Mining the Internet of Things: Detection of False-Positive RFID Tag Reads using Low-Level Reader Data

D I S S E R T A T I O N
of the University of St. Gallen,
School of Management,
Economics, Law, Social Sciences
and International Affairs
to obtain the title of
Doctor of Philosophy in Management

submitted by

Thorben Keller

from

Germany

Approved on the application of

Prof. Dr. Elgar Fleisch

and

Prof. Dr. Frédéric Thiesse

Dissertation no. 3908

Eigendruck, Aachen 2011

The University of St. Gallen, School of Management, Economics, Law, Social Sciences and International Affairs hereby consents to the printing of the present dissertation, without hereby expressing any opinion on the views herein expressed.

St. Gallen, May 13, 2011

The President:

Prof. Dr. Thomas Bieger

Für meine liebevollen Grosseltern Irmgard und Paul Christensen.

Dedicated to my loving grandparents.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors Prof. Dr. Elgar Fleisch and Prof. Dr. Frédéric Thiesse for their great guidance and assistance throughout the period of this research work.

Furthermore I would like to express my special thanks to Jens Kungl, METRO Groups Head of New Technology Platforms, and Frank Schmid, Senior Managing Consultant at IBM, for their outstanding personal commitment which helped pave the way for this dissertation and the underlying research project.

I would also like to thank my coworkers Nicloas Becker, Volker Honds, Ingo Leiking, Falk Nieder and André Rogge for the valuable input, helpful discussions and all the fun we had in the past 3 years working together at the METRO Group RFID Innovation Center in Neuss, Germany.

Most importantly, this dissertation would not have been possible without the love and patience of my parents Inge & Vitus Keller, my sister Silke Keller and my partner Andrea Strobl.

I thank you all.

June, 2011 Thorben Keller

Abstract

The notion of the "Internet of Things" (IoT) describes the vision of ubiquitous sensor technologies that seamlessly link arbitrary physical objects to their digital counterparts in the network. In recent years, it was particularly the rapid diffusion of Radio Frequency Identification (RFID) in supply chain and distribution center processes that has contributed to the emergence and popularity of the IoT concept. However, technological constraints currently delay fully reliable and productive use of the technology. One major constraint is the problem of false-positive RFID tag reads, i.e., when an RFID transponder is read unintentionally by an RFID reader.

This problem is studied in the context of RFID enabled outgoing goods processing at a distribution center in Unna, Germany. The RFID installation in this METRO Group center was part of the largest operational rollout of the technology in the European retail sector. In the examined scenario it is necessary to distinguish between tracked RFID tagged pallets that are loaded onto trucks and other pallets that are also in range of the reader. If all detected pallets are reported to the warehouse management system as being shipped, the resulting incorrect invoices mean stores have to pay for goods that they neither ordered nor received. It is evident, therefore, that a solution to this problem is a must before a reliable and productive use of RFID technology in distribution center processes is feasible.

Currently, the few conceptual approaches presented in the literature that deal with this problem suffer from fundamental weaknesses. This thesis addresses those weaknesses by means of a machine learning based approach that makes use of the low-level reader data collected when scanning for transponders. For this purpose, meaningful attributes have been identified that help describe characteristics specific to pallets that have been loaded and to pallets that cause false-positives.

This approach was able to minimize the number of possible incorrect shipments to only 1 per 4,500 pallet loadings - a solution that clearly exceeded METRO Group's expectations.

Zusammenfassung

Der Begriff “Internet der Dinge” (IoT) beschreibt die Vision von allgegenwärtigen Sensortechnologien, welche beliebige physikalische Objekte nahtlos mit ihrem digitalen Ebenbild verknüpfen. Insbesondere die wachsende Verbreitung der Radio Frequenz Identifikation (RFID) im Bereich des Lieferketten- und Warenhausmanagements hat während der letzten Jahre zur Entstehung und wachsenden Popularität des IoT Konzeptes beigetragen. Leider steht derzeit einer vollfunktionstüchtigen und produktiven Nutzung der Technologie noch eine Reihe von technologischen Einschränkungen im Wege. Eine der bedeutendsten ist das *Problem der falsch-positiven Tag Lesungen*, also der unerwarteten und unerwünschten Erkennung von RFID Tags durch ein RFID Lesegerät.

Diese Problem wird hier im Zusammenhang mit einem RFID gestützten Warenausgang in einem Zentrallager in Unna, Deutschland, untersucht. Die Installation in diesem Lager der METRO Group war Teil der bis dahin grössten und umfangreichsten Einführung der RFID Technologie auf dem europäischen Kontinent. Im untersuchten Szenario ist es von besonderer Bedeutung zu unterscheiden, ob die RFID getaggten Paletten, welche an den Warenausgangstoren erkannt wurden, auch wirklich in diesem Moment verladen wurden, oder ob es sich lediglich um Paletten handelt, die nur zufällig im Scanbereich des RFID Lesegerätes abgestellt oder bewegt wurden. Falls wirklich alle erkannten Paletten dem Lagerverwaltungssystem als verladen und verschickt gemeldet würden, hätte dies falsche Rechnungen zur Folge und Kunden sollten für Paletten bezahlen die sie weder bestellt, noch jemals erhalten haben. Es ist offensichtlich, dass hier Abhilfe geschaffen werden muss, um einen zuverlässigen und produktiven Einsatz der RFID Technologie im Warenlager zu gewährleisten.

Die wenigen bisher veröffentlichten konzeptionellen Lösungsansätze leiden alle unter einer Reihe von fundamentalen Schwächen, die in dieser Dissertation vermieden werden. Auf Basis verschiedener Methoden des maschinellen Lernens wird ein Algorithmus entwickelt, der die während einer Verladung gesammelten Low-Level Daten untersucht und dann entscheidet, ob die Palette wirklich verladen wurde oder nicht. Zu diesem Zweck wurde eine Reihe von bedeutsamen Eigenschaften dieser Daten ermittelt, die die Identifizierung von falsch-positiven Lesungen überhaupt erst ermöglichen.

Der präsentierte Ansatz ist in der Lage, die Anzahl der möglicherweise inkorrekten Verladungen auf ein Minimum zu reduzieren, welches die Erwartungen der METRO Group bei weitem übertroffen hat.

Contents

List of Figures

List of Tables

List of Algorithms

1. Introduction

1.1. The Internet of Things

A plethora of novel terms including *ubiquitous computing* [Wei91], *pervasive computing* [Sat01] and *things that think* [Ger99] abound these days that herald the appearance of a new paradigm shift in information processing. Common to all of these concepts is the shared vision of a future world of everyday physical objects and places equipped with digital logic, sensors, and networking capabilities, forming what is commonly called the *Internet of Things* (IoT) [Ash09]. The sheer number of IoT devices is expected to surpass the current internet infrastructure (comprising servers, personal computers, and mobile phones) by orders of magnitude [CPB+05]. The drivers behind the ongoing trend towards this vision include the miniaturization of microelectronic components, standardization, and price decline, as well as the various new technologies reaching mass-market maturity (for example, in the area of polymer electronics, energy harvesting, or wireless networks) [MF10]. In recent years, the rapid diffusion of *Radio Frequency Identification* (RFID) in an ever-broader range of application areas has, in particular, contributed to the emergence and popularity of the concept [SBA00, TFH+09].

From a management perspective, the appeal of the IoT vision is grounded in the hope of closing the gap between real-world entities and their digital counterparts in the network, ultimately leading to a state of *real world awareness* [Hei05] of enterprise information systems. Today, the divide between the physical and the digital world is still bridged by manual (e.g., keystrokes) or semi-automatic (e.g., bar code scans) input. In contrast, ubiquitous wireless sensor technologies can provide firms with a continuous stream of fine-granular and timely information on the physical operations, both within the organization and beyond [AL05]. RFID technology specifically offers the benefit of *supply chain visibility* [LO07], which can be expected to reduce inventory holding costs, decrease lead times, improve service levels, and protect customers from counterfeits [BP05, MSW08, Ang05, AM05, BP05, KH02, Sri04]. However, despite the strong interest in the value of RFID for academia and in practice, little research has been performed to identify the necessary procedures to derive meaningful information on business processes from large amounts of raw RFID data. In fact, the vast majority of recent RFID deployments

have been limited to mere process automation, leaving the utilization of the collected data for analytical purposes as a widely untapped application [TAKF09].

1.2. RFID Technology

In recent years, the application of Radio Frequency Identification (RFID) technology in supply chain management has attracted the interest of several industries worldwide [Wyl06]. This development has been strongly driven by standardization activities, cost erosion, and the miniaturization of microelectronic components [TFH+09, Wan04].

The availability of low-cost RFID technology today allows for wider use beyond its traditional niche applications such as for animal tagging and access control. In logistics, RFID competes with the omnipresent bar code in identifying arbitrary physical goods along the entire supply chain [MV01]. Unlike bar codes RFID allows for the unique identification of individual items and bulk readings with no line-of-sight required even under harsh environmental conditions [Fin03].

This gives rise to opportunities to collect the kind of fine-granular, real-time information about physical processes in the supply chain which often cannot be monitored using conventional approaches. The hope among its proponents is that RFID will become the technological enabler of an unprecedented level of supply chain visibility [LO07].

Besides its ever broader diffusion across many industries, RFID has also become a fruitful research topic, not only in electrical engineering and computer science, but also in management research [NMRY08]. In particular, a number of valuable models have recently been developed in the fields of information systems and operations management that not only support designers of RFID-based systems and processes but also explain how RFID can generate business value in organizations. However, the majority of this prior work tends to consider RFID as a "next generation bar code" differing only slightly from its predecessor in its enhanced precision and the timeliness of collected data. This narrow view runs the risk of ignoring the fundamentally different levels of data quality associated with these Auto-ID technologies.

1.2.1. Comparison to Bar Codes

By comparing the characteristics of RFID against object identification using bar codes, some major advantages can be identified for both technologies [MS03, WGPR07]. Furthermore, a completely new product identification scheme, the Electronic Product Code (EPC), allows for a massively increased address space.

1.2.1.1. Advantages of RFID

Since RFID uses radio waves, in contrast to bar codes which are optically read, no line-of-sight is required and tags can be in any orientation and attached to an arbitrary side of the object relative to the antenna. The read range of RFID tags is much higher and multiple tags can be automatically read at once without any human involvement. This allows reduced processing time as the contents of various conveyances can be scanned without opening or unloading them.

Further advantages include a greatly enhanced data storage capacity, the ability to rewrite and program a tag, and improved robustness. In contrast to bar codes, appropriate RFID tags can be used in harsh and dirty environments and they still work when painted over, buried in dirt, or covered with mud and snow or if anything interferes with a clear line-of-sight [Bro07].

1.2.1.2. Advantages of Bar Codes

RFID hardware (e.g., tags and scanners) is still more expensive than the cost effective, well understood, and mature technology of bar codes so ubiquitous throughout many supply chains. Furthermore, bar codes are interoperable on a global level and do not require the regulated frequencies RFID does. Their reliability and manageability is well documented and well proven and unlike RFID tags, bar codes are not affected by materials such as metal foils or liquids. Last but not least, bar codes usually have an alphanumeric identifier printed on them, thus allowing a human to read them if necessary [Bro07].

1.2.1.3. The Electronic Product Code

Maybe the most important advantage of RFID over the traditional bar code is the ability to uniquely identify every single product along with the product class. With this in mind, a new identification scheme for products, the *Electronic Product Code* (EPC) created by the Auto-ID center, has become one of the dominant RFID standards. The Auto-ID center was established in 1999 as a research group consisting of seven leading universities and aimed to develop a "low-cost, open standard RFID infrastructure for Supply Chain Management" [Sri04].

An EPC contains an 8-bit header identifying the EPC version and three sets of information: *EPC Manager*, *Object Class* and *Serial Number*. EPC Manager is a 28-bit value representing a specific manufacturer, the 24-bit Object Class refers to the exact product type and the 36-bit Serial Number corresponds to a specific item of that class. An example of an EPC is shown in Figure 1.1 which also depicts the sizes of the individual information sets.

EPC proponents envision that every product in the supply chain will be RFID tagged with its own and unique Electronic Product Code as the virtual representation of the physical object

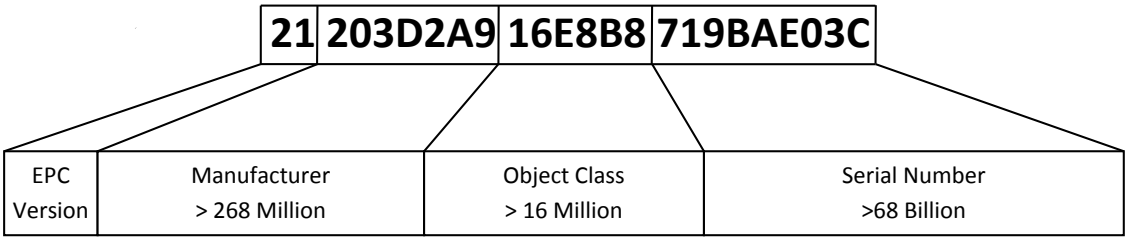


Figure 1.1.: Electronic Product Code Example

it is attached to. Currently however RFID tags are still expensive and there is often little advantage in using tags that may cost the same price as an item itself. So in practice, RFID is often not used on the *item level* but rather on logistical units, for example at the *case* or the *pallet level* [MET07c]. Case or pallet tagging though, does not deliver all of the promised benefits of RFID [AM05].

1.3. RFID in the Supply Chain

RFID can be used in various ways for the optimization of supply chain management and especially for the optimization of processes in distribution centers [DHS07, MM05, MET07c]. A simple retail supply chain is shown in Figure 1.2 [HT06], where producers use the raw materials from their suppliers to create the products which are then distributed to individual retailers where they are ultimately sold to the customer.

Figure 1.2.: Retail Supply Chain

The long-term objective of using RFID in the supply chain is the automatic identification, tracing, and verification of goods along their way from the supplier to the customer. The prerequisite for such an all-embracing process is that the supplier or the producer at the starting point of the supply chain attaches RFID tags to all of their products. This would allow the unique identification of every single product from production to consumption, thus making it much easier, for example, to trace the origin of contaminated food lots or to record the history of a pharmaceutical's buyers and sellers throughout the supply chain [HM08].

The ability to track products all the way along a supply chain provides additional insight into two key metrics (among others): shrinkage rates (theft, damage, etc.) and lead times.

Although RFID is not necessary to determine the amount of shrinkage (it is usually known anyway), it does make it possible to determine exactly where and when in the supply chain the shrinkage occurred [HM08]. Lead times can also now be precisely measured by using the knowledge of how long it took a product to move through the supply chain and the exact time between each of the key read points [DHS07].

1.3.1. RFID enabled Incoming & Outgoing Goods

In order to realize the promised verification and tracking of goods along the supply chain it is necessary to integrate RFID readers at the key points where the goods need to be identified. As can be seen in Figure 1.2 these key points are (among others) the intersections between individual participants, i.e., when goods are handed from one participant to another. For this purpose so-called RFID portals are installed right in front of the incoming and outgoing loading dock doors so that every pallet arriving or leaving has to pass through them.

An example of such an RFID portal is shown in Figure 1.3. As soon as a pallet (⑦) approaches the portal it is automatically recognized by the motion sensor (④) and the RFID reader (①) immediately starts scanning for RFID tags in range of the antennas (⑤). A signal light (③) gives immediate feedback to the warehouseman, telling him, for example, whether a tag has been read or not.

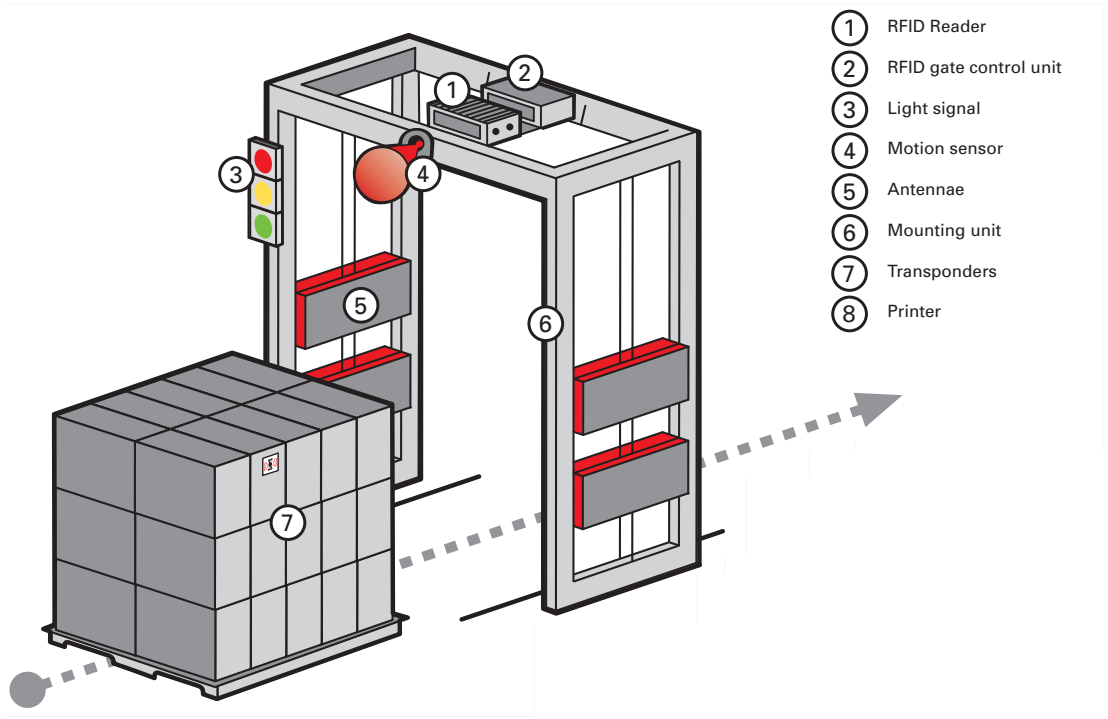


Figure 1.3.: Example of an RFID portal (Source: Metro)

1.3.1.1. Incoming Goods Process

The goods, already tagged with RFID transponders at some preceding step in the supply chain, are automatically detected when unloaded from a truck. As soon as the warehouseman passes through the portal with a pallet on his forklift, the tag together with the corresponding EPC is automatically scanned and the information about the arrival of the pallet is forwarded to the warehouse management system. This contrasts with a bar code based system where each pallet has to be scanned manually by the warehouseman. At this point too, any discrepancies between the expected and the actually received goods are automatically detected.

1.3.1.2. Outgoing Goods Process

RFID also helps ease the shipping process. The warehouseman retrieves a tagged pallet from the staging area and unloads it onto the designated truck. Again the pallet passes through the RFID portal in front of the truck, where it is automatically scanned and the information that the pallet with the corresponding EPC has been loaded is passed on to the warehouse management system. After all the pallets have been loaded onto the truck a request for transportation is sent to a shipper.

1.3.1.3. Benefits

The advantages of RFID enabled incoming and outgoing goods primarily lie in the reduced labor for the warehousemen whose manual work is error-prone and extremely cost intensive [MM05]. Another advantage is that incorrect incoming and outgoing deliveries are recognized immediately and can be dealt with accordingly [MET07a].

1.3.2. Technological Constraints

Although the idea of automatic registration of incoming and outgoing goods seems straight forward, the technology is not completely error free. Generally, the basic principles of radio frequency waves and their limitations have a direct influence upon the readability of RFID tags. These limitations include absorption and reflection of radio waves [JC08] that lead to unexpected read events and ultimately to the problem of *false-positive* and *false-negative* RFID tag reads. The impact of the surrounding environment on the readability has also been studied, for example by [PDG06, GDHK06, DW05].

1.3.2.1. False-Negative RFID Tag Reads

A *false-negative* RFID tag read is when an RFID tag in range of the antennas is not recognized by the reader at all. This corresponds to a pallet that was moved through the RFID portal and loaded onto the truck without being scanned. The reasons for this are manifold. For example, the types of product on the pallet have a significant influence on the readability of RFID tags because water and any other liquids (e.g., shampoo) absorb radio waves and thus greatly reduce the read range [SOVT09]. Other reasons include defective or hard to read RFID tags. To overcome this problem multiple antennas are often installed to increase the chances of reading a tag. This however can lead to another problem, namely the mutual elimination of radio waves due to interference effects [PKSK06].

1.3.2.2. False-Positive RFID Tag Reads

In contrast, the term *false-positive* RFID tag read has two different meanings. On the one hand, the phenomenon is similar to the false-negatives because physical effects influence the readability of the RFID tags. Any metallic material in goods or packages (for example, metal foils and metal ink), the truck itself or anything else within the range of the antennas can unintentionally significantly extend the read range of the antennas. As a consequence tags assumed to be clearly out of range are unexpectedly read by the reader.

On the other hand, the term *false-positive* refers to tags that have been read and are clearly present within the read range but, for whatever reason, should not be read.

1.3.3. Statement of the Problem

The problem of *false-positive RFID tag reads* can easily be illustrated by the scenario of an RFID-enabled outgoing goods process. Because the portal antennas are not directional and have a read range of several meters, not only is the pallet moving through the portal detected, but any others in range are as well.

The problem is that readers cannot distinguish between pallets that are loaded onto trucks and those that appear in the RF field by accident. If all detected pallets were reported to the warehouse management system as being shipped, the resulting incorrect invoices would mean stores have to pay for goods that they neither ordered nor received.

Furthermore, returning wrongly shipped pallets back to the sender incurs very high costs that can often exceed the actual value of the goods. It is evident therefore that as long as this problem remains unresolved, a reliable and productive use of RFID technology in distribution center processes is not feasible.

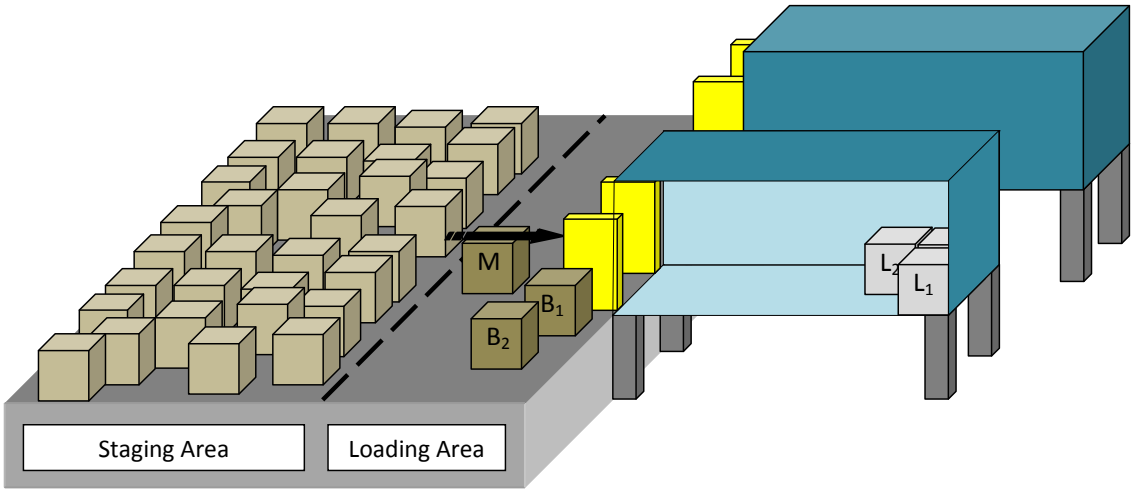


Figure 1.4.: Loading of pallets in a distribution center

Figure 1.4 depicts a distribution center with two containers and their respective RFID portals in front of them. In this scenario, pallet M is about to be loaded into one of the containers. Pallets B_1 and B_2 have been temporarily placed in the loading area by the warehouseman and pallets L_1 and L_2 have already been loaded. It is expected that pallets M, B_1 and B_2 would be detected because they are directly within range of the antennas. However, due to electromagnetic reflections, this range can be significantly extended meaning that pallets L_1 and L_2 as well as any other pallet nearby in the staging area are also recognized.

The question is, with so many pallets recognized by the reader, which of them was actually loaded and should be billed? Scanning L_1 and L_2 is not that much of a problem, as they have already been loaded into the container and are obviously assigned to the corresponding customer. But this isn't necessarily true for pallets B_1 and B_2 or any of the pallets in the staging area. Although B_1 and B_2 might be intended for that specific container, it is also possible that they are intended for the container next to it. Any pallets other than pallet M that were read during the loading are called false-positives; either they were not expected to be read because they were actually out of range or they were within the read range only by accident.

This description of an RFID-enabled outgoing goods scenario leads to the understanding that there is a fundamental difference between the tag of interest and the false-positives: the loaded pallet has obviously been *moved* through the portal while any false-positive pallet has *not moved* and is still somewhere in the read range of the RFID antennas. Therefore, for the sake of simplicity, throughout the rest of this thesis the pallet that was loaded into a container is denote a *moved pallet* and a false-positive read is denoted a *static pallet*.

1.3.4. RFID at METRO Group

In 2007 METRO Group, the world's third-largest retailer, started the operational roll-out of RFID technology along its process chain. In the largest operational rollout of this technology in the European retail sector [MET07b] 180 locations across Germany, including all METRO Cash & Carry markets and the central distribution centers of MGL METRO Group Logistics, dock doors for receiving were equipped with RFID portals to automatically register deliveries. Additionally, all of the 70 shipment dock doors at the METRO Cash & Carry central distribution center in Unna, Germany, were equipped with RFID portals to automatically register any outgoing goods.

The problem of false-positive RFID tag reads soon became apparent in the outgoing goods process and thus METRO sought a solution as to how to differentiate between pallets that had been loaded and other pallets stored near the portal only by accident. Because further investments in additional hardware were out of question a software-based approach was requested to solve this problem. As all the available approaches presented in the literature were far from being useful, a completely new approach had to be taken.

Initial considerations had led to the insight that the RFID readers at the shipment dock doors collect a lot of data when scanning for tags and that this data could possibly be used to approach the problem. Now, whenever a pallet approaches an outgoing goods portal it is recognized and the RFID reader performs a transponder scan for up to 10 seconds. During this period a tag is usually seen multiple times, reporting its Electronic Product Code to the reader every single time. However, in addition to the EPC the following data is also stored for every tag answer:

- A timestamp
- An integer value corresponding to the antenna that has read the tag
- The received signal strength indication (RSSI) which is a measurement of the received signal the tag emits

Most approaches to the problem of detecting false-positive reads in the literature simply count the number of answers during a given period to filter out unexpected or unwanted reads. If a tag answers only sporadically and less often than expected then it is interpreted as noise and consequently a false-positive read. However, these approaches led to only fair success and were often only based on theoretical assumptions rather than practical experience.

Hence, it was decided in the METRO RFID project to create a completely new and practically proven software approach to the problem of false-positive RFID tag reads. The basic idea was

to take all of the data stored by the RFID readers into account and then to identify any patterns or characteristics of false-positive RFID tag reads that would help to deal with this problem.

In order to generate a well founded approach a large amount of real-world data was required to ensure universal validity, so students were assigned to monitor the loading of RFID tagged pallets at the outgoing goods dock in the METRO Cash & Carry distribution center in Unna, Germany. The students kept track of which pallets had actually been loaded and which were considered to be false positives. During the months of observations more than 90,000 pallets were monitored, with approximately 2/3 of them causing false-positives.

This data collected in a productive system allows for greater insights than any simulation under lab conditions would, and provides the foundation for research on the detection of false positive RFID tag reads.

1.4. Scope of the Thesis

1.4.1. Research Question

The problems stated above, paired with the unit of analysis, reveal great research potential. On the one hand there is the real world problem of detecting and reducing false-positive RFID tag reads which is vital to ensure a fully functional RFID enabled outgoing goods process. On the other hand there is a unique and extensive dataset which can be used to identify patterns in the low-level reader data specific to false-positives. With this information a software based algorithm could be created that can immediately distinguish between pallets that have actually been loaded and pallets that are standing within the read range of the antennas by accident. Accordingly, the following research question shall be answered in this thesis:

How can the Low-Level Reader Data be used to detect
False-Positive RFID Tag Reads?

The aim of this thesis is to present a generalized framework to approach the problem of false-positive RFID tag reads. The insights that eventually led to its construction are clearly presented so that researchers and practitioners can understand and reproduce them. The advantage of this generalized type of presentation is that the findings can easily be transferred to other distribution centers or even other processes in the supply chain.

Furthermore, because the detection of false-positive RFID tag reads is approached by mapping it to the problem of movement detection, the knowledge derived from this research could potentially be used in other related processes where RFID tags are moved. Examples of such

processes include incoming goods, storage retrieval, electronic article surveillance (EAS) and point of sales (POS) processes.

Consequently, the result of this thesis is a *framework* for utilizing the low-level reader data to detect false-positive RFID tag reads.

1.4.2. Research Methodology

1.4.2.1. Knowledge Discovery Process

This thesis aims to develop a framework to automatically detect false-positive RFID tag reads by distinguishing between moved and static tags. In order to avoid taking a trial-and-error approach to answering the research question an elaborate and well structured research methodology is required. The process of deriving knowledge from data is commonly known as the *Knowledge Discovery Process* (KDP) or *Knowledge Discovery in Databases* (KDD) and is defined as "the non-trivial extraction of implicit, previously unknown and potentially useful information from data" [FPSM92]. In [CPSK07] the following requirements are defined for the knowledge discovery process as a standardized process model:

- The end product must be useful for the user / owner of the data.
- A well-defined KDP model should have a logical, cohesive, well-thought-out structure and approach that can be presented to decision-makers who may have difficulty understanding the need, value, and mechanics behind a KDP.
- Knowledge discovery projects require a significant project management effort that needs to be grounded in a solid framework.
- Knowledge discovery should follow the example of other engineering disciplines that already have established models.

Various process models have been proposed in the literature. One of the earliest and most accepted is the nine-step KDD process described in [FPSS96] which has an academic origin. In 1996 several major companies from Europe and the US including Daimler Chrysler AG and SPSS Inc. came together, developing *CRISP-DM* (Cross-Industry Standard Process for Data Mining) [CCK+00]. Since CRISP-DM has become the leading Data Mining model in industry [CPSK07] it was therefore used as a template for the research methodology in this thesis.

1.4.2.2. The Cross Industry Process for Data Mining

The Cross Industry Standard Process for Data Mining consists of six phases corresponding to specific projects tasks and the relationships between these phases (see Figure 1.5).

Note that it is sometimes necessary to move back and forth between any of these phases; however, the most important relationships are indicated by arrows. Each of the six phases is briefly described below.

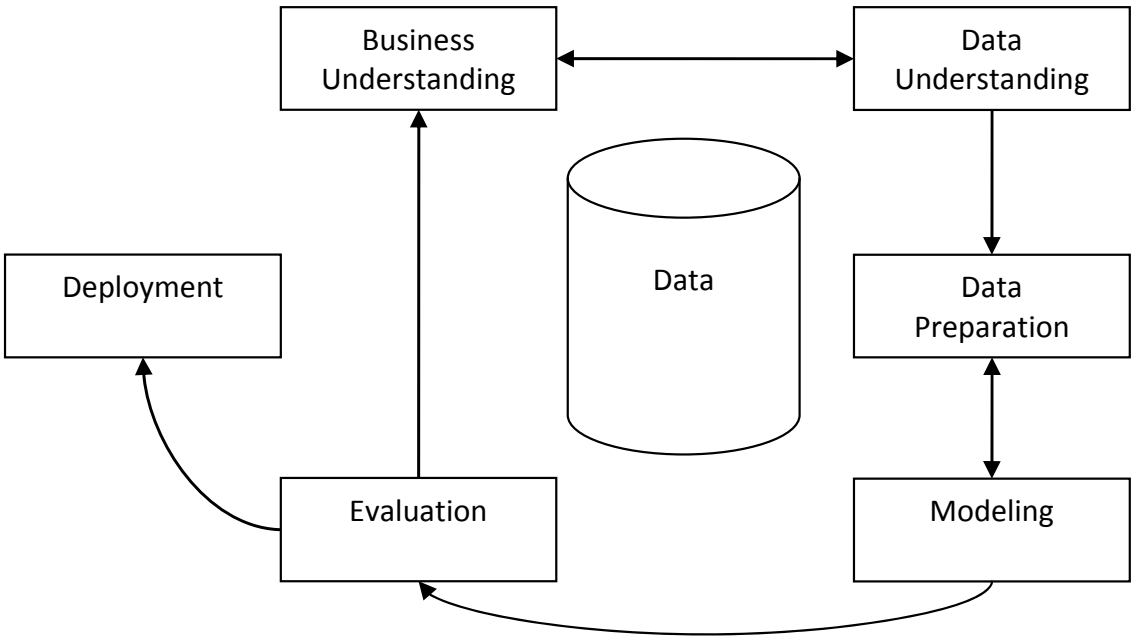


Figure 1.5.: Phases of the CRISP-DM reference model (Based on [CCK+00])

1.4.2.2.1. Business Understanding. The first phase, *business understanding*, is all about understanding from a business perspective what the client really expects from the solution. Not spending enough time and effort on this task may result in "producing the right answers to the wrong questions" [CCK+00]. Usually there are several constraints and requirements on the client side that have to be considered when setting up the research plan and making decisions, such as choosing the data mining model to be used. In order to avoid any confusion, the common terminology used by the client and the researcher also has to be defined at this point. Furthermore, the criteria determining the success or failure of the solution have to be defined too.

1.4.2.2.2. Data Understanding. The second phase, *data understanding*, involves anything related to an initial data overview. First of all, the available data sources have to be identified

and an initial data collection is performed. This helps to identify and avoid in advance any serious problems concerning the data. Furthermore, the data is examined and described on a higher level, including obvious and distinctive features and a preliminary statistical analysis. Finally, the quality of the data has to be evaluated, taking into account considerations like the following: Are all cases covered? Is the data correct? Are there missing values? If any problems are identified a solution must be worked out before moving on.

1.4.2.2.3. Data Preparation. The result of the third phase, *data preparation*, is going to be the final data set (or data sets) used in the succeeding phases. Compilation of the final dataset includes the selection of a data subset to work on and a data cleaning process. The selection of the subset can be justified by data volume constraints or the elimination of (for this specific data mining task) data. Often the dataset is partly incorrect or contradictory; in this case a data cleaning procedure is performed. In addition, the generation of attributes to describe and integrate the data is carried out and it is transformed into a suitable format by syntactic modifications.

1.4.2.2.4. Modeling. The fourth phase, *modeling*, involves three different tasks. Initially, the actual modeling technique to be used is selected (although the model selection is often already complete after the business understanding phase). After the model selection, a test design needs to be generated in order to evaluate the model's quality and validity; this also includes the definition of acceptable error rates. The next task is to actually construct the data mining model based on the specifications and constraints from previous phases.

1.4.2.2.5. Evaluation. The fifth phase, *evaluation*, deals with evaluating the quality of the model on the one hand and how far it meets the business requirements on the other hand. If multiple models were built, they are now assessed in this phase as well. In addition, a review is conducted of the experiences to date, to determine whether anything could be done better in a different way.

1.4.2.2.6. Deployment. In the last phase the results of the evaluation are taken into account and a strategy for *deployment* is prepared. Afterwards, the deployment has to be monitored to identify any malfunctions and to evaluate the model in a productive environment. A final report is usually produced, where the overall project is reviewed and possible improvements are assessed.

1.4.3. Intended Audience

1.4.3.1. Practitioners

The result of this thesis is a framework that practitioners in the field of RFID can easily adopt and examine for applicability to their specific problems. This thesis aims to formulate the insights and findings in a general way, and not specific to the scenario under investigation, so that individual parts of the approach can be easily considered on their own and checked for applicability. In sum, the framework presented here holds the potential to serve practitioners directly or indirectly by providing a complete and adaptable solution applicable to various RFID processes.

1.4.3.2. Researchers

For researchers in the field of RFID the most interesting part of this study is probably the underlying method followed to approach the problem of false-positive RFID tag reads. It fills a gap in the academic literature and is entirely unique in considering the low-level reader data as a valuable source of information rather than as dispensable junk data. The insights are not too specific and can easily be tested for their validity in any other process where RFID tags are moved. Hopefully this will inspire other researchers to adopt the approach and use it to enhance their own work in this area.

1.4.4. Thesis Structure

The thesis is consistently structured according to the Cross Industry Process Model for Data Mining. Table 1.1 shows how the chapters of this thesis are mapped to the individual phases of the process model.

Chapter 2 gives an overview of approaches proposed in the existing literature and then concludes with a rough evaluation of the weaknesses of these approaches that are going to be addressed in this thesis.

Chapter 3 introduces the necessary foundations to understand the problem and how it is approached. First, the problem of false-positive RFID tag reads is discussed from a business perspective, describing in detail the affected processes in the distribution center, and the objectives to solve this problem. Next, the problem is looked at from the data perspective and the low-level reader data used to detect these false-positives is described and illustrated. Based on this, two machine learning approaches are proposed and a general introduction to classification models is given.

Table 1.1.: Structure of the Thesis

CRISP Phase	**Thesis Section**
-	Chapter 2: Related Work
Business Understanding	Chapter 3.1: Understanding Business Requirements
Data Understanding	Chapter 3.2: Understanding Low-Level Reader Data
Data Preparation	Chapter 4: Data Collection and Initial Analysis
Modeling	Chapter 5: Classification Model Building
Evaluation	Chapter 6: Model Evaluation
Deployment	Chapter 6.4: Deployment

Chapter 4 describes how the data needed to train the classification model was collected, and also introduces the three RFID portal types used in the distribution center under consideration, along with their respective peculiarities.

