Misplaced product detection using sensor data without planograms

Misplaced Product Detection Using Sensor Data Without Planograms

Andreas Solti[a], Manuel Raffel[a], Giovanni Romagnoli[b], Jan Mendling[a]

[a] *Vienna University of Economics and Business. Welthandelsplatz 1, 1020 Vienna, Austria*
[b] *Università di Parma. via Università 12, Parma, Emilia-Romagna, 43121,Italy*

Abstract

Accurate and timely provisioning of products to the customers is essential in retail environments to avoid missed sales opportunities. One cause for missed sales is that products are misplaced in the store. This can be addressed by fast and accurately detecting those misplacements. A problem of current detection methods for misplaced products is their reliance on up-to-date planogram information, which is often missing in practice. This paper investigates the effectiveness and efficiency of outlier detection methods for finding misplaced products without planograms. To that end, we conduct simulation studies with realistic parameters for different store parameters and sensor infrastructure settings. We also evaluate the detection methods in a real setting with an RFID inventory robot. The findings indicate that our proposed MiProD aggregation of individual detection methods consistently outperforms individual techniques in detecting misplaced products.

Keywords: Data analysis, Sensors, Outlier detection, Inventory management

1. Introduction

A central challenge of daily operations in brick-and-mortar retail shops is the timely and accurate provision of products to the customers. Retailers try to avoid store execution errors, such as out-of-stock and inventory record inaccuracy that jeopardize their performance due to low on-shelf availability and lost sales [13]. Ironically, there are also missed sales opportunities when products are not truly out-of-stock but have been misplaced. The term "misplaced products", also referred to as "out-of-shelf", describes the situation where a customer does not find on the shelf the product he/she wishes to purchase, even though that product exists in the store, and thus it is not out-of-stock [39]. The numbers are substantial, with misplaced products that can exceed 16 percent of out-of-stock products [43]. Thus, the goal of retailers is to detect misplaced products in a *timely and accurate* way.

One way to manage the misplaced product detection problem is to set up real-time location systems (RTLS). RTLS continuously infer product positions and movements in real time [50]. Even though different RTLS technologies are available, it is often radio frequency identification (RFID) that is used as a locating system technology [15, 45]. However, issues such as misplacements cannot be readily observed in RTLS, but they need to be extracted from raw data. For this knowledge extraction, it is often assumed that complementary information is available, such as planograms [47]. Planograms are layout plans that specify in detail where specific product types shall be placed in a retail store. Although the benefits of planograms have been demonstrated in [11, 7], planograms are hardly systematically and continuously maintained in practice. This means that misplaced product detection with sensor data is required to work even without planogram information. Currently, research into misplacement detection without planograms is missing and it is unclear whether it is feasible with the required level of accuracy.

In this paper, we address the research challenge of detecting misplaced products without planogram information. The proposed approach is called MiProD (Misplaced Products Detection). The goal of MiProD is to detect products misplacements by relying only on potentially noisy RFID sensor readings. To achieve this goal, MiProD systematically compares four different analytical methods, namely (i) distance [18, p. 538f]; (ii) kNN – k-Nearest Neighbors [22]; (iii) LOF – Local Outlier Factor [6]; (iv) GLOSH - Global-Local Outlier Scores from Hierarchies [9]. We investigate the accuracy of each of these methods using simulation and a case study from a European fashion retailer. Our results demonstrate the feasibility of misplaced product detection without planograms, both in the simulated environment and in the industrial case. Also, results from the case study suggest that MiProD can achieve a suitable level of accuracy in everyday fashion and apparel retail operations.

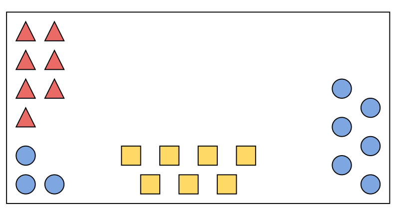
The remainder of this paper is structured as follows. Section 2 describes the background of misplaced product detection and relevant technologies. Section 3 describes our conceptual contribution of misplaced product detection without planograms. Section 4 evaluates the method based on simulation experiments and an application in a real-world fashion store. Section 5 critically discusses the implications of this work. Section 6 concludes the paper.

2. Background

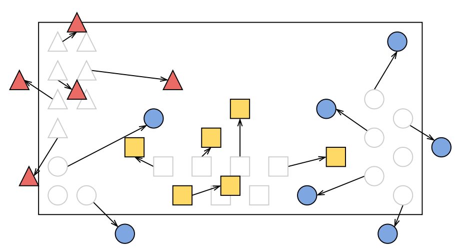
In this work, we define that *a product that is not out of stock, but misplaced in the wrong aisle or location is called misplaced product*, cf. Raman et al. [43]. The problem of misplaced product detection can be formulated as an unsupervised classification problem. Given a set of products and their sensed locations, classify each product into (i) misplaced or (ii) not misplaced.

2.1. The Problem of Misplaced Product Detection

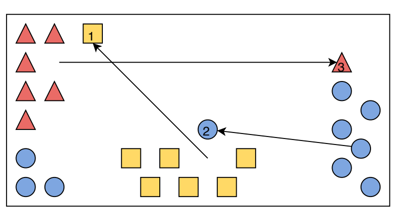
The problem can be illustrated using the planogram examples shown in Figure 1. To organize its inventory, a retailer groups its assortment according to classes. These classes are ordered such that customers can browse products by class on the sales floor. Assume that there are three product classes on the sales floor: t-shirts, pants, and shoes. In Figure 1a, they are illustrated using triangles, circles, and boxes arranged at different locations. Note that products of one class can potentially have multiple positions on the sales floor, where they are displayed (i.e., they can form multiple clusters). When using a sensor infrastructure to locate products, the real positions are not known to the system. In case of RFID-based sensing, reflections from metallic objects, occlusions of other tags, and other disturbances can cause errors in location accuracy and even missingness of reads of a tag [19]. Figure 1b illustrates that the natural grouping of product classes from the system's perspective can be blurry at the boundaries of the product groups.



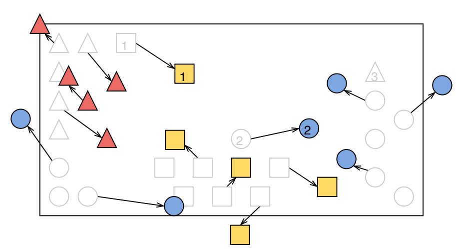
(a) Original inventory



(b) Sensed inventory



(c) Inventory with 3 misplacements



(d) Sensed inventory w. misplacements

Figure 1: Conceptual illustration of the misplaced product detection problem.

The identification of the product classes as clusters becomes more difficult when the initial order of products is disturbed, i.e., when some products are misplaced. In Figure 1c, the arrows depict (customer induced) misplacements of three products. Figure 1d shows how the RTLS might sense the inventory with misplacements. Here, the problem is to identify that products *1*, *2*, and *3* are misplaced from the sensed information in Figure 1d. We identify the following challenges: (i) missing reads (e.g., misplaced product *3*), (ii) inaccuracies in the sensing infrastructure that blur the product group boundaries, (iii) product groups can have multiple distinct areas where they are placed.

Next, we describe RFID technology as an example that enables sensing of products and look at general purpose solutions for outlier detection. These foundations serve as building blocks for our proposed detection system.

2.2. Location Sensing Technologies

RFID and RTLS belong to a rich spectrum of sensor technologies that can be used for location sensing. We refer to the survey by Farid et al. for an overview on indoor localization techniques [20]. We focus on passive RFID tags, which are the most accessible and affordable examples of location sensing technologies.

