Classification Models for RFID-Based Real-Time Detection of Process Events in the Supply Chain: An Empirical Study

THORBEN KELLER, University of St. Gallen
FRÉDÉRIC THIESSE, University of Wuerzburg
ELGAR FLEISCH, University of St. Gallen & ETH Zurich

RFID technology allows the collecting of fine-grained real-time information on physical processes in the supply chain that often cannot be monitored using conventional approaches. However, because of the phenomenon of false-positive reads, RFID data streams resemble noisy analog measurements rather than the desired recordings of activities within a business process. The present study investigates the use of data mining techniques for filtering and aggregating raw RFID data. We consider classifiers based on logistic regression, decision trees, and artificial neural networks using attributes derived from low-level reader data. In addition, we present a custom-made algorithm for generating decision rules using artificial attributes and an iterative training procedure. We evaluate the classifiers using a massive set of data on pallet movements collected under real-world conditions at one of the largest retailers worldwide. The results clearly indicate high classification performance of the classification models, with the rule-based classifier outperforming all others. Moreover, we show that utilizing the full spectrum of data generated by the reader hardware leads to superior performance compared with the approaches based on timestamp and antenna information proposed in prior research.

Categories and Subject Descriptors: H.4.2 [**Database Applications**]: Data Mining; H.4.2 [**Types of Systems**]: Logistics; I.5.2 [**Design Methodology**]: Classifier Design and Evaluation

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: RFID, false-positive reads, machine learning, logistic regression, decision trees, artificial neural networks, time series analysis

ACM Reference Format:
Thorben Keller, Frédéric Thiesse, and Elgar Fleisch. 2014. Classification models for RFID-based real-time detection of process events in the supply chain: An empirical study. ACM Trans. Manag. Inform. Syst. 5, 4, Article 25 (October 2014), 30 pages.
DOI: http://dx.doi.org/10.1145/2629449

1. INTRODUCTION

Radio-frequency identification (RFID) is a technology for the automatic identification by radio of physical objects such as industrial containers, pallets, or sales units. The identification event relies on transponders (“tags”) that are located in or on the respective objects that can be addressed without physical contact over the so-called air interface by an antenna on a reader device. In the past few years, the application of RFID technology in supply chain management has attracted the interest of several

Authors’ addresses: T. Keller (corresponding author) and E. Fleisch, ITEM-HSG, University of St. Gallen, Dufourstrasse 40a, 9000 St. Gallen, Switzerland; emails: {thorben.keller, elgar.fleisch}@unisg.ch; F. Thiesse, Chair of IS Engineering, University of Würzburg, Josef-Stangl-Platz 2, 97070 Würzburg, Germany; email: frederic.thiesse@uni-wuerzburg.de.

DOI: http://dx.doi.org/10.1145/2629449

industries worldwide [Sarac et al. 2010]. Due to standardization activities, cost erosion, and miniaturization of microelectronic components, the availability of low-cost RFID hardware today allows for its use beyond traditional niche applications [Thiesse et al. 2009b; Want 2004]. The technological characteristics of RFID open opportunities for collecting fine-grained, real-time information on physical processes in the supply chain that often cannot be monitored using conventional approaches. The hope among many of its proponents is that RFID will become the technological enabler of an unprecedented level of supply chain visibility [Lee and Özer 2007]. The manifold benefits expected from RFID include labor and time savings, increased service levels, faster reactions to supply chain disruptions, product security, and novel consumer services [Angeles 2005; Asif and Mandviwalla 2005; Bose and Pal 2005].

Several valuable models have been developed in IS and operations management that both explain how RFID can generate business value in organizations and support practitioners designing RFID-based systems and processes [Liao et al. 2011; Ngai et al. 2008]. However, as we argue in the following, most of these prior studies have ignored the fundamentally different data quality levels associated with RFID compared to the classical bar code. In contrast to the latter, RFID as a fully automatic identification technology collects data from any object equipped with an RFID tag within the fuzzy boundaries of the RF field. Because reader devices have no means to distinguish between objects of interest and others located in range by accident, data streams generated outside lab conditions resemble noisy analog measurements rather than the desired recordings of business process activities.

Against this backdrop, the present study examines the issue of effective RFID data filtering in supply chain applications. Our research objective is to design and evaluate models that transform raw RFID data streams generated by reader devices into meaningful information about the physical activities to be monitored. For this purpose, we consider different types of data aggregations—so-called attributes—based on raw RFID data. We use these attributes to generate classification models for the detection of invalid RFID tag reads. We investigate models ranging from single-attribute decision stumps to different standard classifiers such as logistic regression, neural networks, and decision trees. In addition, we present a custom-made algorithm for training classifiers based on decision rules. The dataset underlying the empirical evaluation was collected under real-world conditions at a distribution center of METRO Group, one of the largest retailers worldwide. The results were analyzed with respect to widely accepted performance criteria and demonstrate the effectiveness of the proposed solution.

The remainder of the article is organized as follows. The next section includes information on the practical background to our work and the issues associated with using RFID technology. Section 3 reviews the related work in the academic literature and identifies the research gap addressed by the study. In Section 4, we develop the conceptual foundation of RFID data filtering based on different classification approaches and a set of RFID-specific attributes. In Section 5, we present the results from the empirical evaluation. The article closes with a summary and outlook for further research.

2. CASE BACKGROUND

2.1. RFID in Distribution Center Operations

The ultimate purpose for RFID in the supply chain is the automatic identification and tracking of goods as they move from the supplier to the customer [Bose and Pal 2005; Hardgrave and Miller 2008]. RFID differentiates itself from the bar code through its capability for bulk identification, identification without a line of sight, the unambiguous identification of each individual object, storage of data about the object, and robustness against environmental influences and destruction [Finkenzeller 2010; Shepard 2005].

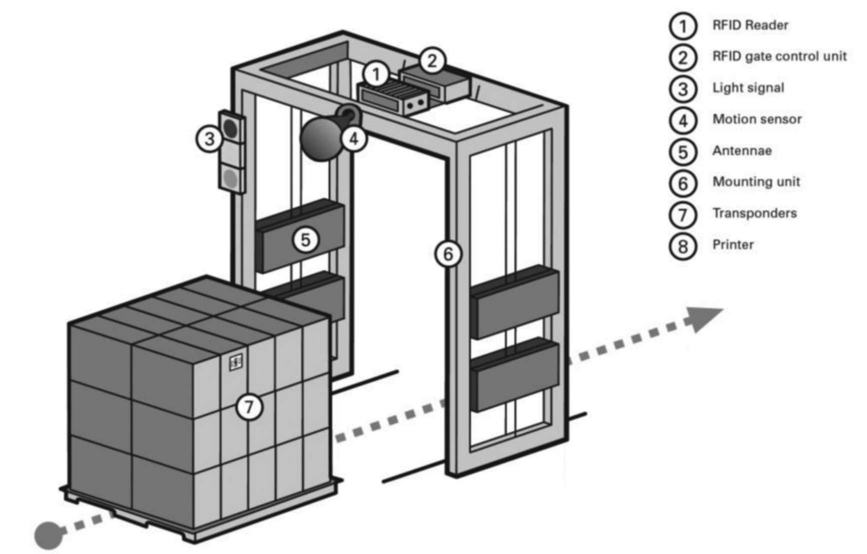

Fig. 1. RFID portal architecture.

The commonality of all RFID transponder types lies in a unique ID number that allows for identification not only of the product type but also at the item level. This allows, in principle, the seamless tracking of physical goods flows, thus making it much easier, for example, to detect the causes of shrinkage, monitor the performance of logistical operations, or trace the origin of contaminated food lots. To collect the required data, it is necessary to deploy RFID readers at specific key locations in the supply chain (e.g., where the goods are handed over from one party to another). Today, the fully automatic collection of RFID data is usually implemented at RFID portals that are available from various technology providers as turnkey hardware components. RFID portals are equipped with at least one reader device that controls one or more antennae to detect any RFID-equipped object moving through the portal.

In the following, we consider the example of METRO Group, an early adopter of RFID technology. The company deployed a solution for automatically identifying RFID-tagged pallets in its distribution center in Unna, Germany. The objective of the project was to track the pallets loaded onto truck trailers that are driven to METRO "Cash & Carry" stores. The underlying economic rationale was to minimize the number of faulty deliveries because the effort to reship those pallets is substantial and costly. Missing pallets also lead to additional economic loss in stores, for example, due to stock-outs. An additional advantage of using RFID technology was evident in the time savings and error reductions compared with the traditional paper-based process. To implement the RFID-supported process, METRO Group uses approximately 1,000,000 passive, preassembled tags annually. All 87 dock doors for outgoing goods are equipped with RFID portals, so every pallet loaded onto a trailer must pass a portal, where it is automatically detected. The main components of an RFID portal include the RFID reader itself, four antennae, a motion sensor to trigger the reader, and a light signal (Figure 1).

An exemplary installation of an RFID portal for the outgoing goods process in the distribution center (DC) is depicted in Figure 2. In this setting, tagged pallets or other logistical units (e.g., furniture and other large stacked objects that carry their own tags) designated for a specific customer are temporarily stored in a staging area, waiting to

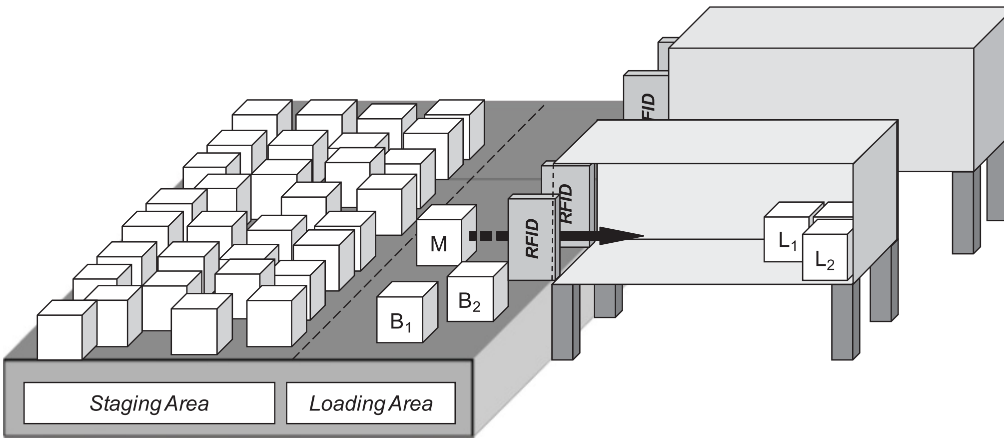

Fig. 2. Identification of outgoing pallets at the METRO DC.

be loaded onto a truck. Several trailers are located outside the dock doors, and each is monitored by an RFID portal. Warehouse people continuously pick one or more pallets from the staging area and transport them to the corresponding trailer. When a pallet passes through the RF field of a portal, the RFID transponder sends its unique ID to the reader device. Thus, the DC operator is provided with detailed information on the time, location, and completeness of his shipments.

The RFID-supported process can be described as follows. To begin, a warehouse person is handed a loading protocol. It contains information on the designated store in terms of a unique numerical identifier and the portal number where the corresponding trailer is waiting. Using a computer in the shipment office, the warehouse person informs the RFID software application that he or she is about to load pallets onto the trailer. At this point, two different workflows are commonly performed:

(1) The warehouse person retrieves a pallet from the staging area, returns to the dock door, and immediately loads it to an appropriate spot in the trailer.
(2) The warehouse person retrieves a pallet from the staging area but places it near the dock door instead of loading it onto the trailer. This is repeated until enough pallets are buffered there and the warehouse person decides to load them onto the trailer.