Chapter 5 presents the core of the thesis, describing in detail the framework used to construct a classification model for detecting false-positive RFID tag reads. The low-level reader data is used on different levels of detail to construct two independent approaches, which may also be combined into a third approach. Additionally, an advanced version of the proposed techniques is presented, which, although it cannot be used in the underlying scenario of this thesis, is likely to be useful in other process or scenarios dealing with similar problems.

The results of the empirical evaluation are presented in Chapter 6. In order to compare the effectiveness of the approaches, data collected in a real world scenario was analysed. To demonstrate a constant performance the results are also presented over a period of several weeks. In contrast to the evaluation of the approaches presented in Chapter 2 this real world data allows reliable and meaningful conclusions.

Finally, a summary is given in Chapter 7 including implications for the intended audience of this thesis, i.e., researchers and practitioners.

2. Related Work

2.1. State of the Art in RFID Research

The various issues surrounding the processing of RFID data have been the subject of a steadily growing body of academic literature. An extensive review of this literature suggests that prior work can be classified roughly into four categories.

First, several authors have discussed requirements and design alternatives for the implementation of specialized RFID middleware components to handle large amounts of raw data collected from distributed RFID readers [FL05, Ye08]. A second area of interest has been the design of algorithms for the filtering and aggregation of RFID data streams in order to derive interpretable information, for example, on business events associated with RFID-equipped products in the supply chain [JGF06, TP08b]. Third, various researchers have proposed approaches for the efficient storage of RFID data, query languages and data structures, and other concepts related to data retrieval and management (e.g., [BS09, MTS07]). A fourth research stream deals with the business value of RFID data in various industrial settings [DHS07, TAKF09].

This thesis contributes to the second category by the development and evaluation of novel filtering and aggregation mechanisms for RFID data cleansing. In particular, it considers the phenomenon of false-positive reads, which denotes the problem of RFID readers detecting not only selected objects of interest but also virtually any other tagged object in range. False-positives are a well known issue in real-world implementations of RFID systems in logistics and beyond [CKRS04, MSW08]. Currently, only a few conceptual approaches have been presented in the literature to deal with this problem, sometimes in combination with the related issue of false-negative or missed reads (e.g.,[BWL06, JAF$^+$06, FL04]). In contrast to false-positives, the latter denote objects that a reader device should detect but cannot because of electromagnetic shielding, dysfunctional tags, or various other reasons [PDG06, GDHK06, DW05]. The few countermeasures against occurrences of false-positive reads found in the literature are:

1. The examination of the number and / or the timestamps of tag reads.

2. The deployment of additional hardware.

2.2. Available Approaches

2.2.1. Efficiently Filtering RFID Data Streams

Bai et al. [BWL06] proposed algorithms for RFID data filtering, including noise removal and duplicate elimination. They identified three typical scenarios concerning the reliability of RFID readings: *false-negative reads* are defined as tags which while present might not be read at all. *False-positive reads* correspond to additional and unexpected reads. In addition, *duplicate reads* are defined as being caused by tags in the scope of a reader for a long time (i.e., in multiple reading frames) or because multiple readers are installed to cover a large area.

Bai et al. state that in practice readings are often performed in multiple cycles to achieve a higher recognition rate. In this way false-negative reads are significantly reduced, unfortunately at the same time false-positive reads are increased. Since these are believed to have lower occurrences only tags with significant reads within a specific time are considered as true reads. This in turn produces more duplicate reads. Based on these observations two types of filtering procedures are studied. Elimination of false-positive reads can be done by *denoising* or *smoothing*, and duplicate elimination by *merging*.

The false-positive elimination algorithm uses a sliding window based approach to solve the problem. A sliding window is one with a certain size that moves over time. The algorithm works as follows: if a tag T is read then a full scan of the preceding time window is performed; if T appears more than a defined threshold of times within that time window, it is concluded that it is not a false-positive so every read of T is outputted. The disadvantage of this approach is that multiple tags might be outputted in an incorrect order, i.e., although a tag might have been seen earlier than another one it might be determined as non-noise at a later time. Another algorithm is presented in the paper to deal with this problem.

The approach to duplicate elimination is even simpler. The authors propose here that only the first read should be retained, the others should be discarded. The algorithm takes only a single input parameter $max_{distance}$. If a reading of the same tag is within $max_{distance}$ in time from the previous reading, this reading is considered a duplicate. Otherwise it is considered a new read and will be outputted.

2.2.2. Efficient Object Identification with Passive RFID Tags

In [Vog02] Vogt proposed a method to reliably identify multiple tags by adapting the number of read cycles performed, depending on the number of tags present in the reading field and the chosen frame size. If the number of tags is high and the frame size is low, for example, then the

percentage of identified tags will fall. The reliable identification of multiple objects is especially challenging if many objects are present at the same time, for example in a supermarket checkout scenario. Vogt further differentiates between two different scenarios: *static* and *dynamic* tag set identification.

The first scenario is used to describe tags that are placed in a radio frequency field until all of them have been correctly identified. Such a scenario might be a shopping bag full of RFID tagged items placed near an automatic checkout counter until all items have been detected. In contrast to that kind of self-contained process the second scenario describes a process in which RFID tags are detected continuously without an explicit termination. Such a scenario might be an RFID portal at the intersection between backroom and sales intended to continuously detect any items passing through it. The author states that in the latter scenario an estimation of the number of tags passing through and an adapting frame size is necessary to maximize the identification rate. However, the author concentrates only on the first scenario.

In the case of the supermarket checkout the number of tags is not known in advance so it is unclear how many read cycles have to be performed in order to scan all items. If too many cycles are performed there might an excessively long delay and if too few are performed not all items might be read. Consequently, the author sought an optimal value for the number of cycles which, nevertheless, will vary with the frame size and the actual number of tags. For this purpose an approach is presented to estimate good values for the frame size and the number of tags present in the reading field.

2.2.3. Reasoning about Uncertainty in Location Identification with RFID

Among other scenarios, Brusey et al. [BFHF03] analyzed false-positive RFID tag reads on the basis of a first in, first out product queue. In this scenario RFID tagged men's shaving items, such as razors and deodorant, are stacked on top of each other. Items are only put on top of the stack and removed from the bottom. An RFID reader scanned the next item to be removed by a robotic arm, i.e., the one at the bottom of the stack. The challenge was that not only was the lowermost item scanned but also various items on top of it. These are considered false-positive reads and need to be filtered out.

The classification procedure uses a sliding window approach supported by a weighting function. A tag is considered to be present if it has been read at least once within that time window. The detection of false-positive reads makes use of 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. Although this procedure yields good results it

unnecessarily delays the removal process because after a product has been removed it takes some time before the number of reads of the next item exceed the previous one.

This problem is resolved by introducing a weighting function that attaches greater weight to more recent reads. Thus, the item with the greatest overall weight is identified as the lowermost.

2.2.4. Reducing False Reads in RFID Embedded Supply Chains

Tu and Pyramuthu analyzed so-called *true* and *false readings* in terms of the presence and absence of RFID tagged objects [TP08b, TP08a]. In their theoretical scenario two readers are used simultaneously and there are two tags expected to be present at the same time. They proposed 3 different algorithms to reduce the read rate error for the tag of interest; these have recently been applied to a healthcare scenario [TZP09].

The first algorithm is used as a base case to compare the results of the other two. If both readers identify a tag as being present, then it is assumed that it is really 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 then it is assumed to be true; if none of them reads the tag it is assumed that it is absent. However, in the case where only one of the readers detects the tag a sliding window approach is used. The window size comprises 15 tag reads. For the first 15 reads it is assumed that the tag is present with a probability of 50%. After that, the 15 immediate past reads are used to determine presence or absence of the tag.

The third algorithm uses information about a second tag that is expected to be read at the same time. Put simply, this means that each object is tagged with two RFID tags. The cases where both or none of the readers recognize a tag is analogous to the first two algorithms. In the case where only one reader detects the tag of interest, information about the other tag is used to come to a decision. If both readers agree that the second tag is present then the first one is assumed to be present as well. In the case where both readers disagree about the presence of both tags, a sliding window approach is used as in algorithm 2.

2.2.5. I Sense a Disturbance in the Force: Unobtrusive Detection of Interactions with RFID-tagged Objects

Jiang et al. analyzed false-positive reads in terms of object interaction [JFRP06]. Their approach relied on the observation that when an object is moved or rotated the distance and the angle between reader and RFID tag change. They state that readers usually report only the presence or absence of tags in terms of *seen* or *not seen*, and that any interaction during a

period of presence is not going to be detected.

Consequently they use the *poll* command of the reader to transmit N polls per second and then report the number of answers per tag. A response rate α is defined as the ratio of answers to polls. If a tag has not been seen at all then α is 1, if it has always been seen then it is 0. First observations have shown that the further away a tag is the lower the response rate α. In this process, a suitable value for the number of polls that are to be sent is required.

However, they found that the response rate not only changes when interacting with an object, but also if additional tags (i.e., false-positives) are in read range. One idea to deal with this kind of problem is to use additional tags per object and multiple readers. If for example an object has two tags attached at different sides then a rotation of the object is recognized by an increased response rate of the first tag and a decreased response rate of the second.

2.3. Summary

This chapter gave an overview of the current approaches found in the literature. However, a closer look at these approaches reveals substantial weaknesses that prevent their productive use. These include, but are not limited to, the following:

- Usually there is no practical evaluation of the approaches in a real world scenario. Working on data acquired under laboratory conditions does not give much insight into how well the approaches will work in the real world.

- Data acquired only under lab conditions leads to very simplified assumptions.

- The approaches are very unlikely to be generalized as they are often based on a very specific scenario and thus are not applicable to other situations.

- The use of a sliding window approach based on the number of answers per tag appears useful at first sight but in fact there is a lot more data available from the RFID reader data which could possibly be used to improve any of the algorithms.

- The absence of an elaborate research methodology means that the threshold values used are almost all based on a trial-and-error approach.

- The use of additional hardware like multiple tags per object or additional readers was proposed several times as the only sound solution to reliably identify false-positive RFID tags. This though, leads to additional and unwanted costs.

3. From Low-Level Reader Data to Detection of Movement

According to the CRISP-DM research methodology a deep understanding of the problem from both the business and data perspective is required. This is the aim of the following sections.

3.1. Understanding Business Requirements

3.1.1. Detailed Description of the RFID-enabled Outgoing Goods Process at METRO Group

The problem of false-positive RFID tag reads was investigated using the RFID enabled outgoing goods process at the METRO distribution center in Unna, Germany. In order to fully understand this process it is necessary to describe the type of pallets used in the distribution center and how and when they are tagged with RFID transponders. Next, the loading of a pallet by a warehouseman is described - similar to the way it is done in the distribution center under consideration. Figure 3.1(a) shows the dock door for outgoing goods in the distribution center. Figure 3.1(b) shows a single portal. In both pictures it is easy to see how close the portals are to each other and to the staging area; this exacerbates the problem of false-positive RFID tag reads, because it is very likely that many pallets are located in range of the antennas.

3.1.1.1. RFID Pallet Label

Attaching the RFID tags to the pallets is usually done immediately after the commissioning by the respective warehouseman. Figure 3.2 shows an example of an RFID label used in the distribution center, which is very similar to the GS-1 Germany recommendation for an RFID transport label [Ger]. The reason why so much information is written on the label is that it serves as a fall-back option, if for any reason the transponder does not work correctly.

The left side of the label is where the actual RFID transponder is located. In the header the distribution center name (*MGL Unna*) with the corresponding address is written. The

(a) (b)

Figure 3.1.: RFID Portals at Shipment Dock Doors

Serial Shipping Container Code (*SSCC*) is a global unique number to identify shipping units (e.g., pallets or containers). *PID* refers to the pallet ID, which is an internally used sequence number to identify the pallet, note that it is a subset of the SSCC and it exists also in a bar code representation. *BKZ* is a numerical representation referencing a specific destination, for example a retailer or another distribution center. This is handwritten on the label by the warehouseman who tagged the pallet. *EPC* is the actual Electronic Product Code that is sent by the transponder to an RFID reader. This is just another coding scheme of the SSCC, thus they can be transformed into one another. In the footer a bar code representation of the SSCC is written and on the right the warehouseman can tear off a part of the label containing the PID, which can also be used as evidence that it was loaded into a container.

3.1.1.2. Truck Loading Preprocessing

Initially, when a warehouseman starts the loading of a container, the shipment office gives him the loading protocol containing information about the designated store and the portal number where the corresponding truck is waiting. He then uses a computer in the shipment office to inform the RFID software application that he is about to load pallets into the container. Next, he moves to the portal and physically opens the dock door. This in turn powers up the motion sensor which then begins to scan for movement in front of the RFID portal. Now the actual loading of pallets into the container can begin.

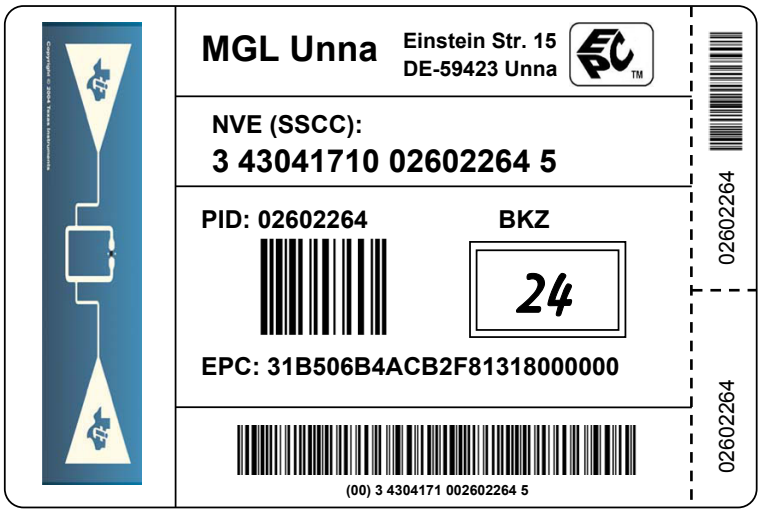


Figure 3.2.: Example of an RFID tag used in the METRO Distribution Center in Unna, Germany

3.1.1.3. Truck Loading

Usually each container can hold around 18 pallets, although this number may vary depending on their weight and size. The task of the warehouseman is to completely fill the container by loading pallets one by one. Up to 40 or 50 pallets designated for the market have already been placed in the staging area directly in front of the shipment dock door so the warehouseman can immediately start the loading process. At this point two workflows are common:

1. The warehouseman retrieves a pallet from the staging area, returns to the dock door, and immediately places it in an appropriate spot in the trailer.

2. The warehouseman retrieves a pallet from the staging area, but instead of loading it into the trailer he places it near the dock door. This is repeated until enough pallets are buffered there and the warehouseman decides to load them into the container. This is done because he needs to presort the pallets to ensure an equally balanced weight distribution in the container.

As soon as the warehouseman approaches the RFID portal, he is recognized by the motion sensor and the RFID reader starts scanning for transponders moving through the portal. The collected IDs of the tags, which uniquely identify the pallets, are sent to the warehouse management system and the warehouseman gets an immediate visual feedback via the signal light:

1. If the loaded pallet has been brought to the right truck, the light flashes green. The loading was valid and the warehouseman may continue with the next pallet.

2. If the pallet was not designated for that particular store, the light flashes yellow. The warehouseman consequently unloads the pallet and continues with another one.

3. In any other scenario (e.g., if a tag is unknown to the warehouse management system), the light flashes red.

There are two special cases that need to be considered. To ensure an unproblematic and secure load, every pallet needs to reach a minimum height so as to minimize the waste of space and prevent pallets falling on each other while being shipped. Consequently, two or three low-height pallets will be stacked on top each other to reach the required height. The important fact about these *stacked pallets* is that each of them has its own RFID tag attached. Under these circumstances it is possible that more than one pallet is loaded at the same time, in which case the visual feedback is altered as the light flashes green in a faster sequence if more than one moved pallet was recognized.

The other special case is the so-called *Mob-Ware* (German: furniture products). Because it is difficult to attach a tag to such things as a bunch of chairs or umbrellas they are dealt with differently. These products are loaded onto the truck just like pallets, but after everything has been loaded, the warehouseman then walks through the RFID portal with the corresponding RFID tags in his hands and puts them in a special pocket inside the container. This way the mob-ware is also detected automatically, because the tags moved through the portal.

Figure 3.3 shows examples of such pallets. In Figure 3.3(a) an ordinary pallet of water bottles is depicted with a clearly visible RFID tag attached to it. Figure 3.3(b) shows a bunch of chairs, i.e., mob-ware, on the left, and two low-height pallets stacked on top of each other on the right.

3.1.1.4. Truck Loading Postprocessing

After the container is fully loaded the warehouseman physically closes the shipment dock door thereby also powering down the motion sensor. He then returns to the shipment office where the RFID software application is informed that the loading process has been completed. The corresponding invoice is issued to the designated store and a request for transportation is sent to a shipper.

3.1.2. Determination of Business Objectives

After this description of the pallet loading process it is necessary to analyze from a business perspective what is really expected from the solution. As previously stated, METRO group's central distribution center in Unna, Germany is fully equipped with RFID hardware to support

(a) (b)

Figure 3.3.: Example Pallets

and enable diverse warehouse processes and in particular the outgoing goods process (described above) to automatically detect any outgoing pallet. However, false-positive RFID tag reads are a barrier to a reliable and fully functional process because pallets actually loaded into containers cannot be always identified correctly.

This inevitably leads to loading errors where pallets are shipped to destination markets that neither ordered the goods nor paid for them. Returning misallocated pallets to the distribution center is very expensive and carried out only in rare cases. The only way to avoid loading errors in advance is if the warehouseman who processed the loading performs a manual check to identify potential mistakes. However, this procedure is not error free and because it requires an extensive time effort still leads to undesired costs.

It was shown in Chapter 2 that various approaches have been proposed to deal with this problem by using additional hardware like multiple tags per pallet or multiple RFID readers at the portals. However, in this study further investment in RFID hardware was not considered a valid option because if possible, the solution should only make use of the existing hardware so as to avoid additional costs.

Although here the problem of false-positive RFID tag reads is approached in the context of an outgoing goods process, the problem exists in a multitude of processes such as electronic article surveillance, point-of-sale processes, pallet retrieval from high-rack storage areas, and so on. Given this, it is desirable to develop a solution that is well understood and could possibly be transferred to other processes with similar problems.

Summing up, from a business perspective, the following expectations of the solution have been identified:

1. Minimize the number of loading errors
    a) Minimize the number of false-positive RFID tag reads
    b) Minimize the time to correct mistakes
2. Avoid any further monetary investment if possible
3. Generate knowledge to transfer the solution to other processes

3.1.3. Determination of Data Mining Goals

The determination of the data mining goals is a very important task as they have a great influence on the data mining model chosen and on the objective success criteria. This is also the process of mapping the business objectives to quantifiable and measurable performance indicators. The overall data mining goal equals the goal of this thesis: namely the development of a software based approach to automatically detect false-positive tag reads in the context of an RFID enabled outgoing goods process.

For a start, there are probably an incredibly large number of possibilities for accomplishing this goal using data mining techniques. Usually this number is considerably reduced because not all data mining methods are suitable for every type of task. Suitability is predominately determined by a method's characteristics. The following characteristics have been identified as most relevant by METRO Group representatives:

- Classification Accuracy
- Classification Speed
- Ability to deal with numerical and nominal attribute values
- Transparent Decision Tracking
- Reliable Classification Performance over time

Naturally *classification accuracy* plays a major role when selecting a data mining technique because it is one of the key performance indicators and is directly related to the overall quality of the classification model. However, it is difficult to compare the methods with one another

based on this characteristic because it heavily depends on various other factors such as the allowed input data type or the vulnerability to *overfitting* (see section 5.1.5).

The second important characteristic is the *classification speed*, especially given that time is always crucial in distribution center processes, since the warehouseman needs immediate feedback to correct errors the moment the loading has finished.

As will be shown in the following chapter, different input data types, i.e., numerical and nominal attributes, are available to describe a pallet. If a classification model is only able to deal with one of them then this automatically leads to a significant information loss. The capability to *deal with numerical and nominal attribute values* is therefore an important decision criterion.

Since one of the business objectives is also to generate knowledge and to possibly transfer the solution to other processes, a profound understanding of why the final data mining model does what it does is necessary. This is equivalent to the characteristic of a data mining model to offer the capability for *transparent decision tracking*.

The data mining model must not only perform well based on some test data; it is essential for a fully functioning outgoing goods process that it shows *reliable performance in the future*. A significant variance in the classification performance cannot be tolerated.

Apart from *classification accuracy* and *classification speed* the identified model characteristics are of a binary type: either a model has a capability or it does not. Accuracy and speed, however, are usually expressed in numerical terms and hence are evaluated in a different manner.

There is no clear definition of the required *classification speed*, but it is known though that the decision has to come *as soon as possible* and within only a few seconds. However, if, for example, one model requires 2 seconds for the decision and another one required 3 seconds but is significantly more accurate, then time does not matter at this magnitude. Consequently, *classification speed* is transformed to a binary characteristic: *fast enough* or *not fast enough*.

This is similar to the *classification accuracy*. The overall goal is to minimize the number of loading errors as they are negatively correlated to the classification accuracy (the more loading errors, the less classification accuracy). However, this correlation is valid only to a limited degree as can be seen from the following example:

Example (Classification Accuracy vs. Number of Loading Errors)

> Suppose during the loading of a pallet the RFID reader detects 3 additional static pallets. In the event that the classification model made a mistake by classifying the moved pallet as static, then the warehouseman is informed by the signal light that no loaded pallet has been detected. Consequently, he immediately reloads

that pallet by moving it through the portal again. This time, the moved pallet is correctly recognized as moved and the static pallets are again correctly classified as static. Summing up these cases then: there were 8 pallets (each of the 4 pallets was classified twice) recognized by the reader where one of them was classified incorrectly. This means the classification accuracy equals 9/10 = 90%. However, the mistake was detected, thus the number of loading errors is not equal to 1 but to 0.

For this reason it is difficult to define a meaningful value for the classification accuracy. As an orientation the observed accuracy of a manual bar-code process was chosen. Because in this case 97% to 98% accuracy could be reached a value of > 99% was chosen for the classification accuracy. The performance of both *classification speed* and *classification accuracy* can only be estimated *after* the model has been built. A meaningful evaluation requires that the model is already deployed and in productive use. Afterwards it can be determined whether anything does not work the way it should. For example, if the warehousemen complain that they always have to wait 10 seconds before the tags are classified and they regularly have to reload pallets because they were not recognized then it is obvious that the model is neither *fast enough* nor *good enough*.

3.2. Understanding Low-Level Reader Data

3.2.1. Data Terminology

After the detailed description of the outgoing goods process it is necessary to consider this process from a technological perspective and to introduce the terminology required for the upcoming chapters. The major relationships between the data terminology are shown in Figure 3.4. Note that this does not reflect the data model of the sample data but is only used for demonstration purposes. The data model is presented in the following section.

The process of loading a container is called a *session* and once the warehouseman initiates the loading of a container by using the RFID software in the shipment office a *SessionID* is generated to identify any events belonging to that loading.

As soon as the warehouseman approaches the portal this is recognized by the motion sensor and an event called *Start-Motion* is triggered. In case he really passed the portal, then at some point he has to leave it again, which is also recognized by the motion sensor, leading to an event called *Stop-Motion*. If no Stop-Motion event happens within 10 seconds after the Start-Motion event, then it is triggered automatically.

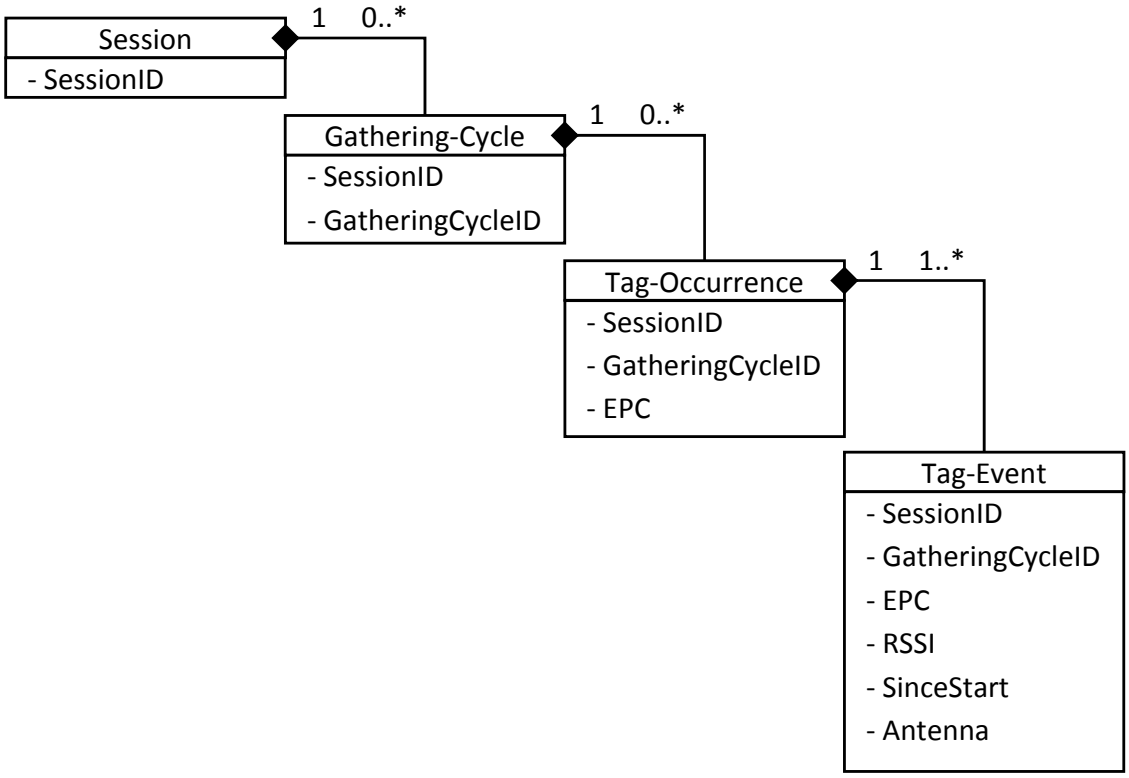


Figure 3.4.: Relationships between Data Terminology

The time period between the Start-Motion and Stop-Motion events is called a *gathering-cycle*. The idea is that whenever a warehouseman approaches the portal it is expected that he is going to load a pallet. Consequently the RFID reader uses its antennas to scan for transponders in range. The entire data collection of a pallet loading is achieved during such a gathering-cycle, which by definition runs for at most 10 seconds. Every event that happens during a gathering-cycle is identified by a combination of *SessionID* and *GatheringCycleID*.

If a transponder, for example an RFID tagged pallet, is read during a gathering-cycle then this is denoted a *tag-occurrence*, which is further identified by the EPC it transmitted to the reader.

Usually, a specific transponder in range is read more often than once during a gathering-cycle. Each of these reads or answers is called a *tag-event*, which is uniquely defined by a combination of *SessionID*, *GatheringCycleID*, *EPC* and a *SinceStart*. Furthermore, for every single tag-event the reader stores the signal strength of the tag answers (RSSI), a timestamp corresponding to the time that has passed since the start-motion event was triggered (SinceStart) and the exact antenna at which it occurred (Antenna).

Of these terms, *gathering-cycle*, *tag-occurrence* and *tag-event* are the most important and thus described in more detail in the following section.

3.2.2. Available Low-Level Reader Data

In this section the data model of the sample low-level reader data is explained. As previously stated, during the data collection in a gathering-cycle a reader usually receives multiple answers from each tag that is in range of the antennas. Depending on a number of external factors, for example, the physical composition of the goods or the packages, the number of answers per tag can be higher or lower. Every tag-event t in terms of the low-level reader data can be represented as the following data tuple:

$$t = [Session_{ID}, GC_{ID}, EPC, Source, SubSource, Time, SinceStart, Antenna, RSSI]$$

The set of all tag-events (i.e., answers) corresponding to a specific tag T during a single gathering-cycle G makes up a *tag-occurrence*. It is defined as

$$TagOccurrence(EPC) = \{TagEvent \in G | TagEvent.EPC = EPC\}$$

When examining the individual elements of the low-level reader data presented in Table 3.1 it becomes obvious that not all of them have the potential to be useful input data for a classification model because they are not specific to the individual tags.

$Session_{ID}, GC_{ID}, EPC$ and $Source$ are all required for identification purposes only. Furthermore, $Session_{ID}, GC_{ID}$ and $Source$ take on the same values for all tags that have been read during a gathering-cycle and thus do not carry any tag specific information. EPC identifies the individual tags but does not say anything about movement. Consequently, these elements of the low-level reader data can not be used as input for a classification model. The element $Time$ corresponds to a global timestamp that also doesn't have any specific information regarding the gathering-cycle. To put it simply, it does not make any difference whether a pallet is loaded in the morning or in the evening - the process is always the same.

However, because the rest of the elements are specific to each tag-event and thus are specific to each tag that was read during the gathering-cycle, they might be useful when trying to use the low-level reader data to discriminate between moved and static tags. Because of their importance they are explained in more detail below.

SubSource The RFID readers that were used in our scenario have the ability to support four different antennas. However, in some cases it might be useful that a portal has more than four antennas attached to it, in which case an additional reader needs to be installed (see section 4.2). The *SubSource* attribute is used to determine which reader the specific antenna where the tag-event occurred is attached to.

Table 3.1.: Overview of Low-Level Reader Data

Element	Data Type	Description
$Session_{ID}$	String	The unique identification number for this session.
GC_{ID}	String	The unique identification number for this gathering-cycle.
EPC	String	The Electronic Product Code to uniquely identify the pallet.
$Source$	String	The identifier of the RFID portal at which the gathering-cycle took place.
$SubSource$	String	If a portal has more than one reader installed then this the identifier of the corresponding reader.
$Time$	Timestamp	The absolute timestamp of the tag-event including time and data.
$SinceStart$	Integer	The time that has passed since the start of the gathering-cycle.
$Antenna$	Integer	An integer representation of the corresponding antenna at which the tag-event occurred.
$RSSI$	Decimal	The received signal strength indication.

SinceStart Each tag-event has two different timestamps assigned. The $Time$ attribute is an absolute timestamp with the corresponding year, month, day and exact time. The $SinceStart$ attribute is relative to the beginning of the gathering-cycle, since it measures how many microseconds have passed since the pallet loading at that portal started.

Antenna Every reader in the METRO distribution center in Unna, Germany has exactly four different antenna attached to it. Whenever a tag responds to the RFID reader the database records which antenna received the answer in the $Antenna$ attribute.

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 antenna. By nature, the RSSI value increases the closer a tag is to the antenna and decreases the further away it is.

3.2.3. Examples of Low-Level Reader Data

An example of the tag-events that occur during a gathering-cycle is depicted in Figure 3.5. In this case, two pallets were present in the RF field: a moved pallet which passed through the RFID portal and was loaded into a container; and a static pallet which was located nearby. The data points shown in this graph correspond to the individual answers the tags gave to the reader. The data given therein includes the signal strength and timestamp of the answers; the information about which antenna read the tag was omitted for reasons of comprehensibility.

The interesting information in this figure is the different low-level reader data specific to the moved and static pallets. Because the static pallet does not change distance from the antennas it is read with an approximately constant signal strength over the entire data collection period. This is completely different for the moved pallet, which is first detected with increasing signal strength, reaching the maximum when the tag actually entered the gate about 1.5 seconds after the start of the gathering-cycle. After leaving the gate, the signal strength decreased again.

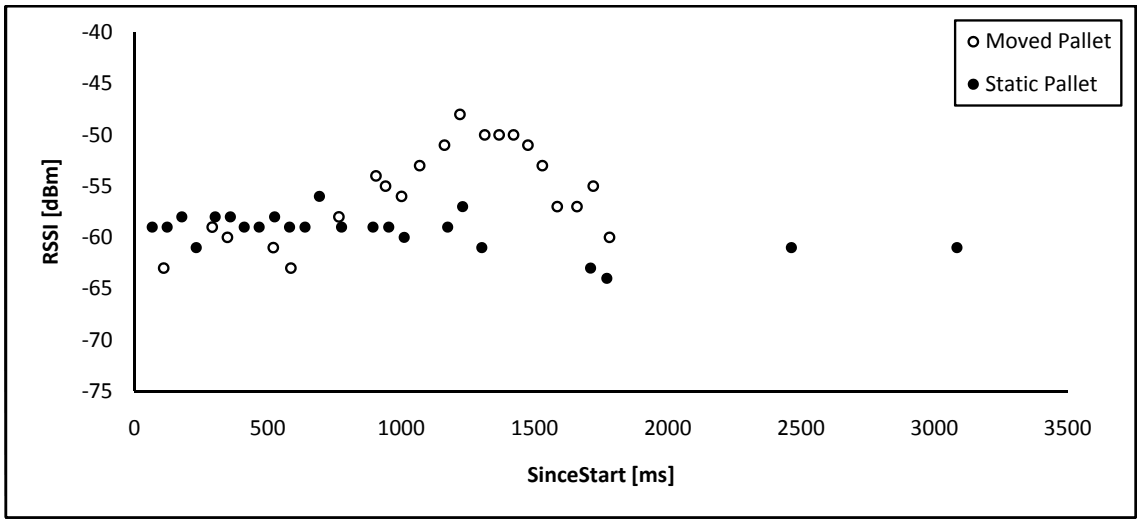
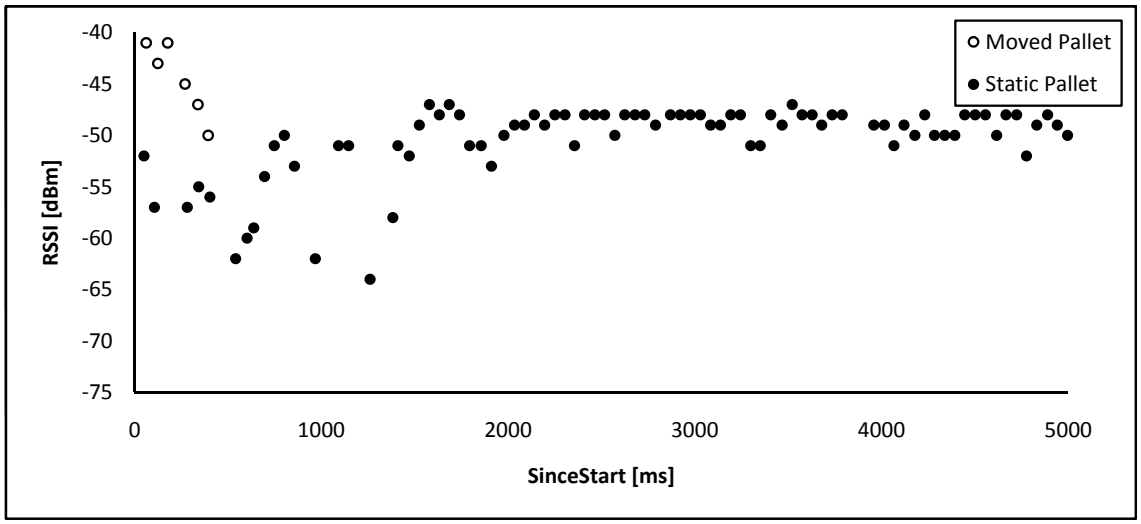


Figure 3.5.: Example of Low-Level Reader Data (Best Case)

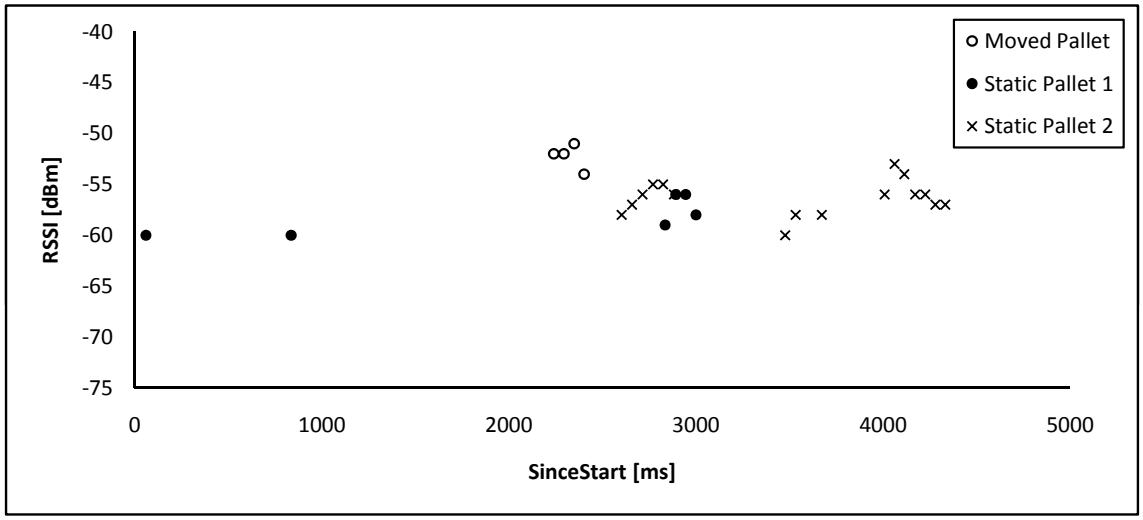
Unfortunately though, while this example poses an illustrative case of RFID based pallet identification, it is by no means typical for the majority of pallet detections. Figures 3.6 and 3.7 depict other examples of low-level reader data that elude the clear interpretation seen in the previous case.