2.2.1. Radio Frequency Identification.

RFID [27, 29] is an acronym for radio frequency identification, a technology for wireless communication that allows us to unequivocally identify objects or people with an assigned tag. It has several applications. For instance, management of supply chains, access control systems or tracking of animals [21].

A system using passive RFID technology is composed of three components: (i) *tags* with a semiconductor chip and an antenna, (ii) *readers* that power the tags and read their response signals and relay that data to a (iii) server. Servers connect several RFID readers and centralize the gathered information for processing. We assume that an RFID system is used to monitor the inventory by location sensing. The readers can be fixed, hand-held, or mounted on robots.

2.2.2. Location Sensing Methods

Location sensing with RFID can be achieved by multiple means, the most common of which are *trilateration* [40], *fingerprinting* (also known as *scene analysis*) [37], or *triangulation* [36]. In case of trilateration, multiple readers with known positions receive a signal from the same tag. Typically the received signal strengths would reflect the distance to the tag from the respective readers. Thus, we can find a position on the map corresponding to the signal strengths. In case of fingerprinting, the signal strengths of the readers are memorized as fingerprints at multiple known positions. This is a preliminary calibration step, also known as the off-line stage. In the on-line stage, when a new measurement is gathered from a tag at an unknown position, this new signal strength fingerprint will be compared to the known patterns. Finally, in case of triangulation, we need to know the angles from at least three reading points to a tag to find the best matching location. Some RFID-equipped robots can perform the latter when their sensing antennas are directional, cf. [46]. This increases the accuracy in the task and can save the costs of installing and maintaining a large array of readers. In practice, the trade-off between RTLS and inventory robots is between timeliness and accuracy of the product positions.

2.3. Outlier Detection

Outlier detection is the process by which elements that do not share the characteristics of their population are identified. There exists a large body of research dealing with the problem of outlier detection in various domains like spatial data [12, 6, 1] or wireless sensor networks [52]. Here, we limit the discussion to clustering and k-nearest neighbors approaches, because these two approaches are very popular and non-model based (i.e. they do not assume or estimate a model that explains data).

- **Clustering**: Clustering refers to dividing a set of elements into disjoint sets, which are called clusters. Elements are assigned to clusters in such a way that they are more similar to one another (intra-cluster similarity) than to other elements outside of the cluster. Clustering can be used for outlier detection, as elements that are dissimilar to the elements in the identified clusters will not be assigned to a cluster, but remain as outliers.

- **k-Nearest Neighbors**: Based on a number (k) of the nearest neighbors, we can classify elements as being outliers or not. In the case that the k-nearest neighbors of an element in a similarity space have another class than the element itself, it can be considered an outlier. If we only have a single type of elements, we can look at the average distance to the nearest neighbors and compare that with the distribution of the remaining elements' average nearest neighbor distance.

Outlier detection bears the potential to identify misplaced products when they are too far away from products of the same group. We will pursue this idea to investigate if misplaced products can be detected without planograms.

2.4. Prior research on misplaced product detection

The topic of misplaced products belongs to the problem areas of out-of-stock situations in retail [16], but on a more general view also applies to other domains, where it is important that certain products are ordered for easier localization, e.g., warehouse management [41]. Prior research on the topic relates to misplaced products, out-of-stock detection, and misplaced product detection.

Managerial considerations of misplaced products. The effects of misplaced products have been captured in mathematical models that show the trade-offs of adopting RFID sensor systems to avoid misplaced products and other inventory inaccuracies. Examples of such research are [31, 44, 3, 8]. These papers do not focus on actually detecting single misplaced products, but rather investigate the relationships between aggregates like inventory count frequency, profit, RFID tag costs, and others. Kang and Gershwin investigate how small stock loss impacts the replenishment process [31]. Rekik et al. [44] analyze three scenarios: (i) where the retailer is unaware of inventory errors, (ii) where a retailer is aware of inventory errors and optimizes operations to take that into account, and (iii) where the retailer knows through RFID-based systems about the errors and is able to eliminate these. Atali et al. [3] specifically separate the sources of inventory inaccuracies in their model into misplacements, shrinkage, and transaction errors. Camdereli and Swaminathan investigate economic considerations of the players involved in RFID adoption to remove inefficiencies by misplaced inventory [8]. In these works the simplifying assumption is mostly that inventory inaccuracies can be avoided with the introduction of RFID, while in reality the RFID technology itself is prone to inaccuracies that impact the replenishment process [49].

Out-of-stock detection. Products being out of stock is a pressing problem causing missed sales opportunities [26]. Several researchers have addressed detection of out-of-stock situations [39, 35, 38]. Papakiriakopoulos et al. [39] rely on a heuristic rule based approach to detect missing products, as at that time they judged RFID to be not yet operational for this purpose. Li et al. [35] focus on improving the detection rate of RFID systems on a technical sensor level to identify missing products and distinguish those from products that are there, but are hidden to the system through the problem of tag collisions during read. Papakiriakopoulos and Georgios [38] investigate classification accuracies of machine learning models trained with data collected from RFID systems to detect out of shelf situations. Out of stock situations are inferred from only limited RFID-enabled interaction points such as the replenishment gate or the point of sales. In contrast, the problem that we consider assumes that a location sensing

Table 1: Overview of prior research on misplaced product detection

Category:	**Focus:**	**decision making** [44, 3, 8]	**operational** (OOS) [39, 35, 38]	**operational** (mispl. prod. detection) [7]	[11]	[47]	This work
decision level	managerial	✓					
	operational		✓	✓	✓	✓	✓
	store-level	✓	✓				
	product-level			✓	✓	✓	✓
infor- mation level	No location	✓	✓				
	RFID location			✓	✓		✓
	visual					✓	
independence of planograms							✓

system is in place and we are interested in finding *concrete* products that are misplaced.

Misplaced product detection.. Bu et al. describe a protocol to locate misplaced products [7]. They make the assumption that all predefined positions of all products are known. Chaves et al. [11] use a heuristic separation of products to classify them as belonging to either RFID antennas in smart shelves. Based on information from planograms for shelves, they can identify products that are located in different shelves. Then, their algorithm uses the distributions of right and wrong reads to decide whether a product identified by several antennas complies with the planogram. More recently, Saran et al. investigated how planogram compliance can be ensured with visual analytics [47] by analyzing pictures of shelves. Planogram-based approaches differ from our approach, as they assume that planograms describing the planned assortment of goods exist, while we compare and present generally applicable methods.

Table 1 summarizes the related approaches and highlights the positioning of our work. The key contribution of this paper is an accurate misplaced product detection technique, that is able to work without planograms. In this way, it is able to cope with the noisy nature of RFID sensor streams, product categories that are spread to multiple locations, while being agnostic of planogram information.

2.5. Requirements for Misplaced Product Detection

Based on the state of the art and the noisy nature of sensor systems, we formulate the following requirements for the misplaced product detection problem.

R1 Accuracy. The misplaced product detector should be able to yield a robust and highly accurate classifier in noisy sensor environments.

R2 No planogram. Due to ever changing layouts and seasonal assortment rearrangements, planograms are hardly kept up to date. The misplaced product detection should be able to detect misplaced products without detailed plans of where each product should be.

R1 is motivated by the fact that sensor based systems (e.g. RFID) are subject to inaccuracies. That is, we cannot simply compare the sensed location of a product with a fixed boundary of an area, where that product should be. Even when the product is orderly in its designated position, the sensing infrastructure could still detect it outside that area and falsely classify it as misplaced. R2 means that we need to rely on methods that do not take into account location plans of products and are able to work only with the sensor data itself. In the following, we discuss how we address these requirements.