When the warehouse person approaches the RFID gate, the motion sensor recognizes him or her, and the RFID reader starts scanning for transponders. The data collection phase is called a *gathering cycle* and takes 5 seconds. As soon as the cycle is completed, the collected pallet ID codes are sent to the warehouse management system, and the warehouse person gets immediate visual feedback from the signal light:

(1) If the detected pallet was brought to the correct truck, the light flashes green. The loading was valid, and the warehouse person may continue with the next pallet.
(2) If the detected pallet was not designated for that particular store, the light flashes yellow. The warehouse person consequently unloads the pallet and continues with another.
(3) In any other case (e.g., if a tag is unknown to the warehouse management system), the signal flashes red.

After all pallets have been loaded onto the trailer, the warehouse person returns to the shipment office, where he or she informs the RFID software application that the loading process has been completed. The corresponding invoice is issued to the store, and a transportation request is sent to a shipper.

2.2. Characteristics of Low-Level RFID Data

Although the automatic detection of goods using RFID seems trivial at first glance, technical constraints limit the quality of the collected data. These constraints are rooted in the basic principles of RF communications, which directly influence the readability of the RFID tags. In particular, the absorption and reflection of radio waves may lead to unexpected read events and, ultimately, to false-negative and false-positive RFID tag reads [Jones and Chung 2008]:

—A *false-negative RFID tag read* describes any RFID tags that are in range as “invisible” to the reader device. This phenomenon corresponds to a pallet that was moved through the RFID portal and loaded onto the truck without being detected. The possible reasons for this event are manifold. For example, the types of products on the pallet have a significant influence on the readability of RFID tags because water and any other liquids (e.g., shampoo) can absorb radio waves, thus severely reducing the read range [Singh et al. 2009]. Other reasons include dysfunctional tags or tags shielding each other. To overcome this problem, multiple antennae are often installed to increase the likelihood of reading a tag. However, this solution may lead to another problem, namely, the mutual elimination of radio waves due to interference effects [Penttila et al. 2006].

—In contrast, *false-positive RFID tag reads* can have two different causes. On the one hand, the phenomenon is similar to false negatives because the physical conditions may influence the readability of the RFID tags. Metal foils and metal ink in goods or packages, the truck itself, or any other metallic object in range of the antennae may significantly extend their read range. As a consequence, tags assumed to be clearly out of range can be unexpectedly read by the reader. On the other hand, false positives can also be due to tags that are clearly present within the configured read range but are read involuntarily.

The latter issue of false-positive RFID tag reads can be illustrated by the previously described scenario of an RFID-enabled outgoing goods process. In the example depicted in Figure 2, a pallet M is about to be loaded into one of the two trailers. Two further pallets (B_1 and B_2) have been temporarily placed in the loading area by the warehouse person. Moreover, pallets L_1 and L_2 have already been loaded. Because the portal antennae are not directed and have a read range of several meters, they detect not only M but also any other pallet in range. The reader device cannot distinguish between moving pallets and those that are located in the RF field only by accident. As a consequence, the warehouse management system might conclude that B_1 and B_2 have also been shipped to the customer. In the worst case, pallets in both the trailers and the nearby staging area might be detected due to electromagnetic reflections. If all detected pallets were reported to the warehouse management system as shipped, incorrect invoices would be issued and stores would be billed for goods that they neither ordered nor received. Until this problem is solved, the reliable and productive use of RFID technology in distribution center processes is not feasible.

In the following, we refer to pallets loaded onto a trailer during a gathering cycle (i.e., the true positives) as *moved pallets*, whereas all others that have been read by accident (i.e., the false positives) are called *static pallets*. Based solely on the knowledge of which RFID tags have been read during a gathering cycle, it is impossible to determine which of them was attached to a pallet loaded onto the truck. The question that we consequently consider is to what extent the analysis of low-level RFID data—so-called

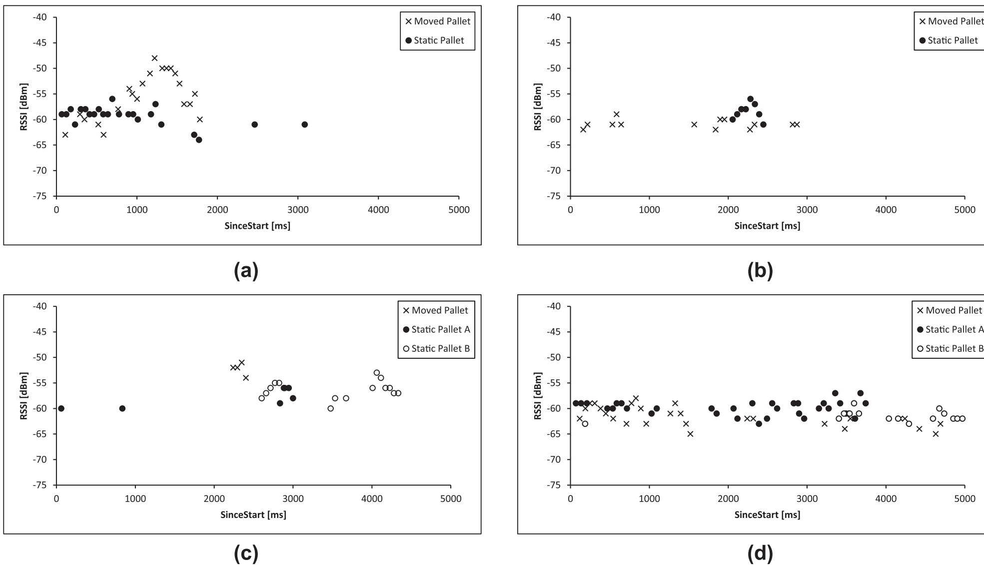

Fig. 3. Exemplary gathering cycles and tag events.

tag events—may allow for developing a solution to the false-positive read problem. Each tag event is characterized by the following types of information:

—*RSSI*. The Received Signal Strength Indication (RSSI) denotes the power of the tag's radio signal measured in dBm, which can intuitively be interpreted as how "loudly" the tag was heard by the antennae. By nature, the RSSI value increases when a tag is closer to the antennae and decreases when it is farther away.
—*SinceStart*. The second piece of information is the timestamp of the tag event relative to the beginning of the gathering cycle, measured in milliseconds.
—*Antenna*. The third piece of information tells us which of the antennae actually detected the tag. The RFID gates under consideration are typically equipped with four antennae, but other configurations are possible with up to eight antennas.

Figure 3(a) depicts an example of tag events that occur during a gathering cycle. The data include RSSI and SinceStart values; antenna data were omitted for clarity. The example highlights the difficulties associated with low-level RFID data and the need for automatic classification mechanisms. In this case, two pallets were present in the RF field, with one pallet being moved through the gate and another one located nearby. The RSSI values of a moving tag show a different behavior over time compared with those of a static tag. The moved pallet is first detected with increasing signal strengths, reaching a maximum when the tag actually enters the gate, approximately 1.5 seconds after the start of the gathering cycle. After leaving the gate, the signal strength decreases again. In contrast, the RSSI values of the static pallet remain approximately constant because its location does not change.

Unfortunately, though this example poses an illustrative case of RFID-based pallet identification, completely different event patterns may also occur. Figures 3(b), 3(c), and 3(d) show gathering cycles in which neither the recorded timestamps nor the signal strength measurements allow for an intuitive interpretation of the monitored real-world activities. It is evident from these examples that the identification of the one pallet of interest among the gathered data poses a nontrivial task. However, without

Table I. Overview of Prior Research

Study	Approach	Raw Data Used	Evaluation
Brusey et al. [2003]	Sliding window	-Timestamp	Lab trial (no results reported)
Bai et al. [2006]	Sliding window	-Timestamp	Simulation
Jiang et al. [2006]	Sliding window, multiple tags/readers	-Timestamp -Antenna	Lab trial
Tu and Piramuthu [2008]	Sliding window with multiple tags/readers	-Timestamp -Antenna	Simulation

an effective filtering procedure, the value of RFID data to any form of process control remains limited.

3. PRIOR RESEARCH

The various issues surrounding the processing of RFID data have been the subject of a steadily growing body of academic literature. The fundamental idea of drawing benefits from RFID beyond those of automation has been discussed in various prior works [Loebbecke and Palmer 2006; Sellitto et al. 2007; Tajima 2007]. In recent years, some initial studies have discussed the value of RFID data analytics against the background of real-world implementations. Delen et al. [2007] identified a number of performance metrics that can be computed from RFID data and discussed how these measures can improve logistical performance at a micro-supply-chain level of operations between a distribution center and a store. Baars and Sun [2009] discussed options for modeling and utilizing multidimensional datasets for analytical applications using two case studies from the retail and automotive industries. Thiesse et al. [2009a] examined the benefits of performance indicators and management reports generated from RFID data in the context of a large implementation project in a department store. In contrast to these works, however, research on the underlying procedures for data cleansing and information extraction from large RFID datasets is still sparse.

An overview of solutions to the problem of false positives proposed in prior studies is given in Table I. The approaches in the literature can roughly be divided into two groups. First, some authors proposed the use of a sliding-window approach. In these studies, a smoothing procedure is applied to the RFID data stream using the number of tag detections in a predefined time interval. The underlying assumption is that false positives may be distinguished from true positives by the smaller number of reads.

—Brusey et al. [2003] analyzed false-positive RFID tag reads in the context of a first-in, first-out product queue. In this setting, RFID-tagged men's shaving items, such as razors and deodorant, are stacked on top of each other. Items are added only at the top of the stack and are removed from the bottom. An RFID reader scans the next lowermost item to be removed by a robotic arm. The challenge is that not only is the lowermost item scanned but also various items on top of it are scanned. These are considered false-positive reads and need to be removed from the output. The detection of false positives uses the fact that only a single item (i.e., the lowermost) needs to be identified. Consequently, the item that has been read most often is classified as the item at the bottom. The procedure is illustrated by an example from a lab trial, but no empirical results are reported.

—Similarly, Bai et al. [2006] proposed algorithms for RFID data filtering, including noise removal and duplicate elimination. They argued that in practice, readings are often performed in multiple cycles to achieve a higher recognition rate. This method significantly reduces false-negative reads but unfortunately also leads to an increased rate of false-positive reads. The authors present a false-positive elimination algorithm based on a sliding-window approach. If the number of readings is greater than a given threshold, the tag is classified as a true positive. In addition, they complement their heuristic by additional procedures for preserving the order of tag reads and removing duplicates. The algorithms are evaluated using a set of simulated tag reads.

A second group of authors proposed the use of the sliding-window approach in combination with multiple tags or readers. In these settings, a reading is classified as a true positive if more than one reader detects the tag or if one reader detects more than one tag belonging to the same object.

—Jiang et al. [2006] considered false-positive reads in terms of object interaction. The authors used the poll command of the reader device to transmit N polls per second. The number of answers per tag was then used to identify the specific interaction with a physical object. However, the authors found that the response rate changes not only when interacting with an object but also if additional tags (i.e., false positives) are in range. They proposed to manage this type of issue by using additional tags per object and multiple readers. If, for example, an object has two tags attached to different sides, a rotation of the object is recognized by an increased response rate of the first tag and a decreased response rate of the second. The authors provided some aggregated performance data from tests in the laboratory but no details on the underlying experimental design.