In these cases it is very difficult if not impossible to decide which of the pallets read during the gathering-cycle actually moved through the portal, and it is obvious that some kind of automatic classification system is necessary.

Figure 3.6(a) shows a moved pallet that was detected only 6 times within the first 0.4 seconds



(a)



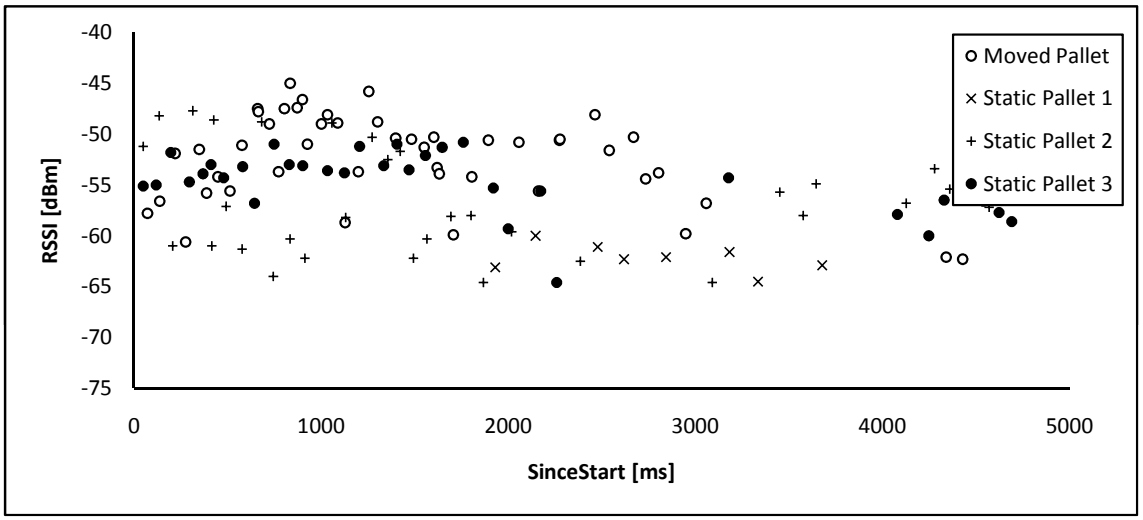
(b)

Figure 3.6.: Examples of Low-Level Reader Data (Normal Cases)

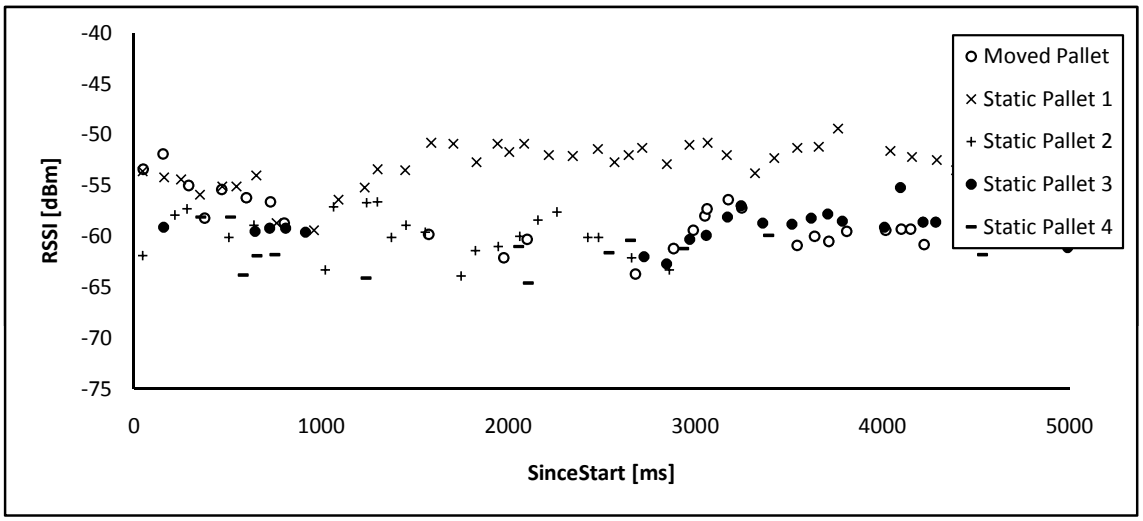
with a strongly decreasing signal strength and a static pallet that had a distinctive variation in the received signal strength indication that evens out at a considerably high level.

Figure 3.6(b) shows a gathering-cycle where 3 different pallets were detected. Again the moved pallet was seen only during a very short time frame, this time for only 0.2 seconds. The first static pallet was seen for 3 seconds, but with only a few reads as well. The second static tag showed a higher variance of RSSI values but was not detected within the first 2.5 seconds after the pallet loading began.

Figure 3.7(a) shows 4 pallets, where the moved one was read over the entire time period and with a high variance. This situation holds also true for static pallets #2 and #3. The only pallet that could be visually determined as a false-positive is static pallet #1 that was read



(a)



(b)

Figure 3.7.: Examples of Low-Level Reader Data (Further Normal Cases)

only during the last 3 seconds, with very low and constant RSSI values.

Figure 3.7(b) shows a similar situation with 5 pallets, where all pallets show a high variance and are recognized throughout the entire time period under consideration. It is notable that static pallet #1 is read with a constant very high signal strength and was likely located really close to one of the antennas.

With respect to the number of tag-events and tag-occurrences generally any situation is possible. There are gathering-cycles where only two tags are recognized and none of them more than 3 times. On the other hand there are gathering-cycles with dozens of recognized RFID tags and hundreds of individual answers. It is very difficult to describe a "normal case" because this is significantly different - in particular this depends on the type of pallet and portal.

In Figure 3.8 the number of tags per gathering-cycle is depicted. It can be seen that up to 18 tags can be detected in a single gathering-cycle. However, in the majority (around 85%) of all gathering-cycles between 1 and 5 tags are read.

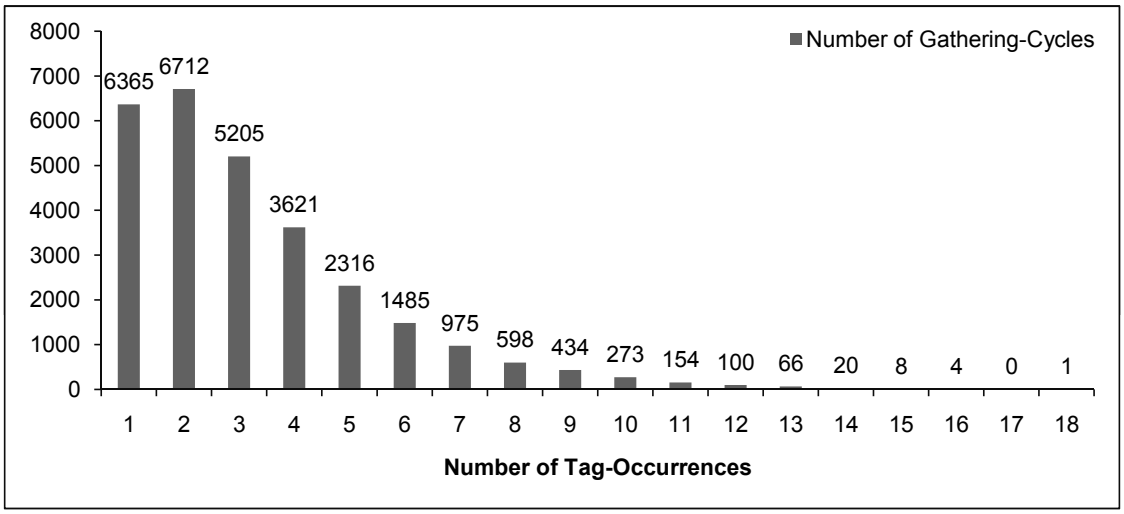


Figure 3.8.: Number of Tags per Gathering-Cycle

In Figure 3.9 the number of tag-events per tag-occurrence is depicted. For presentation purposes the chart has been limited to 50 tag-events per tag-occurrence. However, the median is located at exactly 16.0 tag-events, i.e., 50% of all tag-occurrences are detected at most 16 times during a gathering-cycle. This number can increase to 400 in rare cases.

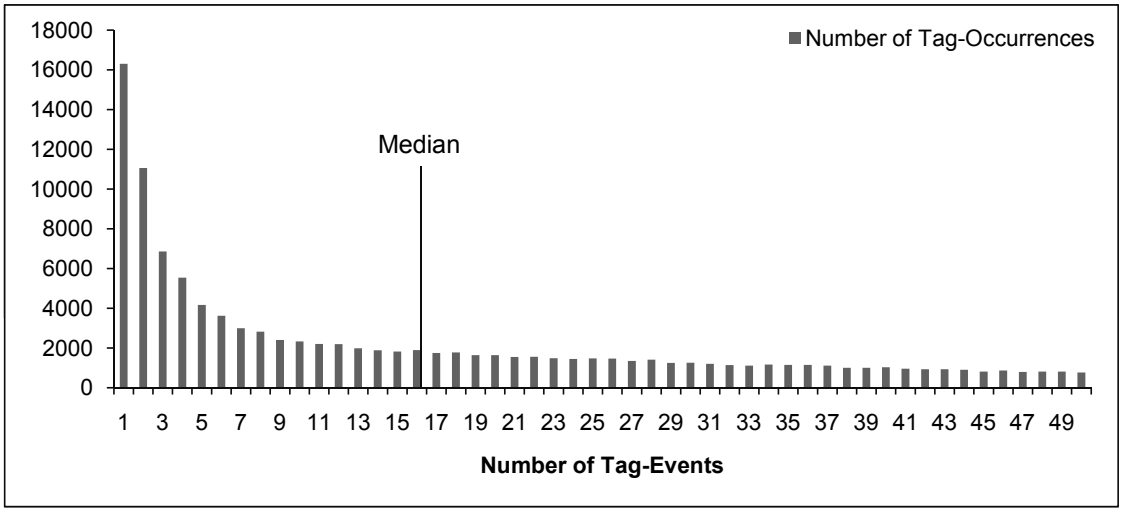


Figure 3.9.: Number of Events per Tag

3.2.4. Movement Detection

Considering the above examples of the low-level reader data, there are generally two alternative ways these can be used for the detection of tag movement. These alternatives differ in complexity and in the level of information granularity that being used. The first alternative is to work with the individual tag-events that are shown in the exemplary gathering-cycles above.

The second alternative is to determine characteristics specific to the set of all tag-events of a certain tag (i.e., the individual tag-occurrences).

3.2.4.1. Tag-Event Level

Figure 3.5 above showed that moving a pallet through an RFID portal changes the received signal strength. Because the individual tag-events are temporally ordered they can be interpreted as a *discrete time-series* of RSSI values. Figure 3.10 shows a selection of the low-level reader data collected during the gathering-cycle shown in Figure 3.5. As can be seen from this Figure, the data can then be transformed into separate time-series for the two tags.

The idea behind this approach is to decide whether the time-series of a tag is most similar to a moved tag or to a static tag. If it looks more like a static time-series then it will be considered a false-positive.

The first problem is to find out what a typical moved or static time-series actually looks like. Several possibilities will be discussed regarding such typical time-series. These are called *reference time-series* in the following and can be derived from the sample data. The second problem is to give the term *similarity* a meaningful definition because determining similarity between time-series is not an easy task. Fortunately, there are a number of approaches available to deal with this problem.

3.2.4.2. Tag-Occurrence Level

In contrast to the tag-event level approach the tag-occurrence level approach works on a higher level of data granularity. On the basis of the single tag-events so called *attributes* are calculated that are generated by applying various aggregation functions and which correspond to specific characteristics. Examples of such characteristics include the maximum, minimum and mean RSSI values or the timestamp of the first or the last recognition of a tag during a gathering-cycle. Figure 3.11 shows how the low-level reader data is used to transform the individual tags into a representation based on these attributes.

The idea behind this approach is to identify false-positive RFID tag reads based solely on the values these attributes take on. If the characteristics are typical for a static tag then it is

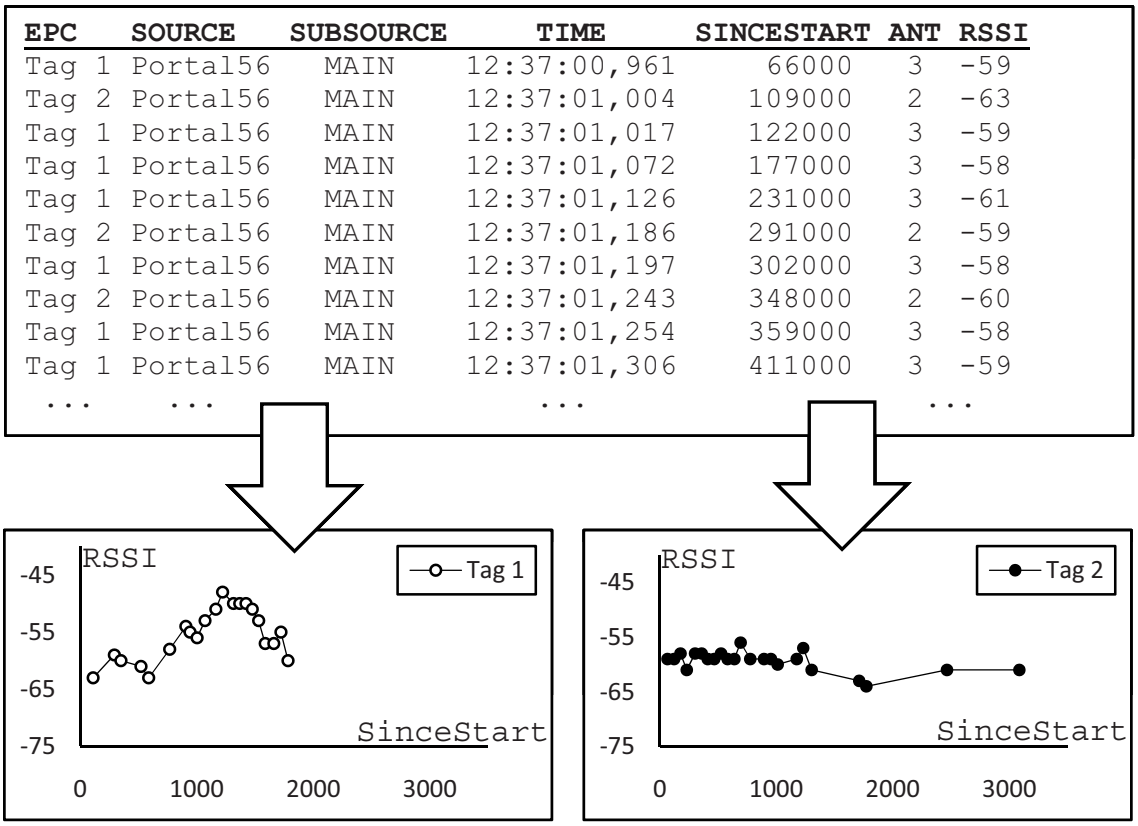


Figure 3.10.: Transformation of Low-Level Reader Data into a Time-Series Representation

going to be considered a false-positive.

The difficulty of this approach lies in determining which attributes are meaningful enough that a significant difference can be observed. The examples shown in Figures 3.6 and 3.7 demonstrate how different the low-level reader data can look like, a situation that requires that a specific technique be used to determine in which cases which attributes are useful, and what values are typical for moved and static tags.

3.3. General Introduction to Classification

After discussing the low-level reader data available, the aim of this section is to introduce the general procedure for constructing a classification model and to present available approaches. By understanding the business and the associated requirements the decision can be made as to which models to use.

Classification is one of the most important tasks in machine learning. It is concerned with assigning objects to classes based on their characteristics (or attributes). In [MB10] the goal of classification is described as being to "build a model which makes it possible to classify future

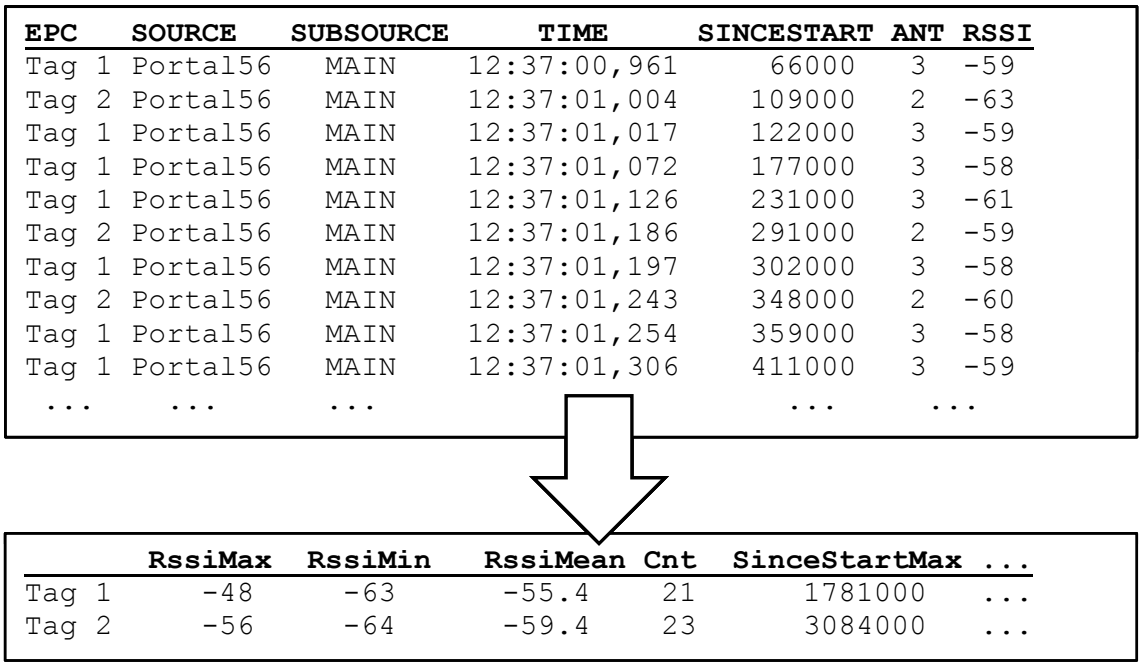


Figure 3.11.: Transformation of Low-Level Reader Data into an Attribute Representation

objects based on a set of specific characteristics in an automated way". Considering the above scenario of RFID enabled outgoing goods, the individual pallets that have been read during a gathering-cycle correspond to the observed objects which can be separated into two disjunctive classes: namely pallets that *have been moved* and pallets that *have not been moved*. These two classes are referred to as *moved pallets* and *static pallets*, respectively.

While it is rather easy for a human observer standing next to the portal to decide whether a pallet has just been moved or not this task is incomparably more difficult for a machine as it only has a limited perception. The human observer can use his eyes to recognize position changes and use available hypotheses like *"If an object changes its position then it has been moved"*; the machine can only use the data that is available through the RFID reader to make such a decision. In particular, the machine does not have any explicit knowledge of any position changes to pallets inside an RFID portal.

3.3.1. Model Training using Supervised Learning

In [Kot07] *Supervised Learning* is described as "the search for algorithms that reason from externally supplied instances to produce general hypotheses, which then make predictions about future instances". An instance in this context is any kind of object (e.g., an RFID tagged pallet) plus a corresponding class label (e.g, a moved pallet).

A hypothesis can be described as a function f that maps an object to a specific class:

$$f(Object) \rightarrow Class$$

If we assume that each object is described by n attributes and there are two classes C_1 and C_2 then the hypothesis becomes as follows:

$$f(Attribute_1, \ldots, Attribute_n) \rightarrow \{C_1, C_2\}$$

The concept of supervised learning is based on the idea that there are a number of instances available for which we know the attribute values as well as the corresponding class labels. When training a classification model these known instances are used to identify attribute values or attribute value combinations that are typical for a certain class and then to derive hypotheses from this knowledge. Later on these hypotheses can be used to decide the class of a previously unknown instance.

3.3.1.1. Attribute Value Types

In general there are two different types of attribute values that can be used to describe an object. On the one hand there are *categorical* or *discrete attributes* which include *nominal*, *ordinal* and *binary* data. Attribute values are called "nominal" if they fall into unordered categories. If the values can be ordered they are called "ordinal". In a case where there are only two possible values the attribute is said to be a binary attribute. On the other hand there are also *continuous attributes* with real numbers as values. The attribute value types of the available sample data are essential in the decision making process since not all classification models support every type.

3.3.1.2. Data Basis

When constructing a classification model it is necessary to determine its overall performance in order to get some idea about how good it will actually perform on unseen data. Testing it based on the data it has been built with will be very misleading and collecting new data every time to verify its quality is out of the question. Therefore this data has to be obtained from somewhere else. There are two common ways of dealing with this problem, one of which, the Train and Test method, is depicted in Figure 3.12

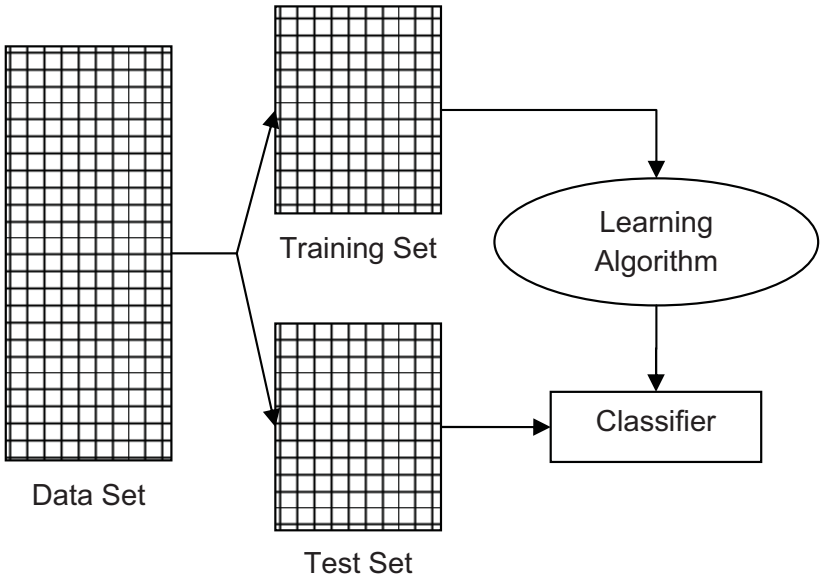


Figure 3.12.: Train and Test [Bra07a]

The sample data is divided into two disjunctive subsets referred to as the Training Set and the Test Set. Three methods may be used to achieve this separation: *linear sampling*, *shuffled sampling* or *stratified sampling*. Linear sampling simply separates the data set by using the first $x\%$ as the test set and the remaining $1 - x\%$ as the training set. Shuffled sampling constructs the two sets by randomly choosing data from the data set. Stratified sampling does the same, however it is ensured that the class distribution in the sample sets is the same as in the whole data set.

The classification model is built on the basis of the training set and then independently tested against the data in the test set. Because using linear and shuffled sampling can sometimes lead to the effect that only samples from a single class can be found in the training or test sets, stratified sampling is favored. However, a single train- and test separation might be misleading as well so this procedure is repeated multiple times with different train- and test sets and the performances are averaged.

The second possibility is called *k-fold cross validation* [Sto74] and is depicted in Figure 3.13. The sample data is divided into k disjunct partitions of approximately equal size. Each of the k partitions is used as a test set while the classification model is trained on the remaining $k - 1$ partitions. The average performance of these classifiers is then returned as the overall performance. A special case of the *k-fold cross validation* is the *leave one out* method [LM68]. This method trains the classification model on all samples except one and then tests it against this single one; it is computationally very expensive though, especially if there are a large number of samples.

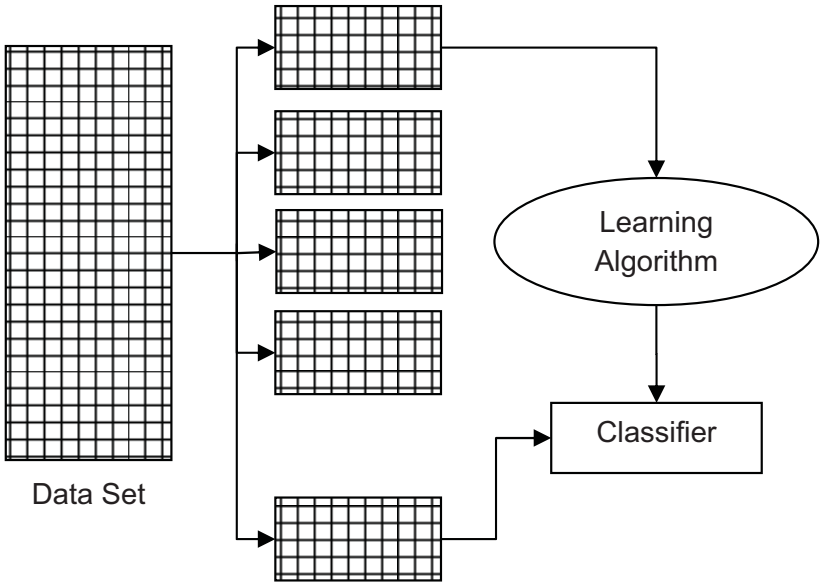


Figure 3.13.: k-fold Cross Validation [Bra07a]

3.3.1.3. Performance Measures

After the classification model has been built on the training set it is necessary to evaluate its performance on the test set. In order to achieve this, performance indicators are needed to answer the following questions:

1. How many pallets are correctly classified as moved and static?
2. How many pallets are incorrectly classified as moved and static?
3. How many of the moved and static pallets are correctly classified as moved and static, respectively.
4. If a pallet is classified as moved or static, how confident is this classification?

For this purpose, several suitable statistical measures have been proposed; the four most common are called *Classification Accuracy*, *Classification Error*, *Class Recall* and *Class Precision*. They are introduced to answer the questions in the order given above.

Classification Accuracy This measure is often simply called *Accuracy* and corresponds to the ratio of correctly classified samples.

$$\text{Classification Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Number of samples}} \tag{3.1}$$

Classification Error This measure depends directly on the classification accuracy because it corresponds to the ratio of incorrectly classified samples. It is usually used for demonstration purposes only.

$$\begin{aligned} \text{Classification Error} &= \frac{\text{Number of incorrectly classified samples}}{\text{Number of samples}} && (3.2) \\ &= 1 - \text{Classification Accuracy} && (3.3) \end{aligned}$$

Class Recall This measure is used to determine the ratio of tags of a specific class C that were classified correctly. Consequently, the class recall for moved and static tags are also known as the *moved detection rate* and the *static detection rate*.

$$\text{Class Recall } (C) = \frac{\text{Number of samples correctly classified as } C}{\text{Number of samples of class } C} \quad (3.4)$$

Class Precision This is a measure of how confident the classifications are, i.e., the ratio of all samples classified as class C that were correctly classified:

$$\text{Class Precision} = \frac{\text{Number of samples correctly classified as } C}{\text{Number of samples classified as } C} \quad (3.5)$$

Example (Classification Performance Measures)

Suppose a classification model was trained and evaluated on a test set containing 2,571 static and 656 moved pallets, and that furthermore, 2,552 of the static pallets were correctly classified as static and 646 of the moved pallets were correctly classified as moved; this leads to the following performance indicator values:

$$\begin{aligned} ClassificationAccuracy &= \frac{646 + 2,552}{3,227} = 99.10\% \\ ClassificationError &= 1 - 99.10\% = 0.90\% \\ ClassRecall(Static) &= \frac{2,552}{2,571} = 99.26\% \\ ClassRecall(Moved) &= \frac{646}{656} = 98.48\% \\ ClassPrecision(Static) &= \frac{2,552}{10 + 2,552} = 99.61\% \\ ClassPrecision(Moved) &= \frac{646}{646 + 19} = 97.14\% \end{aligned}$$

It is common to present these results in a so called *confusion matrix* such as that in Table 3.2 where all relevant information is accessible at at-a-glance. The classification models in Chapter 6 are presented in this way.

Table 3.2.: Example Confusion Matrix

	True Moved	**True Static**	**Class Precision**
Predicted as Moved	646	19	97.14%
Predicted as Static	10	2,552	99.61%
Class Recall	98.48%	99.26%	99.10%

3.3.2. Available Classification Models

There are several different classification models available to choose from. However, not all of them are suitable for any given task because they work under different constraints and differ in complexity and performance. Probably the most popular and commonly used methods are *Decision Tree Learning*, *Neural Networks*, *Naive Bayes Classification* and *Support Vector Machines* (SVM). Table 3.3 shows a rating of the characteristics of these methods. The characteristics that have been identified as the most relevant in Section 3.1.3 are shown in bold.

In terms of *learning speed* neural networks and support vector machines perform much worse compared to decision trees and naive bayes classification. Although this characteristic was not defined as crucial, it is still very important when the data is unknown to the algorithm creator, particularly because in the beginning a large number of tests have to be performed to find the best and most suitable parameter selection. It is obvious that it is helpful if a new test can be set up every few minutes instead of every few days.

Classification speed was defined as an important characteristic because the feedback needs to be available to the warehouseman right after the loading of a pallet. This is not a problem for any of the algorithms once they have been set up.

The ability to *tolerate missing values* and *irrelevant*, *redundant* or *highly interdependent attributes* has a serious effect on the overall classification performance. Decision trees have no problem dealing with missing values or irrelevant attributes because the method has an implicit filter by using only the most interesting attribute subset and thus ignoring useless information.

Table 3.3.: Classification Algorithm Comparison (Source: [Kot07])

Characteristic	**Decision Trees**	**Neural Networks**	**Naive Bayes**	**SVM**
Learning Speed	⋆⋆⋆	⋆	⋆⋆⋆⋆	⋆
Classification Speed	⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆
Tolerance to missing values	⋆⋆⋆	⋆	⋆⋆⋆⋆	⋆⋆
Tolerance to irrelevant attributes	⋆⋆⋆	⋆	⋆⋆⋆⋆	⋆⋆⋆⋆
Tolerance to redundant attributes	⋆⋆	⋆⋆	⋆	⋆⋆⋆
Tolerance to highly interdependent attributes	⋆⋆	⋆⋆⋆	⋆	⋆⋆⋆
Numerical attributes	Yes	Yes	No	Yes
Nominal attributes	Yes	No	Yes	No
Tolerance to noise	⋆⋆	⋆⋆	⋆⋆⋆	⋆⋆
Danger of Overfitting	⋆⋆	⋆	⋆⋆⋆	⋆⋆
Classification Transparency	⋆⋆⋆⋆	⋆	⋆⋆⋆⋆	⋆
Accuracy	⋆⋆	⋆⋆⋆	⋆	⋆⋆⋆⋆

This is similar to naive bayes classification, whereas neural networks have a hard time dealing with such problems.

Because the available attributes are of different types (see next chapter) it is very important that a classification model can *deal with both numerical and nominal data* so that no information loss occurs. As can be seen, only decision trees are able to deal with both types, whereas all others work with either only numerical or nominal attributes. Although it is possible to use tricks to transform, for example, nominal into numerical attributes or the other way around, this does not always work.

Dealing with noise data is less important, the sample data set is expected to be big enough that noise only plays a minor role.

The aim of the classification model to be built is to predict future pallet movements in the distribution center. The ability to classify future data requires that the model is not *overfitted*. The effect is explained in detail in section 5.1.5. The proposed classification models perform more or less similarly, only neural networks have difficulties as they are likely to fit data to irrelevant or redundant attributes.

Another important characteristic was determined to be *transparent decision tracking*, i.e., the classification transparency. This is one of the outstanding strengths of decision trees and naive bays classifiers because the models are self-explanatory by simply looking at them, making it a very easy to track any decisions and to evaluate at which point in the model an incorrect Classification occurs and why this happens. In contrast to this, neural networks and support vector machines are in almost any case absolutely impossible to explain because the models are the results of millions of model building iterations.

Last but not least, the *classification accuracy* has to be compared. It is very difficult to generally decide which classification model shows the best classification performance because this depends heavily on the underlying data and the specific scenario under consideration. Basically, neural networks and support vector machines are considered to be the most powerful classification models. But because these two are not able to deal with both types of data and have very poor transparency they will most likely perform worse in the data sets under consideration and so decision trees are more suitable in this specific case.

Summing up, in this thesis decision trees were chosen over all other models as they are very tolerant of missing values and irrelevant attributes and can also deal with both numerical and nominal data. This will most likely lead to the best classification results, also because there are ways to prevent the effect of overfitting. The explanatory power is also very important because it was the explicit wish of the client to understand how and especially why the model classifies pallets the way it does.

3.4. Summary

The aim of this chapter was to give a detailed overview of the outgoing goods process in a distribution center from a business perspective on the one hand and a data perspective on the other hand.

First of all, an in-depth view of an RFID enabled outgoing goods process was presented. The procedure of loading pallets into a container was described, including the required pallet tagging, loading pre- and postprocessing. The difference between *standard pallets*, *stacked pallets* and *mob-ware* was explained and the fact that sometimes multiple pallets are loaded at the same time was stressed. On this basis, the business objectives were derived. From the client's point of view, the approach to detecting false-positive RFID tag reads needs to minimize the number of loading errors, avoid any further investment and ideally to also generate additional knowledge that could possibly be carried over to other processes with similar problems. Next, these business objectives were transformed into measurable performance indicators that contribute

to the estimation of the solution's success or failure. Furthermore, the requirements of the type of classification model were derived from these objectives.

Secondly, the problem was described from a data perspective by investigating the available low-level reader data. A unifying terminology was defined and the low-level reader data was exemplarily described including the general idea that moved and static pallets show different behaviors - in particular with respect to the received signal strength indication. The two major approaches to detecting movement of RFID tags were then briefly introduced. The idea of working on the *tag-occurrence level* is to identify specific characteristics that help to distinguish between moved and static tags. The idea of working on the *tag-event level* is to consider the development of the signal strength over time and to decide whether this development is more typical of moved or static tags.

Because the overall aim of both approaches is to construct a so-called *classification model* to detect false-positive RFID tag reads, the basics of such models were described including a general introduction to the data mining task of *classification* and the type of allowed input data. Strategies and key performance indicators to measure the quality of such a classification model were described. Finally, based on business and data objectives, the most popular classification techniques were evaluated and it was decided to go for the classification model constructing using *Decision Tree Learning*.

4. Data Collection and Analysis

4.1. Data Collection

The construction of a classification model requires the availability of a massive sample dataset for which the class labels are already known. In the context of this thesis this means that low-level reader data needs to be available and it needs to be known whether it belongs to moved or to static pallets. Generally, there are three different methods to acquire a sample data set.

1. Construct a test environment in a laboratory and then use it for the data collection task.
2. Simulate data on the computer.
3. Gather data in a real world scenario.

The first two methods were mainly used by the approaches presented in Chapter 2 and have a number of drawbacks: for example, a lab environment tends to be very homogenous for every single sample so it is questionable in how far the samples are able to adequately describe the real world; the same applies to any findings under lab conditions because it is nearly impossible to estimate in how far these can be matched to a productive environment. These two problems apply to simulated data to an even more serious degree. It is obvious therefore that it is most desirable to use data samples collected in a real world environment if the findings are to be used there and are to exceed a theoretical or academic level. Furthermore, if access to the productive environment is still available, it is very easy to test the validity of any conclusions.

The aim of this chapter is to describe how the data sets used for the classification model task were acquired. First of all, the general procedure for the data collection is introduced along with a description of how the class labeling of the sample data took place. Next, three different RFID portals, denoted as *Standard-*, *Satellite-* and *Transition Portals*, are introduced; these allow the definition of sub data sets corresponding to their respective types. For these data sets individual classification models can then be built in order to further investigate whether one of the portal types is more suitable for addressing the problem of false-positive RFID tag reads.

4.1.1. Pallet Monitoring at METRO Group Distribution Center

The data set used in this thesis was collected in a productive environment and under real world conditions at the METRO Group central distribution center located in Unna, Germany. This center sees between 3,500 and 8,000 pallet movements a day and all of the 87 shipment dock doors have been equipped with RFID portals to automatically register any outgoing pallets. As stated in Section 1.3.3, the task is to reliably distinguish between pallets that *have been moved* through the RFID portal (and thus were loaded) and pallets that *have not been moved*. In the first case, the pallet is called a *moved pallet* and in the latter case it is called a *static pallet* or a *false-positive read*.

In order to obtain the required sample data set students were assigned to accompany the warehousemen and to monitor the loading of pallets from the distribution center into containers. Their task was to keep track of which pallets that were recognized by the reader during the loading process had actually been moved through the outgoing goods RFID portal and also those which were present in the reading field of the portal antenna only by accident. For this purpose they used a custom developed software called *Varena Analyzer* which immediately after the ending of a gathering-cycle shows a list of all detected pallets. All they had to do then was mark each entry (corresponding to an individual EPC) as either "moved" or "static". A screenshot of this application is depicted in Figure 4.1 where the loading of pallets on August, 17th 2010 is shown.

The relevant parts of the application are highlighted by the red box. In the leftmost column, the students selected either "moved", "static" or "unknown" for each pallet. By default, every pallet is marked as "unknown". The second column shows the result of the classification algorithm currently in use in the distribution center. The value "true" is used to denote a pallet as "moved" and the value "false" is used to denote it as "static". Note that this information in the second column was not shown to the students so as not to influence their decisions. The following data shows information such as the EPC of the pallet, the number of tag detections during that gathering-cycle or the minimum signal strength a pallet was detected with.

