Simple Network Management Protocol (SNMP). *J. Internet Softw. Eng.* **2021**, *2*, 1–8. [CrossRef]
6. Rahim, M.A.; Rahman, M.A.; Rahman, M.M.; Asyhari, A.T.; Bhuiyan, M.Z.A.; Ramasamy, D. Evolution of IoT-Enabled Connectivity and Applications in Automotive Industry: A Review. *Veh. Commun.* **2021**, *27*, 100285. [CrossRef]
7. Ammar, M.; Haleem, A.; Javaid, M.; Bahl, S.; Garg, S.B.; Shamoon, A.; Garg, J. Significant Applications of Smart Materials and Internet of Things (IoT) in the Automotive Industry. *Mater. Today Proc.* **2022**, *68*, 1542–1549. [CrossRef]
8. El Zouka, H.A.; Hosni, M.M. Secure IoT Communications for Smart Healthcare Monitoring System. *Internet Things* **2021**, *13*, 100036. [CrossRef]
9. Mani, N.; Singh, A.; Nimmagadda, S.L. An IoT Guided Healthcare Monitoring System for Managing Real-Time Notifications by Fog Computing Services. *Procedia Comput. Sci.* **2020**, *167*, 850–859. [CrossRef]
10. Keller, T.; Thiesse, F.; Kungl, J.; Fleisch, E. Using Low-Level Reader Data to Detect False-Positive RFID Tag Reads. In Proceedings of the 2010 Internet of Things (IOT), Tokyo, Japan, 29 November–1 December 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–8.
11. Ma, H.; Wang, Y.; Wang, K. Automatic Detection of False Positive RFID Readings Using Machine Learning Algorithms. *Expert Syst. Appl.* **2018**, *91*, 442–451. [CrossRef]
12. Zhu, S.; Wang, S.; Zhang, F.; Zhang, Y.; Feng, Y.; Huang, W. Environmentally Adaptive Real-Time Detection of RFID False Readings in a New Practical Scenario. In Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, 8–12 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 338–345.

13. Alfian, G.; Syafrudin, M.; Yoon, B.; Rhee, J. False Positive RFID Detection Using Classification Models. *Appl. Sci.* **2019**, *9*, 1154. [CrossRef]
14. Motroni, A.; Pino, M.R.; Buffi, A.; Nepa, P. Artificial Intelligence Enhances Smart RFID Portal for Retail. In Proceedings of the 2022 IEEE International Conference on RFID (RFID), Las Vegas, NV, USA, 17 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 53–57.
15. Motroni, A.; Buffi, A.; Nepa, P.; Pesi, M.; Congi, A. An Action Classification Method for Forklift Monitoring in Industry 4.0 Scenarios. *Sensors* **2021**, *21*, 5183. [CrossRef]
16. Tang, J.; Gong, Z.; Wu, H.; Tao, B. RFID-Based Pose Estimation for Moving Objects Using Classification and Phase-Position Transformation. *IEEE Sens. J.* **2021**, *21*, 20606–20615. [CrossRef]
17. Alfian, G.; Syafrudin, M.; Farooq, U.; Ma'arif, M.R.; Syaekhoni, M.A.; Fitriyani, N.L.; Lee, J.; Rhee, J. Improving Efficiency of RFID-Based Traceability System for Perishable Food by Utilizing IoT Sensors and Machine Learning Model. *Food Control.* **2020**, *110*, 107016. [CrossRef]
18. Mizuno, K.; Miwa, Y.; Naito, K.; Ehara, M. State Estimation Scheme for Multiple RF Tags with an Angled Single Antenna. In Proceedings of the 2022 IEEE International Conference on RFID (RFID), Las Vegas, NV, USA, 17 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 64–69.
19. Mani, S.; Chen, Y.; Elasy, T.; Clayton, W.; Denny, J. Type 2 Diabetes Risk Forecasting from EMR Data Using Machine Learning. *AMIA Annu. Symp. Proc. AMIA Symp.* **2012**, *2012*, 606–615.