—Tu and Piramuthu [2008] analyzed the so-called true and false reads in terms of the presence and absence of RFID-tagged objects. In their theoretical scenario, two readers are used simultaneously, and two tags are expected to be present at the same time. A first algorithm is used as a base case to compare the results of two others. If both readers identify a tag as being present, it is assumed to be actually present. In a case where only one or neither of the readers detects the tag, it is assumed that the tag is absent. The second algorithm is similar to the first. If both readers agree that a tag is present, it is assumed to be present; if neither reads the tag, it is assumed to be absent. However, if only one of the readers detects the tag, a sliding-window approach is used. The third algorithm uses information about a second tag that is expected to be read at the same time (i.e., each object is equipped with two RFID tags). In the case where only one reader detects the tag of interest, information from the other tag is used to reach a decision. If both readers agree that the second tag is present, the first one is assumed to be present as well. In the case where the readers disagree about the presence of both tags, a sliding-window approach is used, as in the second algorithm. The three algorithms are evaluated in a simulation study, but no details are given on the experimental design.

These earlier studies suffer from various weaknesses, which we intend to address as follows:

—*Usage of RFID data.* All of the studies mentioned previously use a sliding-window approach based on tag reads and the corresponding timestamps. However, RFID data generated by reader devices are richer than is suggested in the literature. We consider additional low-level reader data in the form of signal strength information. The use of signal strength measurements has been proposed repeatedly in the context of real-time location systems (e.g., Joho et al. 2009), but no prior study has investigated its value for RFID data stream cleansing.

—*Hardware cost.* Some authors propose using additional readers or tags. However, the concept of increasing the number of readers or tags appears rather impractical considering the high costs of RFID hardware components. In contrast to this "brute force" approach, we do not rely on additional hardware but instead apply sophisticated data mining techniques to solve the false-positive read problem.

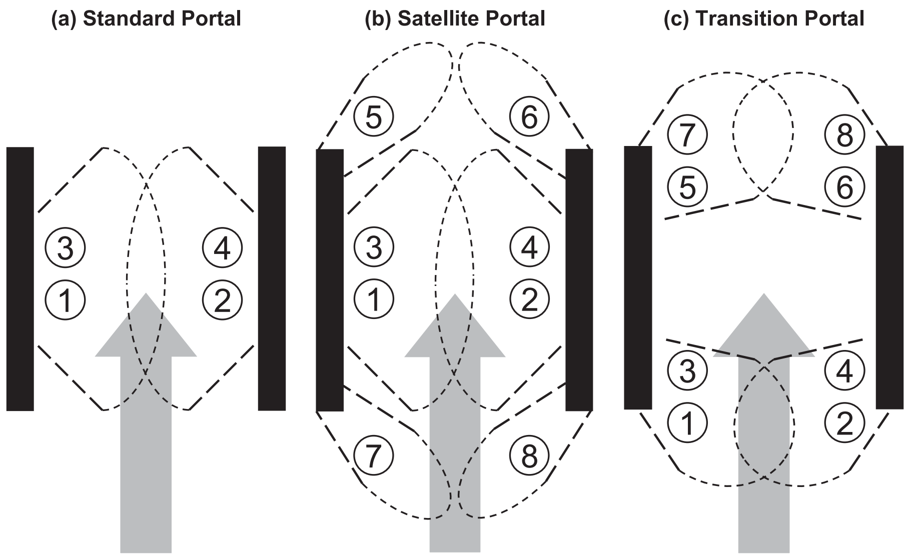

Fig. 4. Alternative RFID portal configurations (dotted lines denote antenna directions).

—*Real-world evaluation.* None of the earlier studies is based on real-world data; instead, they use computer simulations or trials under lab conditions. Assumptions regarding RFID hardware behavior and the generalizability of the results are thus questionable, given the complex physical characteristics of RF communications (e.g., the phenomenon of electromagnetic reflections). We avoid this shortcoming through a massive dataset obtained over a longer period of time in a production environment.

4. CONCEPTUAL APPROACH

4.1. Terms and Definitions

Removing false-positive tag reads poses a binary classification task, where tag detections are assigned to one of two possible classes: (1) “moved” (i.e., true positive) and (2) “static” (i.e., false positive). In this section, we develop the conceptual foundations for the construction of such RFID data classification models. Our starting point is the architecture and the mode of operation of RFID portals. The use of the term “RFID portal” in this context refers to the three portal types depicted in Figure 4.

These portal types contribute to the ability to generalize our results and allow us to evaluate the impact of different hardware configurations on classification performance:

—The most commonly used type of RFID portal in logistics is the *standard portal*, which is available off the shelf from various hardware manufacturers and system integrators. Here, a single reader device has four main antennae attached to it, two at each side of the portal, on top of each other and face to face with the other two. The portal is equipped with a motion sensor, which triggers a gathering cycle. The entire data collection of a pallet loading occurs during this cycle. At some point the warehouse person leaves, which is also recognized by the motion sensor.

—*Satellite portals* are an advanced version of the standard portals that use an additional RFID reader with four more antennae. Two of these antennae (5 and 6) are directed toward the truck trailer and the other two (7 and 8) are directed toward the staging area. Accordingly, the additional antennae are denoted “truck antennae” and “DC antennae,” respectively. The four remaining antennae (1–4) correspond to

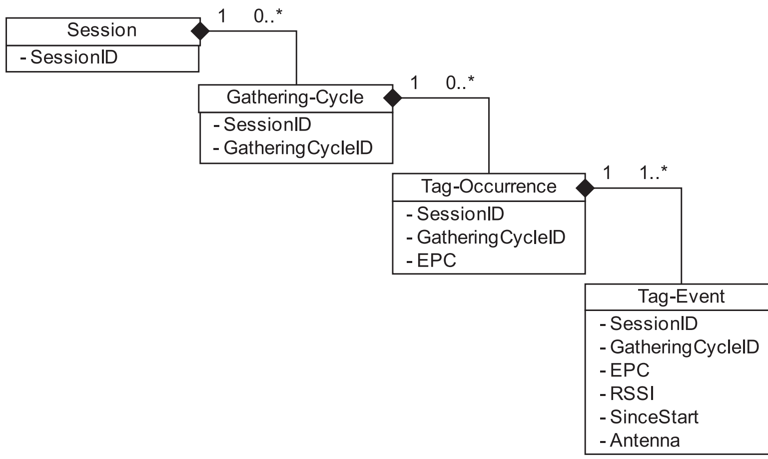

Fig. 5. Logical structure of tag event data.

the main antennae used in the standard portals. This configuration is based on the assumption that a tag that has moved through the portal is expected to be inside the trailer rather than inside the distribution center. Consequently, all tag reads detected by the DC after the end of the actual loading process should be considered false positives.

—Similarly, *Transition Portals* use two different readers, but they do not have any main antennae. The four antennae of the first reader (1–4) are directed toward the DC, and the four antennae of the second reader (5–8) are directed toward the trailer. In this configuration, a tag that moves through the portal is expected to be seen first by the DC antennae and then by the truck antennae.

Each RFID tag can be identified by its unambiguous *Electronic Product Code* (EPC). Usually, a specific transponder in range is read more than only once per cycle. Each of these responses is called a *tag event* that is uniquely characterized by the combination of a session ID, a gathering cycle ID, the EPC, and a timestamp. For each tag event, the reader also records the signal strength of the tag response and the antenna that read the tag. Each tag event t can hence be represented as a three-tuple of signal strength, timestamp, and antenna:

$$t = (RSSI, SinceStart, Antenna).$$

The set of tag events associated to a specific RFID tag during a gathering cycle is called a *tag occurrence*. If a tag has been detected n times, the corresponding tag occurrence T (the entirety of these tag events) can be represented as follows:

$$T = \{t_1, \ldots, t_n\} = \{(RSSI_1, SinceStart_1, Antenna_1), \ldots, (RSSI_n, SinceStart_n, Antenna_n)\}.$$

An overview of the resulting conceptual model of low-level RFID data is given in Figure 5. Not all of the data elements can potentially be used as input data for a classification model because they are not specific to the individual tags. However, some elements are specific to each tag event (RSSI, SinceStart, Antenna) and can help distinguish between moved and static objects.

4.2. Approaches for RFID Data Classification

To build classification models, meaningful attributes must be derived from the RFID data. In this context, "meaningful" denotes the contribution that the value of a

EPC	SOURCE	SUBSRC	TIME	SINCESTART	ANT	RSSI
Tag1	Portal56	MAIN	12:37:00,961	66000	3	-59
Tag2	Portal56	MAIN	12:37:01,004	109000	2	-63
Tag1	Portal56	MAIN	12:37:01,017	122000	3	-59
Tag1	Portal56	MAIN	12:37:01,072	177000	3	-58
Tag1	Portal56	MAIN	12:37:01,126	231000	3	-61
Tag2	Portal56	MAIN	12:37:01,186	291000	2	-59
Tag1	Portal56	MAIN	12:37:01,197	302000	3	-58
Tag2	Portal56	MAIN	12:37:01,243	348000	2	-60
Tag1	Portal56	MAIN	12:37:01,254	359000	3	-58
Tag1	Portal56	MAIN	12:37:01,306	411000	3	-59
...				...		...

EPC	RssiMax	RssiMin	RssiMean	Cnt	SinceStartMax	...
Tag1	-48	-63	-55.4	21	1781000	...
Tag2	-56	-64	-59.4	23	3084000	...

Fig. 6. Calculating tag-occurrence-level attributes from tag events.

particular attribute makes to the correct classification of moved and static pallets. For this purpose, we consider the development of attributes on the levels of both tag events and tag occurrences.

On the *tag occurrence level*, attributes are calculated and generated by applying various aggregation functions that correspond to specific characteristics of the data. Examples of such characteristics include the maximum, minimum, and mean RSSI values, as well as the timestamp of the first or the last recognition of a tag during a gathering cycle. Figure 6 shows examples of how a sequence of low-level reader data is transformed into attribute values. The idea is to identify false-positive RFID tag reads based solely on these attributes. However, the difficulty of this approach lies in determining which attributes are meaningful enough that a significant difference can be observed. It is hence necessary to determine which attributes are useful under certain conditions and what values are typical for moved and static tags.

On the *tag event level*, the development of attributes follows a different logic. Because tag events collected during a gathering cycle are temporally ordered, they can be interpreted as a discrete time series of RSSI values (Figure 7). The idea behind this approach is to examine whether the time series of a particular tag is more similar to a moved tag or a static tag. In the latter case, it is considered a false positive. The first challenge is to determine what a typical moved or static time series actually looks like. The latter requires a reference time series, which must be derived from the sample data.

A second challenge is to agree on a similarity measure. We consider *Dynamic Time Warping* (DTW) to address this problem by using local scaling to determine the distance between two time series [Sakoe and Chiba 1978]. Originally, this approach was introduced as a technique for speech recognition to cope with different speaking speeds. Because we are facing a similar issue with physical processes occurring at different speeds, DTW seems an appropriate choice. Simply put, DTW can cope with the temporary acceleration or deceleration of the warehouse person as he or she moves a pallet through the portal. The same holds for the problem of differing radio signal strengths (i.e., the "loudness" of a tag signal).

EPC	SOURCE	SUBSRC	TIME	SINCESTART	ANT	RSSI
Tag1	Portal56	MAIN	12:37:00,961	66000	3	-59
Tag2	Portal56	MAIN	12:37:01,004	109000	2	-63
Tag1	Portal56	MAIN	12:37:01,017	122000	3	-59
Tag1	Portal56	MAIN	12:37:01,072	177000	3	-58
Tag1	Portal56	MAIN	12:37:01,126	231000	3	-61
Tag2	Portal56	MAIN	12:37:01,186	291000	2	-59
Tag1	Portal56	MAIN	12:37:01,197	302000	3	-58
Tag2	Portal56	MAIN	12:37:01,243	348000	2	-60
Tag1	Portal56	MAIN	12:37:01,254	359000	3	-58
Tag1	Portal56	MAIN	12:37:01,306	411000	3	-59
...			...	...		...