Note that due to the occurrence of stacked pallets it is possible that multiple pallets were moved during the same gathering-cycle. In this case, several pallets, each with their own RFID tag attached, had been stacked on top of each other and were moved through the portal all at once (see Section 3.1.1.3).

The acquisition of the data used in this thesis ran for a period of 30 weeks from the beginning of February to the end of August 2009 at the center's shipment dock doors. Figure 4.2 shows the number of pallets monitored in each calendar week.

It can be seen that the majority of the pallet monitoring took place between the 11th and

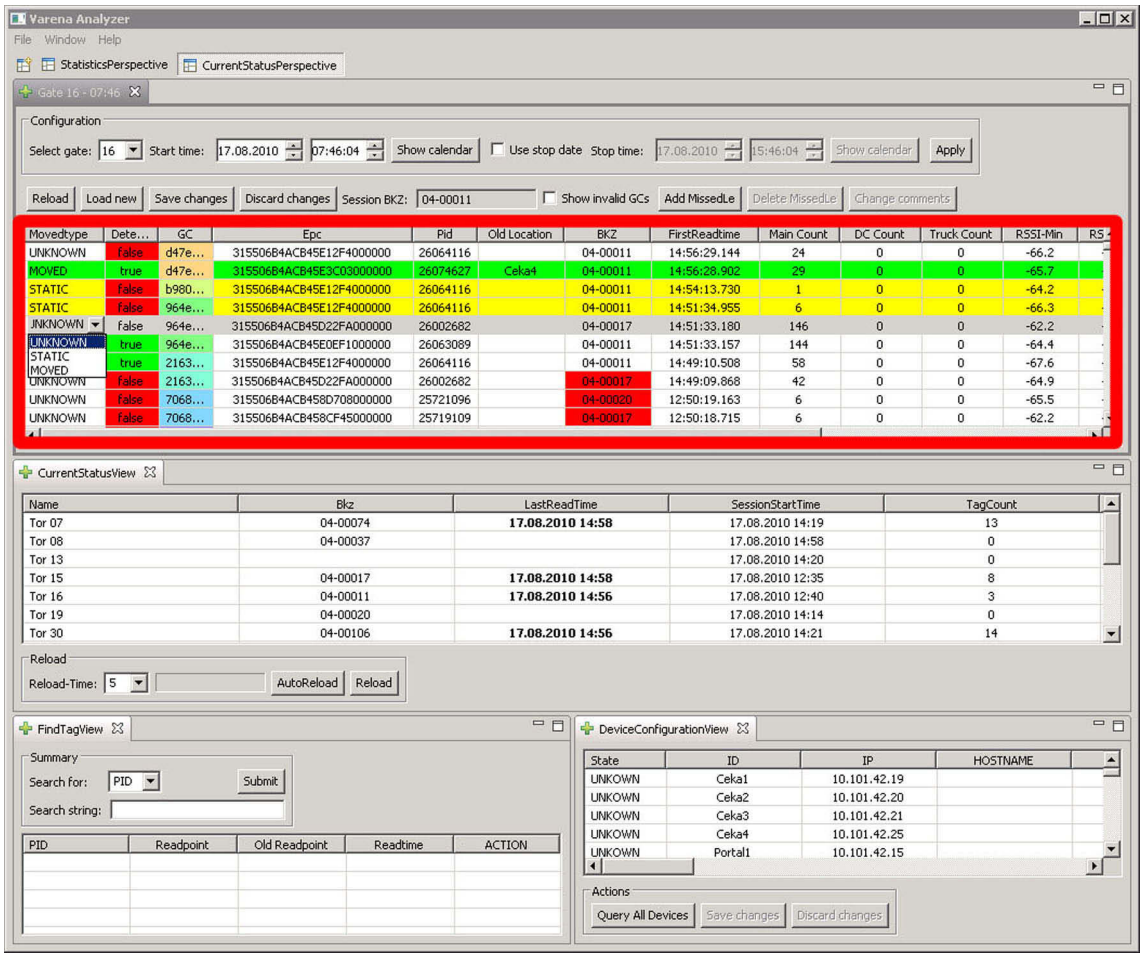


Figure 4.1.: Screenshot of Varena Analyzer Software

27th calendar weeks. This is because in the beginning and at the end of the data acquisition period fewer students were employed to do that task. The presence of the students did not influence the warehousemen or the way they worked, so the sample data is an exact mapping of the real world distribution center process.

In total, 92,857 pallets were monitored with 74,432 classified as "static" and the remaining 18,425 classified as "moved". Usually pallets are detected multiple times during a single loading process so this corresponds to 2,664,621 individual tag detections in total. It is expected that this data set is large enough to cover any possible variances, allows for greater insights than any simulation or lab trial, and thus provides the foundation for our proposed solution.

4.1.2. Data Selection

There is a phrase in the field of computer science, "Garbage in - Garbage out" (GiGo) which is used to describe the effect where a system produces invalid outputs if it receives invalid inputs [Pyl99, Lar05b]. In this context the phrase means that if the samples in the sample data sets

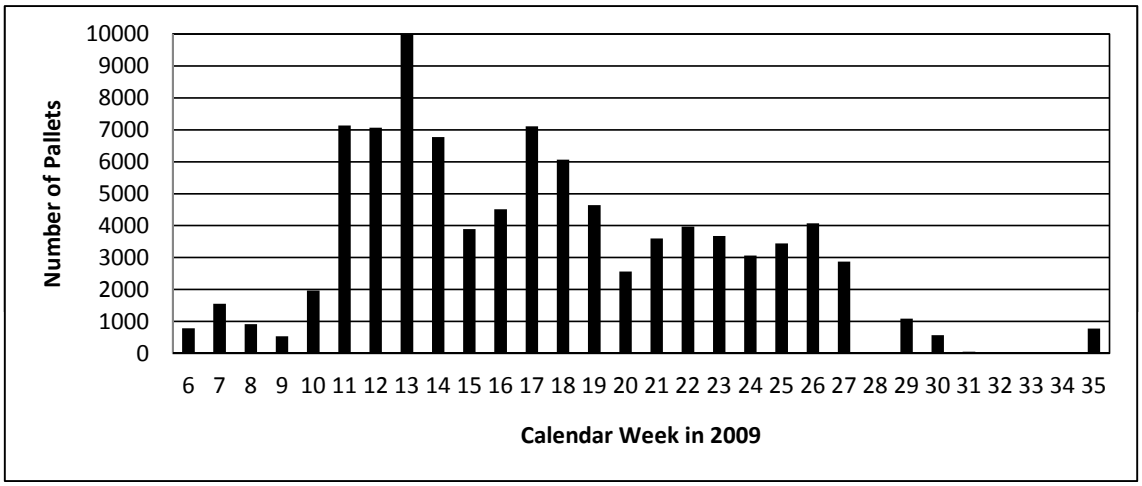


Figure 4.2.: Monitored Pallets per Calendar Week in 2009

were marked incorrectly, then the resulting classification model is also going to be not correct. Thus it is necessary to filter out any data that could possibly negatively affect the quality of the classification model in advance to assure a smooth data set. Two different types of monitored data considered as *garbage* were identified:

1. Tags that have an *incorrect chip-type*.
2. Tags that are called *suspicious* because it was very likely that they were marked or monitored incorrectly by the students.

Both of these types are described below.

4.1.2.1. RFID Tags with incorrect Chip Types

In the METRO Group distribution center where the data collection took place, a variety of different RFID chip-types were in use that differ in impedance and sensitivity as described for example in [NSML09]. This means that different chip types are read with different signal strength and with a different frequency during a loading. Since the classification model presented in this thesis heavily relies on this information it is important to concentrate only on a single chip-type where a homogenous behavior can be expected. Table 4.1 shows the three different chip-types that were in use: Monza 2 and Monza 3 tags developed by Impinj, Inc., along with tags developed by NXP Semiconductors.

Very few tags (110 in total) are of type Monza 2, so they can be easily discarded from our data. The 7,797 NXP tags are much more significant, but even after removing these there are still more than 83,000 monitored tags remaining. For 1,134 monitored tags it was not possible

to determine the corresponding tag type; consequently they are removed as well. This leaves only the remaining set of 83,816 tags using a Monza 3 chip-type as the final data set to continue with.

Table 4.1.: Number of monitored Pallets per Chip-Type

Chip-Type	**Monitored Tags**
(Unknown)	1,134
Monza 2	110
Monza 3	83,816
NXP	7,797
Total	**92,857**

4.1.2.2. Suspicious Tags

The second type of garbage data is connected with the students who monitored the pallets. As with any activity involving manual work the monitoring was error-prone and the resulting data set therefore not completely inaccurate. Two different kinds of problems were identified where it is very likely that a student made a mistake by assigning the wrong class to a pallet (i.e., it was marked as "static" although the pallet was "moved" or the other way around). These data samples are called *suspicious tags*.

The first group of suspicious tags is called *never moved* tags. Suppose that every time a pallet was detected in any gathering-cycle a student was present to mark it as "moved" or "static". In several cases it is definitely known that this pallet has been shipped, for example because one of the destination markets confirmed its arrival. But, it has been marked as "static" every single time. This means that the student made a mistake in at least one of these gathering-cycles where it should have been marked as "moved". But because it is not known for sure which of the pallet occurrences was marked incorrectly, as a precaution all of them are removed from the sample dataset.

The second kind of suspicious tags are called *multiple moved* tags. This term is used to describe a pallet that has been marked as "moved" in *different* gathering-cycles (i.e., in different loadings). There is a slight chance that this might really happen, for example because a pallet has been removed from the container and was reloaded at some point later. However, because

it is not possible to decide whether this really was the case, all occurrences of a tag that has been marked as "moved" multiple times are removed from the data set.

Table 4.2.: Suspicious Tags

Monitored Tags	**Never Moved**	**Multiple Moved**
83,816	865	341

Table 4.2 shows the number of suspicious tags corresponding to each type. In total there were 1,206 monitored pallets that were either *never-* or *multiple moved*. This corresponds to 1.4% of the 83,816 Monza 3 tags monitored altogether. Accordingly, only the remaining 82,610 monitored pallets serve as the sample data set in the following.

4.2. Data Sources

The sample data set of 82,610 monitored pallets can further be separated into three major sub data sets. These correspond with the three different RFID portal types installed at the distribution center in Unna which differ in their configuration and functionality. The naming of the portal types, *Standard Portals*, *Satellite Portals* and *Transition Portals*, is based upon their general antenna adjustment which is described in more detail below.

Figure 4.3 depicts an outline of the distribution center where the data collection took place and Table 4.3 shows which portals belong to which portal type. Note that the Satellite Portals are also listed as Standard Portals and that not all portals were really used for the monitoring task. The reason why Standard- and Satellite Portals overlap is because portals 23-26 were used as Satellite Portals for only a short time before being rebuilt as Standard Portals. Thus, some of the data monitored at these portals belongs to the Standard- type and the rest belongs to the Satellite Portals. A complete list of how many pallets have been monitored at each individual portal can be found in the appendix (Tables B.1, B.2, B.3 and B.4).

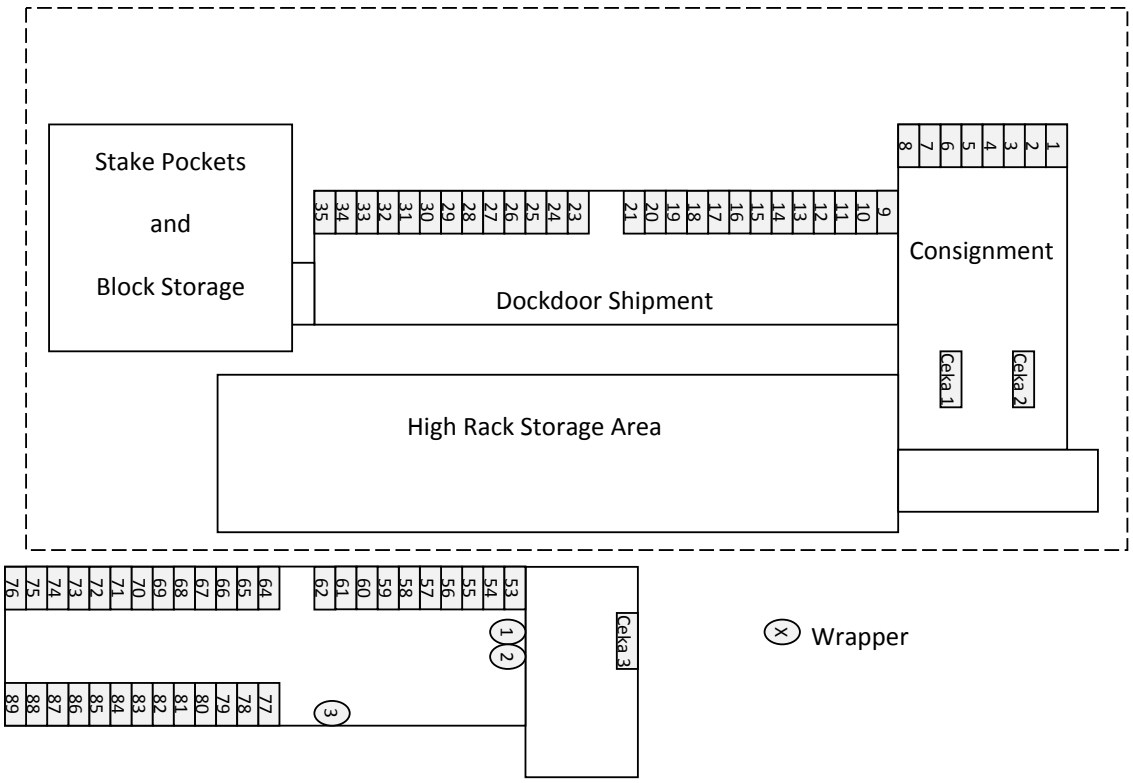


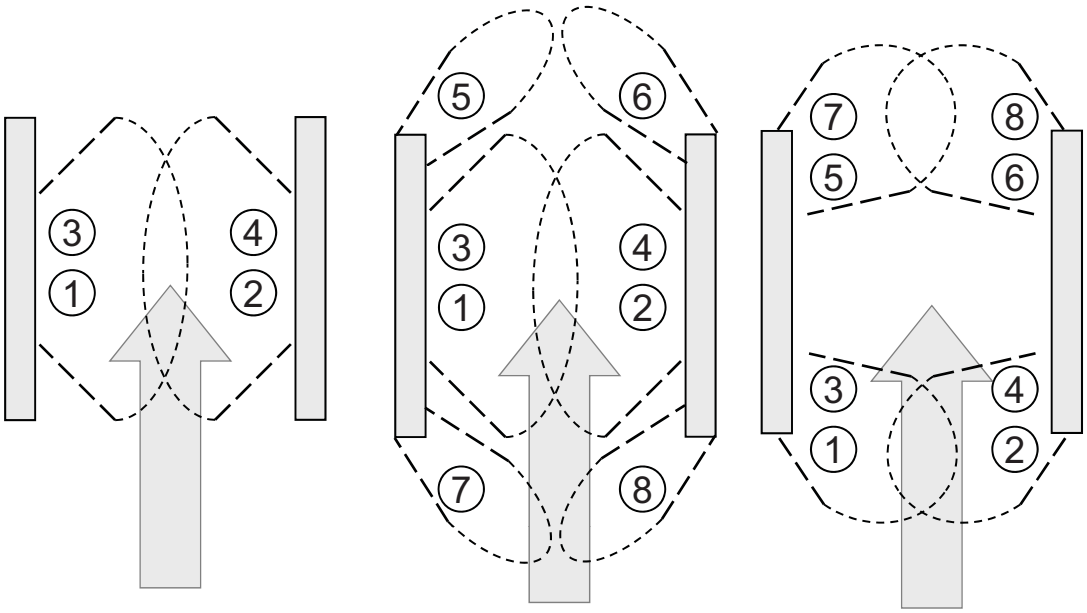
Figure 4.3.: Map of Portal Locations at METRO Distribution Center Unna, Germany

Table 4.3.: Portal Number Overview

Portal Type	**Portal Numbers**	**Total**
Standard Portals	9-21,23-62,64-89	79
Satellite Portals	23-26	4
Transition Portals	1-8	8

4.2.1. Standard Portals

The most commonly used type of RFID portal in the distribution center is the *Standard Portal*; its general antenna configuration is shown in Figure 4.4(a). A single reader is used with four different antennas (called *Main Antennas*) attached to it, two at each side of the portal, on top of each other and face to face with the other two. Antenna #1 is located bottom-left, antenna #2 bottom right, antennas #3 and #4 are located top left and right, respectively. The design is very similar to the RFID portal depicted in Figure 1.3.



(a) Standard Portal (b) Satellite Portal (c) Transition Portal

Figure 4.4.: Antenna Configuration of different Portal Types

The automatic recognition procedure of RFID tagged pallets passing a Standard Portal is shown in Algorithm 1. As soon as a warehouseman approaches the portal this is recognized by the motion sensor, triggering the RFID reader to start scanning for transponders in range. All four antennas start scanning simultaneously and every single tag detection is recorded until the stop trigger is activated and the scan is terminated. The stop trigger activation can be the result of two different events:

1. The motion sensor recognized that the warehouseman has left the container.
2. If the motion sensor does not recognize that the warehouseman has left the container within 10 seconds from the beginning of the gathering-cycle, it is terminated automatically.

Immediately afterwards, all tags that have been read during the loading are marked as "loaded" in the loading protocol.

4.2.2. Satellite Portals

Satellite Portals are an advanced version of the Standard Portals which use an additional RFID reader with 4 more antennas. Two of these antennas are directed toward the distribution center

Algorithm 1: Standard Portal Loading

```
output: Set T of tags read during a gathering-cycle
if Start trigger activated then
    repeat                                          /* Using Main Antennas */
        if Tag t has been recognized then
            if t ∉ T then Add t to T
    until Stop trigger activated
end
foreach Tag t ∈ T do
    Mark t as loaded in the loading protocol
end
return Tags T
```

(antennas #7 and #8) and the other two (antennas #5 and #6) are directed toward the truck - as can be seen in Figure 4.4(b). The four remaining (1-4) correspond to the antennas also used in the Standard Portals and are thus still referred to as *Main Antennas*. The additional antennas #7 and #8 are denoted *DC Antennas* and the other two (#5 and #6) are denoted *Truck Antennas*.

The automatic recognition procedure of RFID tagged pallets passing a Satellite Portal is shown in Algorithm 2. As soon as a warehouseman approaches the portal this is recognized by the motion sensor, triggering the RFID reader to start scanning for RFID tags in range. In addition to the four Main Antennas the two Truck Antennas start scanning simultaneously and every single tag detection is recorded until the stop trigger is activated and the scan is terminated. At that moment the DC Antennas start scanning for transponders still present in the distribution center. The idea behind this is that a tag that moved through the portal is expected to be inside the container rather than inside the distribution center. Consequently, all reads of tags that can be seen in the distribution center after the end of the actual gathering-cycle should be considered false-positive. The logic behind this antenna configuration leads to 7 disjunctive cases that can apply to a transponder detected during a specific loading.

Case 1 The tag has been read by Main-, Truck- and DC Antennas. This first case is interesting: this is a situation where after the loading has finished reflections can cause an effect where a tag can still be seen inside the distribution center even although it had previously moved through the portal and onto the truck. Tags that this case applies to are most likely false-positives.

Case 2 The tag has been read by DC- and Main Antennas only. This means that the tag has been seen during the loading but afterwards appears to be still in the distribution center. Tags in this Case are expected to be static, i.e., false-positives.

Case 3 The tag has been read by DC- and Truck Antennas but not by the Main Antennas. This case is similar to Case 1 since reflections cause the effect that the tag appears to be in two different locations. These tags are expected to be false-positives.

Case 4 The tag has been read by the DC Antennas only. Since the tag has neither been read by the Main- nor by the Truck Antennas, it has likely not been moved through the portal and thus is a false-positive.

Case 5 The tag has been read by Main- and Truck Antennas. This is like the first case, where a tag is expected to have been loaded into the container because it was seen during the loading and afterwards the DC Antennas signal that it is not in the distribution center anymore.

Case 6 The tag has been read by the Main Antennas only. This means that a tag has been read during the loading but then suddenly vanishes. Nevertheless the tag has apparently passed through the portal and is thus expected to be in the container.

Case 7 The tag has been read by the Truck Antennas only. These tags have not been read during the loading and consequently it is most likely that they have been loaded at a previous point in time and thus are false-positives.

4.2.3. Transition Portals

Like the *Standard Portals* the *Transition Portals* make use of two different readers, but they do not have any Main Antennas (see Figure 4.4(c)). The general idea is that the 4 antennas of the first reader (antennas 1-4) are directed toward the distribution center and the 4 antennas of the second reader (antennas 5-8) are directed towards the truck.

The automatic recognition procedure of RFID tagged pallets passing though a Transition Portal is shown in Algorithm 3. As soon as a warehouseman approaches the portal this is recognized by the motion sensor, triggering the RFID reader to start scanning for transponders in range. DC- and Truck Antennas begin scanning simultaneously and every single tag detection is recorded until the stop trigger is activated and the scan is terminated. The idea behind this is that a tag that moved through the portal is expected to be seen first by the DC Antennas

Algorithm 2: Satellite Portal Loading

output: Set T of tags read during a gathering-cycle
if *Start trigger activated* **then**
  repeat /* Using *Main-* and *Truck* Antennas */
    if *Tag t has been recognized* **then**
      if $t \notin T$ **then** Add t to T
  until *Stop trigger activated*
  repeat /* Using *DC* Antennas */
    if *Pallet t has been recognized* **then**
      if $t \in T$ **then** Remove t from T
  until *Stop trigger activated*
end
foreach *Tag* $t \in T$ **do**
  Mark t as loaded in the loading protocol
end
return *Tags T*

and afterwards by the Truck Antennas. Accordingly, all tags seen only by the DC- or only by the Truck Antennas should be considered false-positive reads. The logic behind this antenna configuration leads to 3 disjunctive cases that can apply to a transponder detected during a specific loading.

Case 1 The tag has been read by the DC Antennas only. Since it has not been read by the Truck Antennas it is likely to be a false-positive tag.

Case 2 The tag has been read by the Truck Antennas only. Similar to Case 1 it is likely to be a false-positive.

Case 3 The tag has been read by the DC- Antennas as well as by the Truck Antennas. It is most likely that this tag has been moved through the portal and thus has been loaded into the container.

4.3. Data Set Compilation

On the basis of the three different portal types described above the three major data sets are compiled. In principle, each portal type is represented by its own data set. However, Satellite- and Transition Portals lead to different cases that transponders can apply to, so these in turn are interpreted as distinct sub data sets. Table 4.4 shows the portal types including possible cases, the particular antennas that read the tag and the denomination of the respective data sets.

Algorithm 3: Transition Portal Loading

output: Set T of tags read during a gathering-cycle
if *Start trigger activated* **then**
  repeat /* Using *DC-* and *Truck* Antennas */
    if *Tag t has been recognized* **then**
      if $t \notin T$ **then** Add t to T
  until *Stop trigger activated*
end
foreach *Tag* $t \in T$ **do**
  if *t has not been read by Main Antennas* **then** Remove t from T
  if *t has not been read by Truck Antennas* **then** Remove t from T
end
foreach *Tag* $t \in T$ **do**
  Mark t as loaded in the loading protocol
end
return *Tags T*

Table 4.4.: Data Set Denomination

Portal Type	**Involved Antennas**	**Data Set**
Standard Portals	Main Antennas	STD_COMPLETE
Satellite Portals (all cases)	Union of cases 1-7	SAT_COMPLETE
Case 1	Main-, Truck- and DC Antennas	SAT_ALL
Case 2	DC- and Main Antennas	SAT_DC_MAIN
Case 3	DC- and Truck Antennas	SAT_DC_TRUCK
Case 4	DC Antennas only	SAT_DC_ONLY
Case 5	Main- and Truck Antennas	SAT_MAIN_TRUCK
Case 6	Main Antennas only	SAT_MAIN_ONLY
Case 7	Truck Antennas only	SAT_TRUCK_ONLY
Transition Portals (all cases)	Union of cases 1-3	TRA_COMPLETE
Case 1	DC Antennas only	TRA_DC_ONLY
Case 2	Truck Antennas only	TRA_TRUCK_ONLY
Case 2	DC- and Truck Antennas	TRA_BOTH

4.3.1. Standard Portals

The number of moved and static tags monitored at the Standard Portals including the resulting false-positive rate is shown in Table 4.5.

Table 4.5.: Monitored Pallets at the Standard Portals

Data Set	Moved Tags	Static Tags	Total Tags	False-Positives
STD_COMPLETE	13,245	40,743	53,988	75.47%

A total of 53,988 pallets were observed by the students, of which 40,743 were false-positives. This meant that on average there are slightly more than 3 static tags per moved tag read during a gathering-cycle which corresponds to a false-positive rate of 75.47%. It is obvious then, that in order to attain a reliable and fully functional RFID enabled outgoing goods process these false-positives need to be filtered out.

4.3.2. Satellite Portals

The number of moved and static tags monitored at the Satellite Portals, including the false-positive rate, is shown in Table 4.6. Based upon the possible cases defined in Section 4.2.2 the respective data is shown for each individual sub data set.

A total of 14,777 pallets were monitored by the students, of which 12,806 were false-positives. This means that on average there are almost 6.5 static tags per moved tag read during a gathering-cycle which corresponds to a false-positive rate of 86.66%. The general idea of the Satellite Portals was that tags that were read by the DC Antennas are expected to be static. These tags can be found in the following data sets:

- SAT_ALL,
- SAT_DC_MAIN,
- SAT_DC_TRUCK and
- SAT_DC_ONLY.

In sum there are 3,247 tags in these data sets of which 3,225 were static. This corresponds to a false-positive rate of 99.32%. Of the 5,122 tags that were read only by the Truck Antennas

99.79% were false-positives. Because of the very high false-positive rate, each tag that applies to one of the above cases is automatically considered to be a false-positive. The remaining data sets of interest are

- SAT_MAIN_TRUCK and
- SAT_MAIN_ONLY.

For these tags only, a classification model needs to be constructed.

Table 4.6.: Monitored Pallets at the Satellite Portals

Data Set	Moved Tags	Static Tags	Total Tags	False-Positives
SAT_ALL	10	279	289	96.54%
SAT_DC_MAIN	12	784	796	98.49%
SAT_DC_TRUCK	0	40	40	100.00%
SAT_DC_ONLY	0	2,122	2,122	100.00%
SAT_MAIN_TRUCK	1,282	1,899	3,181	59.70%
SAT_MAIN_ONLY	656	2,571	3,227	79.67%
TRUCK_ONLY	11	5,111	5,122	99.79%
SAT_COMPLETE	1,971	12,806	14,777	86.66%

4.3.3. Transition Portals

The number of moved and static tags monitored at the Transition Portals, including the false-positive rate, is shown in Table 4.7. Based upon the possible cases defined in Section 4.2.3 the respective data is shown for each individual sub data set.

A total of 13,845 pallets were observed by the students, of which 12,487 were false-positives. This means that on average there are around 9.2 static tags per moved tag read during a gathering-cycle which corresponds to a false-positive rate of 90.19%. The general idea of the Transition Portals is that transponders that were seen either by only the DC- or only by the Truck Antennas are expected to be static. These tags can be found in the

- TRA_DC_ONLY and
- TRA_TRUCK_ONLY

Table 4.7.: Monitored Pallets at the Transition Portals

Data Set	Moved Tags	Static Tags	Total Tags	False-Positives
TRA_DC_ONLY	28	5,779	5,807	99.52%
TRA_TRUCK_ONLY	31	4,316	4,347	99.29%
TRA_BOTH	1,299	2,392	3,691	64.81%
TRA_COMPLETE	1,358	12,487	13,845	90.19%

data sets. In sum there are 10,154 reads in these data sets of which 10,095 were static. This corresponds to a false-positive rate of 99.42%. Because of the very high false-positive rate, each tag that applies to one of the above cases is automatically considered to be a false-positive. The remaining data set of interest is

- TRA_BOTH.

For these tags only, a classification model needs to be constructed.

4.3.4. The Final Data Sets

The separation of the monitored data into distinct subsets revealed a number of cases where the application of a classification model is unnecessary because of the very high false-positive rate. For example, every single one of the 2,122 tags that were detected only by the DC Antennas of the Satellite Portals were false-positives, thus tags corresponding to this case can be classified as "static" by definition. The relevant data sets for which a classification model needs to be generated are shown in Table 4.8.

Table 4.8.: Sample Data in the Relevant Data Sets

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

4.4. Summary

This chapter comprised three major parts. First of all, the process whereby the students collected data by monitoring the loading of pallets into containers was described. This exercise ran for almost seven months at the METRO central distribution center in Unna, Germany, and comprised 92,857 pallet observations. It was shown that the major proportion of these were actually false-positive tag reads, i.e., pallets that had not been loaded into containers but were present in the reading field of the antennas only by accident. Under the assumption that any manual work is error-prone, two types of *suspicious tags* were identified that were obviously monitored incorrectly. The first type is denoted *never moved* tags and corresponds to those tags that have never been marked as "moved" although it is known that they were loaded at some point. The second group is denoted *multiple moved* tags and corresponds to those tags that have been marked as "moved" (i.e., as loaded) multiple times. In addition, any transponders using a chip-type other than Monza 3 were removed in order to guarantee a homogenous and meaningful sample data set.

The second part introduced the three different portal types (*Standard-*, *Satellite-* and *Transition Portals*) in use at the distribution center. While *Standard Portals* are the most common and most intuitive portals with 4 antennas scanning whatever moves through the portal, the other two are more advanced versions with additional readers and antennas. These other two types furthermore, encompass an additional but rather simple procedure to filter out false-positive RFID tags on a logical level.

Satellite Portals have antennas that also scan what is inside the truck at the same time the Main Antennas try to read what moves through the portal. Afterwards two more antennas directed towards the distribution center are used to double-check which pallets that have been read before can still be seen there. Any tag that is still present in the distribution center is considered a false-positive.

Transition Portals meanwhile, have four antennas directed towards the distribution center and four more directed towards the truck. It is expected that any pallet moving through this portal is read first at the distribution center and afterwards inside the truck. Pallets read only inside the truck or only in the distribution center are considered false-positives.

The third part was about the compilation of the final data sets for which a classification model is going to be generated. First of all, there is the set comprising the pallets monitored at the Standard Portals and denoted as `STD_COMPLETE`. The second data set comprises the pallets monitored at the Satellite Portals and can be further divided into two disjunctive subsets denoted as `SAT_MAIN_ONLY` and `SAT_MAIN_TRUCK`. As the names indicate, the first set comprises

the tags that were read only by the *Main Antennas* of the Standard Portals and the latter set comprises tags that were read by both *Main-* and *Truck Antennas*. The last data set, `TRA_BOTH`, comprises the tags read by both *DC-* and *Truck Antennas* at the Transition Portals.

The problem of false-positive RFID tag reads could be observed regardless of the type of portal. During the monitoring exercise, a large amount of data was collected for each of the portals; this will be inputted into the classification model in the following chapters.

5. Classification Model Building

It was stated in Section 3.3 that the training of a classification model requires a description of the objects to be classified by means of so called *attributes*. The performance of any classification model depends highly upon the ability of these attributes to successfully map the object's characteristics. Consequently, *attribute identification* and *-evaluation* are the most important tasks when building a classification model and require a dedicated investment of time and effort.

Accordingly, this chapter can be regarded as the core of the thesis as it describes the attributes required for the definition of a classification model framework to distinguish between moved and static pallets. It is organized as follows: first of all, the type of classification model chosen, *Decision Trees* is introduced and described in detail; next, the attributes and their generation on the basis of the *Tag-Event-* and the *Tag-Occurrence Level* is described. Because both approaches result in independent classification models with different strengths and weaknesses an additional approach is presented that combines the two of them.

The chapter closes by proposing a fourth approach that relies on the attributes presented here but can only be used if it is known in advance how many tags need to be classified as moved (or static, depending on the scenario).

5.1. Classification using Decision Tree Learning

The overall target of the classification model is to decide whether a pallet has been loaded into the container or not, solely on the basis of the low-level reader data collected during a gathering-cycle. Among several available classification models, *Decision Tree Classification* [Coh95] was identified as the most appropriate approach to take in the scenario under consideration.

A decision tree is a powerful and commonly used machine learning technique that performs a sequence of attribute value tests to ultimately determine a classification. In contrast to many other machine learning techniques (e.g., Neural Networks [Zha00], Naive Bayes Learning [Hec95] or Support Vector Machines [Bur98]), Decision Tree Classification uses a white-box model, which allows the user to easily replicate classification results. It is this core property in particular, that motivated the decision to use decision tree classification over other approaches.

5.1.1. An Example Decision Tree

An example decision tree is given in Figure 5.1, where each object is tested against the attribute values of attributes A, B, C and D while descending the tree from the top node (called *root node* or simply *root*) to the bottom nodes (called *leaves*). The complexity of a tree is usually described by its depth for the actual classification. Since this depth depends on the maximum number of tests that have to be performed, the one shown here is a depth-3 decision tree. The leaves correspond to the final classification of a pallet.

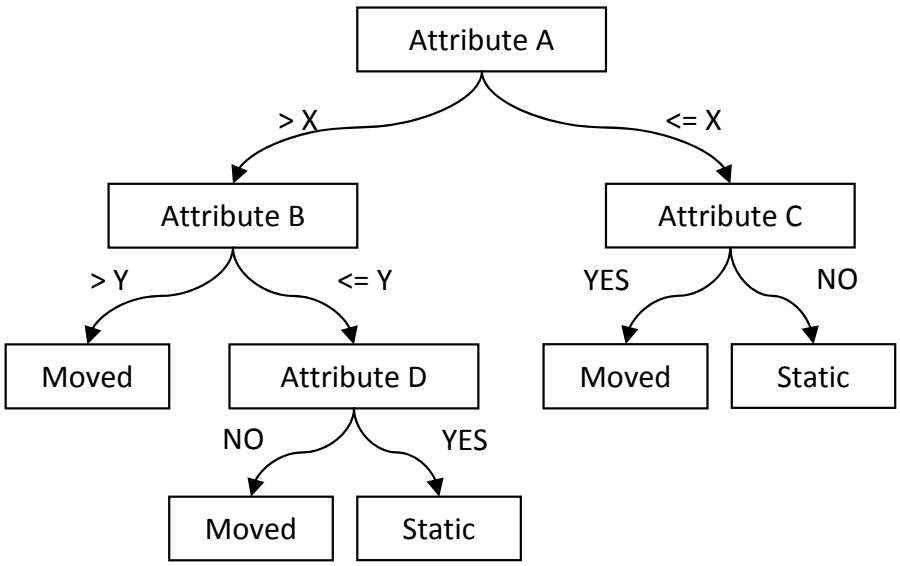


Figure 5.1.: Example Decision Tree

Note that in this example some attributes have a different attribute value type: Attributes A and B are numerical type and, therefore, two thresholds X and Y are tested. Attributes C and D, however, describe a nominal characteristic that takes on only the values *Yes* or *No*. The type of decision tree shown in this figure is called a *binary decision tree* because there is a binary decision made at each node leading to exactly two outgoing branches per attribute. Of course, a decision tree is not limited simply to this. For example, a nominal attribute might be used that takes on 3 different values, e.g., *Yes*, *No* and *Unknown*. In this case there would be three outgoing branches from the corresponding node.

A path from the root to a leaf is called a *rule*. For example, Algorithm 4 shows the rule evaluation for the path from the root to the second left-most leaf. All tags that this rule applies to (i.e., that pass the 3 attribute tests) are said to be *covered* by that rule.

At this point it is reasonable to focus on the question of why at specific nodes a certain attribute is chosen in favor of another one. So why, for example, is attribute A evaluated at the

Algorithm 4: Example Decision Tree Rule Evaluation

if *Attribute* $A > X \wedge$ *Attribute* $B \leq Y \wedge$ *Attribute* $D = No$ **then**
    Moved
end

root level and not attribute B or attribute C? The answer to this question is that any classification algorithm is trying to generate a model that is most confident with its classifications, i.e. in terms of decision trees, that has leaf nodes as *pure* as possible. Consequently, the attributes used in the tree are not selected randomly but constitute the results of the attempt to grow the tree with the purest leaves. Algorithm 5 provides the general pseudo-code for building a decision tree [Kot07].

Algorithm 5: Decision Tree Building

if *All samples belong to the same Class* C **then**
    return *Root Node*
end
foreach *Attribute* A **do**
    Calculate the ability to use A as the Attribute to split on
end
Let A_{Best} be the best attribute to split on.
Create a decision node N that splits on A_{Best}
Recurse on the sub-lists obtained by splitting on A_{Best} and add those nodes as children of N

5.1.2. Rule Expressiveness

In Section 3.1.2 it was mentioned that a key business objective for this research was to understand how and why the model decides the way it does. With this objective in mind, some kind of statistical measure is required to describe the quality of an individual classification rule rather than the entire model. Three measures are commonly used in machine learning and especially in the context of *association rule mining* (e.g. [SON95, LAR02]), these are denoted as *confidence*, *support* and *completeness*. These measures can also be applied to the rules of a decision tree without problems.