3. MiProD: Misplaced Product Detection

In this section, we formalize the problem of misplaced product detection and present a general system architecture to deploy different algorithms in the context of misplaced product detection.

3.1. Problem statement

Let $P = \{p_1, \ldots, p_n\}$ be a set of n products in store. Further, let $C = \{c_1, \ldots, c_l\}$ be a set of l product classes, and $l \leq n$. Each product is assigned to its class through the function $\gamma : P \to C$. For example, let c_1 denote the class of *pants*. Then, $\gamma(p_1) = c_1$ means that the product p_1 is of class pants. The set of products P is partitioned into the disjoint sets of misplaced products M and non-misplaced products $\overline{M}$ (i.e., $P = M \cup \overline{M}$, and $M \cap \overline{M} = \emptyset$). Each product has a real location in three dimensional space captured by the function $\lambda : P \to \mathbb{Q}^3$. We assume that a noisy sensing infrastructure estimates the positions of products. Therefore, each product has a *sensed location* in three dimensional space, captured by $\widetilde{\lambda} : P \to \mathbb{Q}^3 \cup \perp$. Note that the sensing infrastructure can assign the empty position $\perp$ (missing read) to a product. Especially, the inaccuracy of the sensing system affects the discrepancy between actual locations λ and sensed locations $\widetilde{\lambda}$ of products. Provided these notions, we define the static version of the misplaced product detection problem.

Problem 1 (Static Misplaced Product Detection). *Given a set of products P, their classes γ and their sensed locations $\widetilde{\lambda}$, decide which products are misplaced and return the set of estimated misplaced products M'.*

Given the static misplaced product detection problem, we can also define the dynamic version as follows.

Problem 2 (Dynamic Misplaced Product Detection). *Given a set of products P, their classes γ and their sensed locations at two consecutive sensor readings $\widetilde{\lambda}_1$, and $\widetilde{\lambda}_2$, find the misplaced products at the last sensor reading and return them in the set M'.*

In either variant of the problem, we can define the quality of the detection classifier if we know the true set of misplaced products M. Then, the quality of the misplaced product detection can be measured by the *recall* ($\frac{|M' \cap M|}{|M|}$) that captures the fraction of correctly detected misplaced products over all

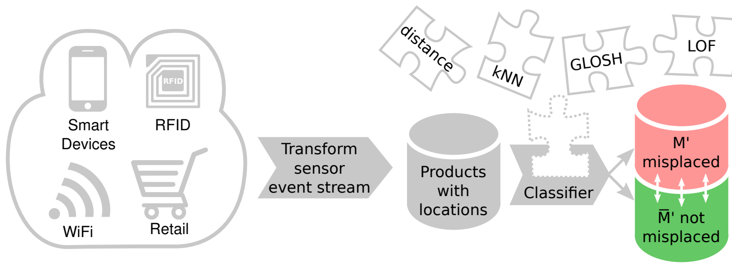


Figure 2: Overview of the approach. Sensor events are turned into products and their locations. Misplaced product detection is applied and a ranked list of products is generated. The discrimination threshold separates products into misplaced and not misplaced.

misplaced products and the *precision* ($\frac{|M' \cap M|}{|M'|}$) that captures the fraction of correctly detected misplaced products over the result set [4].

Besides these basic quality measures, we can analyze the trade-off incurred by adjusting the discrimination threshold of the classifier. That is, we analyze the effect of increasing the number of returned products according to the ordering by a classifier. The receiver operating characteristic (ROC) curve plots the true positive rate against the false positive rate at various threshold settings [28]. In general, the area under the curve (AUC) is a good overall measure to compare the accuracy of classifiers. The trade-off between precision and recall is also of interest. That means, we can plot how precise the result is over varying degrees of recall. We report these curves and graphs because outlier detection methods are threshold-based, and ROC curves and precision/recall graphs show how management decisions can achieve an adequate balance between effectiveness (recall) and efficiency (precision). We use these metrics to discuss performance of misplaced product detectors.

3.2. Misplaced product detection method

Figure 2 shows the overview of the approach. We assume that a sensor event stream exists and captures RFID data (or other sensor data) from products in a store. Then, we define aggregate and convert raw sensor reads of tagged products to locations of products, cf. [25]. Based on the product information and their estimated locations, we use classifiers to separate misplaced products M' from non misplaced products $\overline{M}'$. We can select any classification method in this misplaced product detection architecture. Each classifier produces a ranked result set, that is based on a score $\sigma : P \to \mathbb{Q}^+$ function. The classifiers produce a ranking $\rho : P \to \mathbb{N}$, where the expected outliers (misplaced products) are on top of the ranking. We implemented the following classifiers.

distance One simplistic detection method to detect misplaced products is by considering the distance σ_{dist} between the two consecutive sensed locations λ_1 and λ_2 of a product [18, p. 538f]. Formally, this is: $\sigma_{\text{dist}}(p) = \delta\big(\lambda_2(p), \lambda_1(p)\big)$.

Here, δ represents any distance metric (e.g., the Euclidean distance between two points in Cartesian space). Considering two points (x_1, y_1) and (x_2, y_2) in a two dimensional space, yields the distance:

$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Note that if any of the two sensor readings failed for product p (i.e., $\lambda_1(p) = \perp \vee \lambda_2(p) = \perp$), the distance based method fails to produce a score for that product ($\sigma_{\text{dist}}(p) = 0$).

kNN A solution to the static misplaced product detection is the k-nearest neighbors approach. Given the k-nearest neighbors of a product p in the store as $N_k(p)$, we assign the outlier score σ_{kNN} of a product p as follows:

$$\sigma_{\text{kNN}}(p) = \frac{|\{p_i \in N_k(p) \mid \gamma(p) \neq \gamma(p_i)\}|}{k}$$

Note that σ_{kNN} assigns a value between 0 and 1 to each product, and the more neighbors of a product have a different class, the higher the outlier score σ_{kNN}. Intuitively, this measure reflects the idea that based on the class of the nearest neighbors, we can decide whether a product is in distribution or not [22].

LOF Another method is the local outlier factor (LOF), as defined by Breunig et al. [6]. The LOF depends on the density of the cluster and the distance of individual elements to the nearest clusters. We separately apply the LOF to the products of one class in isolation.

GLOSH Furthermore, we compare our results with the recently proposed outlier detection method called "Global-Local Outlier Scores from Hierarchies" (GLOSH) by Campello et al. [9]. It computes a range of density based clusterings and analyses them to derive the final outliers.

MiProD classifier The proposed approach in this work. It is an aggregate method by aggregating the ranked result lists of distance, kNN, and LOF. The aggregation is chosen to be sensitive to either perspective, that is, we use a maximum aggregation of the resulting ranks.

Given two ranking functions that order products according to their outlier scores $\rho_1 : P \to \mathbb{N}$ and $\rho_2 : P \to \mathbb{N}$, the maximum aggregation assigns the scores: $\rho_{\max}(p) = \max(\rho_1(p), \rho_2(p))$. This concept naturally translates to more than two rankings.

We provide a brief motivation and example for aggregating ranked lists with a maximum rank. Consider the example of four products $\{p_1, p_2, p_3, p_4\}$ that are ranked by two different classifiers that yield ranks ρ_1 and ρ_2 as described in Table 2. We can see that for product p_3 and p_4 the two rankings agree, but one classifier orders p_1 on top and p_2 at position three, while the other does the opposite. Further, in Table 2, we see the resulting ordering by using a maximum rank (i.e., preferring products that *any* classifier preferred), a minimum rank (i.e., penalizing products that any classifier penalized), and an average rank (i.e., mixing the two rankings equally).