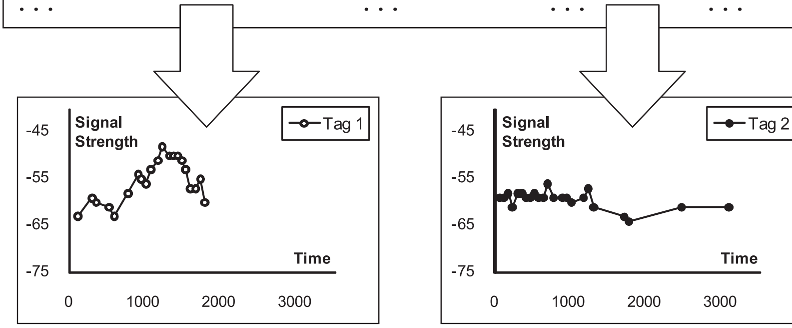

Fig. 7. Time-series analyses based on sequences of tag event.

In general, a reference series R for a tag class C must satisfy the following two conditions:

(1) R is as similar as possible to all time series in C.
(2) R is as dissimilar as possible to all time series not in C.

From all available samples, an average time series can be calculated and returned as the reference. This approach leads directly to the question of how the average of a set of time-series is defined. Let $T = (t_1, \ldots, t_n)$ and $U = (u_1, \ldots, u_n)$ be two time series. Then, the average time series V of T and U can be calculated by averaging the respective data points:

$$V = \left(\frac{t_1 + u_1}{2}, \ldots, \frac{t_n + u_n}{2} \right).$$

More generally, if there are k different time series $= \{T_1, \ldots, T_k\}$, an average data point v_i is calculated as

$$v_i = \frac{\sum_{j=1}^{k} t_{j_i}}{k}.$$

However, this technique requires that all time series have the same length because only in this case can an average value be computed. To address this issue, another approach is presented here to interpolate a time series while keeping the temporal order of the individual tag events. The entire gathering cycle is divided into k time intervals of equal length t. Consequently, the reference series R has a length of k data points. If $M = (m_1, \ldots, m_n)$ is a time series with corresponding timestamps $(t_1, \ldots, t_n)$, then the kth data point of R is the average of all data points of M that lie within the interval $I = [\Delta t \cdot (k); \Delta t \cdot (k+1)]$. In a case where no tag event occurred within a specific interval, the two preceding and two succeeding tag events are averaged and used as

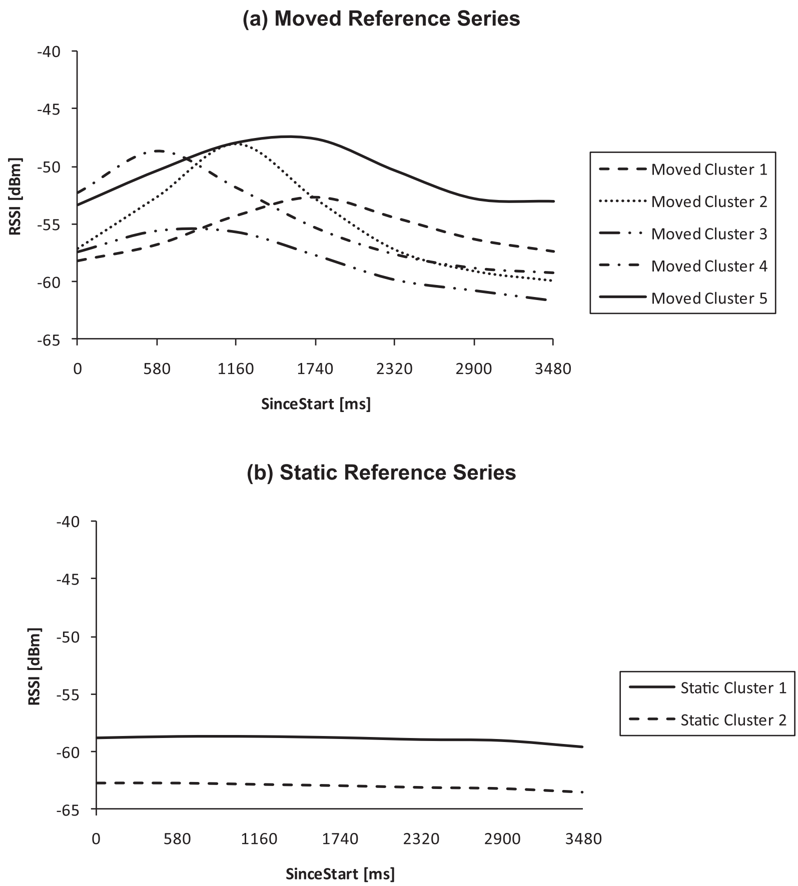

Fig. 8. Reference time series for standard portals.

an interpolation. If there are no preceding or succeeding tag events, the first or last tag event is used, respectively.

Given the variety of conceivable loading processes, it is evident that more than one reference time series exists for moved or static tags. In fact, further subclasses can be found within each of the two tag classes. To improve the classification of moved and static tags, these subclasses must be identified and their respective reference time series generated. For this purpose, we consider cluster analysis using k-Means [MacQueen 1967] and k-Medoid partitioning [Kaufman and Rousseeuw 1990]. An example of a reference series generated in this way from RFID data collected by standard portals is given in Figure 8. These reference series reflect the actual behaviors of moved and static tags, as expected.

The similarity between the reference series and a given time series is calculated using a similarity query. For this purpose, we consider the k-nearest neighbor query. Under this approach, the k nearest neighbors—which correspond to the references that are closest to the query tag—are retrieved. For $k = 1$, only the closest (i.e., the most similar) reference is returned. The *k-NN* query is formally defined as

$$k - NN(t) = \{R \subseteq \mathbb{M} \cup \mathbb{S} || R| = k \wedge \forall r \in R, u \in (\mathbb{M} \cup \mathbb{S}) \backslash r : d(t, r) \leq d(t, u)\},$$

Table II. Domain Attributes Based on the RSSI Information

Attribute	Description
$RSSI_{Min}$	The minimum signal strength measured during a gathering cycle. Because moved tags are expected to be located slightly closer to the antennae at the beginning of the pallet loading, a higher minimum RSSI value should be observed relative to the static pallets.
$RSSI_{Max}$	The maximum signal strength measured during a gathering cycle. Because moved pallets pass through the portal and move closer to the antennae at this time, they are expected to have a higher maximum RSSI value.
$RSSI_{Diff}$	The difference between the highest and lowest signal strengths that is measured during a gathering cycle. The value range is a dispersion measure of the RSSI values. Because moved pallets continuously change their distance to the antennae, they are expected to have a higher dispersion and thus a higher $RSSI_{Diff}$ attribute value than are static tags.
$RSSI_{Mean}$	The average signal strength that was measured during a gathering cycle. Because moved pallets spend more time closer to the antennae while they pass through the portal, it is expected that they have a higher average RSSI value relative to static pallets.
$RSSI_{StDev}$	The standard deviation of the RSSI values. Similar to $RSSI_{Diff}$, this is a dispersion measure; therefore, a higher attribute value is expected for moved pallets than for static pallets.
$RSSI_{CoV}$	The coefficient of variation of the RSSI values, which is defined as the ratio between the standard deviation and the average RSSI value. The mathematical expression can be converted to a form that solely depends on the $RSSI_{Mean}$ attribute. Because this attribute is expected to take on higher values for moved pallets, the coefficient of variation is expected to be lower for moved pallets than for static pallets.

where $\mathbb{M}$ and $\mathbb{S}$ correspond to the set of typical moved and static references, t is the tag of interest, and d is a distance function (e.g., DTW). This query type can be used to determine whether the k most similar references correspond to moved or static time series. If the ranking is ambiguous (i.e., if there are several references with similar distance but different class types), a majority vote determines the class type of the query tag. Choosing $k = 1$ returns only the nearest neighbor; consequently, the query tag is classified as the corresponding class.

4.3. Attribute Development

Using the two approaches described in the previous subsection, we propose the use of the following types of attributes for the classification task: (1) domain attributes, (2) logical attributes, and (3) time-series attributes.

4.3.1. Domain Attributes. *Domain attributes* are based on the experience and knowledge of the people working in the environment under consideration. For example, one obvious difference between moved and static pallets that can be derived from the business process is that the former get closer to the RFID antennae than do the latter. Based on this information and the characteristics of the RSSI data, one might conclude that the maximum RSSI value measured during a gathering cycle is a valuable attribute to distinguish between the moved and static pallets because the former are expected to show a higher attribute value than are the latter.

Consequently, the first types of domain attributes that we consider are the *RSSI-based attributes* (Table II). The calculation of RSSI attributes is based on the unordered set of the corresponding RSSI values. Underlying the definition of these attributes is the observation that many static pallets are farther away from antennae than are the moved pallets. Because the received signal strength depends on the distance between the sender and receiver, it is expected that the RSSI attributes successfully reflect this insight and therefore play a major role in the ability to distinguish between moved and static tags.

Table III. Domain Attributes Based on the SinceStart Information

Attribute	Description
$\text{Read}_{\text{First}}$	The time since the beginning of the gathering cycle until the tag is first read. Moved pallets are typically read starting at the very beginning of the cycle, while certain false positives are detected only after a few seconds.
ReadLast	The time since the beginning of the gathering cycle until the tag is last read. At the end of a cycle, basically any type of pallet may be read due to electromagnetic reflections. Consequently, the timestamps of the last tag read are not expected to differ substantially. However, this attribute might be helpful in combination with another one.
ReadDiff	The time that has passed between the first and the last detection of the tag. Because moved pallets are often read at both the beginning and the end of a gathering cycle, this attribute is expected to take on higher values for moved tags than for static tags.

Table IV. Domain Attributes Based on the Antenna Information

Attribute	Description
CountX	The number of tag events that were recorded by each of the antennae, where X is the identifier of the corresponding antenna. Because many static tags remain close to specific antennae, it is expected that these antennae detect these static tags more often than they detect the moved tags.
AntCount	The number of antennae that were able to detect the tag. Because many static tags are close to a particular antenna, it is expected that moved pallets are read by more antennae than are static tags.
CountMain	The total number of reads the tag gave to all of the antennae combined. Because moved tags pass through the portal and are thus very close to the antennae, it is expected that they are read more often in total than static tags are.

A second group of domain attributes can be generated from the SinceStart information associated with every tag event (Table III). The calculation of the *SinceStart attributes* is based on the unordered set of the corresponding SinceStart values. The purpose behind these attributes is the observation that certain static pallets are detected only occasionally by specific antennae. For example, some false-positive reads are the result of unexpected reflections that only occur randomly. In contrast to moved pallets, these are usually not detected over the entire duration of a gathering cycle.

Furthermore, the information from the antennae that received a tag's response signal might be useful for classification purposes. The calculation of *antenna attributes* is based on the unordered set of the corresponding antenna values (Table IV). The motivation for defining these attributes is the same that leads to the definition of the SinceStart attributes, namely, the fact that certain static pallets are detected only occasionally. Hence, it can be expected that a moved tag would be detected more often and by more antennae than would a static tag.

4.3.2. Logical Attributes. Because satellite and transition portals are equipped with different antenna types (i.e., main antennae, truck antennae, DC antennae), it is also possible to analyze the order in which a tag was read by these antennae. For example, it can be expected that a pallet being loaded into the trailer would first be read by the DC antennae and later by the truck antennae. Following this rationale, a number of so-called *logical attributes* can be defined, as shown in Table V.

4.3.3. Time-Series Attributes. In contrast to the previously proposed attributes based on tag occurrences, it is also possible to define attributes using the similarity of a sequence of tag events to a set of reference time series. This allows for combining the tag-event-level with the tag-occurrence-level classification and for integrating the results from time-series analysis of RFID data into our classification models. A list of the *time-series attributes* used in this study is given in Table VI.