Every rule (or path in a decision tree) has by definition the following form:

if *Condition A* **then** *Classification C.*

Note that A can easily correspond to multiple conditions (compare Algorithm 4). Let N_{Cond} denote the number of samples matching condition A and N_{Class} denote the number of samples of class C. Furthermore, let denote N_{Both} the number of samples matching both condition A and class C and N_{Total} denote the total number of samples. The meaning of the statistical measures can then be explained as follows:

Confidence This is a measure of how confident the rule is, i.e., how many of the samples covered by that rule are classified correctly:

$$Confidence = \frac{N_{Both}}{N_{Cond}}$$

Support This is a measure of generality, i.e., how many of the total number of samples are covered by that rule and are classified correctly:

$$Support = \frac{N_{Both}}{N_{Total}}$$

Completeness This is another measure of generality, i.e., how many of the samples of a specific class are covered by that rule and are classified correctly:

$$Completeness = \frac{N_{Both}}{N_{Class}}$$

After the rule evaluation measures are introduced, in the following section two approaches to decision tree learning are presented. Called *C4.5* and *CART* , they can be used independently of each other to discriminate between moved and static pallets. The difference between them lies in how they select attributes and corresponding thresholds at the individual nodes. However, in most cases it is a good idea to use both of them and then evaluate which one returns the best classification model.

5.1.3. C4.5 Algorithm

The *C4.5 algorithm* [Qui93] is a more advanced version of the seminal *ID3 algorithm* presented in 1986 [Qui86]. For each node in the decision tree, C4.5 determines the optimal attribute by making use of two common concepts from information theory and machine learning, called *entropy* and *information gain* (also known as Kullback-Leibler divergence) [HMS66, Lar05c].

In information theory the term *entropy* is used to measure the amount of *uncertainty* contained in a data set because of the presence of multiple object classes (in this case moved and static tags). Because an in-depth knowledge of data coding techniques is required to fully

understand the theory behind entropy, only a brief introduction is given in this thesis; a very good introduction to the concepts of entropy and information gain can be found in [Bra07b] and [Bra07c]:

Suppose a data set D contains objects of two different classes O and P with frequencies p_O and p_P. If an object of class O is drawn randomly then the probability for each object $o \in O$ to be selected is $\frac{1}{p_O}$. To distinguish these individual objects $\log_2(\frac{1}{p_O}) = -\log_2(p_O)$ bits are required. This is the *called self-information* of class O that can further be weighted by p_O to acquire the *average self-information* of class O. Summing up the average self-information of every class results in the entropy H of the entire data set:

$$H(D) = -\sum_{i=1}^{n} p(i) \log_2 p(i)$$

Without loss of generality, the dataset collected at the standard portals is used to illustrate these calculations. There are two different classes (moved and static) in the data set D with 75.8% static and 24.2% moved tags. Thus, the entropy of the dataset equals

$$H(D) = -(0.7542 \cdot \log_2 0.7542 + 0.2458 \cdot \log_2 0.2458) = 0.805$$

Let A be an attribute and x a threshold under consideration. Then $D_{\leq}$ is the set containing all tags having a value for attribute A less than or equal to x and $D_{>}$ is the set of all tags with an attribute value greater than x:

$$D_{\leq}(A, x) := \{Tag \in D | Tag(A) \leq x\} \qquad D_{>}(A, x) := \{Tag \in D | Tag(A) > x\}$$

The information gain can now be interpreted as a measure of how well the two tag types can be separated using a specific value as threshold and is a common method in machine learning [GH07]. Illustrated below is how the information gain obtained for specific values on the basis of the entropy is used to determine the best attribute and the optimal threshold to separate moved tags from the false-positives. The information gain I obtained by splitting attribute A on x is defined as

$$I(A, x) = H(D) - \left[\frac{|D_{\leq}(A, x)|}{|D|} \cdot H(D_{\leq}(A, x)) + \frac{|D_{>}(A, x)|}{|D|} \cdot H(D_{>}(A, x)) \right].$$

The optimal threshold o to separate moved and static tags is the one where the information gain is maximized, i.e.,

$$o(A) = \max_{x \in X} \{I(A, x)\}.$$

If there were a perfect attribute, i.e., one with a threshold value perfectly splitting static and moved tags, the maximum information gain of the attribute is equal to the entropy of the dataset, 0.805. In C4.5 the preceding calculations are repeated for every available attribute. The attribute with the highest information gain is then returned as the best attribute to split on.

5.1.4. CART Algorithm

A further approach is based on Breiman's *Classification and Regression Trees* [BFSO84]. Just like before, the algorithm tries to determine the attribute to split on by evaluating all possible threshold values. However, here it uses a different way to measure the *goodness* of a candidate threshold to split on [Lar05c].

Let D be the set of all sample data and x a threshold under consideration for an attribute A. Without loss of generality, let $D_{\leq}$ and $D_{>}$ be the set of tags with an attribute value less than and greater than x, respectively. Then the quality measurement G of splitting an attribute A on x is calculated as

$$G(A, x) = 2 \cdot P_{\leq} \cdot P_{>} \cdot \sum_{c=1}^{n} |P(c|D_{\leq}) - P(c|D_{>})|$$

where c is the number of classes and $P_{\leq}$ and $P_{>}$ correspond to the weights of the two subsets. $P(c|D_{\leq})$ and $P(c|D_{>})$ are defined as

$$P(c|D_{\leq}) = \frac{|\{Tag \in D_{\leq}| \text{ class of } Tag = c\}}{|D|}$$
$$P(c|D_{>}) = \frac{|\{Tag \in D_{>}| \text{ class of } Tag = c\}}{|D|}.$$

The optimal threshold value used to split on attribute A is the one that maximizes G. These calculations are repeated for every attribute and the attribute with the highest *goodness* value G is then returned as the best attribute to split on.

5.1.5. Overfitting

A very important issue that needs to be dealt with with any classification model, including the decision trees created using the two above approaches, is the problem of *overfitting*. In [Bra07a] it is stated that a classification model "is said to overfit to the training data if it generates a decision tree [...] that depends too much on irrelevant features of the training instances, with the result that it performs well on the training data but relatively poor on unseen instances".

In the context of the scenario presented in this thesis, this means that an overfit classification model would have a very low error rate with respect to the sample data but would not perform well in a productive system because it won't classify future pallet loadings correctly.

There is a significant relationship between the complexity of a classification model and its error rate as can be seen in Figure 5.2. In Section 3.3.1.2 the concept of *train and test* was described to divide the sample data into a *training set* and a *test set* (also called *validation set*). This figure shows how the error rate in the training data decreases with increasing model complexity. In the beginning, the same applies to the test data but at some point this error rate increases again when the classification model becomes too complex. This is exactly the point where the model begins to overfit to the training data. Consequently, the optimal model complexity is where the error rate in the test data is minimal. However, the complexity of the model must also not be too restrictive because in this case, the model is said to be *underfit* and won't show optimal performance on the training or on the test sets.

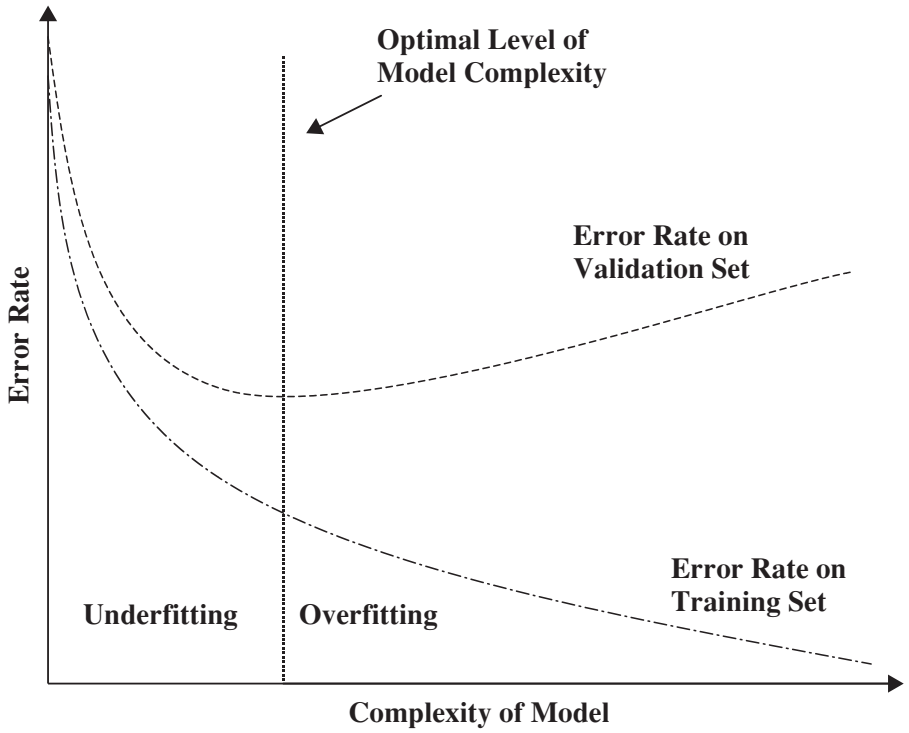


Figure 5.2.: Relationship between Model Complexity and Overfitting. Source: [Lar05a]

Summing up, the complexity of a decision tree must be limited in order to avoid or minimize the effect of overfitting. This is often done by *pruning*, i.e., simplifying the decision tree by removing certain branches or nodes. There are two different types of pruning strategies, *pre-pruning* and *post-pruning*; a good overview is given in [Mar97]. Pre-pruning means that at

some point during the decision tree generation phase it is decided to not separate the dataset any further. This requires some kind of stopping criterion to decide whether it is reasonable to continue the tree generation at this point or not.

There are several stopping criteria that can be used with *size cutoff* and *maximum depth cutoff* being two of them [Bra02]. Size cutoff avoids further tree generation if the resulting nodes contain less than a given threshold number of objects. Maximum depth cutoff stops the tree generation if the rule length (i.e., the tree depth) is longer than a predefined threshold. In this thesis, the decision trees try to ensure a number of at least 5% of the total sample data per leaf and are ultimately limited to a depth of 4. However, in some cases the leaf size constraint cannot be guaranteed, for example if there is less than 5% of the sample data left to finish the tree generation.

In contrast, post-pruning involves building a complete decision tree and then removing specific branches and nodes to simplify the tree and to improve the classification rate on the test set. There are a number of different post-pruning strategies available. The C4.5 algorithm uses an approach called *error based pruning* [Qui93], while CART's pruning strategy is called *cost complexity pruning* [BFSO84]. Because these two are used in our proposed decision tree learning techniques they are described in the following sections.

5.1.5.1. Error Based Pruning

Error based pruning takes the *error of an inner node* and the *error of the subtree* rooted at that node into account. For a node o containing N_o samples of which m_o belong to the majority class (i.e., the class with the most representatives in that node) its error E_o is estimated by

$$E_o = \frac{N_o - m_o + 0.5}{N_o}.$$

For a subtree S originating at node o and with k leaf nodes the error E_S is calculated by

$$E_S = \frac{\sum_{i=1}^{k} (N_i - m_i + 0.5)}{\sum_{i=1}^{k} N_i}.$$

The standard error of the subtree, SE_S, is estimated by

$$SE_S = \sqrt{\frac{E_S \cdot (1 - E_S)}{\sum_{i=1}^{k} N_i}}.$$

If the error of the subtree plus the standard error is greater than the error at that inner node,

then the subtree is entirely pruned. Accordingly, subtree S is pruned if

$$E_o \leq E_S + SE_S$$

holds true. In this case the inner node o becomes a leaf node classifying each object as the majority class in o.

5.1.5.2. Cost Complexity Pruning

In contrast to error based pruning, cost complexity pruning requires a test set and tries to find a good trade-off between the error estimated in the test set and the complexity of the decision tree. The complexity C_S of a tree S is determined by the number of leaves within that tree. If o is an inner node serving as the root for S, then E_o and E_S correspond to the error of o and S, respectively. If α is some real number and a measure of tree complexity, the total cost $Cost_S$ of S is then estimated by

$$Cost_S = E_S + \alpha \cdot C_S$$

If S is pruned and replaced by a node o then the costs $Cost_o$ of node o are estimated by

$$Cost_o = E_o + \alpha$$

Equalizing the two cost functions leads to

$$\alpha = \frac{E_o - E_S}{C_S - 1}$$

Cost complexity pruning involves calculating α for every node. Starting with the original tree having the maximum size, a sequence of trees is generated by pruning the subtree with the lowest value of α in each step. The algorithm stops with the minimal tree containing only the root node. This sequence of trees is then ranked according to the cost complexity and the best trade-off between complexity and number of nodes has to be found. There are several rules available to estimate the best trade-off. For example, in the original work, Breiman et al. proposed selecting the tree within one standard error of the minimum cost tree that also has the least number of nodes.

5.1.6. Attribute Type Categorization and Investigation

In the preceding sections the foundations of decision tree learning, including evaluation of rule expressiveness and avoidance of overfitting, were explained. The next step is the identification

of attributes that can be used to describe moved and static pallets and that serve as input to the decision trees.

Each tag-event t, i.e., each single detection of a tag during a gathering-cycle can be represented as a three-tuple of signal strength, timestamp and antenna:

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

Next, if a specific RFID tag has been detected n-times during a gathering-cycle, the corresponding tag-occurrence T as the whole of these tag-events can be represented as

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

On this basis the following types of attributes can be distinguished:

Domain Attributes These are characterizations of moved and static pallets mainly based on the manual observation of pallet loadings in the distribution center. Depending on the type of low-level reader data used they have different denominations.

RSSI Attributes These attributes are intuitive aggregations of the RSSI values. For example, the maximum signal strength a tag has been read with is defined as:

$$RSSI_{Max} := \max\{RSSI_1, \ldots, RSSI_n\}$$

SinceStart Attributes These attributes are intuitive aggregations of the SinceStart values. For example, the time since the beginning of the gathering-cycle that passed before a tag is first detected is defined as:

$$Read_{First} := \min\{SinceStart_1, \ldots, SinceStart_n\}$$

Antenna Attributes These attributes are intuitive aggregations of the Antenna values. For example, the number of detections involving antenna 1 is defined as:

$$AntCount_1 := |\{Antenna_i | 1 \leq i \leq n \wedge Antenna_i = 1\}|$$

Artificial Attributes These characterizations are automatically derived using a sequence of operators on arbitrary Domain Attributes. In contrast to the latter they do not have an intuitive semantic. For example, an artificial attribute A could be defined as:

$$A := \frac{RSSI_{Max}}{RSSI_{Mean}} + \sqrt{RSSI_{Min}}$$

where $RSSI_{Mean}$ and $RSSI_{Min}$ denote the average and the minimum received signal strength during that gathering-cycle.

Logical Reader Attributes These characterizations describe the order in which a tag was read by different readers of the same portal. Because only Satellite- and Transition Portals have multiple readers, these attributes are not available at the Standard Portals.

Time-Series Attributes These characterizations describe the development of the signal strength over the period of a gathering-cycle. Generally, it is examined whether this development is more typical of a moved or of a static tag.

As stated previously, two different approaches, *Tag-Occurrence Level Classification* and *Tag-Event Level Classification* are proposed in this thesis to discriminate between moved and static tags. The particular difference between these two lies in the type of attributes they use and the way they are ultimately calculated. Considering the above attributes it will become clear in the following sections that the underlying calculations of *Domain-*, *Artificial-* and *Logical Reader Attributes* are very similar. Furthermore they are all based on the same idea of calculating specific aggregation functions over the entirety the individual tag-events, i.e., on the tag-occurrences. Consequently, these attributes types are pooled in the *Tag-Occurrence Level Approach*. The Time-Series Attributes on the other hand, are pooled in the *Tag-Event Level Approach*.

In the following sections a list of attributes for each of the above attribute types is presented. Furthermore, the underlying rationale is described and their principal applicability for classification purposes is discussed. The analysis is based on the descriptive statistics of the collected sample data in the relevant data sets shown in Table 4.8. Because the different portal types have different readers it is furthermore necessary to distinguish between them, so all affected attributes are indexed by the corresponding reader *DC*, *Main* or *Truck* where the tag-events occurred (e.g., $RSSI_{Max,Main}$ for the maximum signal strength measured by the Main Antennas).

The above attributes can be of either *numerical-* or *nominal type* and thus need to be examined in different ways. *Numerical attributes* can be investigated by calculating the minimum,

maximum and average attribute value of each attribute together with the standard deviation. This is done for both moved and static tags and all four data sets. These measures are shown in Figure 5.3 where the distribution of the average RSSI values is depicted for both moved and static tags monitored at the Standard Portals. Note that the minimum and maximum attribute values shown in this Figure do not represent the absolute minima and maxima, as due to outliers or incorrectly monitored pallets the respective minima and maxima for moved and static tags are the same or at least very close to each other. Consequently, the attribute value a is chosen as the minimum representative where 99% of the values lie above it and the attribute value b where 99% of the values lie below it is chosen as the maximum value. In descriptive statistics, a and b are often called the 1st and the 99th percentile, respectively.

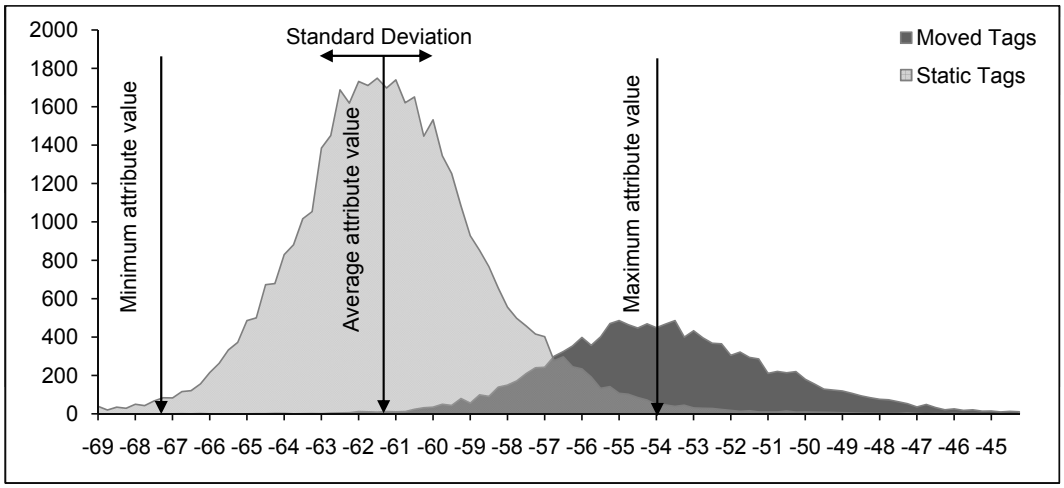


Figure 5.3.: Example Numerical Attribute Investigation

Because *nominal attributes* have by definition a discrete value space, the investigation can be carried out without using the above statistical measures. The whole distribution can easily be presented in a way similar to the confusion matrices described in section 3.3.1.3 on page 41.

Table 5.1 shows an example investigation of a nominal attribute called $Seen_{First}$ corresponding to the specific reader that first sees a tag at the Transition Portals. The values shown there can be interpreted as follows: 1,191 of all moved tags were seen first by the DC Antennas and 1,370 of all static tags were seen first by the Truck Antennas, this corresponds to a moved detection rate of 91.7% and a static detection rate of 57.3%. The confidence of classifying a tag correctly (i.e., the classification precision) is 53.8% if it was first seen by the DC Antennas and 92.7% if was seen first by the Truck Antennas.

Table 5.1.: Example Nominal Attribute Investigation

Attribute	Value	Moved Tags	Static Tags	Precision
$Seen_{First}$	DC	1,191	1,022	53.8%
	Truck	108	1,370	92.7%
	Recall	91.7%	57.3%	

5.2. Tag-Occurrence Level Classification

Tag-occurrence Level Classification is the first approach presented in this thesis to identify meaningful attributes that can be used to discriminate between moved and static RFID tag reads. As stated above, these attributes can be separated into disjunctive subgroups denominated *Domain-*, *Artificial-* and *Logical Reader Attributes*. In the following sections the different attributes belonging to each of the groups are identified, described and investigated.

5.2.1. Domain Attributes

Domain Attributes are derived from the experience and knowledge of people working in the environment under consideration. For example, if we talk to a warehouseman and ask him about the difference between moved and static pallets, he might answer that the first get closer to the RFID antennas than the latter do. Based on this information and the characteristics of the RSSI data we would then conclude that the maximum RSSI value measured during a gathering-cycle poses a valuable attribute to distinguish moved and static pallets because the first are expected to show a higher attribute value than the latter.

5.2.1.1. RSSI Attributes

In accordance with the above explanations, a tag-occurrence T can be described as a sequence of individual tag-events:

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

where n is the number of tag-events. Calculation of the *RSSI Attributes* is based on the unordered set of the corresponding RSSI values:

$$\{RSSI_1, \ldots, RSSI_n\}$$

The main point behind the definition of these attributes is the observation that many static pallets are further away from antennas than moved pallets are. Because the received signal strength indication depends on the distance between sender and receiver it is expected that the RSSI Attributes are able to successfully map this insight and that they play a major role in being able to discriminate between moved and static tags.

$\mathbf{RSSI_{Min}}$ The minimum signal strength measured during a gathering-cycle. Because at the beginning of the pallet loading process both moved and static tags can be far away from the antennas, no significant differences in the minimum signal strength were expected. However, it was found that moved pallets seem to have a slightly higher minimum RSSI value than static pallets do.

$$RSSI_{Min} := \min\{RSSI_1, \ldots, RSSI_n\}$$

$\mathbf{RSSI_{Max}}$ The maximum signal strength measured during a gathering-cycle. Because moved pallets pass the portal and therefore have a smaller distance to the antennas at this time, they are expected to have a higher maximum RSSI value.

$$RSSI_{Max} := \max\{RSSI_1, \ldots, RSSI_n\}$$

$\mathbf{RSSI_{Diff}}$ The difference between the highest and lowest signal strength that was 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 antennas they are expected to have a higher dispersion and thus a higher $RSSI_{Diff}$ attribute value than static tags.

$$RSSI_{Diff} := RSSI_{Max} - RSSI_{Min}$$

$\mathbf{RSSI_{Mean}}$ The average signal strength measured during a gathering-cycle. Because moved pallets spend more time closer to the antennas while they pass the portal it is expected that they have a higher average RSSI value than static pallets.

$$RSSI_{Mean} := \left(\sum_{i=1}^{n} RSSI_i\right) \cdot \frac{1}{n}$$

$\mathbf{RSSI_{StDev}}$ The standard deviation of the RSSI values. Similar to $RSSI_{Diff}$ this is a dispersion measure and therefore a higher attribute value is expected for moved pallets than for static pallets.

$$RSSI_{StDev} := \sqrt{\sum_{i=1}^{n}(RSSI_i - RSSI_{Mean})^2}$$

$\mathbf{RSSI_{CoV}}$ The coefficient of variation of the RSSI values, it is defined as the ratio between 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 is expected to take on higher values for moved pallets the coefficient of variation is expected to be lower for these than for static pallets.

$$RSSI_{CoV} := \frac{RSSI_{StDev}}{RSSI_{Mean}} = \sqrt{\sum_{i=1}^{n}\left(\frac{RSSI_i}{RSSI_{Mean}} - 1\right)^2}$$

Table 5.2 shows the minimum, maximum, average and standard deviation of the RSSI attribute values for moved and static tags monitored at the *Standard Portals*. It is notable that moved tags are usually read with a significantly higher signal strength, as was expected. Consequently, $RSSI_{Max}$ and $RSSI_{Mean}$ take on higher values for moved tags than for static tags. It was also expected that moved tags are subject to a higher variance in the signal strength. This also holds true because the attributes $RSSI_{Diff}$ and $RSSI_{StDev}$, as a measure of variance, take on considerably lower values for static tags. The minimum received signal strength, denoted as $RSSI_{Min}$, does not show much divergence, as was also expected. However, another significant difference can be observed by examining the coefficient of variation. This attribute tends to be 0 for many static tags and is much lower for moved tags.

Table 5.3 shows the RSSI attribute values for moved and static tags monitored at the *Satellite Portals* that have been read only by the *Main Antennas*. It can be seen that the above expectations hold true also for this data set. Moved tags are read with a higher minimum, maximum and average signal strength than static tags. The standard deviation and the $RSSI_{Diff}$ attribute prove that static tags also have a lower signal strength variance than moved tags.

Table 5.4 shows the RSSI attribute values for moved and static tags monitored at the *Satellite Portals* that have been read by both *Main-* and *Truck Antennas*. Considering the attribute values that correspond to the Main Antennas it becomes clear that the above expectations still hold true. Moved tags are read by these antennas with higher minimum, maximum and average signal strengths than static tags. The standard deviation and the $RSSI_{Diff}$ attribute prove

Table 5.2.: RSSI Attribute Values (`STD_COMPLETE` Data Set)

Attribute	**Moved Tags**				**Static Tags**			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$RSSI_{Min,Main}$	-68.6	-53.4	-62.8	2.9	-69.4	-57.0	-63.8	2.4
$RSSI_{Max,Main}$	-56.5	-38.3	**-44.1**	4.2	-66.6	-47.3	**-59.1**	4.0
$RSSI_{Diff,Main}$	2.9	27.5	**18.7**	5.2	0.0	16.7	**4.7**	4.0
$RSSI_{Mean,Main}$	-60.7	-46.0	**-54.1**	3.0	-67.4	-54.2	**-61.5**	2.6
$RSSI_{StDev,Main}$	1.0	8.5	**5.3**	1.6	0.0	4.4	**1.4**	1.1
$RSSI_{CoV,Main}$	-0.164	-0.018	-0.100	0.031	-0.077	**0.000**	-0.023	0.018

Table 5.3.: RSSI Attribute Values (`SAT_MAIN_ONLY` Data Set)

Attribute	**Moved Tags**				**Static Tags**			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$RSSI_{Min,Main}$	-67.8	-50.7	-62.7	3.0	-66.9	-58.5	-63.4	1.8
$RSSI_{Max,Main}$	-55.6	-38.3	**-43.8**	4.2	-65.9	-50.3	**-59.5**	3.5
$RSSI_{Diff,Main}$	2.0	26.6	**18.9**	5.6	0.0	14.0	**3.9**	3.5
$RSSI_{Mean,Main}$	-60.8	-45.6	**-54.4**	3.0	-66.0	-56.2	**-61.5**	2.1
$RSSI_{StDev,Main}$	0.8	8.6	**5.4**	1.7	0.0	3.6	**1.2**	0.9
$RSSI_{CoV,Main}$	-0.165	-0.015	-0.101	0.034	-0.061	0.000	-0.020	0.016

that the latter also have a lower signal strength variation than moved tags. Looking at the attribute values that correspond to the Truck Antennas, the same observations can be made. However, the differences between static and moved tags in this situation are considerably less significant. Although the Main Antennas of the Satellite Portals are identical to the Main Antennas at the Standard Portals a slight variation can be observed due to the different tag populations they represent. Table 5.3 shows tags that were detected only by the Main Antennas, Table 5.4 shows the tags that are detected as moved by both Main and Truck Antennas and finally Table 5.2 shows some kind of average of these two tag types because using Standard Portals it is not possible for it to differ between the other cases due to the lack of the Truck Antennas.

Table 5.4.: RSSI Attribute Values (SAT_MAIN_TRUCK Data Set)

Attribute	**Moved Tags**				**Static Tags**			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$RSSI_{Min,Main}$	-66.3	-50.6	-61.0	3.3	-66.7	-54.2	-61.8	2.7
$RSSI_{Max,Main}$	-54.5	-38.3	**-43.1**	3.8	-65.1	-44.3	**-58.1**	4.0
$RSSI_{Diff,Main}$	3.7	26.0	**17.9**	4.7	0.0	15.2	**3.7**	3.7
$RSSI_{Mean,Main}$	-58.8	-44.4	**-51.7**	3.2	-65.3	-50.7	**-59.9**	2.9
$RSSI_{StDev,Main}$	1.3	8.6	**5.3**	1.5	0.0	4.8	**1.3**	1.2
$RSSI_{CoV,Main}$	-0.168	-0.026	-0.104	0.030	-0.087	0.000	-0.023	0.021
$RSSI_{Min,Truck}$	-64.9	-53.6	-60.7	2.3	-65.4	-50.4	-61.1	2.9
$RSSI_{Max,Truck}$	-58.6	-38.4	-45.2	4.6	-62.9	-42.1	-53.4	4.7
$RSSI_{Diff,Truck}$	1.0	24.2	15.5	5.0	0.0	19.0	7.7	4.4
$RSSI_{Mean,Truck}$	-59.8	-44.9	-52.2	3.3	-63.2	-45.5	-56.9	3.7
$RSSI_{StDev,Truck}$	0.6	7.1	4.1	1.3	0.0	4.9	1.9	1.1
$RSSI_{CoV,Truck}$	-0.147	-0.010	-0.080	0.028	-0.095	0.000	-0.034	0.020

Table 5.5 shows the RSSI attribute values for moved and static tags monitored at the *Transition Portals*. Considering the attribute values that correspond to the DC Antennas it becomes clear that the above expectations hold true. Moved tags are read by these antennas with higher minimum, maximum and average signal strengths than static tags. The standard deviation and the $RSSI_{Diff}$ attribute prove that the latter also have a lower signal strength variation than moved tags. Considering the attribute values that correspond to the Truck Antennas, the same observations can be made. However, the differences between static and moved tags are somewhat less significant. In contrast to Standard- and Transition Portals it is notable that the minimum signal strength appears to be more meaningful while the maximum signal strength and the standard deviation become less expressive.

Table 5.5.: RSSI Attribute Values (TRA_BOTH Data Set)

Attribute	**Moved Tags**				**Static Tags**			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$RSSI_{Min,DC}$	-65.8	-48.8	-59.8	3.6	-66.4	-53.9	-62.0	2.6
$RSSI_{Max,DC}$	-58.1	-38.3	**-43.6**	4.7	-65.1	-39.3	**-56.4**	5.5
$RSSI_{Diff,DC}$	0.0	25.5	**16.1**	5.6	0.0	21.3	**5.6**	5.4
$RSSI_{Mean,DC}$	-60.4	-43.0	**-50.7**	3.7	-65.2	-46.3	**-59.1**	3.7
$RSSI_{StDev,DC}$	0.0	8.5	**4.7**	1.7	0.0	5.9	**1.7**	1.4
$RSSI_{CoV,DC}$	-0.172	0.000	-0.094	0.035	-0.115	0.000	-0.030	0.027
$RSSI_{Min,Truck}$	-66.0	-49.4	-59.9	3.8	-66.2	-50.4	-60.3	3.8
$RSSI_{Max,Truck}$	-56.6	-38.3	-43.4	4.3	-64.6	-40.3	-51.7	5.6
$RSSI_{Diff,Truck}$	1.1	25.8	16.5	5.7	0.0	21.9	8.6	5.6
$RSSI_{Mean,Truck}$	-59.0	-44.1	-51.5	3.3	-64.9	-45.9	-55.8	4.3
$RSSI_{StDev,Truck}$	0.6	8.0	4.6	1.6	0.0	6.0	2.3	1.4
$RSSI_{CoV,Truck}$	-0.159	-0.010	-0.090	0.031	-0.118	0.000	-0.042	0.027

5.2.1.2. SinceStart Attributes

On the basis of a tag-occurrence

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

encompassing n individual tag-events, calculation of the *SinceStart Attributes* is based on the unordered set of the corresponding SinceStart values:

$$\{SinceStart_1, \ldots, SinceStart_n\}$$

The main point behind the definition of these attributes is the insight that certain static pallets are detected only occasionally. For example, some false-positive reads are the result of unexpected reflections that occur only randomly. In contrast to moved pallets these are not expected to be read over the entire period of a gathering-cycle.

Read$_{\text{First}}$ The time since the beginning of the gathering-cycle that passed before the tag is *first* read. Moved pallets are typically read from the very beginning, while certain false-positives are detected only after a couple of seconds.

$$Read_{First} := \min\{SinceStart_1, \ldots, SinceStart_n\}$$

Read$_{\text{Last}}$ The time since the beginning of the gathering-cycle that passed before the tag is *last* read. At the end of a gathering-cycle basically any type of pallet may be read due to electromagnetic reflections. Consequently, the time stamps of the last tag answer are not expected to differ much. However, this attribute might be helpful in combination with another one.

$$Read_{Last} := \max\{SinceStart_1, \ldots, SinceStart_n\}$$

Read$_{\text{Diff}}$ The time that has passed between the first and the last detection of the tag. Because moved pallets are often read at the beginning as well as at the end of a gathering-cycle, this attribute is expected to take on higher values for moved than for static tags.

$$Read_{Diff} := Read_{Last} - ReadFirst$$

Table 5.6 shows the minimum, maximum, average and the standard deviation of the SinceStart attribute values for moved and static tags monitored at the *Standard Portals*. Investigation of the $Read_{First}$ attribute reveals that on average moved tags are read around 0.7 seconds sooner than static tags are. The maximum value of this attribute indicates that 99% of the moved tags were first detected within 2.43 seconds, while this percentage is reached only after 7.41 seconds for the static tags. As it was expected, the$Read_{Last}$ attribute does not appear to carry much information on its own. Looking at the $SinceStart_{Diff}$ attribute, it can be seen that on average moved tags are seen over a period of 3.32 seconds while static tags are seen for only 2.61 seconds, which is 0.71 seconds less. The minimum value of this attribute indicates that 99% of the moved tags are seen over a period of at least 0.29 seconds. The value of 0.00 for the static tags implies that these are often seen only during a very short time window or for a single instance, resulting in a $SinceStart_{Diff}$ value of 0 by definition. Table 5.7 shows the SinceStart attribute values for moved and static tags monitored at the *Satellite Portals* that were read only by the *Main Antennas*. As expected these values do not differ much from the values observed at the Standard Portals. However, on average the last tag detection of moved tags occurs after 3.82 seconds while static tags are on average read last 0.4 seconds

Table 5.6.: SinceStart Attribute Values (STD_COMPLETE Data Set)

Attribute	Moved Tags				Static Tags			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$Read_{First,Main}$	0.03	**2.43**	**0.38**	0.63	0.03	**7.41**	**1.10**	1.45
$Read_{Last,Main}$	1.06	9.98	3.70	1.90	0.22	9.99	3.71	2.59
$Read_{Diff,Main}$	**0.29**	9.93	**3.32**	2.02	**0.00**	9.94	**2.61**	2.77

sooner, namely after 3.43 seconds. This effect could not be observed at the Standard Portals at all. Furthermore, most of the differences between moved and static tags are considerably more significant. For example, in this data set moved tags are seen for 1.22 seconds longer than static tags on average (3.48s vs. 2.26s) which is an increase of almost 0.5 seconds. Table

Table 5.7.: SinceStart Attribute Values (SAT_MAIN_ONLY Data Set)

Attribute	Moved Tags				Static Tags			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$Read_{First,Main}$	0.03	2.32	0.33	0.56	0.03	7.43	1.17	1.49
$Read_{Last,Main}$	0.99	9.99	**3.82**	2.08	0.21	9.99	**3.43**	2.57
$Read_{Diff,Main}$	0.18	9.93	**3.48**	2.23	0.00	9.92	**2.26**	2.68

5.8 shows the SinceStart attribute values for moved and static tags monitored at the *Satellite Portals* that were read by both *Main-* and *Truck Antennas*. Looking at the Main Antennas, not much difference from the above two cases can be observed, except that the last detection of 99% of the moved tags occurs within the first 2.96 seconds of a gathering-cycle in this data set. Furthermore, moved tags are read over an average period of 2.61 seconds which is significantly less when compared to the observations made above. Looking at the Truck Antennas it can be seen that moved tags are seen for the first time after 1.48 seconds on average compared to static tags that are seen for the first time after 0.72 seconds. Furthermore, on average static tags are read over a longer period of time (4.65 seconds) compared to moved tags (3.37 seconds). It can be assumed that these static tags are mainly tags that were loaded inside the container already. Table 5.9 shows the SinceStart attribute values for moved and static tags monitored at

Table 5.8.: SinceStart Attribute Values (SAT_MAIN_TRUCK Data Set)

Attribute	Moved Tags				Static Tags			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$Read_{First,Main}$	0.03	2.95	0.35	0.65	0.03	8.42	1.37	1.73
$Read_{Last,Main}$	0.78	9.81	**2.96**	1.74	0.10	9.99	3.48	2.71
$Read_{Diff,Main}$	0.33	9.71	**2.61**	1.79	0.00	9.87	2.11	2.64
$Read_{First,Truck}$	0.03	5.04	**1.48**	1.07	0.02	5.77	**0.72**	1.34
$Read_{Last,Truck}$	1.52	10.00	4.85	2.01	0.79	10.00	5.37	2.76
$Read_{Diff,Truck}$	0.07	9.80	**3.37**	2.11	0.00	9.97	**4.65**	3.07

the *Transition Portals* that have been read by both *DC-* and *Truck Antennas*. Looking at the DC Antennas it can be seen that some attribute values differ significantly for moved and static tags. For example, the maximum of the $Read_{First,DC}$ attribute reveals that 99% of the static tags are seen for the first time within 8.01 seconds after the start of the gathering-cycle but 99% of the moved tags are seen first within 3.34 seconds. Furthermore, they are last read after an average 4.18 seconds, which is 1.01 seconds after the moved tags (3.17 seconds). Looking at the Truck Antennas it is notable that on average the static tags can still be detected after moved tags (5.41s vs. 4.98s) and that they are read over a longer period of time (4.50s vs. 3.99s).