Table 2: Two example rankings of four products $\{p_1, p_2, p_3, p_4\}$ and the aggregate maximum, minimum, and average rankings. Ties in the aggregate ranking are highlighted. The table is sorted with the outliers at the top. Rankings are indicated in brackets.

ρ_1	ρ_2	$\rho_{\max}$	$\rho_{\min}$	ρ_{avg}
p_1 (4)	p_2 (4)	p_1 (4)	p_3 (3)	p_3 (3)
p_3 (3)	p_3 (3)	p_2 (4)	p_1 (2)	p_1 (3)
p_2 (2)	p_1 (2)	p_3 (3)	p_2 (2)	p_2 (3)
p_4 (1)	p_4 (1)	p_4 (1)	p_4 (1)	p_4 (1)

Results in the context of aggregating outliers [33, 48] suggest that there is no general optimal way of aggregating different outlier detection methods. It depends on the application, how ranking methods should be aggregated. In our case, we want to find products that moved a distance and ended up in a neighborhood unlike their class, *or* were already misplaced in the first place. Following this line of reasoning, we selected $\rho_{\max}$. We checked multiple combination options of normalization and aggregation of outlier scores and report the results in the online appendix to this paper. For brevity, the summary of those experiments is that it can happen that the average of the scores can occasionally yield better results than using the maximum of the ranked scores. However, the maximum of the ranked scores is more robust against the selection of different outlier methods and on average outperforms all score based methods.

4. Evaluation

To evaluate the approaches, we first conceptually test the accuracy in an artificial setting. This way, we can test a high number of configurations that would be infeasible to explore in real settings. To also validate the approach in realistic conditions, we performed a misplacement experiment in a retail store.

4.1. Generating artificial data - Variables

In order to generate a sufficient amount of test data to validate our algorithm, we developed a software application to simulate arbitrary retail store setups based on various parameters. To ensure realistic parameter ranges, we analyzed the literature for corresponding information. Table 3 summarizes our findings in the literature. We found that the references more or less agree on an inaccuracy between 0 and 5 meters, while contradicting ranges for missingness were reported in different studies. Latter extremes for missingness occur in settings such as trying to read RFID tags behind water basins such that a range between 0 percent and 50 percent seems to be worthy to investigate in our setup.

Table 3: Summary of reported accuracy values in meters and missingness in percent.

Parameter	Range	Ref.
Accuracy	0.5 to 4.5 m	[34]
	0.07 to 0.91 m	[23]
	1.92 to 4.69 m	[5]
Missingness	Read rate drops to 0 after 18m / 16dB of attenuation	[42]
	Between 60 and 100%	[14]
	Up to 13%	[45]
	Between 33 and 95%	[10]

These parameters are used in the evaluation. Figure 3 gives an overview of the structure of our evaluation. Within the developed "RTLS-Simulator", three subsequent steps can be identified. First, the initial store setup (Inventory I) is generated based on the following four parameters:

- *Products [#]*: The number of individual products, which are to be placed within the store. We explore the range between 1 000 products and 50 000 products (1 000, 2 000, 5 000, 10 000, 20 000, 50 000).

- *Classes [#]*: The number of classes (i.e. groups of products in a shop) to be generated. Each product is randomly uniformly assigned to one class out of this set. Each class is assigned a random location on the floor layout with configurable overlap at the boundaries. We investigate situations from 10 classes to 200 classes with step size 50 (10, 50, 100, 150, 200).

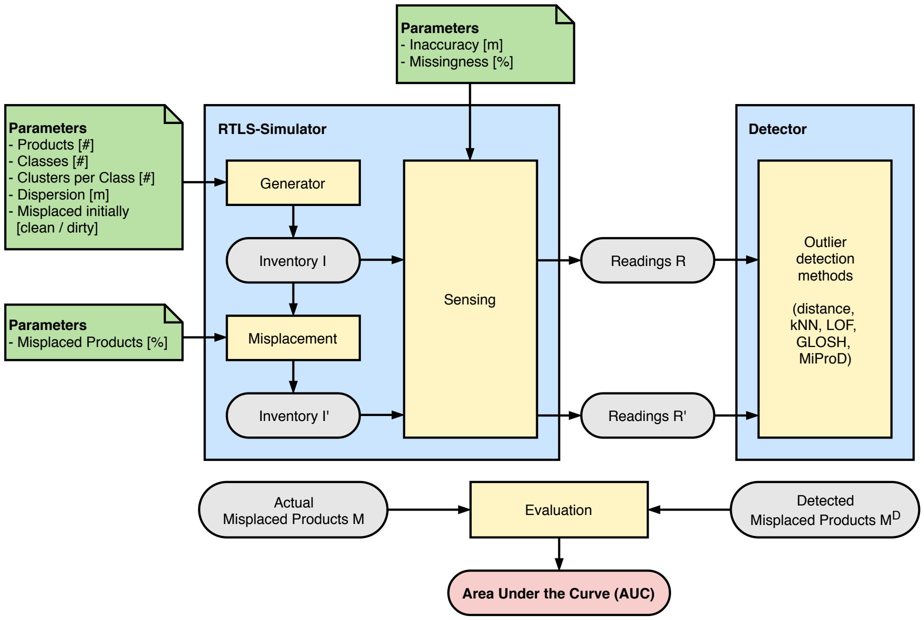


Figure 3: Structure of the performed evaluation.

- *Clusters per Class [#]*: A factor which determines the number of areas that will be generated within a shop (#Areas = #Classes · #Clusters per Class). A factor of 1.0 will generate one area for each class, while a higher factor will allocate the areas such that some classes will have two or more distinct areas on the floor. This means that even in an ordered store, there might be certain classes that are displayed at multiple positions. We vary the cluster per class between 1 and 3 (1, 1.5, 2, 2.5, 3).

- *Dispersion [m]*: The amount of spatial overlap between adjacent class areas. When we look at a 2D-projection of a shop in which shirts are placed above pants, the classes of shirts and pants overlap in their floor area. Dispersion is a means to allow for fuzzy boundaries between classes by extending the class boundaries. Dispersion ranges between 0 and 5 meters ($0m$, $1m$, $2m$, $3m$, $4m$, $5m$).

- *Misplaced initially [clean/dirty]:* If set to *dirty*, 10 percent of the products is randomly misplaced in the initial inventory. Misplacement means relocation of a product to a random point on the sales floor. This is a logical parameter and we denote it as "clean" (i.e., not misplaced initially) and "dirty" (i.e., misplaced initially).

In a second step, the generated inventory I is manipulated based on the given parameter *Misplaced Products [%]*. In general, the given percentage of all products is randomly taken and moved to arbitrary positions within the store's boundaries resulting in inventory I'. A product is *misplaced*, if it is misplaced initially in inventory I, or if it is misplaced in inventory I'. Formally, let M_1 be the products initially misplaced (note that M_1 is empty in the clean experiment) and let M_2 be the products misplaced according to the parameter *Misplaced Products [%]*. Then, the set of misplaced products M is the set union of M_1 and M_2 (i.e., $M = M_1 \cup M_2$).

The third step simulates physical sensor hardware by taking the following two parameters into account. We select the variable ranges based on the literature findings in Table 3, and on our own experiences with sensing hardware.

- *Inaccuracy [m]*: The maximum deviation between a product's actual position in the store and the detected position (in meters). We explore inaccuracy values between 0m (perfect) and 5m (low accuracy) in steps (0m, 1m, 2m, 3m, 4m, 5m).

- *Missingness [%]*: The percentage of products that are not sensed in one sensor reading due to reading collision [19], or reflection, and other reasons [17]. We cover the range between 0% and 50% in steps (0%, 5%, 10%, 20%, 30%, 40%, 50%).