Table V. Logical Attributes

Attribute	Description
$\text{Where}_{\text{Read}}$	This attribute is an integer representation of the antennae that have read the tag.
$\text{Seen}_{\text{First}}$	This attribute examines the $\text{Read}_{\text{First}}$ attribute values to determine whether a tag was seen first by the main or truck antennae.
$\text{Seen}_{\text{Last}}$	This attribute examines the $\text{Read}_{\text{Last}}$ attribute values to determine whether a tag was last seen by the DC, main, or truck antennae.
$\text{Seen}_{\text{Longer}}$	This attribute examines the $\text{Read}_{\text{Diff}}$ attribute values to determine which reader read the tag over the longest period of time.
$\text{First}_{\text{Main}}\text{Last}_{\text{Truck}}$	This attribute determines whether a tag was first read by the main antennae and last read by the truck antennae. It is likely that such a tag moved through the portal.
$\text{First}_{\text{Truck}}\text{Last}_{\text{Main}}$	This attribute determines whether a tag was first read by the truck antennae and last read by the main antennae. It is unlikely that this would happen, but if it does, it is presumably a static tag.
$\text{First}_{\text{Main}}\text{Last}_{\text{Main}}$	This attribute determines whether the first and last detection of a tag occurred at the main antennae. It is unlikely that this would happen, but if it does, such a tag is presumably a static tag located somewhere in the DC.
$\text{First}_{\text{Truck}}\text{Last}_{\text{Truck}}$	This attribute determines whether the first and last detection of a tag occurred at the truck antennae. If this is the case, it is likely that the tag was already inside the container during the entire gathering cycle and is hence static.
$\text{Disjoint}_{\text{Main,Truck}}$	This attribute determines whether a tag was read only by the main antennae in the beginning and later only by the truck antennae. Because this may be considered the optimal case for a loaded pallet, it is expected that tags for which this condition holds have been moved through the portal.
$\text{Disjoint}_{\text{Truck,Main}}$	This attribute determines whether a tag was read only by the truck antennae in the beginning and later only by the main antennae. It is unlikely that this would happen, but if it does, it is presumably a static tag.

4.4. Classifier Development

The construction of a classification model follows a generic train and test procedure, as depicted in Figure 9. To form a judgment on how well a classifier will actually perform on unseen data, it is necessary to determine its overall performance. Testing the model on the data with which it was built is misleading, and collecting new data every time to verify its quality is not a feasible option. Therefore, a dataset is divided into two disjoint subsets referred to as the *Training Set* and the *Test Set*. For our three standard classifiers, the classification model is built on the training set and is then independently tested against the data in the test set.

Several different classification model types might be used to decide whether a pallet has been moved through an RFID portal. Within the scope of the present study, we consider (1) decision stumps, (2) three different standard classifiers, and (3) a custom-made classifier using artificial attributes and an iterative procedure for the generation of decision rules.

4.4.1. Decision Stumps. The simplest type of classification model relies on only one attribute and a threshold value. These so-called decision stumps correspond to the heuristics proposed in prior studies. For example, a decision stump using the Count-Main attribute (i.e., the total number of tag reads per cycle) is equivalent to a sliding-window approach, where a true positive is characterized by a minimum number of reads. Similarly, a decision stump using the AntCount attribute (i.e., the number of antennae that detected a tag) is equivalent to the idea of using more than one reader to distinguish true and false positives. For this reason, we use decision stumps as the benchmark against which all other approaches are compared.

Table VI. Time-Series Attributes

Attribute	Description
D_M, D_S	The distance between the tag and the reference time series of all moved and static tags, respectively. By nature, moved tags should have a shorter distance to this time series than do static tags and vice versa.
$D_{M,Ci}$, $D_{S,Ci}$	In cases where i subclasses have been identified for the moved tags and j subclasses for the static tags, this attribute corresponds to the distance to the respective cluster reference time series. By nature, moved tags should have a shorter distance to the moved reference time series, and static tags should have a shorter distance to the static reference time series.
$D_{M,Min}$, $D_{S,Min}$	The minimum of the distances to all available moved and static reference time series, respectively. By nature, moved tags should have a shorter minimum distance to the moved reference time series, and static tags should have a shorter minimum distance to the static reference time series.
$D_{M,Max}$, $D_{S,Max}$	The maximum of the distances to all available moved and static reference time series, respectively. By nature, moved tags should have a shorter maximum distance to the moved reference time series, and static tags should have a shorter maximum distance to the static reference time series.
$D_{M,Mean}$, $D_{S,Mean}$	The average of the distances to all available moved and static reference time series, respectively. By nature, moved tags should have a shorter average distance to the moved reference time series, and static tags should have a shorter average distance to the static reference time series.
$D_{M,StDev}$, $D_{S,StDev}$	The standard deviation of the distances to all available moved and static reference time series, respectively. By nature, moved tags should have a smaller standard deviation of the distances to the moved reference time series. The same applies to static tags.
$D_{M,CoV}$, $D_{S,CoV}$	The coefficient of variation of the distances to all available moved and static reference time series, respectively. By nature, moved tags should have a lower CoV value with respect to the distances to the moved reference time series. The same applies to static tags.
NN	The class of the nearest neighbor (i.e., the class of the reference series to which the distance is minimal) of the tag. By nature, the nearest neighbor of a moved tag should be a moved reference series, and the nearest neighbor of a static tag should be a static reference series.
FN	The class of the furthest neighbor (i.e., the class of the reference series to which the distance is maximal) of the tag. By nature, the furthest neighbor of a moved tag should be a static reference series, and the furthest neighbor of a static tag should be a moved reference series.
$\text{Agree}_{FN,NN}$	Indicates whether NN and FN agree, that is, whether the nearest neighbor and the furthest neighbor correspond to different tag classes. If the nearest and furthest neighbor correspond to the same tag class, then a decision is not possible. In any other case, the class of the nearest neighbor is returned. Note that only for the corresponding object classes (i.e., moved and static) can the class precision be calculated.

4.4.2. Standard Classifiers. A second group of models makes use of different combinations of attributes. Based on the recommendations for classifier selection made by Kiang [2003], our analysis considers three types of standard classifiers: (1) logistic regression, (2) neural networks, and (3) decision trees.

—*Logistic regression* is a straightforward extension of conventional linear regression that allows for binary dependent variables and hence suits a two-class classification problem. It employs the linear predictor function $g = \beta_0 + \sum_i \beta_i X_i$ as the argument of a nonlinear logistic function. Maximum likelihood optimization estimates the coefficients β_i for given training data. The output of the resulting logistic regression function is interpreted as the probability of event occurrence given a predictor variable vector X [Christensen 1997]. This value corresponds to the classifier score; for classification purposes, a simple threshold applied to it yields the predicted class labels.

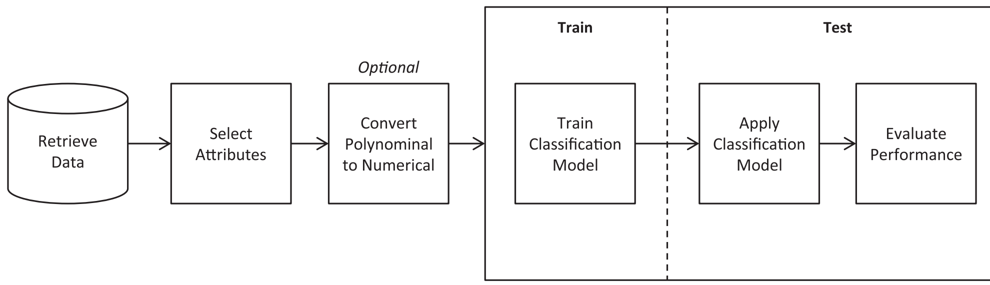

Fig. 9. Classifier development and evaluation.

—For a second classifier, we consider a *neural network* in the form of a multilayer perceptron with a single hidden layer and a sigmoid activation function [Rogers et al. 1999; Wilson and Sharda 1994]. Depending on the specific sigmoid used, this type of neural network corresponds to a "stacking" of logistic regression operators. In each node, the weighted sum of continuous node outputs from the previous layer (or the predictor variables) is the argument of the activation function, where the coefficients β_i correspond to the weights of the edges connecting nodes between layers [Dreiseitl and Ohno-Machado 2002]. The comparison of neural networks with logistic regression is common, and previous work has shown that the more complex model structure of neural networks often outperforms the latter in classification tasks [Chiang et al. 2006; Kim 2006; Swiderski et al. 2012].

—*Decision trees* as the third classifier separate cases in a sample according to simple one-dimensional thresholds applied to predictor variables [Cohen 1995]. The resulting tree structures allow the user to reconstruct decisions made by the classification model, which seems useful against the background of the dynamic advancements in the area of RFID technology. Moreover, they are very tolerant of missing values and irrelevant attributes and can handle both categorical and numerical data.

4.4.3 Rule-Based Classifier. In addition, we consider a classifier based on decision rules. While the mere use of decision rules as an alternative to decision trees is not unusual, we employ an advanced procedure for classifier development that differs in two respects from the previously described standard classifiers:

—On the one hand, we use an iterative training algorithm that repeatedly generates decision trees, analyzes the classification results for each rule (i.e., each path from root to leaf), and discards rules that do not show acceptable classification rates. The procedure is then repeated for the remaining data subsets that are not covered by one of the selected rules. As a result of these iterations, we obtain a classification algorithm containing a set of rules that cover the entire sample.

—On the other hand, we wish to investigate to what extent the classification performance may benefit from the use of so-called artificial attributes. Artificial attributes are generated from the domain attributes using unary or binary mathematical operations. These attributes are then used in the training phase in addition to the other attribute types. Artificial attributes are a common approach to improving the performance of decision trees [Kamath 2009].

Our rationale behind the rule-based approach with the previous two enhancements is to generate classifiers that are as comprehensive as decision trees but offer a better classification performance. The individual steps in training a rule-based classifier are as follows (Figure 10):

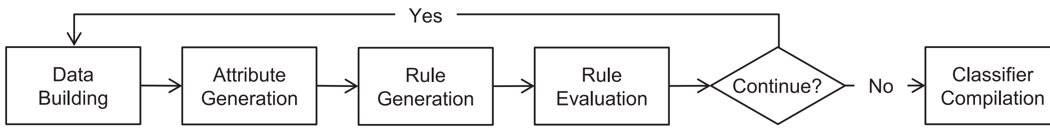

Fig. 10. Generation of decision rules.

(1) *Data Building*. We start by separating the available data into training and test sets. If this is not the first iteration, all pallets that have already been covered by a rule are removed from the dataset.
(2) *Artificial Attribute Generation*. In the second step, artificial attributes are generated for the dataset built in the previous step. Because this sample changes after each iteration, previously created artificial attributes may no longer be suitable for the new data. Accordingly, new artificial attributes are created that allow for a better classification of the current subsample data.
(3) *Rule Generation*. Using all domain and artificial attributes, a decision tree is generated. Each leaf can be interpreted as a rule composed of a sequence of attribute tests.
(4) *Rule Evaluation*. To evaluate the rules, two common quality measures in data mining, *Confidence* and *Support*, are used. Let R be a classification rule and N be the total number of observed sample pallets used to build the classification model. The Confidence and Support of the rule are defined as follows:

$$Confidence(R) := \frac{Number\ of\ Tags\ classified\ correctly\ by\ R}{Number\ of\ Tags\ classified\ by\ R}$$
$$Support(R) := \frac{Number\ of\ Tags\ classified\ correctly\ by\ R}{N}.$$

The Support of a rule describes its generality. If we use 10,000 samples and the rule covers 1,000 of them with 950 classified correctly, the Support value is 950/10,000 = 9.5%, and the Confidence value is $950/1,000 = 95\%$. Confidence tells us about the classification quality of a rule, meaning that the rule would correctly classify 95% of all tags covered by this rule. When building a classification model, we try to maximize Confidence while keeping the Support of all rules over 5%. Below this threshold, we run the risk of overfitting if a rule cannot generalize. Eventually, every rule with a Confidence over 99% and a Support over 5% is marked as "useful" and becomes part of the classifier.
(5) *Abort Criterion*. Generating new rules and creating new artificial attributes are repeated until one of the following two criteria is fulfilled:
(a) *All pallets in our dataset are covered by a rule.* In this case, no new rules can be created. The rule generation procedure described previously ensures that exactly one rule covers any pallet detected in the future.
(b) *During the last evaluation round, no rule has been marked as "useful."* This happens if the remaining pallets in our dataset cannot be separated for any reason. In this case, a new rule is created that classifies all remaining pallets based on a majority decision. If most pallets are static, then all pallets are classified as "static"; otherwise, all are classified as "moved."