Table 5.9.: SinceStart Attribute Values (TRA_BOTH Data Set)

Attribute	Moved Tags				Static Tags			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$Read_{First,DC}$	0.03	**3.34**	0.19	0.59	0.03	**8.01**	1.10	1.55
$Read_{Last,DC}$	0.48	9.99	**3.17**	2.18	0.11	10.00	**4.18**	2.95
$Read_{Diff,DC}$	0.00	9.93	2.99	2.19	0.00	9.94	3.09	3.11
$Read_{First,Truck}$	0.03	4.80	0.99	0.94	0.03	6.72	0.91	1.49
$Read_{Last,Truck}$	1.61	9.99	**4.98**	1.99	0.38	10.00	**5.41**	2.69
$Read_{Diff,Truck}$	0.15	9.81	**3.99**	2.10	0.00	9.94	**4.50**	3.14

5.2.1.3. Antenna Attributes

On the basis of a tag-occurrence

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

encompassing n individual tag-events, calculation of the *Antenna Attributes* is based on the unordered set of the corresponding Antenna values:

$$\{Antenna_1, \ldots, Antenna_n\}$$

The main point behind the definition of these attributes is the same insight that lead to the definition of the SinceStart Attributes, namely that certain static pallets are detected only occasionally. Thus, it is expected in general that a moved tag is detected more often and by more antennas than a static tag.

Count$_x$ The number of tag-events that were recorded by each of the antennas, where x is the identifier of the corresponding antenna. Because many static tags are close to specific antennas it is expected that certain antennas will detect these tags more often than they detect the moved tags.

$$Count_x := |\{Antenna_1, \ldots, Antenna_n | Antenna_i = x \wedge 1 \leq i \leq n\}|$$

AntCount The number of antennas that were able to detect the tag. Because many static tags are close to specific antennas it is expected that moved pallets are read by more antennas than static tags are.

$$AntCount := |\{Count_x | Count_x > 0\}|$$

Count The total number of answers the tag gave to all of the antennas together. Because moved tags pass the portal and are thus very close to the antennas it is expected that they are read more often in total than static tags are.

$$Count := \sum_{i=1}^{x} Count_x$$

Table 5.10 shows the Antenna attribute values for moved and static tags monitored at the *Standard Portals*. It is notable that the number of tag detections significantly differs with respect to the specific antenna and furthermore with respect to moved and static tags. For

example an average moved tag is detected 24.3 times by antenna 1 but only 5 times by antenna 4. Furthermore, moved tags are detected by antennas 1 and 2 significantly more often than static tags (24.3 vs. 8.4 and 10.5 vs. 4.2).

Table 5.10.: Antenna Attribute Values (`STD_COMPLETE` Data Set)

Attribute	**Moved Tags**				**Static Tags**			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$Count_{1,Main}$	0.0	117.0	**24.3**	21.7	0.0	118.0	**8.4**	20.3
$Count_{2,Main}$	0.0	59.0	**10.5**	13.3	0.0	74.0	**4.2**	14.1
$Count_{3,Main}$	0.0	56.0	7.0	11.8	0.0	65.0	4.1	13.4
$Count_{4,Main}$	0.0	56.0	**5.0**	11.4	0.0	76.0	4.4	14.8
$AntCount_{Main}$	1.0	4.0	**3.2**	0.9	1.0	4.0	**1.8**	1.0
$Count_{Main}$	4.0	195.0	**46.8**	39.9	1.0	161.0	**21.0**	33.9

The reason for these heterogeneous attribute values can be found in the antenna configuration. As stated in Section 4.2.1 antennas 1 and 2 are located at the bottom of the portal, while antennas 3 and 4 are installed on at the top. When a pallet moves through the portal it covers the lower antennas, while the top antennas still have a free line-of-sight to register arbitrary static pallets. This distinction does hold true, because in the table it can be seen that the top antennas detect static tags more often than they detect moved tags. It is notable furthermore, that the maximum number of tag reads per antenna is higher for static than for moved tags, but the average number of tag reads at the specific antennas is higher for the latter. The reason for this is that often a pallet is placed directly next to an antenna and is consequently read by it almost right through the gathering-cycle. However, many other static pallets are out of read range of that specific antenna and consequently are never read by it. Thus, the maximum number of reads can be much higher for static tags but this effect is cancelled out when averaging over all static tags including the ones that never read by the antenna. Another very important piece of information in this table is the *Count* attribute which tells that moved tags are detected an average 46.8 times which is significantly more often compared to static tags that are detected only 21 times on average. Furthermore, moved tags are detected on average by 3.2 antennas compared to static tags that are detected on average by only 1.8 antennas.

Table 5.11 shows the Antenna attribute values for moved and static tags monitored at the *Satellite Portals* that were detected only by the *Main Antennas*. Compared to the values of the Standard Portal attributes some differences can be observed. On average moved tags are detected by antenna 1 more often, but by antennas 2, 3 and 4 less often. Static tags are detected by all antennas less often on average and the total number of average detections drops from 21.0 to 15.5 while this attribute value remains almost constant for the moved tags (46.8 vs. 45.4 detections).

Table 5.11.: Antenna Attribute Values (`SAT_MAIN_ONLY` Data Set)

Attribute	**Moved Tags**				**Static Tags**			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$Count_{1,Main}$	0.0	92.4	**27.2**	21.2	0.0	83.3	**6.3**	15.5
$Count_{2,Main}$	0.0	40.5	**9.0**	10.2	0.0	28.3	**1.9**	8.1
$Count_{3,Main}$	0.0	32.0	**6.3**	7.6	0.0	44.3	**2.8**	9.8
$Count_{4,Main}$	0.0	19.8	**2.8**	8.0	0.0	84.3	**4.5**	15.8
$AntCount_{Main}$	1.0	4.0	3.2	1.0	1.0	4.0	1.6	0.9
$Count_{Main}$	4.0	140.1	**45.4**	27.8	1.0	137.0	**15.5**	25.3

Table 5.12 shows the Antenna attribute values for moved and static tags monitored at the *Satellite Portals* that were detected by both *Main-* and *Truck Antennas*. Looking at the Main Antennas it can be observed that both moved and static tags are read less often on average at either antenna and also in total ($Count_{Main}$ attribute). Looking at the Truck Antenna attributes it can seen that these read both types of tags significantly more often. However, Truck Antenna 6 in particular reads static pallets even more often than it does moved pallets. Furthermore it is notable that moved tags are read more often than static pallets by the Main Antennas ($Count_{Main}$), but less often than static tags by the Truck Antennas ($Count_{Truck}$). In total though, moved tags are still detected more often than static tags ($Count_{Total}$).

Table 5.12.: Antenna Attribute Value Distribution (SAT_MAIN_TRUCK Data Set)

Attribute	**Moved Tags**				**Static Tags**			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$Count_{1,Main}$	0.0	61.0	**16.0**	12.1	0.0	51.0	**3.8**	9.4
$Count_{2,Main}$	0.0	28.0	**6.0**	7.3	0.0	24.0	**1.7**	5.5
$Count_{3,Main}$	0.0	18.0	**3.5**	5.0	0.0	26.0	**1.7**	5.2
$Count_{4,Main}$	0.0	14.0	**2.6**	3.2	0.0	30.0	**1.8**	5.5
$Count_{5,Truck}$	0.0	154.0	34.8	31.4	0.0	200.0	29.1	43.5
$Count_{6,Truck}$	0.0	82.1	14.7	18.8	0.0	227.0	31.5	46.9
$AntCount_{Main}$	1.0	4.0	3.2	0.9	1.0	4.0	1.6	0.9
$AntCount_{Truck}$	1.0	2.0	1.7	0.5	1.0	2.0	1.5	0.5
$AntCount_{Total}$	2.0	6.0	4.9	1.2	2.0	6.0	3.2	1.0
$Count_{Main}$	4.0	87.2	**28.0**	15.5	1.0	75.0	**8.9**	14.7
$Count_{Truck}$	2.0	198.1	**49.5**	36.1	1.0	251.0	**60.6**	56.8
$Count_{Total}$	12.0	240.0	**77.5**	41.0	3.0	262.0	**69.5**	58.7

Table 5.13 shows the attribute values for moved and static tags monitored at the *Transition Portals* that have been read by both *DC*- and *Truck Antennas*. Looking at the DC Antennas it can be seen that only antennas 1 and 2 show significantly different values for moved and static tags. However, on average moved tags are still seen by more antennas and more often in total. Looking at the maximum values of the Truck Antenna attributes it becomes obvious that these four antennas read static tags more often than they do moved tags.

5.2.2. Artificial Attributes

As has been shown in the previous section, some domain attributes are likely suitable candidates for classifying moved and static pallets. However, in many cases Domain Attributes alone do not lead to sufficiently acceptable classification rates. A common approach to overcome this problem is to construct additional *Artificial Attributes* derived from the available Domain Attributes [Kam09, SP06] using unary or binary mathematical operations. The Artificial Attribute generation procedure comprises two distinct phases denoted *Attribute Generation* and *Attribute Evaluation*.

Table 5.13.: Antenna Attribute Value Distribution (TRA_BOTH Data Set)

Attribute	**Moved Tags**				**Static Tags**			
	Min	**Max**	**Avg**	**SD**	**Min**	**Max**	**Avg**	**SD**
$Count_{1,DC}$	0.0	100.2	**12.0**	16.9	0.0	94.2	**6.9**	16.9
$Count_{2,DC}$	0.0	59.1	**7.1**	12.1	0.0	88.1	**5.0**	14.6
$Count_{3,DC}$	0.0	25.0	4.1	7.1	0.0	51.1	3.6	10.0
$Count_{4,DC}$	0.0	21.0	3.4	4.5	0.0	38.0	3.0	8.2
$Count_{5,Truck}$	0.0	66.0	15.8	15.2	0.0	99.2	13.5	21.2
$Count_{6,Truck}$	0.0	40.0	6.9	8.9	0.0	71.0	6.3	14.3
$Count_{7,Truck}$	0.0	37.0	7.8	9.5	0.0	85.1	7.8	15.9
$Count_{8,Truck}$	0.0	26.0	5.5	6.1	0.0	60.2	5.6	12.1
$AntCount_{DC}$	1.0	4.0	**2.9**	1.0	1.0	4.0	**1.8**	1.0
$AntCount_{Truck}$	1.0	4.0	**3.1**	1.0	1.0	4.0	**2.3**	1.1
$AntCount_{Total}$	2.0	8.0	**6.0**	1.7	2.0	8.0	**4.1**	1.5
$Count_{DC}$	1.0	146.1	**26.6**	22.7	1.0	140.0	**18.4**	29.2
$Count_{Truck}$	2.0	108.0	**36.0**	21.5	1.0	137.1	**33.1**	31.8
$Count_{Total}$	9.0	175.0	**62.5**	31.8	2.0	173.2	**51.6**	39.0

5.2.2.1. Attribute Generation

In the first step, a sequence of mathematical operators is repeatedly applied to the Domain Attributes to create new combinations. The operators can be separated into two different groups, namely *Binary Operators* (Table 5.14) and *Unary Operators* (Table 5.15). The first group takes two different Domain Attributes, A_1 and A_2, as input and generates a new attribute by applying the corresponding operator. The attribute generation algorithm would, for example, at some point combine the *Count* attribute with the $Read_{Diff}$ attribute using the division operator:

$$Attribute_1 = \frac{Read_{Diff}}{Count}$$

This attribute has not been thought of before when identifying the Domain Attributes. The meaning of this new, artificial attribute can be interpreted as "*the average time between two different tag reads*". Although this attribute appears to be quite reasonable at first sight, the semantic of an artificial attribute can be determined only in very few cases. Like in the

Table 5.14.: Binary Attribute Generation Operators

Operator	**Attribute Construction**
Addition	$Attribute_{New} = A_1 + A_2$
Subtraction	$Attribute_{New} = A_1 - A_2$
Multiplication	$Attribute_{New} = A_1 \cdot A_2$
Division	$Attribute_{New} = \frac{A_1}{A_2}$
Hypothenuse	$Attribute_{New} = \sqrt{(A_1)^2 + (A_2)^2}$

motivating example above, where two different attributes were combined, it is also possible to influence the suitability of a single Domain Attribute, A, to classify moved and static tags by altering its attribute value space. This is done by the use of one of the unary operators shown in Table 5.15.

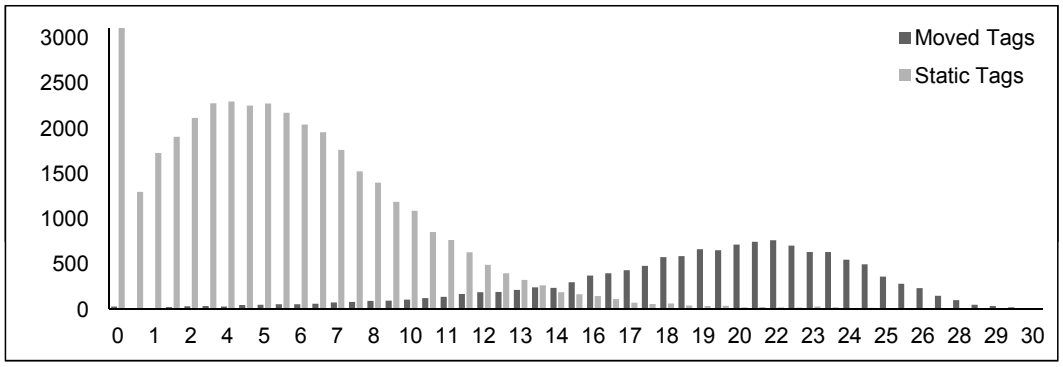
Table 5.15.: Unary Attribute Generation Operators

Operator	**Attribute Construction**
Sine	$Attribute_{New} = \sin A$
Cosine	$Attribute_{New} = \cos A$
Tangent	$Attribute_{New} = \tan A$
Square Root	$Attribute_{New} = \sqrt{A}$
Weighting	$Attribute_{New} = A \cdot n, n \in \mathbb{N}$
Reciprocal	$Attribute_{New} = \frac{1}{A}$
Logarithm	$Attribute_{New} = \log A$
Exponential Function	$Attribute_{New} = e^A$
Power Function	$Attribute_{New} = A^x, x \in \mathbb{N}$

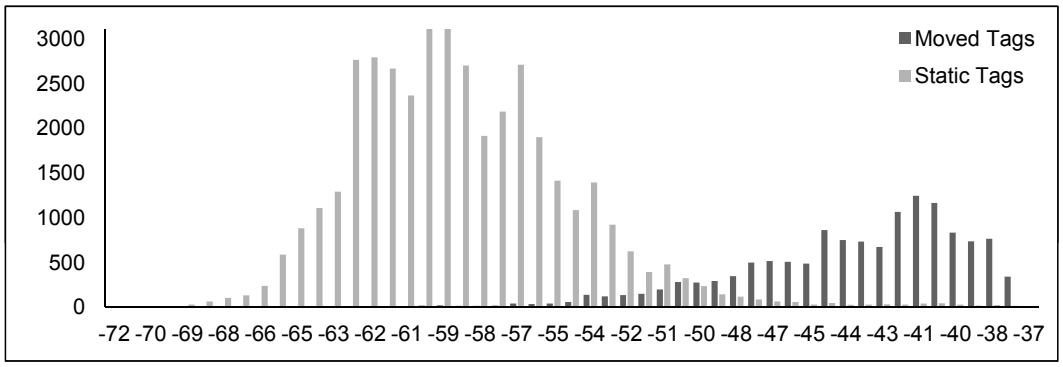
If unary and binary operators are combined, then a number of new attributes are created similar to the following:

$$Attribute_2 = \sin \sqrt{(RSSI_{Diff}) + (RSSI_{Max})}$$

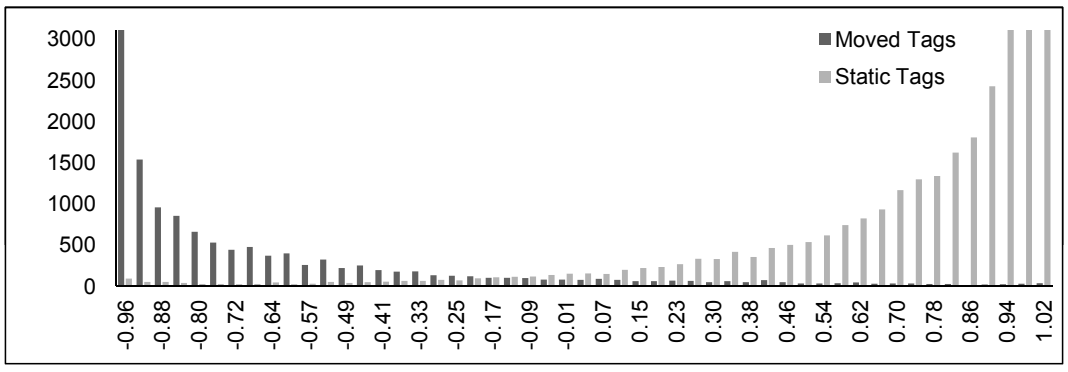
These have been created by a sequence of unary operators (sin-function and taking the square root) as well as one binary operator (addition). The effect of these operations is shown in Figure 5.4. Figures 5.4(a) and 5.4(b) depict the attribute value distributions of the two underlying



(a) Distribution of $RSSI_{Diff}$ Attribute Values (`STD_COMPLETE` Data Set)



(b) Distribution of $RSSI_{Max}$ Attribute Values (`STD_COMPLETE` Data Set)



(c) Distribution of Artificial $Attribute_2$ Values (`STD_COMPLETE` Data Set)

Figure 5.4.: Value Distribution of an Artificial Attribute in contrast to the two originating Domain Attributes

Domain Attributes that were used to create $Attribute_2$ i.e., $RSSI_{Diff}$ and $RSSI_{Max}$. The distribution of the $RSSI_{Diff}$ attribute values of the individual distributions of moved and static tags resemble two gaussian distributions. Another important piece of information in this histogram is the remarkable number of static tags that have a $RSSI_{Diff}$ attribute value of 0 (other than depicted this corresponds to more than 9000 static tags and not only 3000, this was done for presentation purposes only). This is because it turns out that a large number of static tags responded only a single time to the reader thus having a RSSI difference of 0 by definition. The distribution of the maximum RSSI values shows a similar behavior, although due to quantization effects the two gaussian distributions don't appear to be as smooth.

Figure 5.4(c) shows the distribution of attribute $Attribute_2$ which has been created using these two Domain Attributes (similar to Figure 5.4(a) the maxima have been cut for presentation purposes). The important information in this figure is that the distribution of moved and static tags does not resemble a Gaussian distribution anymore and the set of tags having an RSSI standard deviation of 0 has been smoothed. Furthermore, while the maxima of the two distributions previously lay pretty close to each other with respect to the Domain Attributes, this effect has been eliminated as moved tags now cluster at the minimum value of $Attribute_2$ and static tags cluster at the maximum value of $Attribute_2$. It is notable that a distribution similar to the sample artificial attribute cannot be observed for any of the Domain Attributes already introduced.

5.2.2.2. Attribute Evaluation

The attribute generation procedure provides us with a large number of attributes, of which only a small fraction are actually helpful for distinguishing moved and static tags. For example, if only 4 unary and 4 binary operators are used to combine 5 different Domain Attributes, this results in

$$(5 \cdot 5)^2 \cdot 4 = 2,500$$

new attributes. One additional application of the operators to these attributes gives us an additional

$$(((5 \cdot 5)^2 \cdot 4)^2) \cdot 4 = 25,000,000$$

new attributes. In order to reduce the complexity for building the classification model it becomes necessary to reduce the number of attributes to a manageable quantity [Bel61, CFZ09]. It is understood that one tries to keep the best attributes and discard the useless ones. At

this point, a quality measure definition is needed to evaluate how good an attribute is and when to favor an attribute over another. In Section 5.1.3 *Entropy* and *Information Gain* were introduced as the C4.5's method to evaluate the quality of an attribute. Because this is well known and proven to be useful [GH07], it is also the method of choice for the artificial attribute evaluation. The Information Gain is calculated for every artificial attribute with only the 500 with the highest gain being kept; the rest are discarded.

5.2.3. The Tag-Occurrence Count

Observations at the METRO distribution center in Unna have shown that pallets which have already been seen in a preceding gathering-cycle (at the same portal) were static most of the time. Consequently, an attribute that counts the number of previous occurrences of the pallets, called TO_{Count}, is introduced which corresponds to the number of gathering-cycles in which a tag has been seen before, including the current one. If a tag is seen for the first time, then it is called a *first-sight* (FS), otherwise it is called a *non first-sight* (NFS). Table 5.16 shows the distribution of moved and static tags in the relevant sample data sets based on the TO_{Count} values. It can be seen that the initial assumption holds true. At either portal, at least 90% of

Table 5.16.: Tag Distribution based on Tag-Occurrence Count

Data Set	TO_{Count}	Moved Tags	Static Tags	Precision
STD_COMPLETE	First-Sights	11,835	3,065	**79.4%**
	Non First-Sights	1,410	37,678	**96.4%**
	Recall	**89.4%**	**92.5%**	
SAT_MAIN_ONLY	First-Sights	567	89	**86.4%**
	Non First-Sights	268	2,303	**89.9%**
	Recall	**67.9%**	**96.3%**	
SAT_MAIN_TRUCK	First-Sights	1,081	201	**84.3%**
	Non First-Sights	67	1,832	**96.5%**
	Recall	**94.2%**	**90.1%**	
TRA_BOTH	First-Sights	1,172	127	**90.2%**
	Non First-Sights	181	2,211	**92.4%**
	Recall	**86.6%**	**94.6%**	

all static tags were seen in a previous gathering-cycle. In turn, at around 90% of all moved tags had not been seen before (except for the tags monitored at the Satellite Portals). With respect to the question of how confident a classification using this attribute is, it is notable that only the Transition Portals reach an accuracy of at least 90% for both values FS and NFS.

5.2.4. Logical Attributes

5.2.4.1. Satellite Portal Logic

Satellite Portals consist of Main- as well as Truck- and DC Antennas as described in Section 4.2.2. Because moved tags pass the portal it is expected that they are read first by the Main- and subsequently by the Truck Antennas (remember that if they are read by the DC Antennas after the loading then they are classified as static by definition). Accordingly, besides the definition of an attribute to determine which antennas a tag was read by, a number of additional characteristics are introduced to map the order at which it was read against meaningful attributes. From all the possible cases, only the tags read by both Main- and Truck Antennas are of interest at this point because for tags read only by the Main Antennas no order can be defined and all other cases are automatically classified as static by definition.

WhereRead This attribute corresponds to the cases defined in Section 4.3.2 and has already been investigated there.

$$WhereRead := \begin{cases} 1, & \text{if read only by DC Antennas} \\ 2, & \text{if read only by Main Antennas} \\ 3, & \text{if read only by Truck Antennas} \\ 4, & \text{if read by Main-, then by DC Antennas} \\ 5, & \text{if read by Truck-, then by DC Antennas} \\ 6, & \text{if read only by Main- and Truck Antennas} \\ 7, & \text{if read by Main- and Truck-, then by DC Antennas} \end{cases}$$

$\mathbf{Seen_{First}}$ This attribute examines the $Read_{First}$ attribute values to determine whether a tag was seen first by the Main- or by the Truck Antennas. Because in the beginning of a loading moved tags are not inside the container, then by definition it most likely they are seen first by the Main Antennas.

$$Seen_{First} := \begin{cases} Main, & \text{if } Read_{First,Main} \leq Read_{First,Truck} \\ Truck, & \text{otherwise} \end{cases} \tag{5.1}$$

$\mathbf{Seen_{Last}}$ This attribute examines the $Read_{Last}$ attribute values to determine whether a tag was last seen by the Main or the Truck Antennas. Because at the end of a loading the moved as well as some static tags are present inside the container there is no clear expectation of what is typical for either tag type. Still it is reasonable to consider this attribute as it might be helpful in combination with another one.

$$Seen_{Last} := \begin{cases} Main, & \text{if } Read_{Last,Main} \geq Read_{Last,Truck} \\ Truck, & \text{otherwise} \end{cases} \tag{5.2}$$

$\mathbf{Seen_{Longer}}$ This attribute examines the $Read_{Diff}$ attribute values to determine which reader the tag has been read by over the longer period of time. Both moved and static tags are usually read for a longer period of time by the Truck Antennas but there is no clear expectation of what is typical for moved and static tags. Still it is reasonable to consider this attribute as it might helpful in combination with another one.

$$Seen_{Longer} := \begin{cases} Main, & \text{if } Read_{Diff,Main} \geq Read_{Diff,Truck} \\ Truck, & \text{otherwise} \end{cases} \tag{5.3}$$

$\mathbf{First_{Main}Last_{Truck}}$ This attribute determines whether a tag has first been read by the Main Antennas and last by the Truck Antennas. It is likely that such a tag is one that moved through the portal.

$$First_{Main}Last_{Truck} := \begin{cases} Yes, & \text{if } Seen_{First} = Main \text{ and } Seen_{Last} = Truck \\ No, & \text{otherwise} \end{cases} \tag{5.4}$$

$\mathbf{First_{Truck}Last_{Main}}$ This attribute determines whether a tag has first been read by the Truck Antennas and last by the Main Antennas. It is rather unlikely that this happens, but if it does, then such a tag is likely static.

$$First_{Truck}Last_{Main} := \begin{cases} Yes, & \text{if } Seen_{First} = Truck \text{ and } Seen_{Last} = Main \\ No, & \text{otherwise} \end{cases} \tag{5.5}$$

$\mathbf{First_{Main}Last_{Main}}$ This attribute determines whether the first and last detection of a tag occurred at the Main Antennas. It is rather unlikely that this happens, but if it does, then such a tag is likely a static tag located somewhere in the distribution center (i.e., outside of the container).

$$First_{Main}Last_{Main} := \begin{cases} Yes, & \text{if } Seen_{First} = Truck \text{ and } Seen_{Last} = Main \\ No, & \text{otherwise} \end{cases} \tag{5.6}$$

$\mathbf{First_{Truck}Last_{Truck}}$ This attribute determines whether the first and last detection of a tag occurred at the Truck Antennas. If this is the case then it is likely that the tag was already inside the container during the entire gathering-cycle and is thus static.

$$First_{Truck}Last_{Truck} := \begin{cases} Yes, & \text{if } Seen_{First} = Truck \text{ and } Seen_{Last} = Main \\ No, & \text{otherwise} \end{cases} \tag{5.7}$$

$\mathbf{Disjoint_{Main,Truck}}$ This attribute determines whether a tag has been read only by the Main Antennas in the beginning and after that only by the Truck Antennas. Because this could 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.

$$Disjoint_{Main,Truck} := \begin{cases} Yes, & \text{if } Read_{Last,Main} \leq Read_{First,Truck} \\ No, & \text{otherwise} \end{cases} \tag{5.8}$$

$\mathbf{Disjoint_{Truck,Main}}$ This attribute determines whether a tag has been read only by the Truck antennas in the beginning and after that only by the Main Antennas. It is rather unlikely that this happens, but if it does, then such a tag is likely static.

$$Disjoint_{Truck,Main} := \begin{cases} Yes, & \text{if } Read_{Last,Truck} \leq Read_{First,Main} \\ No, & \text{otherwise} \end{cases} \tag{5.9}$$

Tables 5.17 and 5.18 show the investigation of the *Logical Satellite Attributes*. For presentation purposes they have been split into two parts. In Table 5.17 it can be seen that more than 93% of all moved tags are first seen by the Main- and more than 95% are last seen by the Truck Antennas - as expected. Because static tags can be located inside the container or inside the distribution center no clear distinction is possible for these. Both moved and static tags are seen longer by the Truck Antennas so the $Seen_{Longer}$ attribute appears to be not so useful. Note that the attribute characteristics

- $Seen_{First} = Main$,
- $First_{Main}Last_{Truck} = Yes$ and
- $Disjoint_{Main,Truck} = Yes$

constitute an order in the way that the succeeding attribute further reduces the set of tags covered by the previous. The same holds true for the attribute characteristics

- $Seen_{First} = Truck$,
- $First_{Truck}Last_{Main} = Yes$ and
- $Disjoint_{Truck,Main} = Yes$.

Table 5.17.: Logical Satellite Attribute Value Distribution Part 1

Attribute	**Value**	**Moved Tags**	**Static Tags**	**Precision**
$Seen_{First}$	Main	1,195	629	**65.5%**
	Truck	87	1,270	**93.6%**
	Recall	**93.2%**	**66.9%**	
$Seen_{Last}$	Main	53	310	**85.4%**
	Truck	1,229	1,589	**56.4%**
	Recall	**95.9%**	**83.7%**	
$Seen_{Longer}$	Main	420	328	**56.1%**
	Truck	862	1,571	**64.6%**
	Recall	**67.2%**	**82.7%**	

Table 5.18.: Logical Satellite Attribute Value Distribution Part 2

Attribute	Value	Moved Tags	Static Tags	Precision
$First_{Main}Last_{Truck}$	Yes	1,153	499	**69.8%**
	No	129	1,400	**91.6%**
	Recall	**89.9%**	**73.7%**	
$First_{Truck}Last_{Main}$	Yes	11	180	**94.2%**
	No	1,271	1,719	**57.5%**
	Recall	**99.1%**	**90.5%**	
$First_{Main}Last_{Main}$	Yes	42	130	**75.6%**
	No	1,240	1,769	**58.8%**
	Recall	**96.7%**	**93.2%**	
$First_{Truck}Last_{Truck}$	Yes	76	1,090	**93.5%**
	No	1,206	809	**59.9%**
	Recall	**94.1%**	**57.4%**	
$Disjoint_{Main,Truck}$	Yes	246	256	**51.0%**
	No	1,036	1,643	**61.3%**
	Recall	**80.8%**	**86.5%**	
$Disjoint_{Truck,Main}$	Yes	0	70	**100.0%**
	No	1,282	1,829	**58.8%**
	Recall	**100.0%**	**96.3%**	

5.2.4.2. Transition Portal Logic

Transition Portals consist of DC- and Truck Antennas. Because moved tags pass the portal it is expected that they are read first by the DC- and subsequently by the Truck antennas (if they are read by only one of them they are classified as static by definition). Consequently, besides the definition of an attribute to determine at which antennas a tag was read, a number of additional attributes are introduced to map the order at which it was read against meaningful attributes.

WhereRead This attribute corresponds to the cases defined in Section 4.2.3 and was already investigated there.

$$WhereRead := \begin{cases} 1, & \text{if read only by DC Antennas} \\ 2, & \text{if read only by Truck Antennas} \\ 3, & \text{if read by both DC- and Truck Antennas} \end{cases}$$

Seen$_{\text{First}}$ This attribute examines the $Read_{First}$ attribute values to determine whether a tag has been seen *first* in the distribution center or in the truck. Moved tags are usually read first by the DC antennas and it is expected that tags read first by the Truck Antennas are static. Calculation is done analogous to Equation 5.1.

Seen$_{\text{Last}}$ This attribute examines the $Read_{Last}$ attribute values to determine whether a tag has been seen *last* in the distribution center or in the truck. Moved tags are usually read last by the Truck Antennas and it is expected that tags read last by the DC Antennas are static. Calculation is done analogous to Equation 5.2.

Seen$_{\text{Longer}}$ This attribute examines the $Read_{Diff}$ attribute values to determine at which antennas the tag has been read over the longer period of time. Moved and static tags are usually read for longer by the Truck Antennas but there is no clear expectation of what is typical for moved and static tags. Still it is reasonable to consider this attribute as it might helpful in combination with another one. Calculation is done analogous to Equation 5.3.

Disjoint$_{\text{DC,Truck}}$ This attribute determines whether a tag has been read only by the DC Antennas in the beginning and after that only by the Truck Antennas. It is expected that such a tag is one that moved through the portal. Calculation is done analogous to Equation 5.8.

Disjoint$_{\text{Truck,DC}}$ This attribute determines whether a tag has been read only by the Truck Antennas in the beginning and after that only by the DC Antennas. It is rather unlikely that this happens, but if it does, then such a tag is expected to be static. Calculation is done analogous to Equation 5.9.

First$_{\text{DC}}$Last$_{\text{Truck}}$ This attribute determines whether a tag has first been read by DC Antennas and last by the Truck Antennas. It is expected that such a tag is one that moved through the portal. Calculation is done analogous to Equation 5.4.

$\mathbf{First_{Truck}Last_{DC}}$ This attribute determines whether a tag has first been read by the Truck Antennas and last by the DC Antennas. It is likely that such a tag is static. The calculation is done analogous to Equation 5.5.

$\mathbf{First_{DC}Last_{DC}}$ This attribute determines whether the first and last detection of a tag occurred at the DC Antennas. Such a tag is likely a static tag located somewhere in the distribution center (i.e., outside of the container).

$\mathbf{First_{Truck}Last_{Truck}}$ This attribute determines whether the first and last detection of a tag occurred at the Truck Antennas. If this is the case then it is likely that the tag was already inside the container during the entire gathering-cycle and is thus static.

Tables 5.19 and 5.20 show the investigation of the *Logical Transition Attributes*. For presentation purposes they have been split into two parts. It can be seen in Table 5.19 that almost 92% of all moved tags are first seen by the DC- and more than 90% are seen last by the Truck Antennas as it was expected. Because static tags can be located inside the container or inside the distribution center no clear distinction is possible for these. Both moved and static tags are seen longer by the Truck Antennas so the $Seen_{Longer}$ attribute appears to be not so useful. Note that the attribute characteristics $Seen_{First} = DC$, $First_{DC}Last_{Truck} = Yes$ and $Disjoint_{DC,Truck} = Yes$ constitute an order in the way that the succeeding attribute further reduces the set of tags covered by the previous. The same holds true for the attribute characteristics $Seen_{First} = Truck$, $First_{Truck}Last_{DC} = Yes$ and $Disjoint_{Truck,DC} = Yes$.

Table 5.19.: Logical Transition Attribute Value Distribution Part 1

Attribute	Value	Moved Tags	Static Tags	Precision
$Seen_{First}$	DC	1,191	1,022	**53.8%**
	Truck	108	1,370	**92.7%**
	Recall	**91.7%**	**57.3%**	
$Seen_{Last}$	DC	127	732	**85.2%**
	Truck	1,172	1,660	**58.6%**
	Recall	**90.2%**	**69.4%**	
$Seen_{Longer}$	DC	367	707	**65.8%**
	Truck	932	1,685	**64.4%**
	Recall	**71.7%**	**70.4%**	

Table 5.20.: Logical Transition Attribute Value Distribution Part 2

Attribute	Value	Moved Tags	Static Tags	Precision
$First_{DC}Last_{Truck}$	Yes	1,087	581	**65.2%**
	No	212	1,811	**89.5%**
	Recall	**83.7%**	**75.7%**	
$First_{Truck}Last_{DC}$	Yes	23	291	**92.7%**
	No	1,276	2,101	**62.2%**
	Recall	**98.2%**	**87.8%**	
$First_{DC}Last_{DC}$	Yes	104	441	**80.9%**
	No	1,195	1,951	**62.0%**
	Recall	**92.0%**	**81.6%**	
$First_{Truck}Last_{Truck}$	Yes	85	1079	**92.7%**
	No	1,214	1,313	**52.0%**
	Recall	**93.5%**	**54.9%**	
$Disjoint_{DC,Truck}$	Yes	123	245	**66.6%**
	No	1,176	2,147	**64.6%**
	Recall	**90.5%**	**89.8%**	
$Disjoint_{Truck,DC}$	Yes	7	96	**93.2%**
	No	1,292	2,296	**64.0%**
	Recall	**99.5%**	**96.0%**	

5.3. Tag-Event Level Classification

The decision tree approach used a rule based system to identify moved and static pallets on the tag-occurrence level. If the low-level reader data collected for an RFID tagged pallet exhibits specific characteristics such as a predetermined maximum RSSI value the decision whether a pallet has been moved or not can easily be derived. In the end, the decision finding process is a simple sequence of Yes/No questions that have to be answered. In contrast to this, building a classification model on the basis of the tag-event level is more complex. The example in Figures 3.6 and 3.7 show how scattered the distribution of tag-events during a gathering-cycle really is. Nevertheless, in many cases it is possible for an experienced human observer to distinguish between moved and static tags optically, just by looking at the tag-event development over

time. Therefore, the second approach presented in this thesis aims at reproducing this human capability by identifying and classifying the typical behavior of moved and static tags over the time of a gathering-cycle.

The tag-event sequence of a specific RFID tag during a gathering-cycle is temporally ordered and can thus be considered as a so called *time-series*. Generally, a time-series TS consists of an ordered sequence of $n > 0$ data points d_i, which are usually real or integer values:

$$TS = (d_1, \ldots, d_n)$$

Similar to the Domain- and Artificial Attributes the basis of the Time-Series Attributes is a Tag-Occurrence

$$T = \{(RSSI_1, SinceStart_1), \ldots, (RSSI_N, SinceStart_n)\}$$

encompassing n individual tag-events. Note that in contrast to the tag-occurrence level approach the antenna data is omitted. The basic idea of the *tag-event level classification* approach presented here is to analyze these time-series and then decide whether they are more similar to a typical moved time-series $M = (m_1, \ldots, m_o)$ or to a typical static time-series $S = (s_1, \ldots, s_p)$. However, the prerequisite for such a decision is that there is a formal understanding of the terms *similarity* and *similarity of time-series* in particular.