I and I' go through this last step separately, potentially resulting in different reading positions for the products depending on the specified parameters.

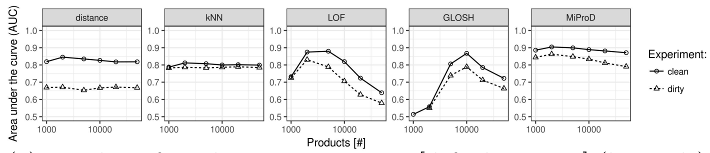
4.2. Experiment Results

Given the number of variables and their ranges, an exhaustive exploration of the parameter space yields a combinatorial explosion of settings. Therefore, we set realistic default values for all parameters except for the controlled variable that is varied in its range. This way, we can isolate the effects of individual variables on the accuracy of the misplaced product detection methods.

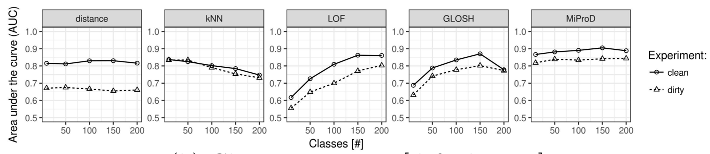
Figure 4 shows the results for the different types of scenarios (indicated in the legend of each figure). The resulting scores depict the area under the curve (AUC) of the receiver operating characteristic (ROC) curve [28].

Varying number of products.. We can see in Figure 4a that the *distance* and *kNN* approaches deliver constant results with respect to the number of products in the store. The *LOF* and *GLOSH* have a peak at averages of 5.000-10.000 products, while delivering worse results on the ends of the spectrum. This result is surprising as it indicates that these two approaches are tuned to work well at a particular density of products. The proposed rank average *MiProD* between *distance*, *kNN* and *LOF* outperforms the other approaches on average.

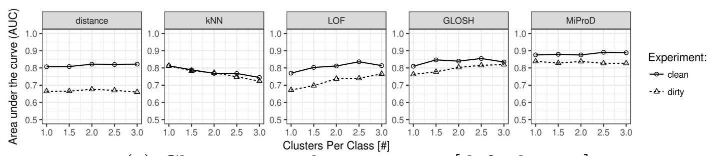
Varying number of classes.. In Figure 4b, we see that *distance* and the *MiProD* approaches are not affected by the variation of class numbers. The *kNN* suffers from an increasing number of classes, as the chances that the neighbor is of the same class gets lower with an increased number of classes. The *LOF* and *GLOSH* approaches gain from an increased variation and a relatively smaller area per class that is entailed by a growing number of classes. However, we see that at 200 classes, there is a drop in performance for *GLOSH*, as the number of products per class is reduced as well, which apparently reduces the accuracy of the hierarchical classifiers.



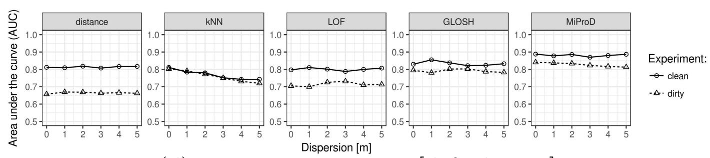
(a) Number of products 1000 - 50 000 [default: 10 000] (log-scale).



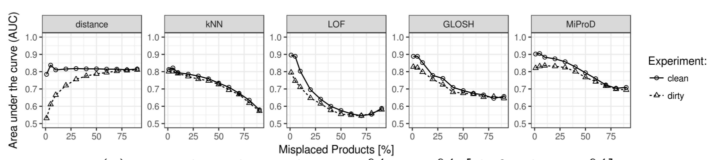
(b) Classes 10 to 200 [default: 100].



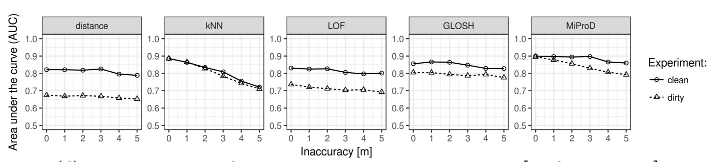
(c) Clusters per class 1.0 - 3.0 [default: 1.5].



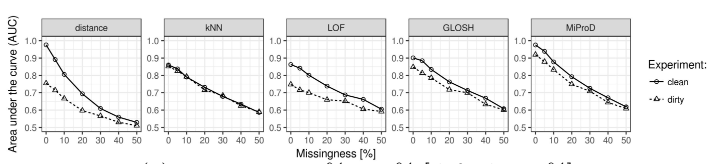
(d) Dispersion 0m - 5m [default: 1m].



(e) Misplaced products 1% - 90% [default: 10%].



(f) Inaccuracy of reading position 0m - 5m [default: 3m].



(g) Missingness 0% - 50% [default: 10%].

Figure 4: Resulting area under the curve values for two experiments: (○) clean at start, and (△) dirty with 10% initially misplaced products. All parameters are set to their default value except for the variable varied on the x-axis. Ranges are noted in the caption and the default value is noted in square brackets.

Varying clusters per class.. We can see in Figure 4c similar trends as in Figure 4b, but less pronounced. Also here, the local clusters of products become smaller, as they are increasingly spread throughout the area. In both cases (varying the classes and the clusters per class), the aggregate classifier *MiProD* yields the best overall performance.

Varying dispersion of class areas.. In Figure 4d we can see that most of the classifiers are not much affected by dispersion. However, the *kNN* based approach shows a decreasing accuracy with increasing dispersion. This is expected, as with increasing dispersion products at the boundaries can become surrounded by neighboring products of other classes. This leads to false positives.

Varying percentage of misplaced products.. Figure 4e depicts the number of misplaced products in percentage of the total number of products. The performance of all classifiers decreases rapidly, as the number of misplaced products increases. Only the *distance* based classifier shows an increasing trend for the dirty case. The distance measure helps distinguishing truly moved products from stationary inaccurate readings. If we limit our attention to values between 1% and 30%, we observe that *MiProD* outperforms the other approaches.

Varying inaccuracy of the sensor system.. We see in Figure 4f that the approaches *kNN* and indirectly also *MiProD* slightly suffer in their classification accuracy with increasing sensor inaccuracy. The trends are comparable to the dispersion experiment. The *kNN* based approach that looks at immediate neighbors of a product is most affected, as with lower accuracy, the chances increase that the sensed position of a product is within a neighborhood of a different class, which renders it a false positive. The combination of classifiers in *MiProD* outperforms the individual classifiers.

Table 4: Average AUCs for the two scenarios and the different methods.

Experiment	**distance**	**kNN**	**LOF** [6]	**GLOSH** [9]	**MiProD**
clean	0.802	0.769	0.757	0.784	0.855
dirty	0.675	0.759	0.684	0.743	0.810
both	0.739	0.764	0.720	0.763	0.832

Varying missingness of reads of the sensor system.. Last, in Figure 4g, we see that the percentage of missing reads severely impacts the classification accuracy of the compared methods. The distance based classifier suffers the most, as it depends on having two consecutive reads for each product. In a clean state with no missingness issues, however, this is the best method. When some products are already misplaced, or there is at least a 5% chance for missed sensor readings per product, the *MiProD* approach yields the best results.

Summary of the experiment.. Table 4 summarizes the experiment results and shows the competing methods' aggregate AUC values. We can see that the proposed rank-aggregate method *MiProD* yields the overall best results in finding misplaced products. In a clean state, the *distance* based approach detects the misplaced products second best, while yielding the worst results in a dirty state, where 10% of the products are initially misplaced. The *kNN* based approach is the least susceptible to a dirty state and is overall second best, directly followed by *GLOSH*, and then *distance*. The local outlier factor *LOF* yields the overall lowest scores in this experiment.