If at least one rule is useful and pallets remain to be classified, the procedure returns to the data-building step. Here, tags covered by one of the new rules are removed from the original dataset, and the iteration restarts.

(6) *Classifier Compilation*. All rules marked as "useful" in the rule evaluation step pose the main component of the classifier. Making a decision on an arbitrary pallet T that was detected during a gathering cycle is now simple, as follows:
   (a) Calculate all domain and artificial attributes used in any rule.
   (b) Find the rule that covers T. The corresponding rule is the one where all attribute tests can be answered with "yes." Let R be the rule that covers T.
   (c) Return the classification result (i.e., "static" or "moved") characterizing T as defined by R.

5. EMPIRICAL EVALUATION

5.1. Data Collection

The construction of a classification model requires a massive set of sample data for which the class labels are already known. In the context of the present study, low-level RFID reader data must be available, and it must be known whether specific reads belong to moved or static pallets. The dataset that we used for this purpose was collected over the course of 30 weeks at one of METRO Group's distribution centers. The center sees between 3,500 and 8,000 pallet movements a day, and all 87 shipment dock doors have been equipped with RFID portals to automatically register any outgoing pallets.

To obtain the required dataset, METRO employees were assigned to accompany the warehouse people and monitor the loading of pallets from the distribution center into containers. Their task was to track which of the pallets that the reader had recognized during the loading process had actually been moved through the outgoing goods RFID portal, as well as those pallets that were present in the reading field of the portal antenna only by accident.

In total, 92,857 pallets were monitored, corresponding to a total of 2,664,621 individual tag detections. Among these, 74,432 were classified as "static"—that is, 80.2% of pallet identifications were false positives—and the remaining 18,425 were classified as "moved." It can be assumed that this dataset is large enough to cover any possible process variants, allows for greater insights than any computer simulation or laboratory experiment, and thus provides an appropriate foundation for evaluating the proposed classification models. All of the pallets loaded onto trucks during the data collection period were also identified by the RFID system, which indicates an excellent detection rate. However, the figures clearly indicate that this result comes at the cost of a very high number of undesired false positives.

To ensure the high quality of the data used for the classification model building and testing, any data that could negatively affect the quality of the model were filtered out in advance to ensure a smooth dataset. Two different types of monitored data were identified as problematic. First, RFID transponder types may differ in impedance and sensitivity. For this reason, we concentrated only on the RFID readings of "Monza 3" tags, which accounted for 83,816 of the 92,857 monitored pallets. Second, some issues in the monitoring of pallet movements were identified, indicating that an employee possibly made a mistake by assigning the wrong class to a pallet. In some cases, it was definitely known that a specific pallet has been shipped, for example, because one of the destination stores confirmed its arrival; however, the recorded data state that it was always marked as "static." In other cases, tags were marked multiple times as "moved" in different gathering cycles. As a consequence, 1,206 tags were removed from the sample in cases of likely employee mistakes, leaving 82,610 tags for the actual classification study.

The descriptive statistics given in Figures 11 and 12 provide an overview of the number of tag events and tag occurrences in our dataset. First, we see that the number of different tags detected in a gathering cycle may vary between one and 18 (Figure 11).

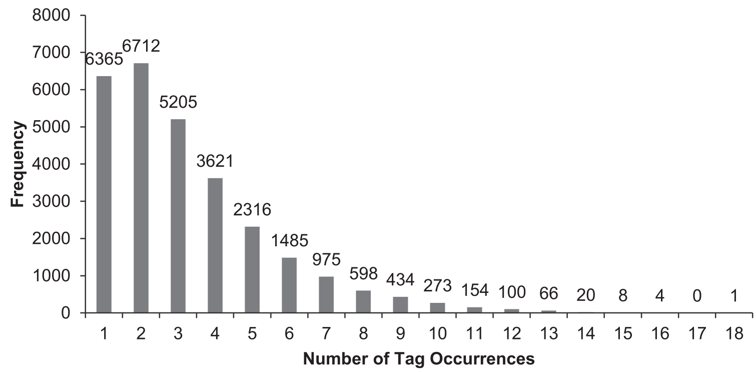

Fig. 11. Number of tag occurrences per gathering cycle.

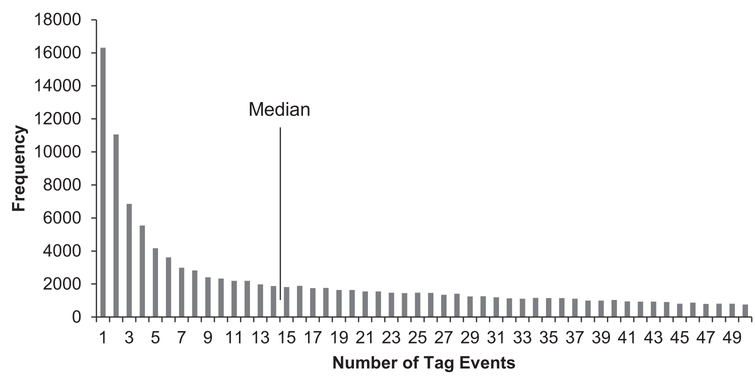

Fig. 12. Number of tag events per tag and cycle.

In approximately 85% of all cycles, between one and five tags are read. The number of true positives per cycle is not necessarily one because not only pallets but also large stacks of large objects were loaded into trucks from time to time. That is, more than one tag occurrence per cycle does not necessarily indicate the presence of false positives. Second, Figure 12 indicates that the number of events per tag in a cycle shows high variance. While the median is located at 16 tag events, this number may increase to 400 in some rare cases (the x-axis has been limited to 50 events per cycle).

The sample dataset can be separated into three major groups. These groups correspond to the three different RFID portal types installed at the distribution center. The data collected by the satellite and transition portals can be further divided into several subsets, depending on the involved antenna types. As depicted in Table VII, some datasets show very high fractions of false positives. For example, the figures show that for the satellite portals, some pallets were detected by the DC antennae and the truck antennae, but not the main antennae (“Case 3”). Within this specific dataset (SAT_DC_TRUCK), 100% of the pallet identifications were false positives. This example indicates that it is, to some extent, possible to achieve a first filtering of false positives based solely on the antenna information generated by the reader hardware without any deeper analysis of the data.

However, the figures also show that the hardware configuration alone is not sufficient to cleanse the complete dataset from false positives. For example, in the case of the transition portals, pallets that were identified by both types of antennae (TRA_BOTH) were false positives in 64.81% of all cases, whereas the remaining pallets

Table VII. Data Sets

Portal Type	Involved Antennae	Data Set	False Positives
Standard portals	Main antennae	STD_COMPLETE	75.47%
Satellite portals (all cases)	Union of cases 1–7	SAT_COMPLETE	86.66%
Case 1	*Main, Truck, DC*	*SAT_ALL*	*96.54%*
Case 2	*DC, Main*	*SAT_DC_MAIN*	*98.49%*
Case 3	*DC, Truck*	*SAT_DC_TRUCK*	*100.00%*
Case 4	*DC*	*SAT_DC_ONLY*	*100.00%*
Case 5	*Main, Truck*	*SAT_MAIN_TRUCK*	*59.70%*
Case 6	*Main*	*SAT_MAIN_ONLY*	*79.67%*
Case 7	*Truck*	*SAT_TRUCK_ONLY*	*99.79%*
Transition portals (all cases)	Union of cases 1–3	TRA_COMPLETE	90.19%
Case 1	*DC*	*TRA_DC_ONLY*	*99.52%*
Case 2	*Truck*	*TRA_TRUCK_ONLY*	*99.29%*
Case 3	*DC, Truck*	*TRA_BOTH*	*64.81%*

Table VIII. Descriptive Statistics on the Critical Datasets

Data Set	Moved Tags	Static Tags	Total Tags	False Positives
STD_COMPLETE	13,245	40,743	53,988	75.47%
SAT_MAIN_ONLY	656	2,571	3,227	79.67%
SAT_MAIN_TRUCK	1,282	1,899	3,181	59.70%
TRA_BOTH	1,299	2,392	3,691	64.81%

were identified correctly. The same issue can be observed for the other two types of portals (STD_COMPLETE, SAT_MAIN_ONLY, SAT_MAIN_TRUCK), which supports our argument that a more sophisticated filtering procedure on the software level is needed to ensure high data quality. In the following, we evaluate the performance of three classifiers using the different RFID-based attributes outlined in the previous section. We concentrate on the four critical subsets of our data, which show a balanced distribution of false and true positives (Table VIII).

5.2. Classification Performance

The performance of a classifier is usually summarized in the form of a so-called confusion matrix. In the case of our classification problem with two classes, the confusion matrix consists of the following four elements:

—*True Positives* (TP) denotes the number of moved pallets that were correctly classified as "moved."
—*False Positives* (FP) denotes the number of static pallets that were wrongly classified as "moved."
—*True Negatives* (TN) denotes the number of static pallets that were correctly classified as "static."
—*False Negatives* (FN) denotes the number of moved pallets that were wrongly classified as "static."

From these results, a number of performance metrics can be calculated, with the accuracy being the most important. The accuracy of a classification model is defined as the number of correct classifications relative to the total size of the dataset:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

However, the accuracy statistic might not necessarily be the most appropriate measure because it weights false positives and false negatives equally. In fact, the economic

Table IX. Performance Measures for Decision Stumps
(A = Accuracy, P = Precision, R = Recall)

Type	Attribute	A	P	R
RSSI	RSSIMin	75.47%	–	0%
	RSSIMax	95.83%	92.99%	89.78%
	RSSIDiff	93.87%	91.86%	82.32%
	RSSIMean	92.39%	84.02%	85.21%
	RSSIStDev	94.62%	92.40%	85.06%
	RSSICoV	95.23%	91.87%	88.37%
SinceStart	ReadFirst	75.47%	–	0%
	ReadLast	75.47%	–	0%
	ReadDiff	75.47%	–	0%
Antenna	Count1	78.84%	54.99%	75.73%
	Count2	77.17%	52.67%	68.31%
	Count3	75.47%	–	0%
	Count4	75.47%	–	0%
	AntCount	82.94%	70.57%	52.25%
	CountMain	75.47%	–	0%

consequence of a false positive detection may be completely different from a false negative. If a static pallet is wrongly classified as "moved," the warehouse management system assumes that it has been loaded into the trailer and sends incomplete shipments to the respective store. In the worst case, stock-outs and lost sales may be the consequence. In contrast, false negatives lead to surplus inventory in the store. While both cases are evidently undesirable, the latter may be considered less critical from a practitioner's perspective than the former. Performance measures that make the two issues quantifiable are (1) precision and (2) recall:

$$Precision = \frac{TP}{TP+FP}$$
$$Recall = \frac{TP}{TP+FN}.$$

Precision sets the focus on the moved pallets and denotes the corresponding fraction of correctly classified pallets. Hence, this statistic indicates the risk of incomplete shipments. Recall, in turn, considers only those pallets that were classified as "moved" and denotes the corresponding fraction of correct classifications. This statistic indicates the risk of shipping pallets that were not ordered by the respective store.