5.3.1. About the Similarity between Time-Series

5.3.1.1. Distance Functions

In contrast to the decision tree classification approach using time-series analysis there is no clear class determination in the form of a rule and a leaf (at least in the beginning). Rather, there is a decision to be made as to whether a series is more similar to one than to another. Generally, two objects are said to be *similar* to each other if they have a small *distance*. The distance between two objects of class O is determined by evaluating a so called distance function

$$d : O \times O \to \mathbb{R}.$$

If o, p and q are objects of class O, then a distance function has to satisfy the following conditions:

Non-Negativity $d(o, p) \geq 0$. The distance between two objects cannot be negative.

Identity of Indiscernibles $d(o,p) = 0 \Leftrightarrow 0 = p$. The two objects have a distance of 0 - if, and only if, they are identical.

Symmetry $d(o,p) = d(p,o)$ The distance from o to p is always the same as the distance from p to o.

Triangle Inequality $d(o,q) \leq d(o,p) + d(p,q)$. The distance between o and p is always determined by the shortest connection between the two objects.

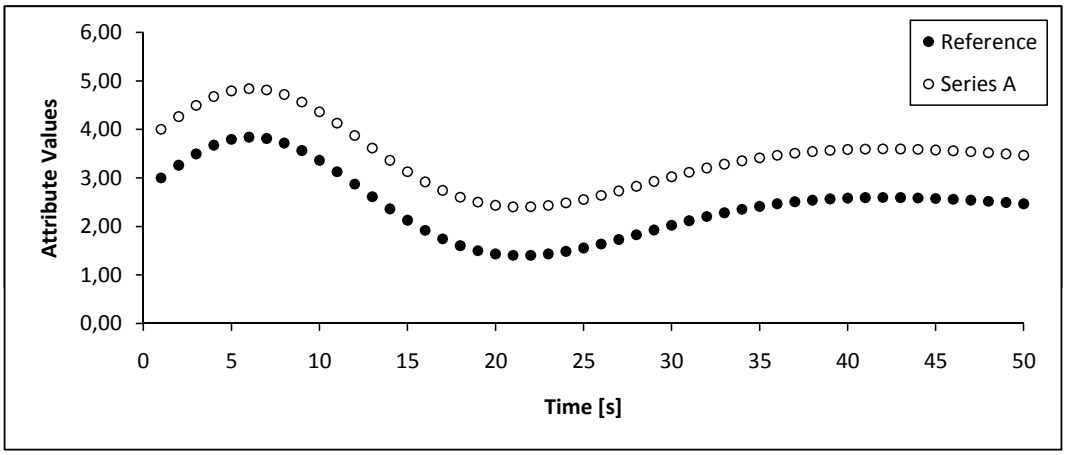
A common method to determine the similarity between two time-series is to interpret them as vectors in a metric space [ZADB06] and then to calculate one of the *Minkowski Distances*. Given two time-series $T = (t_1, \ldots, t_n)$ and $U = (u_1, \ldots, u_n)$ then these distances (also called L_p *distances*) are defined as follows:

$$L_p(T,U) = \sqrt[p]{\sum_{i=1}^{n} |t_i - u_i|^p}$$

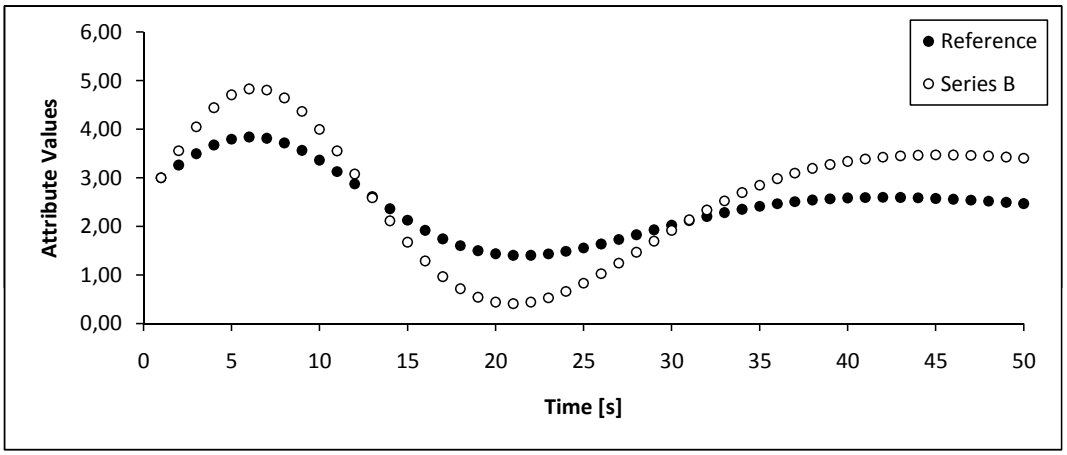
where L_1 is called the *Manhattan Distance*, L_2 the *Euclidean Distance* and L_∞ is known as the *Chessboard Distance*. In any case the similarity is determined by summing up the distance between two corresponding data points of T and U. For optimization reasons the calculation of the square root can be omitted because this does not alter the relative similarity ranking of objects according to a reference. Initial tests have shown that from the L_p distances using the Euclidean Distance leads to the best classification results and therefore it was chosen for the distance calculation of individual data points over Manhattan and Chessboard Distance.

5.3.1.2. Offset Translation and Amplitude Scaling

As stated above, two time-series are said to be similar if they have a small distance between each other. Intuitively, the distance between two time-series is small if they have the same shape. In Figure 5.5 some fictitious sample time-series are depicted with developing values of an arbitrary attribute over a period of 50 seconds. Considering Figure 5.5(a) it is obvious that the reference series and series A are similar because they have the exact same shape. Series B in Figure 5.5(b) has the same shape as the reference series, the only difference being that it has a different amplitude. But nevertheless, one would also consider it as similar (though not as similar as series A).



(a) Offset Translation



(b) Amplitude Scaling

Figure 5.5.: Distortions leading to high distance despite similar shapes.

5.3.1.2.1. Time Series Normalization In both cases the Euclidean Distance would determine a high distance although the respective series have obviously similar shapes. Strictly speaking, calculating the Euclidean distance between the reference series and series A yields a distance of 44.83. However, the distance between the reference series and series B amounts to only 40.75. Intuitively, one would have expected a distance of 0 between the reference series and series A and a small distance between the reference and series B. In order to deal with these two types of time-series distortions known as *offset translation* and *amplitude scaling* a normalization before calculating the distance appears to be reasonable (e.g. [ER08]).

Let

$$T = (t_1, \ldots, t_n)$$

be a time-series. Then the normalized time-series $\widehat{T}$ is acquired by subtracting the average

value of the series, $\bar{t}$, from the individual data points and then dividing them by the standard deviation of the values, $\sigma(t)$ [GK95].

$$\widehat{T} = \left(\frac{t_1 - \bar{t}}{\sigma(t)}, \ldots, \frac{t_n - \bar{t}}{\sigma(t)} \right)$$

with

$$\bar{t} = \frac{1}{n} \cdot \sum_{i=1}^{n} t_i \qquad \text{and} \qquad \sigma(t) = \sqrt{\sum_{i=1}^{n} (t_i - \bar{t})^2}$$

However, this procedure makes sense if, and only if, the shape of the two series is relevant and not their absolute values. To demonstrate the effect Figure 5.6 shows series B after the normalization. The shape has been kept the same but the amplitude scaling effect has almost disappeared.

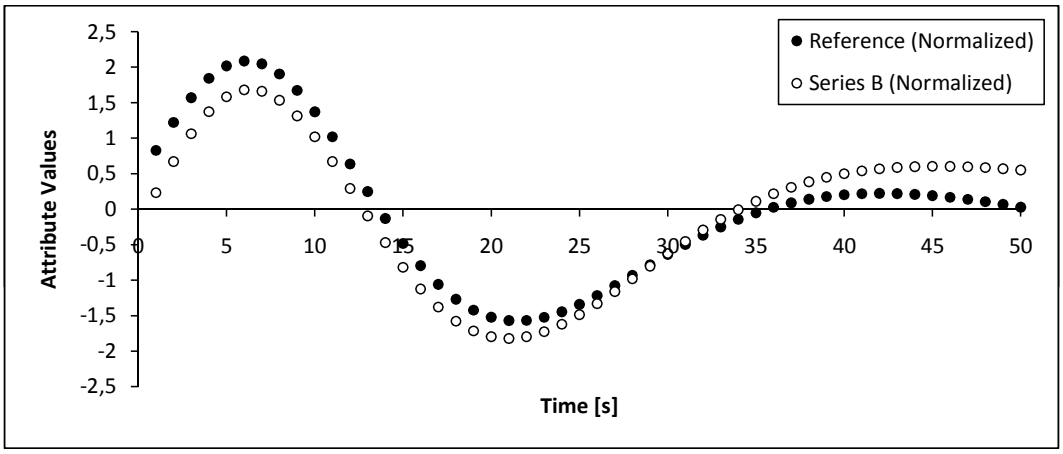


Figure 5.6.: Normalization of time-series

Table 5.21 shows the different distances that were calculated between the reference series and series B before and after the normalization. It can be seen that the series has a distance almost twice as great before the normalization. Therefore it holds true that the normalization helps in matching similar shapes despite time series distortions like offset translation and amplitude scaling.

5.3.1.3. Stretching and Compression

Another important problem with time-series similarity is the occurrence of *stretching* or *compression* which can be present either locally or globally. If the time-series is based on data

Table 5.21.: Impact of Normalization on Distances to a Reference Series

Series	**Distance**
Series B	40.75
Series B (normalized)	21.85

acquired from a human interaction, e.g., the movement of a pallet through an RFID portal, then compression can be the result of a faster movement and stretching the result of a slower movement by the warehouseman. This effect is described for example in [KPZ+04, PB02]. Figure 5.7 shows the reference series together with a compressed version of itself denoted as series C. In this case the Euclidean Distance is not defined at all, because after 35 seconds no data exists for series C. This is problematic because it is necessary to calculate the distance between two corresponding data points from each series.

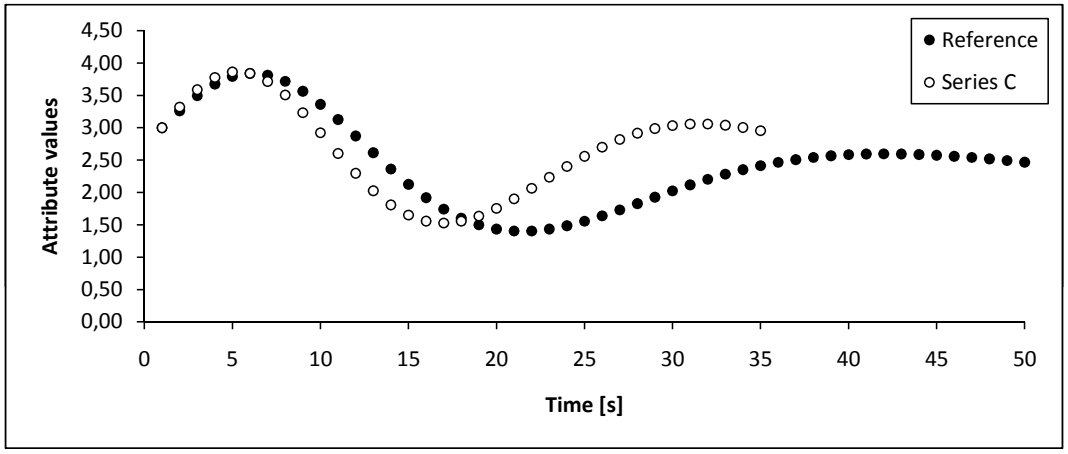


Figure 5.7.: Compressed time-series

It would be intuitively reasonable to *stretch* series C to the length of the reference series or to *compress* the reference series to the length of series C. However, two more meaningful approaches to deal with this problem can be found in the literature: *Uniform Scaling* and *Dynamic Time Warping* [FKL+05]. The main difference between them is that Uniform Scaling tries to perform a *global* compression or stretching and Dynamic Time Warping does this only *locally*. Both approaches are explained in the following.

5.3.1.3.1. Uniform Scaling The idea of Uniform Scaling is to perform a uniform warping of time to address the effect of shrunken or stretched time-series and was first proposed in [Keo03]. In order to calculate the similarity between two different time-series using some kind of distance function like the Euclidean Distance introduced above, it has to be clear which data point of the one series has to be compared to which data point of the other series. Let $T = (t_1, \ldots, t_n)$ and $U = (u_1, \ldots, u_m)$ be two different time-series with $n < m$. Shrinking U to the size of T is not a valid option because this would mean a loss of information and so T has to be stretched to the size of U. Consequently $m - n$ data points have to be inserted into T resulting in a new time-series $T' = (t'_1, \ldots, t'_m)$ with length m. The distance d between T and U is then calculated by

$$d((t_1, \ldots, t_n), (u_1, \ldots, u_m)) = d((t'_1, \ldots, t'_m), (u_1, \ldots, u_m))$$

where the individual data points t'_j are calculated as follows:

$$t'_j = u_{\lfloor \frac{j \cdot n}{m} \rfloor} \qquad \text{where } 1 \leq j \leq m$$

The resulting time-series after applying Uniform Scaling to series C is shown in Figure 5.8.

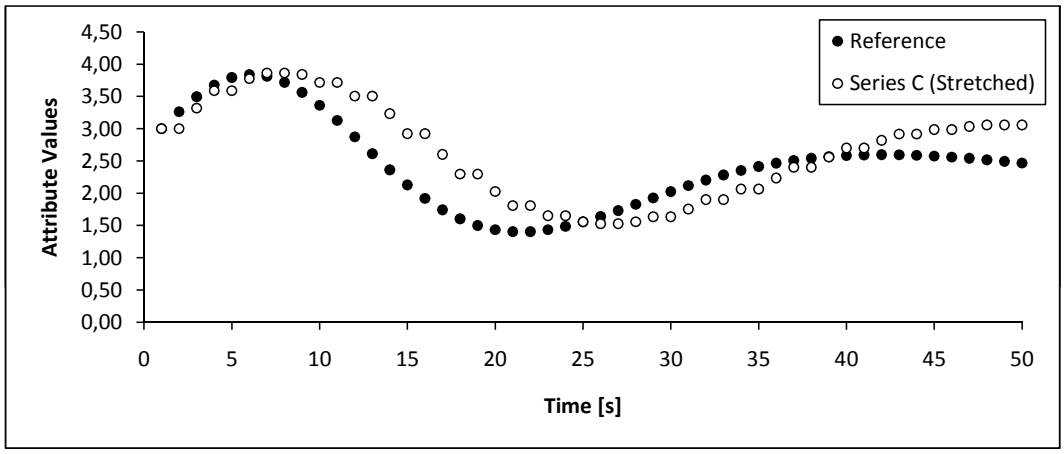


Figure 5.8.: Uniform Scaling

Because every data point in the reference series now has a corresponding data point in the scaled series C, the Euclidean Distance can be calculated in the usual way.

5.3.1.3.2. Dynamic Time Warping Rather than using a global stretching factor to scale a time-series, Dynamic Time Warping (DTW) uses local scaling to determine the distance between two time-series. This can be interpreted as a temporary acceleration or deceleration of

the warehouseman moving the pallet through a portal. Originally, this approach was introduced as a technique for speech recognition to cope with different speaking speeds [SC78]. Today, Dynamic Time Warping is successfully applied to all kinds of data, including for example audio, video and graphics data, in multiple disciplines such as computer science, biology, medicine and economics. The most important difference compared to all other distance measures described so far is that Dynamic Time Warping is far more flexible as it dynamically chooses which data point pairs are compared to each other.

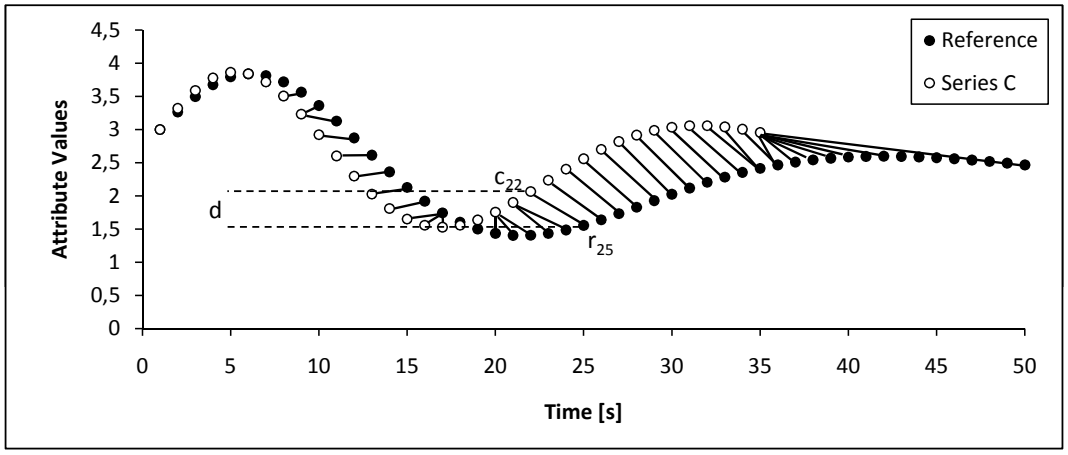


Figure 5.9.: Dynamic Time Warping

Dynamic Time Warping is formally defined as

$$
\begin{aligned}
DTW((),()) &= 0 \\
DTW(T,()) &= DTW((),U) = \infty \\
DTW(T,U) &= d(t_n,u_m) + \min \begin{cases} DTW((t_1,\ldots,t_{n-1}),(u_1,\ldots,u_{m-1})) \\ DTW((t_1,\ldots,t_n),(u_1,\ldots,u_{m-1})) \\ DTW((t_1,\ldots,t_{n-1}),(u_1,\ldots,u_m)) \end{cases}
\end{aligned}
$$

where d is called the *ground distance function* as it is used to determine the distance between two single data points and not the entire time-series. Usually the Euclidean Distance or the Manhattan distance is used for this purpose. Figure 5.9 shows Dynamic Time Warping applied to the compressed series C. Note that the distance is now determined between different data points than in the previous similarity measures. In particular, a single data point can be used multiple times as a reference to a data point of the other time series. This effect becomes most apparent at the last data point of C where it is compared to the remaining data points of the reference series. Note that the line drawn between two data points does not correspond to their

distance but is used only for presentation purposes to show which points are compared. The distance is still determined by the difference between their absolute values. For example, DTW would compare the two data points c_{22} to r_{25} which, using Euclidean Distance, results in a distance of

$$d(c_{22}, r_{25}) = d(2.06, 1.55) = \sqrt{(2.06 - 1.55)^2} = 0.51$$

Table 5.22.: Distances between Reference Series and compressed Series C

Approach	**Distance**
Euclidean Distance	Not defined
Uniform Scaling	29.28
Dynamic Time Warping	7.42

Initial experiments were performed to determine whether Dynamic Time Warping or Uniform Scaling would lead to the best classification results with respect to the problem of stretched and/or compressed time-series. It turned out that Dynamic Time Warping was able to handle the variance in speed during a gathering-cycle much better than Uniform Scaling could. The reason for this is that the first has the ability to deal with complex local distortions (e.g., spontaneous acceleration or deceleration of the warehouseman), while the latter, in contrast, can only deal with global distortions. A comparison of the distances to the reference series resulting from Uniform Scaling and Dynamic Time Warping is shown in Table 5.22.

5.3.2. Generation of Reference Time-Series

As previously stated, the tag-event level approach tries to determine whether the time-series of a tag is more similar to a typical moved or more similar to a typical static time-series. The tricky part of this approach, besides the concept of similarity, is the identification of such *typical* time-series. In the previous section different time-series were compared to a so-called reference series. This section therefore aims at the generation of such a reference series for moved and static time-series, respectively, against which the tags can be compared afterwards. In general, a reference series R for a tag class C has to 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

The question is how to find such a reference series. Basically there are two possibilities:

- Use an existing time-series from the sample data set as a reference (called *median approach*)

- Construct a new reference from the existing sample data (called *mean approach*)

5.3.2.1. Median Approach

The first possibility is rather simple and the pseudo code of the procedure for retrieving the reference time-series for the moved tags is shown in Algorithm 6. For each moved time-series m the average distance to all other moved series $\mathbb{M} \setminus m$ is calculated. The time-series which has the least average distance to all others is obviously the most typical and is thus chosen as the reference time-series for all moved tags.

Algorithm 6: Median approach to reference series generation

```
Let 𝕄 be the set of all moved time-series.
Let d be a distance function (e.g., Dynamic Time Warping)
foreach time-series m ∈ 𝕄 do
    m_d = 0
    foreach time-series s ∈ 𝕄 \ m do
        m_d = m_d + dist(m, s)
    end
    m_d = m_d / |𝕄|
end
return m ∈ 𝕄 with minimum m_d
```

An alternative way of identifying the reference series is not to calculate the average *minimum distance to all moved series* but to calculate the average *maximum distance to all static series*. However, initial tests have shown that the first approach results in a much better classification performance as a reference series having a high distance to all static tags does not necessarily need to have a low distance to the moved tags. Or in other words: dissimilarity to static tags does not implicate similarity to moved tags. Consequently such a reference series is useless.

5.3.2.2. Mean Approach

The major drawback of the median approach is that the reference series has to be one that already exists in the sample data set. Consequently the approach is both limited by, and depends upon, the number of sample time-series available for the two tag classes. It seems likely that creating a completely new reference series is probably going to yield better results.

The second approach presented here is called *mean approach* and the pseudo code is shown in Algorithm 7. The idea is that from all available samples an average time-series is calculated and returned as the reference. This 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:

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

Or more generally if there are k different time-series $\mathbb{M} = \{T_1, \ldots, T_k\}$ then an average data point v_i is calculated by

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

Algorithm 7: Mean approach to reference series generation

```
input  : Number of Intervals k,
         Interval length Δt,
         Set 𝕄 of sample time-series with |𝕄| = n
output: Reference time-series R = (r_1, ..., r_k)
foreach time-series M ∈ 𝕄 do
    Interpolate M
end
for i = 1 to k do
    r_i = 0
    foreach time-series M ∈ 𝕄 do
        r_i = r_i + m_i
    end
    r_i = r_i / n
end
return R = (r_1, ..., r_n)
```

However, this technique requires that all time-series have the same length, since only in this case can an average value be computed. Uniform Scaling was presented above as an approach to compress or stretch time-series to the same length. However, compressing and stretching

each and every time-series in the sample data set to a specific length is computationally very expensive - especially because a suitable length is very difficult to choose. Another problem with this approach is that every single data point (i.e., tag-event) is taken into account and is sometimes repeated multiple times in order to stretch a time-series. This in turn means that the information about the timestamp showing exactly when the tag-event occurred is blurred and can hardly be reconstructed. To deal with this problem another approach is presented here to interpolate a time-series while keeping the temporal order of the individual tag-events.

5.3.2.2.1. Time-Series Interpolation The entire gathering-cycle is divided into k time intervals of equal length Δt. Consequently the reference series R is going to have a length of k data points. If

$$M = (m_1, \ldots, m_n)$$

is a time-series with corresponding timestamps

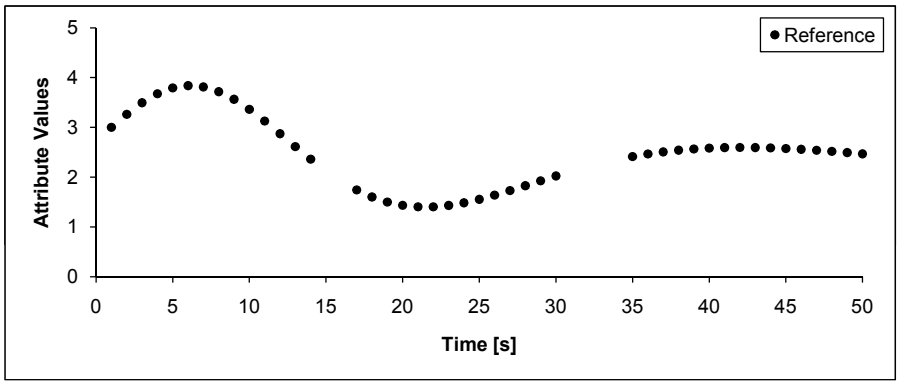
$$(t_1, \ldots, t_n)$$

then the k-th data point of R is the average of all data points of M that lie within the interval

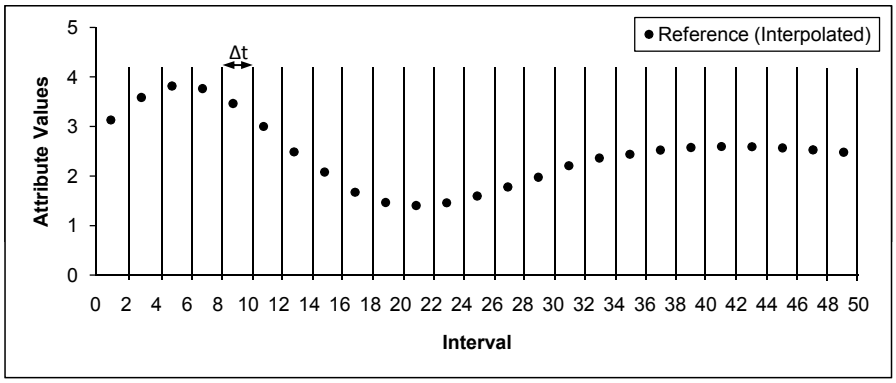
$$I = [\Delta t \cdot (k); \Delta t \cdot (k+1)].$$

In a case where no tag-event occurred in a specific interval then the two preceding and the two succeeding tag-events will be averaged and used as the interpolation. If there are no preceding or succeeding tag-events then the first or the last tag-event is used, respectively.

An example of this procedure is shown in Figure 5.10. Figure 5.10(a) shows the reference time-series introduced above before the interpolation. Note that six data points have been removed from the series (seconds 15-16 and seconds 31-34) to show the effect of interpolating missing data points in an interval. Figure 5.10(b) shows the same series after the interpolation. The period of 50 seconds has been divided into $k = 25$ intervals with an equal length of $\Delta t = 2$ seconds. Some important effects can be observed here: for example on the one hand, the shape of the time-series has been retained, including the temporal order of the individual tag-events; and on the other hand, in the intervals without any tag-events the mean approach is able to successfully interpolate data points. This also holds true for the case where multiple intervals in a row are missing tag-events.



(a) Before Interpolation



(b) After Interpolation

Figure 5.10.: Interpolation of a time-series

5.3.2.3. Identification of different Reference Series per Class

Observations in the METRO distribution center in Unna have shown that there is no one and only typical reference time-series for moved or static tags. Rather, further *sub classes* exist within each of the two tag-classes. For example, in Section 3.1.1.3 two different ways to retrieve a pallet were described. In the one case, the warehouseman retrieves a pallet from the staging area and then directly loads it into the container. In the other case, the warehouseman loads one of the pallets that he previously buffered near the portal before. It is obvious that in these two cases different shapes exist for the corresponding time-series, because in the second case the warehouseman needs to rearrange the pallet in front of the portal before moving through, in the first case he can pass it through the portal directly. Still, both pallets have been moved. Furthermore it is, for example, possible that static tags which have already been loaded into the truck show a different behavior than static tags buffered near the portal. In order to improve the classification of moved and static tags these sub-classes have to be identified and the respective reference time-series have to be generated.

A common method used in machine learning to find such sub-classes is called *Cluster Analysis*.

Various clustering methods exist with k-Means [Mac67] and k-Medoid [KR90] being the most popular. These two methods are called *partitioning* methods because they aim at partitioning the data set into k disjunctive sub sets where each is represented by an individual cluster center. The major drawback of these two algorithms compared to other clustering algorithms is that the number of clusters, k, has to be chosen in advance. To solve this problem the performances of the algorithms using different k values are compared. The general algorithm for creating such a clustering is shown in Algorithm 8.

Algorithm 8: Partitioning Clustering

input : Set $\mathbb{M}$ of sample time-series with $|\mathbb{M}| = n$
Number of clusters k
choose k random cluster centers $c_1, \ldots, c_k$ for clusters $C_1, \ldots, C_k$
repeat
  foreach *time-series* $m \in \mathbb{M}$ **do**
    assign m to the closest cluster C, i.e., where $d(m, c)$ is minimal
  end
  foreach *clusters* $C \in \mathbb{C}$ **do**
    calculate new cluster center c
  end
until *No more changes in clustering*

Initially the k cluster centers are chosen randomly. Then every time-series in the data set is assigned to the cluster where the distance to the cluster center is minimal. After all time-series have been assigned to a cluster the cluster centers for each cluster are recalculated. Again all time-series are assigned to the new clusters where the distance to the center is minimal. This procedure is repeated until there are no more changes in the clustering, i.e., the cluster centers do not move after the recalculation.

Another drawback of these partitioning cluster algorithms is that the result depends on the initial choice of the k cluster centers which means that different initial cluster centers yield a different result. Consequently, the clustering is repeated multiple times for each value of k and the best clustering is returned. At this point it is necessary to clarify how to measure the goodness of a clustering and when to choose one clustering over another. The quality of a clustering depends on how similar time-series in the same cluster are to each other and how dissimilar they are to time-series in other clusters.

A common method to determine this is to use the Davies-Bouldin-Index DB [DB79]. For a

clustering with k clusters it is defined in the following way:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \left\{ \frac{\sigma(C_i) + \sigma(C_j)}{\delta(C_i, C, j)} \right\}$$

where $\sigma(C_i)$ is denoted *intra-cluster distance* and is a measure of scatter of the objects within a cluster and $\delta(C_i, C_j)$ is denoted *inter-cluster distance* and corresponds to the distance between two clusters. For a cluster C with n objects o_i and a cluster center c, $\sigma(C)$ is defined as follows:

$$\sigma(C) = 2 \left(\frac{1}{|C|} \sum_{i=1}^{n} d(o_i, c) \right)$$

where $d(o_i, c)$ corresponds to the distance between an object o_i and the cluster center c. Theoretically any distance measure (for example the Euclidean Distance) can be used here, but since Dynamic Time Warping showed the best results it is chosen here. The distance between two clusters C_i and C_j, denoted as $\delta(C_i, C_j)$ is defined as the distance between their cluster centers c_i and c_j:

$$\delta(C_i, C_j) = d(c_i, c_j)$$

Small DB values correspond to compact clusters where the centers of different clusters are far away from each other. Consequently, the clustering minimizing DB is considered the best clustering.

As stated above, the initial choice of the k cluster centers as well as the number of clusters, k itself, is unknown in advance. Calculating DB multiple times for different k values results in the best choice for these parameters.

5.3.3. Classification using Time-Series Analysis

In the previous section it was shown how the similarity between time-series is calculated and how reference time-series are created that correspond to typical moved and static tags. This section is going to show how the concept of similarity is used for building a classification model. Usually, when dealing with similarity, a so called *similarity query* is performed. The most common types of similarity queries are *ε-range query* and *k-nearest neighbor query*, which are both explained below.

5.3.3.1. ε-Range Query

The general idea of the ε-range query is to retrieve all objects that have a similarity of at least ε to the query object. If t is the tag to classify, $\mathbb{M}$ and $\mathbb{S}$ are the sets of moved and static reference series, respectively, and d is the distance function (e.g., Dynamic Time Warping), then the ε-range query R is formally defined as

$$R(t, \varepsilon) = \{r \in \mathbb{M} \cup \mathbb{S} | d(t, u) \leq \varepsilon\}$$

If a small value is chosen for ε, then the resulting set contains all reference time-series that are very similar to the tag of interest. In turn, choosing a higher value for ε results in a set containing more references that are similar to the tag of interest. In both cases a majority vote is performed, thus classifying the tag as the class of the majority of the references in this set. This majority voting can be tuned further by ranking the references according to their distance.

5.3.3.2. k-Nearest Neighbor Query

Choosing a meaningful value for ε is often not an easy task. If the value is too small, then the resulting set is empty. In the case of a too large ε value the resulting set may contain non-significant references. Therefore, an alternative way of retrieving similar references is presented, the k-nearest neighbor query (k-NN). The general idea of this approach is to retrieve the k nearest neighbors, which correspond to the references which are closest to the query tag. In the case of $k = 1$ only the closest, i.e., the most similar reference is returned. The k-NN query is formally defined as

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

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., Dynamic Time Warping). First of all, this query type can be used to find out 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, then usually a majority voting is performed to determine the class type of the query tag. Secondly, choosing $k = 1$ returns only the nearest neighbor and consequently the query tag is classified as the corresponding class.

5.3.3.3. Time Series Attributes

The similarity queries presented above can be used in various ways to determine the class type for a tag of interest and it is reasonable to use an approach that is able to combine the strengths of both of them. In Section 5.1 decision trees were introduced to find the best combination of attributes and their values. Consequently, the same approach is used here to combine different similarity queries, especially because it is easy to eventually combine both classification techniques (i.e., tag-occurrence level classification and tag-event level classification). However, in order to successfully apply decision tree learning it is required that attributes in the form of the Domain Attributes exist. In the following the list of attributes to describe a tag on the basis of the time-series similarity queries is presented.

$\mathbf{D_M}, \mathbf{D_S}$ The distance between the tag and the reference time-series of *all* moved and static tags, respectively. By nature, moved tags should have a lower distance to this time-series than static tags do and vice-versa.

$\mathbf{D_{M,C_i}}, \mathbf{D_{S,C_j}}$ In cases where the i sub-classes have been identified for the moved tags and j sub-classes for the static tags then this attribute corresponds to the distance to the respective cluster reference time-series. By nature, moved tags should have a lower distance to the moved reference time-series and static tags should have a lower distance to the static reference time-series.