20. López, B.; Torrent-Fontbona, F.; Viñas, R.; Fernández-Real, J.M. Single Nucleotide Polymorphism Relevance Learning with Random Forests for Type 2 Diabetes Risk Prediction. *Artif. Intell. Med.* **2018**, *85*, 43–49. [CrossRef]
21. Sun, J.; McNaughton, C.D.; Zhang, P.; Perer, A.; Gkoulalas-Divanis, A.; Denny, J.C.; Kirby, J.; Lasko, T.; Saip, A.; Malin, B.A. Predicting Changes in Hypertension Control Using Electronic Health Records from a Chronic Disease Management Program. *J. Am. Med. Inform. Assoc.* **2014**, *21*, 337–344. [CrossRef]
22. Alam, M.Z.; Rahman, M.S.; Rahman, M.S. A Random Forest Based Predictor for Medical Data Classification Using Feature Ranking. *Inform. Med. Unlocked* **2019**, *15*, 100180. [CrossRef]
23. Salazar, L.H.A.; Leithardt, V.R.Q.; Parreira, W.D.; da Rocha Fernandes, A.M.; Barbosa, J.L.V.; Correia, S.D. Application of Machine Learning Techniques to Predict a Patient's No-Show in the Healthcare Sector. *Future Internet* **2021**, *14*, 3. [CrossRef]
24. Omasheye, O.R.; Azi, S.; Isabona, J.; Imoize, A.L.; Li, C.-T.; Lee, C.-C. Joint Random Forest and Particle Swarm Optimization for Predictive Pathloss Modeling of Wireless Signals from Cellular Networks. *Future Internet* **2022**, *14*, 373. [CrossRef]
25. Heigl, M.; Anand, K.A.; Urmann, A.; Fiala, D.; Schramm, M.; Hable, R. On the Improvement of the Isolation Forest Algorithm for Outlier Detection with Streaming Data. *Electronics* **2021**, *10*, 1534. [CrossRef]
26. Chang, K.; Yoo, Y.; Baek, J.-G. Anomaly Detection Using Signal Segmentation and One-Class Classification in Diffusion Process of Semiconductor Manufacturing. *Sensors* **2021**, *21*, 3880. [CrossRef]
27. Hu, S.; Gao, J.; Zhong, D.; Deng, L.; Ou, C.; Xin, P. An Innovative Hourly Water Demand Forecasting Preprocessing Framework with Local Outlier Correction and Adaptive Decomposition Techniques. *Water* **2021**, *13*, 582. [CrossRef]
28. Chen, J.; Zhang, J.; Qian, R.; Yuan, J.; Ren, Y. An Anomaly Detection Method for Wireless Sensor Networks Based on the Improved Isolation Forest. *Appl. Sci.* **2023**, *13*, 702. [CrossRef]
29. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-Sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
30. Sun, J.; Lang, J.; Fujita, H.; Li, H. Imbalanced Enterprise Credit Evaluation with DTE-SBD: Decision Tree Ensemble Based on SMOTE and Bagging with Differentiated Sampling Rates. *Inf. Sci.* **2018**, *425*, 76–91. [CrossRef]
31. Le, T.; Lee, M.; Park, J.; Baik, S. Oversampling Techniques for Bankruptcy Prediction: Novel Features from a Transaction Dataset. *Symmetry* **2018**, *10*, 79. [CrossRef]
32. Jin, O.; Qu, L.; He, J.; Li, X. Recognition of New and Old Banknotes Based on SMOTE and SVM. In Proceedings of the 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, China, 10–14 August 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 213–220.
33. Liu, F.T.; Ting, K.M.; Zhou, Z.-H. Isolation Forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 413–422.
34. Breiman, L. Random Forest. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
35. Alfian, G. RFID Reading Dataset. Available online: https://github.com/ganjar87/RFID_reading_dataset (accessed on 18 February 2023).
36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Müller, A.; Nothman, J.; Louppe, G.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2012**, *12*, 2825–2830. [CrossRef]
37. Sokolova, M.; Lapalme, G. A Systematic Analysis of Performance Measures for Classification Tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
38. Keller, T.; Thiesse, F.; Ilic, A.; Fleisch, E. Decreasing False-Positive RFID Tag Reads by Improved Portal Antenna Setups. In Proceedings of the 2012 3rd IEEE International Conference on the Internet of Things, Wuxi, China, 24–26 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 99–106.

39. Rau, H.-H.; Hsu, C.-Y.; Lin, Y.-A.; Atique, S.; Fuad, A.; Wei, L.-M.; Hsu, M.-H. Development of a Web-Based Liver Cancer Prediction Model for Type II Diabetes Patients by Using an Artificial Neural Network. *Comput. Methods Programs Biomed.* **2016**, *125*, 58–65. [CrossRef]
40. Ahmed, N.; Ahammed, R.; Islam, M.M.; Uddin, M.A.; Akhter, A.; Talukder, M.A.-A.; Paul, B.K. Machine Learning Based Diabetes Prediction and Development of Smart Web Application. *Int. J. Cogn. Comput. Eng.* **2021**, *2*, 229–241. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.