5.3. Results

Our performance evaluation started with the decision stumps, each of which was based on only one of the previously described domain attributes. We considered the STD_COMPLETE dataset because it poses the base case with a standard hardware configuration. To minimize the impact of a potential bias in the data, we applied 10-fold cross-validation. The corresponding results in Table IX allow for drawing a number of important conclusions.

We see that even a single-attribute classifier may achieve a rather high accuracy of up to 95.83%. With the exception of the $RSSI_{Min}$ attribute, all RSSI-based decision stumps achieve results greater than 90%. This outcome supports our initial assumption regarding the value of the signal strength information. We also see that the other decision stumps using timestamp and antenna information achieve much worse results, with AntCount showing the highest accuracy (82.94%). Most classifiers based on the SinceStart and Antenna attributes were unable to distinguish between true and false positives at all. In fact, all tag occurrences were classified as “static,” and the accuracy equaled the total fraction of false positives in the sample. In these cases, the precision measure cannot be calculated, and the recall measure equals 0%. The classifier based on the CountMain attribute corresponds to the use of the sliding-window approach proposed in prior studies. Similarly, the AntCount attribute corresponds to the idea of using more than one reader. The results for both classifiers support our argument that the two approaches found in the literature may not be able to achieve satisfactory performances under real-world conditions.

In a second step, we compared these results to the three standard classifiers. For this purpose, we considered seven groups of attributes, which again allows us to compare the value of the different information types associated with a tag event to the classification task:

—*RSSI, SinceStart, and Antenna*: Domain attributes
—*Logical*: Logical attributes based on the antenna information
—*Tag Occurrence*: The combination of RSSI, SinceStart, Antenna, and Logical
—*TimeSeries*: Attributes based on the similarity of the tag event sequence and a reference time series
—*All*: The combination of Tag Occurrence and TimeSeries

The three types of classifiers were separately trained using each of the four datasets. For each combination of classifier and dataset, the train and test procedure was conducted for the seven groups of attributes with the exception of the logical attributes, which are only applicable for SAT_MAIN_TRUCK and TRA_BOTH. Again, we applied 10-fold cross validation. As a result, we generated 78 confusion matrices. We report the results for all combinations of classifier, dataset, and attribute group in Table X. The measures indicate that the best classification performance depends on the use of RSSI- and time-series-based attributes. In contrast, the sole use of SinceStart, Antenna, and Logical attributes does not allow for the same level of accuracy. This finding indicates that the full spectrum of information generated by the reader hardware must be used to achieve high classification performance.

We also see that the results show a slight margin for the neural network classifier, followed by decision trees. This performance difference may be expected, given the varying complexity of the three classifiers, and is consistent with the performances reported in several other data mining studies. Moreover, we see that the classifiers achieved the best results for the dataset generated by the satellite portal, which can be attributed to the fact that this portal type with eight antennae provided the richest information. In contrast, the worst performance can be observed for the transit portals, which were not equipped with any main antennae.

We next evaluated the performance of our rule-based classifier using artificial attributes. The results given in Table XI show that the custom-made model outperformed even the best-performing standard classifier. For the standard and satellite portals, we found that a combination of tag-occurrence- and tag-event-level data led to the best performance. For the transition portals, the classification model based only on tag-occurrence data achieved the best results. We thus conclude that the combination of an iterative algorithm for the generation of decision rules with artificial attributes leads to a substantial overall improvement in the achievable classification performance. Though the difference between 95.83% accuracy in the case of the RSSI_{Max} decision stump for the STD_COMPLETE subset and 98.00% in the case of the rule-based classifier may seem small, it should be noted that this improvement corresponds to a relative error reduction of more than 52%.

Any classification model must show reliable performance over time because a significant variation in classification performance is unacceptable in production use. For this

Table X. Comparison of Standard Classifiers (A = Accuracy, P = Precision, R = Recall)

		Logistic Regression			Neural Network			Decision Tree		
		A	P	R	A	P	R	A	P	R
STD_COMPLETE	RSSI	95.91%	95.04%	87.90%	96.43%	92.86%	92.56%	95.80%	91.21%	91.73%
	SinceStart	75.47%	0.00%	0.00%	75.62%	53.93%	4.25%	75.47%	0.00%	0.00%
	Antenna	78.04%	79.73%	14.04%	86.49%	77.21%	63.76%	84.54%	69.76%	65.25%
	Logical	–	–	–	–	–	–	–	–	–
	Tag Occurrence	95.81%	95.51%	87.02%	97.32%	95.02%	93.99%	96.54%	93.85%	91.91%
	Time-Series	95.55%	91.19%	90.59%	96.27%	93.10%	91.58%	95.59%	91.81%	90.05%
	All	96.38%	94.85%	90.14%	97.06%	94.69%	93.25%	96.65%	94.59%	91.57%
SAT_MAIN_ONLY	RSSI	97.55%	94.18%	93.75%	97.64%	95.03%	93.29%	97.21%	93.27%	92.99%
	SinceStart	79.67%	0.00%	0.00%	79.39%	47.87%	15.40%	79.58%	20.00%	0.15%
	Antenna	87.60%	67.68%	74.70%	89.80%	81.02%	65.09%	89.90%	83.95%	62.20%
	Logical	–	–	–	–	–	–	–	–	–
	Tag Occurrence	97.95%	95.11%	94.82%	98.14%	95.57%	95.27%	97.33%	95.24%	91.46%
	Time-Series	97.40%	94.00%	93.14%	97.27%	94.10%	92.38%	96.87%	93.56%	90.85%
	All	97.43%	96.74%	90.40%	97.77%	94.11%	94.97%	97.03%	92.42%	92.99%
SAT_MAIN_TRUCK	RSSI	95.57%	96.80%	92.04%	95.72%	94.42%	95.01%	94.22%	90.43%	95.79%
	SinceStart	73.03%	72.70%	52.96%	86.92%	83.31%	84.48%	80.89%	71.28%	88.07%
	Antenna	85.00%	78.89%	85.73%	90.10%	87.63%	87.83%	88.02%	84.57%	85.96%
	Logical	80.57%	78.87%	70.75%	80.73%	73.61%	81.36%	80.26%	69.79%	89.94%
	Tag Occurrence	97.01%	97.67%	94.85%	97.58%	96.60%	97.43%	96.76%	95.38%	96.65%
	Time-Series	91.13%	84.01%	96.33%	93.21%	91.77%	91.34%	93.93%	92.71%	92.20%
	All	97.14%	97.22%	95.63%	97.67%	96.75%	97.50%	96.98%	95.83%	96.72%
TRA_BOTH	RSSI	89.81%	83.91%	87.91%	91.63%	88.19%	87.99%	88.27%	80.67%	87.68%
	SinceStart	67.57%	52.27%	90.38%	84.69%	80.33%	74.83%	76.65%	65.08%	72.59%
	Antenna	77.40%	70.13%	62.36%	85.61%	79.45%	79.75%	77.73%	81.18%	47.81%
	Logical	69.38%	53.82%	91.69%	81.77%	73.60%	75.13%	78.52%	65.17%	83.68%
	Tag Occurrence	91.33%	83.97%	93.15%	93.12%	89.92%	90.61%	92.68%	89.67%	89.53%
	Time-Series	88.30%	80.34%	88.38%	88.73%	84.15%	83.76%	88.59%	82.81%	85.30%
	All	92.22%	85.79%	93.38%	92.79%	89.70%	89.84%	92.20%	88.09%	89.99%

Table XI. Performance Measures for the Rule-Based Classifier
(A = Accuracy, P = Precision, R = Recall)

		A	P	R
STD_COMPLETE	Tag Occurrence	97.57%	95.38%	94.66%
	Time-Series	97.17%	93.76%	94.78%
	All	98.00%	94.54%	97.49%
SAT_MAIN_ONLY	Tag Occurrence	98.82%	95.71%	98.63%
	Time-Series	98.98%	96.70%	98.32%
	All	99.10%	97.14%	98.48%
SAT_MAIN_TRUCK	Tag Occurrence	97.77%	95.36%	99.30%
	Time-Series	97.45%	96.23%	97.50%
	All	97.71%	98.01%	96.26%
TRA_BOTH	Tag Occurrence	96.15%	95.30%	93.69%
	Time-Series	95.04%	91.52%	94.69%
	All	93.96%	89.10%	94.38%

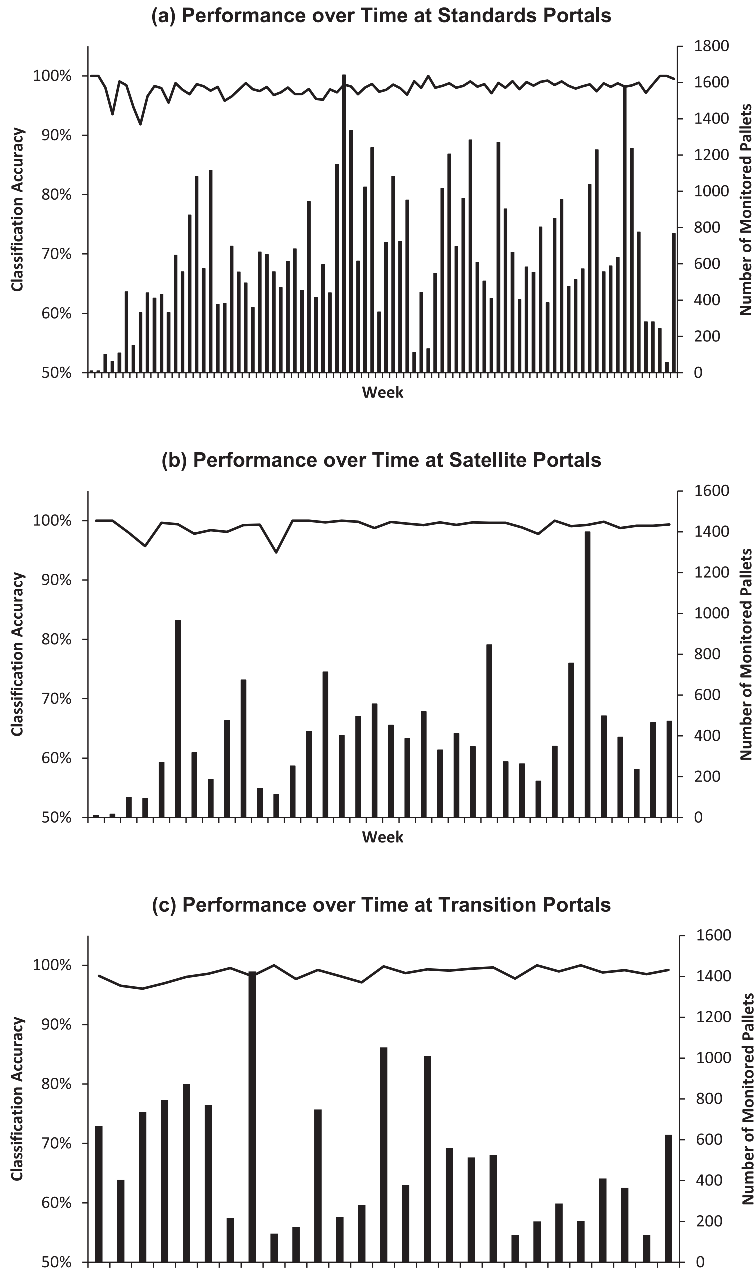

Fig. 13. Performance of the rule-based classifier over time (30 weeks).

reason, the best algorithms (i.e., the combined approach for the standard and satellite portals and the tag-occurrence approach for the transition portals) were again applied to the collected data. The performance results depicted in Figure 13 were averaged over individual days so the occasional performance outliers could be more easily identified. In addition, the number of pallets monitored on each particular day is also included.