4.3. Robot-based case study in retail

We performed a controlled experiment in a retail store, where inventory counting and locating is done by a robot. The inventory robot collects the positions of the products on the sales floor by triangulation of a product's RFID tag. To this end, the robot has directed antennas that have a characteristic signal strength depending on the angle of a tag to it. Readings can be collected at multiple known positions, at which the robot is passing, and aggregated to an estimated location, e.g. the way it is described in [46]. The setting is similar to the experiments we reported above. We start with a rather clean state, as we specifically advised the store staff to diligently clean up the store before the experiment. In this clean state, we took the first sensor reading of the products and their location over night with the help of an inventory robot.

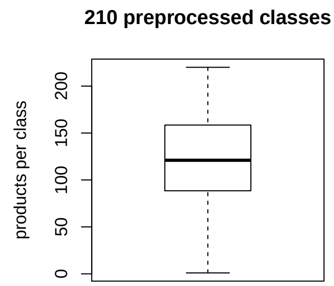
We simulated customer behavior by misplacing 55 products M (the set M is the ground truth for evaluation). Then, we took a second sensor reading of the products and their location with the help the inventory robot. At this point at least the products in M are misplaced. The two readings are input to the classification task of finding the misplaced products M.

Estimated parameters.. The number of products in this setting is 25,570. The initial state can be considered clean. The inaccuracy of the robot is low, as the median difference of the products' estimated location between the two reads is 0.4 meters. Also, the two subsequent sensed product sets' overlap is 94% (based on the first reading R and second reading R', the overlap is $\frac{|R \cap R'|}{|R \cup R'|}$). Assuming a completely random missingness process, this yields an estimated missingness of less than 3 percent per read. The number of classes as organized by the retailer is 39 with a heterogeneous distribution.

The inspection of the classes showed that the distribution areas of some classes are spread throughout the entire shop. This is expected to impact the outlier detection quality of spatial outlier detection methods like *LOF* and *GLOSH*. Furthermore, the classes are rather imbalanced in size, as depicted in Figure 5. Therefore, we applied a preprocessing step to better split the products into logical classes.

Preprocessing the product classes. We applied a hierarchical clustering [30] on all products using a distance measure that takes into account the names of

(a) Box plot of products per class before preprocessing (average products per class 656, standard deviation 796).



(b) Box plot of products per class after preprocessing (average products per class 121, standard deviation 45).

Figure 5: Box plots of products per original (a) and preprocessed (b) classes

compared products, and the distance of products to one another on the map. The hierarchical clustering produces a dendrogram tree. Each node in the tree is annotated with the number of contained products. We extracted the classes from that tree by trying to split large nodes that have a product count larger than 200, and request that splitting does not result in classes smaller than 50 products. In this clustering, we respected the initial categorization of the retailer and only partitioned the larger classes further. The resulting class count after preprocessing is 210, as shown in Figure 5b.

4.3.1. Case study results

The area under the curve for the misplacement with and without preprocessing the classes is depicted in Table 5. We see that by preprocessing the product classes, we can increase the quality of the spatial outlier detectors. The methods *LOF* and *GLOSH* significantly benefit from this step. For brevity, we only investigate the better results based on preprocessed classes. In this case, most methods yield already very high areas under the curve, with the *MiProD* rank aggregate method showing top performance at 0.998, while *GLOSH* and *LOF* closely follow with AUC values of 0.994 and 0.990, respectively. The distance based detector yields an AUC of 0.946 although it has outperformed the other

Table 5: AUCs for the different methods in the robot experiment. The two rows capture the results for 39 original classes, and for the 210 preprocessed classes.

Experiment	**distance**	**kNN**	**LOF** [6]	**GLOSH** [9]	**MiProD**
original classes	0.946	0.706	0.806	0.802	0.987
preproc. classes	0.946	0.706	0.990	0.994	0.998

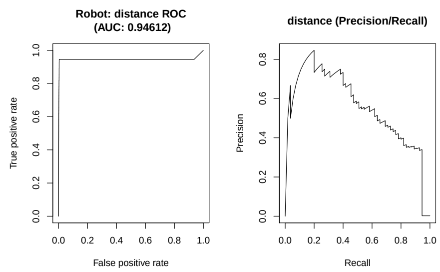
outlier classifiers in the clean setup, as we have seen in the results in Table 4.

Next, we discuss the results on a disaggregated level. Figure 6 shows the shape of the ROC curves and also the corresponding trade-off between recall and precision. Note that the count of positives (55 misplaced products) is only a small subset of the total number of products (25570 products on the sales floor). Therefore, not only recall is interesting, but precision is equally important.

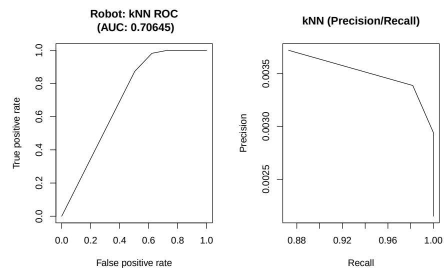
The *distance* method depicted in Figure 6a shows good precision, which is to be expected, as misplaced products tend to have moved a larger distance than the not-misplaced counterparts. We see in the precision/recall graph that a high recall (about 0.9) can be achieved with a still high precision (close to 0.4). This means that a shop employee working through the ordered list of potential outliers would find 90 percent of misplaced products with an average of 3 non-misplaced products per five products checked in this case.

The *kNN* method (Figure 6b) turns out to be unreliable in this particular case. The precision is around 0.004 at best, which means that an employee would need to check 250 products to find a single misplaced product. Latter low precision makes *kNN* alone impracticable in this setting.

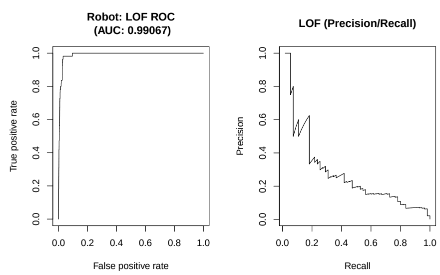
The local outlier factor (LOF) [6] shown in Fig. 6c is the only one that correctly positions the first few misplaced products on the top of its ranking (the precision/recall graph starts at 1 and stays there for a few of the 55 misplaced products). After that however, the precision rapidly decreases and a recall of 0.5 (identifying 50 percent of the misplaced products) entails browsing through five times the number of misplaced products at a precision of 0.2. The precision gets



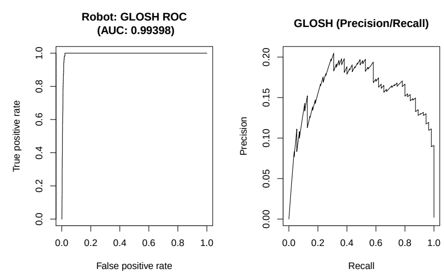
(a) Distance method.



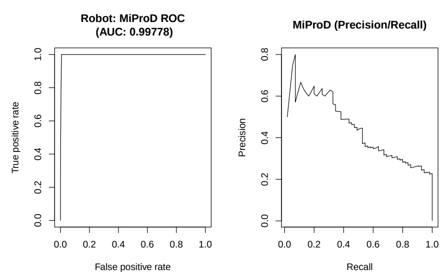
(b) kNN method.



(c) LOF method.



(d) GLOSH method.



(e) MiProD method.

Figure 6: ROC curves for the different methods with the inventory robot experiment, and their performance in terms of trade off between precision and recall (PR space).

only worse with increasing the result set, but this approach is able to rank some of the products higher, which the *distance* method did not discover. The ROC curve reaches 1 at a false positive rate of around 0.1, while with the *distance* method we need to check *almost every* product to find all outliers.