The data collection periods for the three portal types differ in the length and number of collected RFID reads. It can be seen that implementing the approaches presented here on the different portal types led in each case to reliable and robust classification performance over time. In only a few cases did the performance show a notable drop of more than 2 to 3 percentage points. However, in almost all of the cases, the sample size collected on these days was very small, so they may be considered outliers. In total, the average classification accuracy per day achieved by the classification models was 97.30% for the standard portals, 99.06% for the satellite portals, and 98.62% for the transition portals.

6. SUMMARY AND CONCLUSIONS

Interest in the applications of RFID in the supply chain has been growing steadily over the past 10 years. Today, several examples of RFID applications can be found in retail and other industries. The data generated by the underlying hardware infrastructures pose an unprecedented opportunity to gain insights into the reality of a company's physical operations on a fine-grained level of detail. However, the noisy nature of RFID data streams hinders their immediate processing in transactional or analytical information systems. As the present study has shown, the phenomenon of false-positive reads poses a nontrivial challenge for the necessary RFID data cleansing, but virtually no satisfactory procedures for dealing with this issue have been presented in the literature.

To fill this research gap, we investigated concepts of RFID filtering based on the foundation of data mining techniques. We developed several attributes on the level of tag occurrences and individual tag events, which allow for the construction of different types of classification models. These were then used to extract information on physical events from large amounts of RFID data. For evaluation purposes, we considered the example of RFID-equipped logistical assets using a large sample of low-level RFID data gathered under real-world conditions. Though a first filtering of the data was already possible depending on the antenna configuration, some critical subsamples clearly highlighted the need for more complex filtering procedures.

For this purpose, we evaluated three different standard classifiers. Their results yielded high classification performances compared with single-attribute classifiers (i.e., decision stumps), thus supporting our assumption that the utilization of the full spectrum of data generated by the reader hardware leads to superior performance. In particular, we could show that the approaches proposed in prior research based on timestamp and antenna information do not allow for acceptable levels of classification accuracy. In addition, we presented a procedure for generating decision rules that goes beyond the concepts of the standard classifier in the use of artificial attributes and an iterative training procedure. The rule-based classifier achieved better accuracies than did any other model, which highlights the performance potential of custom-made classifiers. A further advantage may be seen in the easy interpretability of decision rules compared to other advanced classification models (e.g., neural networks).

To fill this research gap, we investigated concepts of RFID filtering based on the foundation of data mining techniques. We developed several attributes on the level of tag occurrences and individual tag events, which allow for the construction of different types of classification models. These were then used to extract information on physical events from large amounts of RFID data. For evaluation purposes, we considered the example of RFID-equipped logistical assets using a large sample of low-level RFID data gathered under real-world conditions. Though a first filtering of the data was already possible depending on the antenna configuration, some critical subsamples clearly highlighted the need for more complex filtering procedures. For this purpose, we evaluated three different standard classifiers. Their results yielded high classification performances compared with single-attribute classifiers (i.e., decision stumps), thus supporting our assumption that the utilization of the full spectrum of data generated by the reader hardware leads to superior performance. In particular, we could show that the approaches proposed in prior research based on timestamp and antenna information do not allow for acceptable levels of classification accuracy. In addition, we presented a procedure for generating decision rules that goes beyond the concepts of the standard classifier in the use of artificial attributes and an iterative training procedure. The rule-based classifier achieved better accuracies than did any other model, which highlights the performance potential of custom-made classifiers. A further advantage may be seen in the easy interpretability of decision rules compared to other advanced classification models (e.g., neural networks).

In light of our findings, we see opportunities for future research in various directions. First, new fields of application should be investigated to support the transferability of our approach. Though we considered a scenario with nonoverlapping gathering cycles, our concepts may also be transferred with no modifications to the processing of continuous data streams. Examples include RFID-based self-checkouts, the detection of misplacements on the sales floor, and production lot tracking in complex manufacturing systems. Second, our results might be extended to use the ancillary conditions typical for some specific application settings. The picking process in high-rack storage areas poses an example, where it can be assumed that all logistical units detected must be false positives except for one. Third, the concepts presented here might become a promising foundation for research on the mining of other forms of sensor data beyond RFID. The long-term emergence of a so-called Internet of Things will successively lead to the deployment of many other sensor technologies that organizations might want to leverage. Last but not least, more research will be required to develop a better understanding of the value provided by these novel sources of information to the firm, for example, in operations, marketing, or innovation management.

ACKNOWLEDGMENTS

The authors would like to thank METRO Group and IBM Germany for making their RFID dataset available. In particular, we are deeply indebted to Jens Kungl (METRO Group) and Frank Schmid (IBM) for their support, which made the present study possible in the first place.

REFERENCES

Rebecca Angeles. 2005. RFID Technologies: Supply chain applications and implementation issues. *Information Systems Management* 22, 1, 51–65.

Henning Baars and Xuanpu Sun. 2009. Multidimensional analysis of RFID data in logistics. In *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS'09)*.

Yijian Bai, Fusheng Wang, and Peiya Liu. 2006. Efficiently filtering RFID data streams. In *Proceedings of the1st International Very Large Data Base Workshop on Clean Databases (CleanDB'06)*.

Indranil Bose and Raktim Pal. 2005. Auto-ID: Managing anything, anywhere, anytime in the supply chain. *Communications of the ACM* 48, 8, 100–106.

James Brusey, Christian Floerkemeier, Mark Harrison, and Martyn Fletcher. 2003. Reasoning about uncertainty in identification with RFID. In *Proceedings of the Workshop on Reasoning with Uncertainty in Robotics at the International Joint Conferences on Artificial Intelligence (IJCAC'03)*.

Sellitto Carmine, Stephen Burgess, and Paul Hawking. 2007. Information quality attributes associated with RFID-derived benefits in the retail supply Chain. *International Journal of Retail & Distribution Management* 35, 1, 69–87.

Wei-Yu Kevin Chiang, Dongsong Zhang, and Lina Zhou. 2006. Predicting and explaining patronage behavior toward web and traditional stores using neural networks: A comparative analysis with logistic regression. *Decision Support Systems* 41, 2, 514–531.

Ronald Christensen. 1997. *Log-Linear Models and Logistic Regression*. Springer, New York.

William W. Cohen. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning.*

Dursun Delen, Bill C. Hardgrave, and Ramesh Sharda. 2007. RFID for better supply-chain management through enhanced information visibility. *Production and Operations Management* 16, 5, 613–624.

Stephan Dreiseitl and Lucila Ohno-Machado. 2002. Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics* 35, 5–6 (2002), 352–359.

Klaus Finkenzeller. 2010. *RFID Handbook*. John Wiley & Sons, New York.

Bill C. Hardgrave and Robert Miller. 2008. RFID in the retail supply chain: Issues and opportunities. In *RFID Technology and Applications,* S. Miles, S. Sarma and J. Williams (Eds.). Cambridge University Press, Cambridge, MA, 113–120.

Bing Jiang, Kenneth P. Fishkin, Sumit Roy, and Matthai Philipose. 2006. I sense a disturbance in the force: Unobtrusive long-range detection of passive RFID tag motion. *IEEE Transactions on Instrumentation and Measurement* 55, 1, 187–196.

Erick C. Jones and Christopher A. Chung. 2008. *RFID in Logistics: A Practical Introduction.* CRC Press, Boca Raton, FL.

Chandrika Kamath. 2009. *Scientific Data Mining: A Practical Perspective.* Society for Industrial and Applied Mathematics, Philadelphia, PA.

Leonard Kaufman and Peter J. Rousseeuw. 2005. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons Inc., Hoboken, NJ.

Melody Y. Kiang. 2003. A comparative assessment of classification methods. *Decision Support Systems* 35, 4, 441–454.

YongSeog Kim. 2006. Toward a successful CRM: Variable selection, sampling, and ensemble. *Decision Support Systems* 41, 2, 542–553.

Hau Lee and Özalp Özer. 2007. Unlocking the value of RFID. *Production and Operations Management* 16, 1, 40–64.

Wei-Pang Liao, Tom M. Y. Lin, and Shu-Hsien Liao. 2011. Contributions to radio frequency identification (RFID) research: An assessment of SCI-, SSCI-indexed papers from 2004 to 2008. *Decision Support Systems* 50, 2, 548–556.

Claudia Loebbecke and Jonathan W. Palmer. 2006. RFID in the fashion industry: Kaufhof Department Stores AG and Gerry Weber International AG, Fashion Manufacturer. *MIS Quarterly Executive* 5, 2, 15–25.

James MacQueen. 1967. Some methods of classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. 281–297.

E. W. T. Ngai, Karen K. L. Moon, Frederick J. Riggins, and Candace Y. Yi. 2008. RFID research: An academic literature review (1995–2005) and future research directions. *International Journal of Production Economics* 112, 2, 510–520.

Katariina Penttilä, Mikko Keskilammi, Lauri Sydänheimo, and Markku Kivikoski. 2006. Radio frequency technology for automated manufacturing and logistics control. Part 2: RFID antenna utilisation in industrial applications. *International Journal of Advanced Manufacturing Technology* 31, 2, 116–124.

Seth Rogers, Pat Langley, and Christopher Wilson. 1999. Mining GPS data to augment road models. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 104–113.

Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 1, 43–49.

Aysegul Sarac, Nabil Absi, and Stéphane Dauzère-Pérès. 2010. A literature review on the impact of RFID technologies on supply chain management. *International Journal of Production Economics* 128, 1, 77–95.

Steven Shepard. 2005. *Radio Frequency Identification*. McGraw-Hill, New York.

Jay Singh, Eric Olsen, Keith Vorst, and K. Tripp. 2009. RFID tag readability issues with palletized loads of consumer goods. *Packaging Technology and Science* 22, 8, 431–441.

Bartosz Swiderskia, Jarosław Kureka, and Stanislaw Osowski. 2012. Multistage classification by using logistic regression and neural networks for assessment of financial condition of company. *Decision Support Systems* 52, 2, 539–547.

May Tajima. 2007. Strategic value of RFID in supply chain management. *Journal of Purchasing & Supply Management* 13, 4, 261–273.

Frédéric Thiesse, Jasser Al-Kassab, and Elgar Fleisch. 2009a. Understanding the value of integrated RFID systems: A case study from apparel retail. *European Journal of Information Systems* 18, 6, 592–614.

Frédéric Thiesse, Christian Floerkemeier, Mark Harrison, Florian Michahelles, and Christof Roduner. 2009b. Technology, standards, and real-world deployments of the EPC network. *IEEE Internet Computing* 13, 2, 36–42.

Yu-Ju Tu and Selwyn Piramuthu. 2008. Reducing false reads in RFID-embedded supply chains. *Journal of Theoretical and Applied Electronic Commerce Research* 3, 2, 60–70.

Roy Want. 2004. The magic of RFID. *ACM Queue* 2, 7, 41–48.

Rick L. Wilson and Ramesh Sharda. 1994. Bankruptcy prediction using neural networks. *Decision Support Systems* 11, 545–557.

Asif Zaheeruddin and Mandviwalla Munir. 2005. Integrating the supply chain with RFID: A technical and business analysis. *Communications of the AIS* 15, 393–427.

Received March 2013; revised March 2014; accepted April 2014