The *GLOSH* [9] method's performance in Figure 6d displays a more steady precision graph over the entire set of misplaced products. However, it fails to single out the misplaced products at the start of the list. The best precision/recall trade-off is perhaps at 0.55 recall with 0.2 precision, which is comparable to the result achieved by the *LOF* method. However, it shows more consistent results over the entire set of misplaced products and is able to locate the last misplaced products sooner than the *LOF* which yields a superior AUC.

Finally, the rank aggregate method based on the maximum ranking of distance, kNN, and LOF that we propose outperforms the individual detection techniques. The MiProD approach depicted in Figure 6e yields a remarkable AUC of 0.998, which is close to the optimal score of 1.0. We see in the precision/recall graph that while the precision suffers a little in comparison with the distance method, it is able to find *all misplaced products* with a precision of around 0.25. This means that an employee can find all 55 misplaced products by checking around 220 products.

5. Implications

Our experiment investigated in how far misplaced products can be detected using outlier detection techniques without having a planogram available. Our results demonstrate that this problem can be tackled using outlier detection techniques [2] in an accurate way. The best results were achieved using our *MiProD* aggregation technique. Our results have implications for research into sensor based locating systems [36, 32], misplaced product detection [24, 43], and for their joint application in practice [8, 11, 7].

The important implication of our work for research is that machine learning techniques can be effectively used to harness sensor systems for improved operational use cases. More specifically, this finding is important for research into misplaced products–a stream of research that up until now assumed that planogram information was required [11, 7]. In our experiments, we observed diverging strengths and weaknesses of existing techniques, which we managed to balance using our *MiProD* aggregation technique.

Some observations can be made on the applicability of the existing techniques. We found that in our simulated setting with the collected parameters, the *kNN* method [22] performed mostly better than the spatial outlier techniques *LOF* [6] and *GLOSH* [9]. However, in a more intricate store layout as observed in the real-world experiment, its performance deteriorated. This deterioration implies that products projected on a 2D plane are more heterogeneous in reality than in the generated clusters, where initially most areas are exclusively filled with products of a single class. Notwithstanding, the proposed *MiProD* rank aggregation method works well within the scope of the investigated real-world setting, as it is able to compensate the flaws of one classifier by the strengths of another. Furthermore, we saw that some spatial outlier detection methods can be improved by preprocessing the data before applying the anomaly detection methods.

In the context of misplaced product detection, we first note that the outlier detection problem can be tackled in a binary setting. From the retailer's point of view, in fact, items are either misplaced, or they are not, and it makes not much sense for store managers to assign to each item a degree or measure of being an outlier. On the contrary, it is more interesting for retailers to investigate the precision of the result set. Thus, we investigated the precision/recall graphs as well. We found that even though the *distance* method was outperformed by the spatial methods in the AUC values, its precision for the largest part of the resulting misplaced items was higher. Therefore, for a successful implementation, the distance based classifier might be preferable in the trade-off between precision and recall, when one is willing to compromise on the (hopefully) few products that are missing in either of two consecutive sensor reads.

Our work has also implications for practice. The results clearly demonstrate the potential of improving the analysis of the raw data provided by RTLS. Vendors of such systems might be better advised in fine-tuning their analytical software than investing in more powerful hardware. The results also show that accuracy (Requirement 1) can be achieved without having to rely on planogram information (Requirement 2). This aspect substantially extends the applicability of misplaced product detection using RFID sensor systems to settings in which planograms are not available or not continuously kept up to date.

Furthermore, an accurate insight into the misplaced products can yield operational benefits on the managerial level [51]. The number of product misplacements per product become visible to the decision makers of the stores and indirectly relate to customer interactions with the products. An investigation of the ratio of the number of misplacements and the number of sales per product looks promising. For example, knowing that a product is often misplaced but rarely sold would indicate a discrepancy between customer interest in a product and the willingness to buy it. This valuable knowledge can be used to optimize sales strategies and also inventory assortments.

Also some notes on potential limitations are warranted. The results cover bread ranges of plausible characteristics of retail shops and common sensor technology. Nevertheless, we need to be careful when extrapolating the results to settings in other domains with characteristics beyond the ranges that we investigated. Sensor systems significantly vary in their reading accuracy, missingness rate and other characteristics like the time interval between sensor readings.

6. Conclusion

In this work, we investigated the problem of detecting misplaced products without planogram data in order to reduce the amount of missed sales opportunities in retail stores. We investigated methods of spatial outlier detection, and also a means of misplaced product detection based on consecutive sensor readings based on the distance. In extensive experiments, we investigated the influence of different parameters in the setup of a store and sensing environment on the effect on misplaced product detection and also proposed a novel aggregation method for misplaced product detection (MiProD) that outperformed individual methods. Our results emphasize that misplaced product detection is accurately feasible in practice even if planogram information is not available.

Acknowledgments

We thank David Riobó Barba and Manuel Stochlinski for their assistance in implementing the prototypes and simulators. We thank Alexander Weinhard, and Matthias Hauser for helping with conducting the experiments on site. This work was supported by the European Union's Seventh Framework Programme (FP7/2007-2013) grant 612052 (SERAMIS).

References

[1] Elke Achtert, Ahmed Hettab, Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. 2011. Spatial outlier detection: data, algorithms, visualizations. In *International Symposium on Spatial & Temporal Databases.* Springer, 512–516.

[2] Charu C. Aggarwal. 2017. *Outlier Analysis* (2nd ed.). Springer International Publishing.

[3] Aykut Atali, Hau L Lee, and Özalp Özer. 2009. *If the inventory manager knew: Value of visibility and RFID under imperfect inventory information.* Technical Report. Graduate School of Business, Stanford University.

[4] R.A. Baeza-Yates and B.A. Ribeiro-Neto. 1999. *Modern Information Retrieval.* ACM Press / Addison-Wesley.

[5] Paramvir Bahl and Venkata N. Padmanabhan. 2000. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proceedings IEEE INFOCOM 2000, Tel Aviv, Israel, 2000.* 775–784.

[6] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD.* ACM Press.

[7] Kai Bu, Bin Xiao, Qingjun Xiao, and Shigang Chen. 2012. Efficient Misplaced-Tag Pinpointing in Large RFID Systems. *IEEE Transactions on Parallel Distribited Systems* 23, 11 (2012), 2094–2106.

[8] Almula Z Camdereli and Jayashankar M Swaminathan. 2010. Misplaced Inventory and Radio-Frequency Identification (RFID) Technology: Information and Coordination. *Production and Operations Management* 19, 1 (2010), 1–18.

[9] Ricardo J. G. B. Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. 2015. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Transactions on Knowledge Discovery from Data* 10, 1, Article 5 (2015), 51 pages.

[10] Leonardo Weiss Ferreira Chaves, Erik Buchmann, and Klemens Böhm. 2008. Tagmark: reliable estimations of RFID tags for business processes. In *Proceedings of ACM SIGKDD'08.* 999–1007.

[11] Leonardo Weiss Ferreira Chaves, Erik Buchmann, and Klemens Böhm. 2010. Finding misplaced items in retail by clustering RFID data. In *Proceedings of EDBT 2010, Lausanne, Switzerland.* 501–512.

[12] Sanjay Chawla and Pei Sun. 2006. SLOM: a new measure for local spatial outliers. *Knowledge and Information Systems* 9, 4 (2006), 412–429.

[13] Howard Hao-Chun Chuang. 2015. Mathematical modeling and Bayesian estimation for error-prone retail shelf audits. *Decision Support Systems* 80 (2015), 72–82.

[14] Robert H Clarke, Diana Twede, Jeffrey R Tazelaar, and Kenneth K Boyer. 2006. Radio frequency identification (RFID) performance: the effect of tag orientation and package contents. *Packaging Technology and Science* 19, 1 (2006), 45–54.

[15] Cosmin Condea, Frédéric Thiesse, and Elgar Fleisch. 2012. RFID-enabled shelf replenishment with backroom monitoring in retail stores. *Decision Support Systems* 52, 4 (2012), 839–849.

[16] Daniel Corsten and Thomas Gruen. 2003. Desperately seeking shelf availability: an examination of the extent, the causes, and the efforts to address retail out-of-stocks. *International Journal of Retail & Distribution Management* 31, 12 (2003), 605–617.

[17] Daniel M Dobkin and Steven M Weigand. 2005. Environmental effects on RFID tag antennas. In *IEEE Microwave Symposium*. 135–138.

[18] Richard O Duda, Peter E Hart, and David G Stork. 2012. *Pattern classification* (2nd ed.). John Wiley & Sons.

[19] Daniel W Engels and Sanjay E Sarma. 2002. The reader collision problem. In *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, Vol. 3. IEEE, 6–pp.

[20] Zahid Farid, Rosdiadee Nordin, and Mahamod Ismail. 2013. Recent advances in wireless indoor localization techniques and system. *Journal of Computer Networks and Communications* 2013 (2013).

[21] Klaus Finkenzeller. 2010. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons.

[22] Evelyn Fix and Joseph L Hodges Jr. 1951. *Discriminatory analysis-nonparametric discrimination: consistency properties*. Technical Report. California Univ Berkeley.

[23] Andre Gleser and Oldrich Ondracek. 2014. Real time locating with RFID: Comparison of different approaches. In *Proceedings Radioelektronika'14*. IEEE, 1–4.

[24] Tahi J Gnepa. 1996. An empirical investigation of deliberate shopper misplacement of products on grocery store shelves. *Journal of Food Products Marketing* 3, 3 (1996), 3–21.

[25] Hector Gonzalez, Jiawei Han, Xiaolei Li, and Diego Klabjan. 2006. Warehousing and analyzing massive RFID data sets. In *Proceedings of International Conference of Data Engineering'06*. IEEE, 83–83.

[26] Thomas W Gruen and Daniel S Corsten. 2007. *A comprehensive guide to retail out-of-stock reduction in the fast-moving consumer goods industry*. Technical Report. Grocery Manufacturers of America Washington, DC.

[27] Jeffrey Hightower, Roy Want, and Gaetano Borriello. 2000. *SpotON: An indoor 3D location sensing technology based on RF signal strength*. Technical Report. UW CSE.

[28] D.W. Hosmer and S. Lemeshow. 2000. *Applied Logistic Regression* (2nd ed.). John Wiley & Sons.

[29] V Daniel Hunt, Albert Puglia, and Mike Puglia. 2007. *RFID: a guide to radio frequency identification*. John Wiley & Sons.

[30] Stephen C Johnson. 1967. Hierarchical clustering schemes. *Psychometrika* 32, 3 (1967), 241–254.

[31] Yun Kang and Stanley B Gershwin. 2005. Information inaccuracy in inventory systems: stock loss and stockout. *IIE transactions* 37, 9 (2005), 843–859.

[32] Hakan Koyuncu and Shuang Hua Yang. 2010. A survey of indoor positioning and object locating systems. *International Journal of Computer Science and Network Security* 10, 5 (2010), 121–128.

[33] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. 2011. Interpreting and Unifying Outlier Scores. In *Proceedings of SDM'11, Mesa, Arizona, USA*. 13–24.

[34] Nan Li and Burcin Becerik-Gerber. 2011. Performance-based evaluation of RFID-based indoor location sensing solutions for the built environment. *Advanced Engineering Informatics* 25, 3 (2011), 535–546.

[35] Tao Li, Shigang Chen, and Yibei Ling. 2010. Identifying the missing tags in a large RFID system. In *Proceedings of ACM International Symposium on mobile ad hoc networking and computing*. ACM, 1–10.

[36] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. 2007. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37, 6 (2007), 1067–1080.

[37] Lionel M Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P Patil. 2004. LANDMARC: indoor location sensing using active RFID. *Wireless networks* 10, 6 (2004), 701–710.

[38] Dimitris Papakiriakopoulos and Georgios Doukidis. 2011. Classification performance for making decisions about products missing from the shelf. *Advances in Decision Sciences* 2011 (2011).

[39] Dimitrios Papakiriakopoulos, Katerina Pramatari, and Georgios Doukidis. 2009. A decision support system for detecting products missing from the shelf based on heuristic rules. *Decision Support Systems* 46, 3 (2009), 685–694.

[40] Luis Peneda, Abílio Azenha, and Adriano Carvalho. 2009. Trilateration for indoors positioning within the framework of wireless communications. In *Proceedings of IECON'09*. IEEE, 2732–2737.

[41] TC Poon, King Lun Choy, Harry KH Chow, Henry CW Lau, Felix TS Chan, and KC Ho. 2009. A RFID case-based logistics resource management system for managing order-picking operations in warehouses. *Expert Systems with Applications* 36, 4 (2009), 8277–8301.

[42] Karthik Moncombu Ramakrishnan and Daniel D. Deavours. 2006. Performance benchmarks for passive UHF RFID tags. In *Proceedings of MMB'06*. 137–154.

[43] Ananth Raman, Nicole DeHoratius, and Zeynep Ton. 2001. Execution: The missing link in retail operations. *California Management Review* 43, 3 (2001), 136–152.

[44] Yacine Rekik, Evren Sahin, and Yves Dallery. 2008. Analysis of the impact of the RFID technology on reducing product misplacement errors at retail stores. *International Journal of Production Economics* 112, 1 (2008), 264 – 278.

[45] Antonio Rizzi and Giovanni Romagnoli. 2016. Testing and deploying an RFID-based Real-Time Locating System at a fashion retailer: A case study. In *Workshop on Business Models and ICT Technologies for the Fashion Supply Chain*. Springer, 201–214.

[46] Samer S. Saab and Zahi S. Nakad. 2011. A standalone RFID indoor positioning system using passive tags. *IEEE Transactions on Industrial Electronics* 58, 5 (2011), 1961–1970.

[47] Anurag Saran, Ehtesham Hassan, and Avinash Kumar Maurya. 2015. Robust visual analysis for planogram compliance problem. In *Proceedings of MVA'15*. IEEE, 576–579.

[48] Erich Schubert, Remigius Wojdanowski, Arthur Zimek, and Hans-Peter Kriegel. 2012. On Evaluation of Outlier Rankings and Outlier Scores. In *Proceedings of SIAM International Conference on Data Mining*. 1047–1058.

[49] Frédéric Thiesse and Thomas Buckel. 2015. A comparison of RFID-based shelf replenishment policies in retail stores under suboptimal read rates. *International Journal of Production Economics* 159 (2015), 126–136.

[50] Dieter Uckelmann and Giovanni Romagnoli. 2016. RF-based locating of mobile objects. In *Proceedings of the 6th International Conference on the Internet of Things*. ACM, 147–154.

[51] Ilias P Vlachos. 2014. A hierarchical model of the impact of RFID practices on retail supply chain performance. *Expert Systems with Applications* 41, 1 (2014), 5–15.

[52] Yang Zhang, Nirvana Meratnia, and Paul J. M. Havinga. 2010. Outlier Detection Techniques for Wireless Sensor Networks: A Survey. *IEEE Communications Surveys and Tutorials* 12, 2 (2010), 159–170.