$\mathbf{D_{M,Min}}, \mathbf{D_{S,Min}}$ The minimum of the distances to all available moved and static reference time-series, respectively. By nature, moved tags should have a lower minimum distance to the moved reference time-series and static tags should have a lower minimum distance to the static reference time-series.

$$D_{M,Min} = \min D_M \cup \{D_{M,C_i}\} \qquad D_{S,Min} = \min D_S \cup \{D_{S,C_j}\}$$

$\mathbf{D_{M,Max}}, \mathbf{D_{S,Max}}$ The maximum of the distances to all available moved and static reference time-series, respectively. By nature, moved tags should have a lower maximum distance to the moved reference time-series and static tags should have a lower maximum distance to the static reference time-series.

$$D_{M,Max} = \max D_M \cup \{D_{M,C_i}\} \qquad D_{S,Max} = \min D_s\{\cup D_{S,C_j}\}$$

$\mathbf{D_{M,Mean}}, \mathbf{D_{S,Mean}}$ The average of the distances to all available moved and static reference time-series, respectively. By nature, moved tags should have a lower average distance to the moved reference time-series and static tags should have a lower average distance to the static reference time-series. If there are n moved an m static sub-classes then this attribute is defined as follows:

$$D_{M,Mean} = \frac{1}{n+1}\left(D_M + \sum_{i=1}^{n} D_{M,C_i}\right) \qquad D_{S,Mean} = \frac{1}{m+1}\left(D_S + \sum_{j=1}^{m} D_{S,C_j}\right)$$

$\mathbf{D_{M,StDev}}, \mathbf{D_{S,StDev}}$ The standard deviation of the distances to all available moved and static reference time-series, respectively. By nature, moved tags should have a lower standard deviation of the distances to the moved reference time-series (analogous with static tags). If there are n moved an m static sub-classes then this attribute is defined as follows:

$$D_{M,StDev} = \sqrt{|D_M - D_{M,Mean}|^2 + \sum_{i=1}^{n}(D_{M,C_i} - D_{M,Mean})^2}$$

$$D_{S,StDev} = \sqrt{|D_S - D_{S,Mean}|^2 + \sum_{j=1}^{m}(D_{S,C_j} - D_{S,Mean})^2}$$

$\mathbf{D_{M,CoV}}, \mathbf{D_{S,CoV}}$ The coefficient of variation of the distances to all available moved reference time-series. By nature, moved tags should have a lower CoV value with respect to the distances to the moved reference time-series (analogous with static tags).

$$D_{M,CoV} = \frac{D_{M,StDev}}{D_{M,Mean}} \qquad D_{S,CoV} = \frac{D_{S,StDev}}{D_{S,Mean}}$$

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 and the nearest neighbor of a static tag should be a static reference.

$$NN = \begin{cases} Moved, & \text{if } D_{M,Min} < D_{S,Min} \\ Static, & \text{if } D_{M,Min} \geq D_{S,Min} \end{cases}$$

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 and the furthest neighbor of a static tag should be a moved reference.

$$FN = \begin{cases} Moved, & \text{if } D_{S,Max} < D_{M,Max} \\ Static, & \text{if } D_{S,Max} \geq D_{M,Max} \end{cases}$$

Agree$_{\mathbf{NN,FN}}$ Indicates whether NN and FN agree, i.e., whether the nearest neighbor and the furthest neighbor correspond to different tag classes. Where the nearest and the furthest neighbor correspond to the same 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.

$$Agree_{NN,FN} = \begin{cases} Unknown, & \text{if } NN = FN \\ Moved, & \text{if } NN \neq FN \text{ and } NN = Moved \\ Static, & \text{if } NN \neq FN \text{ and } NN = Static \end{cases}$$

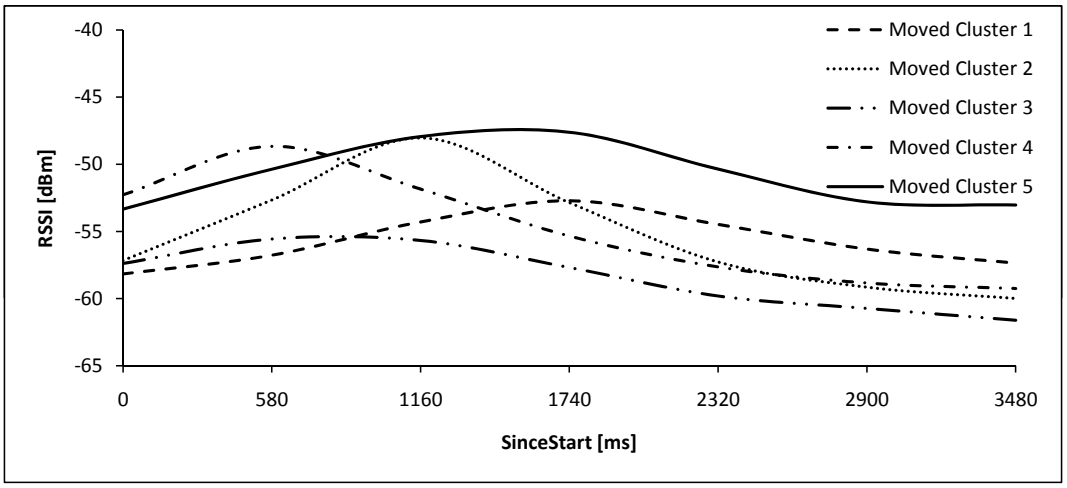
Agree$_{\mathbf{Main,Truck}}$ This attribute indicates whether the classification at the Main- and Truck antennas agree, i.e., whether the nearest neighbors determined at both the Main- and Truck antennas are of the same class. It is only applicable to the Satellite Portals. Note that only for the corresponding object classes, i.e., *moved* and *static* can the class precision be calculated.

$$Agree_{Main,Truck} \begin{cases} Moved, & \text{if } NN_{Main} = Moved \text{ and } NN_{Truck} = Moved \\ Static, & \text{if } NN_{Main} = Static \text{ and } NN_{Truck} = Static \\ Unknown, & \text{if } NN_{Main} \neq NN_{Truck} \end{cases}$$

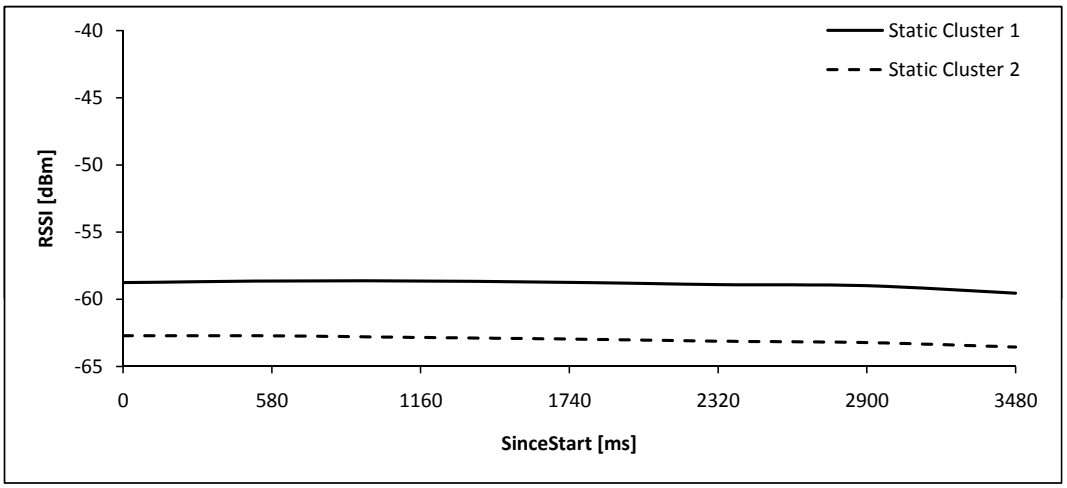
Agree$_{\mathbf{DC,Truck}}$ This attribute indicates whether the classification at the Main and Truck antennas agree, i.e., whether the nearest neighbors determined at both the Main and Truck antennas are of the same class. It is only applicable to the transition portals. Note that only for the corresponding object classes, i.e., *Moved* and *Static* can the class precision be calculated.

$$Agree_{Main,Truck} \begin{cases} Moved, & \text{if } NN_{DC} = Moved \text{ and } NN_{Truck} = Moved \\ Static, & \text{if } NN_{DC} = Static \text{ and } NN_{Truck} = Static \\ Unknown, & \text{if } NN_{DC} \neq NN_{Truck} \end{cases}$$

Figure 5.11 shows the reference series that have been generated for the Standard Portals. The clustering has led to 5 different moved reference series and 2 different static reference series. It is notable that the shapes do indeed match the original expectation stated in Section 3.2.3 of how moved and static tags *should* behave. While there are two different cases of false-positives that differ only in the overall signal strength they are read with, the moved cases are significantly different. As was expected, they all have a maximum RSSI value that corresponds to the point in time where the pallet is just passing the portal. However, the actual maximum RSSI values differ from -56dBm up to -47dBm and occur after different periods of time, ranging from 0.5 seconds up to 1.8 seconds after the beginning of the gathering-cycle. It is also interesting to see that only the first 3.5 seconds of the gathering-cycle carry meaningful information.



(a) Moved Reference Series



(b) Static Reference Series

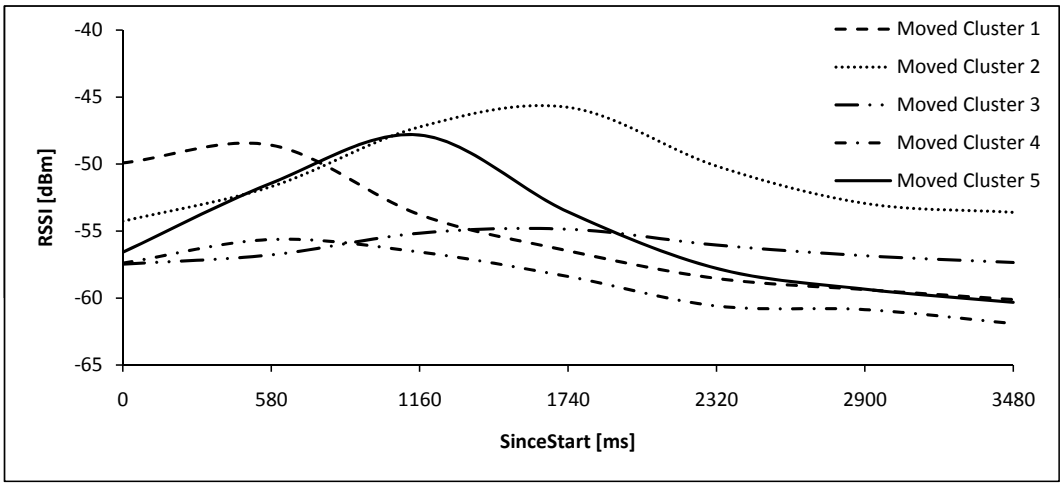
Figure 5.11.: Reference Time-Series (`STD_COMPLETE` Data Set)

The investigation of the corresponding attribute values is shown in Table 5.23. The detailed distances between the individual reference series can be found in the appendix in Table C.1. It can be observed that moved tags have a moved *nearest neighbor* in 96.6% and a static *furthest neighbor* in 92.3% of all cases. Although 12,230 (92.3%) moved tags have a moved reference as *nearest-* and a static reference as *furthest neighbor* it is notable that in 446 (3.4% of all) cases this is the other way around. However, these rates are significantly worse for the static tags, as although in 92.0% of all cases they have a moved *furthest neighbor*, only in 84.8% do they have a static *nearest neighbor*. It is notable too, that in 3,251 (7.9% of all) cases a static tag has a moved reference as *nearest-* and a static reference as *furthest neighbor*.

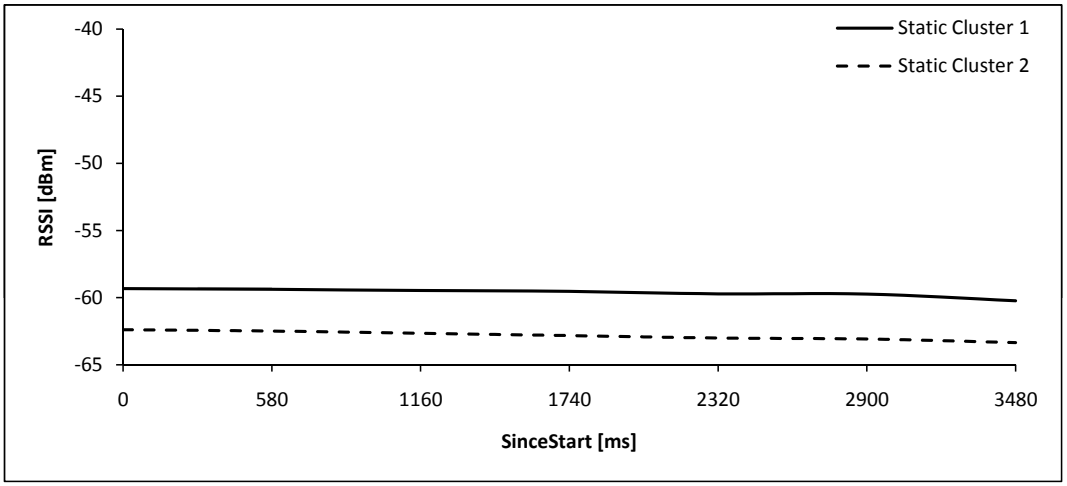
Table 5.23.: Time-Series Attribute Value Investigation (`STD_COMPLETE` Data Set)

Attribute	**Value**	**Moved Tags**	**Static Tags**	**Precision**
NN	Moved	12,799	6,206	**67.3%**
	Static	446	34,537	**98.7%**
	Recall	**96.6%**	**84.8%**	
FN	Moved	1,015	37,487	**97.4%**
	Static	12,230	3,256	**79.0%**
	Recall	**92.3%**	**92.0%**	
$Agree_{NN,FN}$	Moved	12,230	3,251	**79.0%**
	Static	446	34,532	**98.7%**
	Unknown	569	3,529	
	Recall	**92.3%**	**83.6%**	

Figure 5.12 shows the reference series that have been generated for the tags read only by the *Main Antennas* of the *Satellite Portals*. Like the Standard Portals the clustering has led to 5 different moved and 2 different static reference series. In general, as could be expected, the moved and static references appear to be very similar to those generated for the Standard Portals.



(a) Moved Reference Time-Series



(b) Static Reference Time-Series

Figure 5.12.: Moved and Static Reference Time-Series (`SAT_MAIN_ONLY` Data Set)

The investigation of the corresponding attribute values is shown in Table 5.24. The detailed distances between the individual reference series can be found in the appendix in Table C.2. It can be observed that the moved detection rate (i.e., the moved recall) is similar to the rates observed at the Standard Portals, except for the nearest neighbor rate which is a little bit worse. However, it is interesting that the static detection rate significantly increased. In contrast to the Standard Portals, the static detection rate increased by 2.8 percentage points for the *nearest-* and by 3.3 percentage points for the *furthest neighbor* attribute. In the case where a static reference was the *nearest-* and a moved reference was the *furthest neighbor* ($Agree_{NN,FN}$) the rate increased by an even 4.0 percentage points.

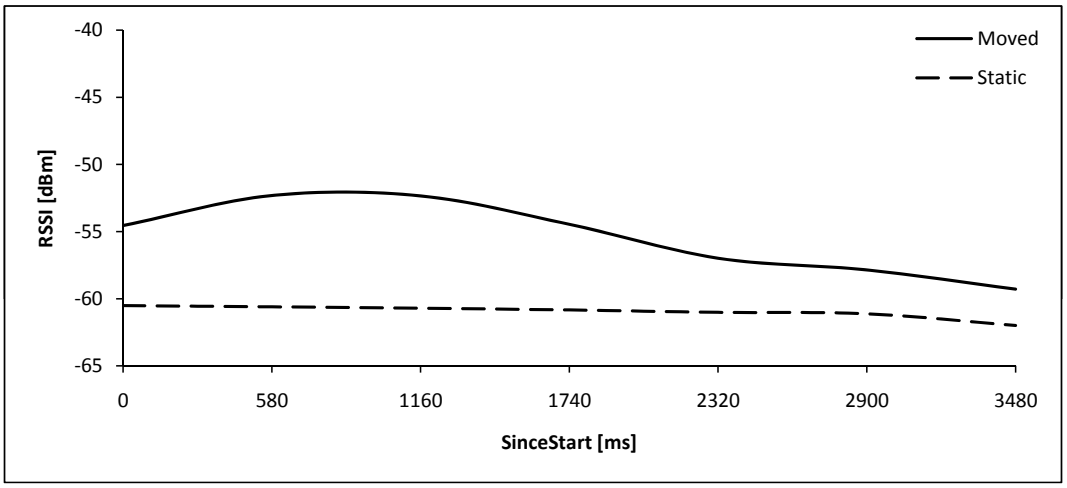
Figure 5.13 shows the moved and static reference series for the tags that were seen by both *Main-* and *Truck Antennas* at the *Satellite Portals*. In contrast to the two cases described

Table 5.24.: Time-Series Attribute Value Investigation (SAT_MAIN_ONLY Data Set)

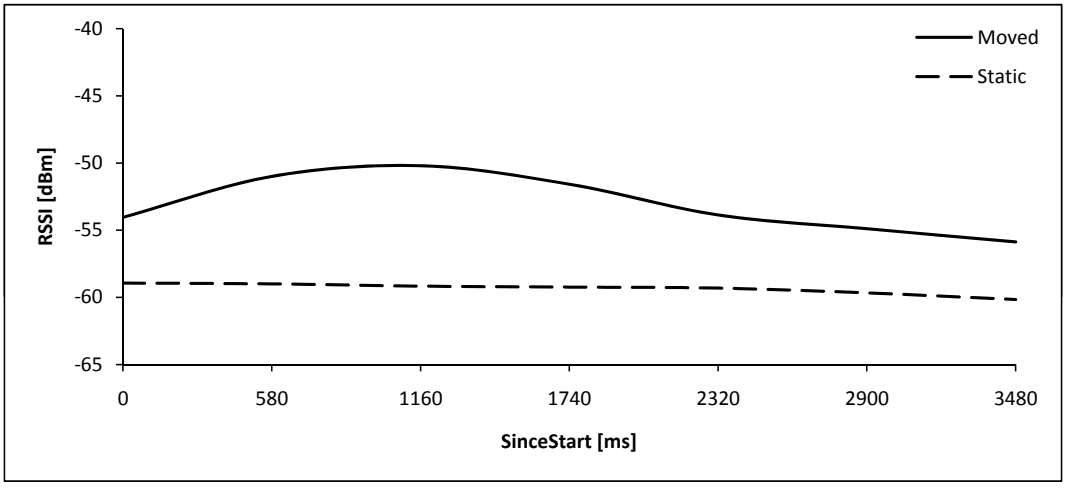
Attribute	Value	Moved Tags	Static Tags	Precision
NN	Moved	629	318	**66.4%**
	Static	27	2,253	**98.8%**
	Recall	**95.9%**	**87.6%**	
FN	Moved	51	2,450	**98.0%**
	Static	605	121	**83.3%**
	Recall	**92.2%**	**95.3%**	
$Agree_{NN,FN}$	Moved	605	121	**83.3%**
	Static	27	2,253	**98.8%**
	Unknown	24	197	
	Recall	**92.2%**	**87.6%**	

above only a single moved and static reference was generated for each antenna type because the cluster analysis did not improve the results. The data collected at the Main Antennas was then compared to the moved and static references of the Main Antenna and the data collected at the Truck Antennas compared to the references of the Truck Antennas. It is notable that the expectation of an increasing RSSI value can also be extended to the Truck Antennas.

The investigation of the corresponding attribute values is shown in Table 5.25. The detailed distances between the individual reference series can be found in the appendix in Table C.3. Note that the *furthest neighbor* attributes were omitted because there are only two references available. Consequently, if the one reference series is the *nearest-* then the other one is *furthest neighbor* by definition. Thus, the *nearest-* and *furthest neighbor* attributes carry the exact same information, thus leading to the exact same recall and precision rates. Looking at the *nearest neighbor* attribute of the Main Antennas, it can be observed that in contrast to the above cases the moved detection rate is somewhat worse while the static detection rate is slightly better. Looking at the Truck Antennas only, for 84.2% of the moved and 69.6% of the static tags the correct *nearest neighbor* could be determined. Only 79.6% of all moved tags were assigned the correct neighbor class by both the Main and Truck antennas. This ratio is even worse for the static tags where only 65.5% of all tags were classified correctly. However, if a tag is classified as static by the $Agree_{Main,Truck}$ attribute, then this decision has a precision of 92.2% ie., very confident.



(a) Main Antennas



(b) Truck Antennas

Figure 5.13.: Moved and Static Reference Time-Series (`SAT_MAIN_TRUCK` Data Set)

Figure 5.14 shows the moved and static reference series for the tags that were seen by both *DC-* and *Truck Antennas* at the *Transition Portals*. Like the Main- and Truck Antennas of the Satellite Portals, individual references have been generated for DC- and Truck Antennas. Moved pallets start off with a higher RSSI value that decreases over time while passing the DC Antennas. Subsequently the RSSI values measured at the Truck Antennas increase and then decrease around 1.5 seconds later. It is also interesting to see that the static tags are read with a constant RSSI value that eventually decreases towards the end.

The investigation of the corresponding attribute values is shown in Table 5.26. The detailed distances between the individual reference series can be found in the appendix in Table C.4. It is particularly notable that none of the attributes is able to achieve a classification rate over 88% on its own. However, the static precision of the NN_{DC} and $Agree_{DC,Truck}$ attributes is

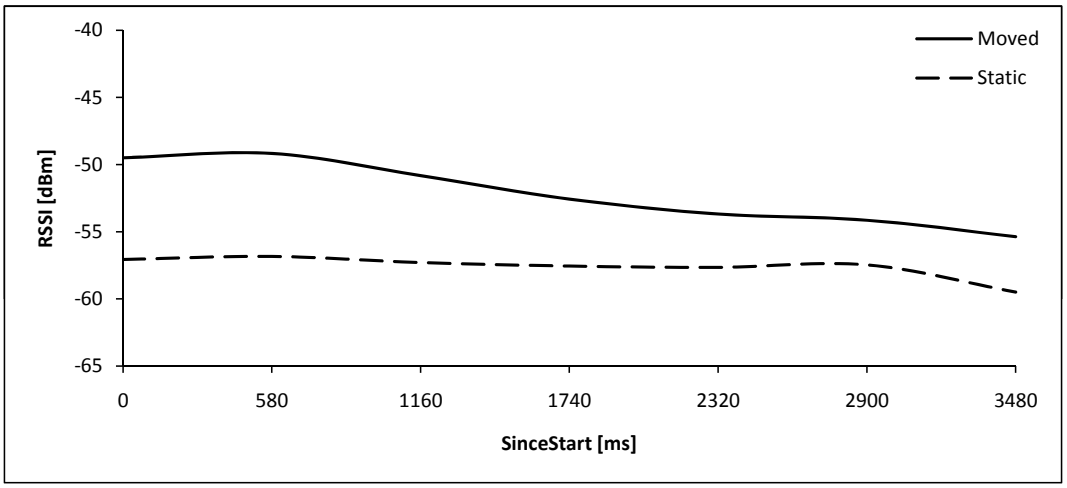
Table 5.25.: Time-Series Attribute Value Investigation (SAT_MAIN_TRUCK Data Set)

Attribute	Value	Moved Tags	Static Tags	Precision
NN_{Main}	Moved	1,187	216	**84.6%**
	Static	95	1,683	**94.7%**
	Recall	**92.6%**	**88.6%**	
NN_{Truck}	Moved	1,080	577	**65.2%**
	Static	202	1,322	**86.7%**
	Recall	**84.2%**	**69.6%**	
$Agree_{Main,Truck}$	Moved	1,021	138	**88.1%**
	Static	36	1,244	**97.2%**
	Unknown	225	517	
	Recall	**79.6%**	**65.5%**	

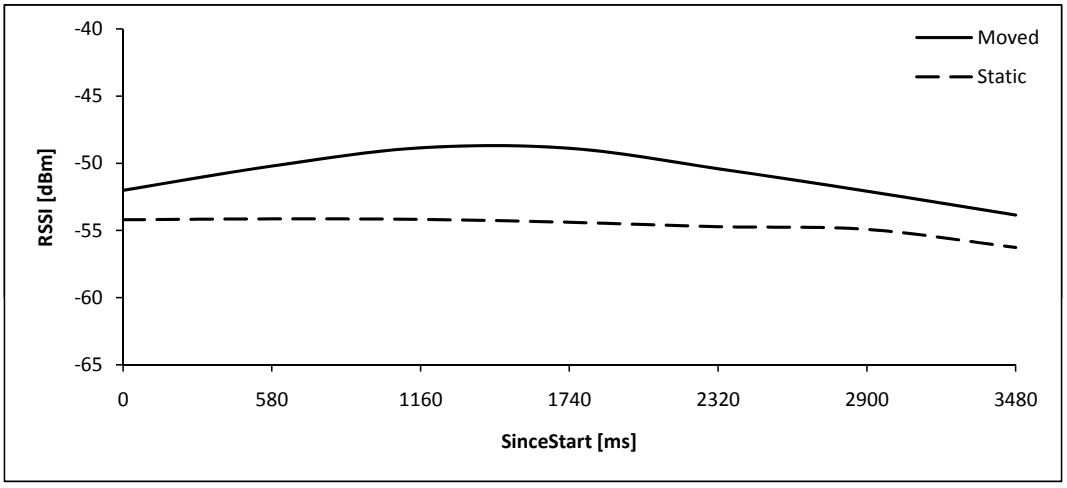
above 92.0% and is thus quite good.

Table 5.26.: Time-Series Attribute Value Investigation (TRA_BOTH Data Set)

Attribute	Value	Moved Tags	Static Tags	Precision
NN_{DC}	Moved	1,115	285	**79.6%**
	Static	184	2,107	**92.0%**
	Recall	**85.8%**	**88.1%**	
NN_{Truck}	Moved	1,089	819	**57.1%**
	Static	210	1,573	**88.2%**
	Recall	**83.8%**	**65.8%**	
$Agree_{DC,Truck}$	Moved	990	152	**86.7%**
	Static	85	1,440	**94.4%**
	Unknown	224	800	
	Recall	**76.2%**	**60.2%**	



(a) DC Antennas



(b) Truck Antennas

Figure 5.14.: Moved and Static Reference Time-Series (`TRA_BOTH` Data Set)

5.4. Further Classification Approaches

In the following sections the *combined classification approach* is presented together with the *exclusive approach* that can be used based on the assumption that the number of tags to be classified as either "moved" or "static" is known in advance.

5.4.1. Combined Classification Approach

What Tag-occurrence level classification and tag-event level classification have in common is that they describe a tag using a number of attributes. There are two different types of attributes, *numerical attributes* on the one hand and *nominal attributes* on the other. In both approaches these attributes are evaluated using a decision tree learner which eventually classifies a tag as

being either moved or static. It is a logical consequence that both attribute sets can be used as input to a combined decision tree. However, additional attributes can also be used to further improve the classification. First of all, the decisions of the two approaches themselves can serve as attributes. Secondly, some kind of a combined decision can be employed. The attributes that were used in the combined approach are described in the following:

C_{TO} The decision of the tag-occurrence based classification model. Results (see Chapter 6) have shown that this classification is already very confident.

$$C_{TO} = \begin{cases} Moved, & \text{if tag-occurrence based classification was “moved”} \\ Static, & \text{if tag-occurrence based classification was “static”} \end{cases}$$

C_{TE} The decision of the tag-event based classification model. Results (see Chapter 6) have shown that this classification is already very confident.

$$C_{TE} = \begin{cases} Moved, & \text{if tag-event based classification was “moved”} \\ Static, & \text{if tag-event based classification was “static”} \end{cases}$$

$Agree_{TO,TE}$ Indicates whether both classification models came to the same decision. Usually, if they both agree then the decision is very confident. If this is not the case, then a classification is performed on the basis of the other available attributes (i.e., tag-occurrence, tag-event and so on).

$$Agree_{TO,TE} = \begin{cases} Moved, & \text{if } C_{TO} = Moved \wedge C_{TE} = Moved \\ Static, & \text{if } C_{TE} = Static \wedge C_{TE} = Static \\ Unknown, & \text{if } C_{TE} \neq C_{TO} \end{cases}$$

5.4.2. The Exclusive Approach

5.4.2.1. Background

The framework presented here is capable of classifying moved and static tags based solely on the low-level readings collected during a gathering-cycle and independently of each other. This means that if multiple tags are moved through an RFID portal then the classification model is able to classify all of them as “moved”. However, there is another approach that was used for experimental purposes and that led to even better results on a selected subset of the sample data. Although it can’t be used in the scenario studied here, the RFID-enabled outgoing goods

process in the METRO distribution center, it is likely to be helpful for other processes. The approach is based on the attributes presented above and thus requires only a little adaption.

In many scenarios it is known in advance how many tags of interest have to be identified, for example in processes such as removing pallets from a high-rack storage area where exactly one pallet is retrieved every time. This is different from the scenario that this thesis is based on, because of the occurrence of *stacked pallets*. This term is used to describe a situation where sometimes multiple pallets with a very low height are stacked on top of each other and are then moved through the portal at the same time. Since each of these low-height pallets also still has its own RFID tag attached to it, it is possible that more than one tag seen during a gathering-cycle has been moved. Furthermore, in many cases no pallet was moved at all but because somebody was passing by the portal the motion sensor signaled the RFID reader to scan for pallets. However, where it is known and can be assured that a fixed number of tags has to be identified, then the following approach can be used:

5.4.2.2. Distance Ranking Classification

The idea is to interpret every tag as a vector with a fixed length corresponding to the number of attributes used to describe it. If, for example all RSSI attributes defined in Section 5.2.1.1 are used, then each tag T is a vector of length six:

$$T = [RSSI_{Min}, RSSI_{Max}, RSSI_{Diff}, RSSI_{Mean}, RSSI_{StDev}, RSSI_{CoV}]$$

Like the time-series similarity queries, some kind of reference object is needed to estimate the degree of similarity. Therefore, two reference vectors corresponding to typical moved and static tags are calculated. Without loss of generality the procedure is shown exemplarily for the class of moved tags. Let $T = [A_1, \ldots, A_k]$ be the representation of a tag using k different attributes. Then the average moved tag R is represented by

$$R = [\mu(A_1), \ldots, \mu(A_k)]$$

where $\mu(A_i)$ is calculated by averaging the attribute values of the n moved tag samples $\mathbb{M} = \{M_1, \ldots, M_n\}$ in the sample data set

$$\mu(A_i) = \frac{1}{n} \cdot \sum_{i=1}^{n} M_i.A$$

Similarity between a query tag $T = [A_1, \ldots, A_k]$ and the typical moved reference vector $R = [\mu(A_1), \ldots, \mu(A_k)]$ is then determined using the cosine distance

$$sim(T, R) = \cos(T, R) = \frac{T \times R}{|T| \cdot |R|} = \frac{\sum_{i=1}^{k} A_i \cdot \mu(A_i)}{\sqrt{\sum_{i=1}^{k} A_i^2} \sqrt{\sum_{i=1}^{k} \mu(A_i)^2}}$$

or the Euclidean distance $d(T, R)$:

$$sim(T, R) = 1 - d(T, R) = \sum_{i=1}^{k} |A_i - \mu(A_i)|$$

As stated in Section 5.3.1.1 the square root can be omitted without losing the similarity order. Because different attributes have different value ranges a normalization of the value space is required. The normalization of an attribute value v of an attribute A, denoted as v', is based on the interval [0,1] and is estimated by

$$v' = (v - \min A) \cdot \frac{1}{\max A - \min A}$$

where $\max A$ and $\min A$ correspond to the maximum and minimum attribute values of A. To classify a tag as moved or static, the cosine or Euclidean distance between the tag and the moved and static references is calculated. From all tags read during a gathering-cycle only the one which is most similar to the moved reference or most dissimilar to the static reference is classified as the moved tag. All other are classified as static, i.e., false-positives.

Example (Distance Ranking Classification)

The distance ranking approach is demonstrated by using three attributes $RSSI_{Max}$, $RSSI_{Diff}$ and $RSSI_{StDev}$ to describe each tag. Let T_1, T_2 and T_3 be the three tags read during a gathering-cycle with the following attribute values:

Tag	$\mathbf{RSSI_{Max}}$	$\mathbf{RSSI_{Diff}}$	$\mathbf{RSSI_{StDev}}$
$T_1 =$	[-49.2dBm,	15.2dBm,	4.9dBm]
$T_2 =$	[-39.2dBm,	27.0dBm,	6.3dBm]
$T_3 =$	[-43.7dBm,	19.6dBm,	9.5dBm]

According to Table 5.2 the moved reference vector M equals

$$M = [-44.1, 18.7, 5.3]$$

Normalizing all tags using the respective minimum and maximum values in Table 5.2 yields

$$\begin{aligned} T_1' &= [0.61, 0.55, 0.58] \\ T_2' &= [0.97, 0.98, 0.74] \\ T_3' &= [0.81, 0.71, 0, 80] \\ M' &= [0.80, 0.68, 0.62] \end{aligned}$$

The tag which is classified as moved is the one that has the minimum distance (or maximum similarity) to the moved reference vector. Using Euclidean distance this yields the results shown in Table 5.27.

Table 5.27.: Distance Ranking Classification Results

Tag	$\mathbf{d(Tag, M')}$	**Similarity**
T_1'	0.35	0.65
T_2'	0.59	0.41
T_3'	0.23	0.77

Because of the three tags T_3 is the one that is most similar to the reference vector, is it classified as the moved tag. T_1 and T_2 are consequently classified as false-positives.

5.5. Summary

This chapter introduced *Decision Trees* as the classification model of choice for our framework to distinguish between moved and static RFID tags. Two concurring approaches to decision tree learning, *C4.5* and *CART*, were introduced. Such decision trees consist of leaves and branches which correspond to a sequence of Yes/No questions which eventually lead to the final classification of a tag as being either moved or static. The individual Yes/No questions are representations of attribute value tests according to typical moved and static tag characteristics.

In addition, three major approaches for describing a tag's characteristics were introduced. The first approach, *Tag-Occurrence Level Classification*, examines the entirety of all the answers a specific tag gave to the RFID reader during a gathering-cycle. This then leads to the definition of a large number of so called *attributes* which can be used to describe specific characteristics of a tag. Since every tag-event is represented by the three dimensions *signal strength*, *timestamp* and corresponding RFID *reader antenna* there are three different groups of attributes.

Next, the difference between *domain attributes* (derived from experience and knowledge of people working in the environment) and *artificial attributes* (automatically constructed on the basis of the domain attributes) was explained. The aim and purpose of constructing and using artificial attributes was described in detail. Furthermore, a unique attribute, TO_{Count} was introduced which states how often a tag has been seen during previous loadings. Its special position is owed to the fact that it is the only attribute taking knowledge from the past (i.e., preceding pallet loadings) into account. Besides these, attributes on the basis of the low-level reader data so called *Logical Reader Attributes* were introduced. They are used to describe the order in which a tag was read by different readers installed at the same portal, if available.

The second approach, *Tag-Event Level Classification*, examines the individual answers (i.e., *tag-events*) a tag gave during a gathering-cycle. This sequence of temporally ordered data points can be interpreted as a time-series. Consequently, typical time-series for moved and static tags have been extracted from the sample data set. These series are called *reference series* and they are compared to the time-series originating from the tag-events of the tag of interest. It was explained in detail how the similarity of two time-series can be computed and *Dynamic Time Warping* was introduced as the major similarity measure used. Observations in the distribution center have shown that there are different types of moved and static tags. Consequently, *partitioning clustering* was used to identify such sub classes.

The third approach is a combination of the two preceding ones. The decisions of the tag-occurrence and tag-event level classification models are used as binary attributes themselves. These decisions can also be tested to see if both approaches agree as to whether a tag has been

moved or is static. Furthermore, all other attributes from the tag-occurrence and tag-event level approaches are used as input to the decision tree.

The chapter closed by proposing a fourth approach that relies on the attributes presented here but can only be used if it is known in advance how many tags need to be classified as moved (or static, depending on the scenario). In the scenario of an outgoing goods process this a priori knowledge cannot be assured, due to situations such as the occurrence of stacked pallets, for example. However, in several other processes, for example retrieving pallets from high-rack storage areas, where it is known in advance that exactly one pallet was retrieved and needs to be discriminated from among all other detected tags, the approach is likely to be very useful.

6. Evaluation

In Sections 3.1.2 and 3.1.3, the business objectives and the corresponding measurable data mining goals were identified. The most important business objective was, of course, to minimize the number of false-positive RFID tag reads and consequently the number of erroneously shipped pallets. In the first instance this is measured by the moved, static and overall detection rate (i.e., the classification accuracy) as defined in Section 3.3.1.3, as with this information the total number of actual incorrectly shipped pallets can be estimated.

Furthermore, it was stated that the classification model has to show a reliable performance over time because any significant variance in the classification performance inevitably prevents a productive use. In order to evaluate this quality measure the performance is averaged over individual days so that occasional performance outliers can be easily identified.

In total three different classification models were presented: *tag-occurrence-*, *tag-event-* and a *combined approach*; along with three different portal types, denoted as *Standard-*, *Satellite-* and *Transition Portals*. Because every classification approach was applied to every portal type, the best combination(s) could therefore be identified. Initially, each portal type with its corresponding classification model is evaluated and afterwards a summary is given comparing the results.

Although *classification accuracy* and *classification performance* over time act as the primary evaluation measures, it is also necessary to evaluate the remaining and secondary business objectives. Accordingly, to it was also evaluated to what extent the solution was able to generate additional knowledge and what further costs might be expected.

In addition, because the framework is supposed to be used in a productive environment, the deployment of the classification models in the METRO distribution center in Unna, Germany, and the resulting experiences are briefly described.

6.1. Evaluation of Classification Accuracy

In order to evaluate the ability to detect false-positive RFID tags, the moved, static and overall detection rates are calculated. The *detection rate* for a class is synonymous with the denom-

ination of the *class recall* as defined in Section 3.3.1.3, the *overall detection rate* corresponds to the *classification accuracy*. At this point, only these three performance measures plus the corresponding error rates are of interest, and these are therefore presented. In the following sections the different detection rates are presented for each approach and portal type.

6.1.1. Standard Portals

Table 6.1 shows the detection rates and the corresponding error rates that were achieved for moved, static and the whole of all tags on the `STD_COMPLETE` data set. Of the three classification models the *tag-occurrence level* is the most suitable for detecting false-positive RFID tag reads as it offers an error rate of only 1.49%. However, it also shows the worst moved detection rate with an error rate of 5.34%. In terms of the overall detection rate (2.00% error rate) and for the detection of moved tags (2.51% error rate) the combined approach shows the best performance. It is interesting to note that it is able to significantly improve the moved detection rate but cannot achieve such an improvement for the static detection rate.

Table 6.1.: Standard Portals - Detection Rates (`STD_COMPLETE` Data Set)

Approach	**Classification**	**Moved**	**Static**	**Overall**
Tag-Occurrence Level	Accuracy	94.66%	**98.51%**	97.57%
	Error	5.34%	**1.49%**	2.43%
Tag-Event Level	Accuracy	94.78%	97.95%	97.17%
	Error	5.22%	2.05%	2.83%
Combined Approach	Accuracy	**97.49%**	98.17%	**98.00%**
	Error	**2.51%**	1.83%	**2.00%**

6.1.2. Satellite Portals

In Section 4.3.2 it was stated that the data collected at the Satellite Portals was divided into 7 disjunctive data sets corresponding to the respective antennas that read a tag. It was further stated that only in the cases where either only the Main Antennas (SAT_MAIN_ONLY data set) or only Main- and Truck Antennas (`SAT_MAIN_ TRUCK` data set) read a tag was the development of a classification model necessary. The reasons for this were that in any other case (for example only Truck Antennas read a tag) hardly any moved tags were found.

Consequently, the detection rates achieved with these two individual data sets and the detection rates for the whole data set `SAT_COMPLETE` are presented.

Table 6.2.: Satellite Portals - Detection Rates (`SAT_MAIN_ONLY` Data Set)

Approach	**Classification**	**Moved**	**Static**	**Overall**
Tag-Occurrence Level	Accuracy	**98.63%**	98.87%	98.82%
	Error	**1.37%**	1.13%	1.18%
Tag-Event Level	Accuracy	98.32%	99.14%	98.98%
	Error	1.68%	0.86%	1.02%
Combined Approach	Accuracy	98.48%	**99.26%**	**99.10%**
	Error	1.52%	**0.74%**	**0.90%**

Table 6.2 shows the detection rates and the corresponding error rates that were achieved for moved, static and the entirety of the tags on the `SAT_MAIN_ONLY` data set. From the three classification models the *combined approach* is the most suitable for detecting false-positive RFID tag reads, with an error rate of only 0.74%. This also holds true for the overall detection rate, with an error rate of only 0.9%. However, the *tag-occurrence* approach outperforms the *combined approach* in terms of the moved detection rate.

It is interesting too, to note that taking the *combined approach* slightly outperforms the moved detection rate of the *tag-event approach* but is still worse than the *tag-occurrence approach*. However, it does have a significantly better static detection rate - leading in turn to a better overall detection rate.

Table 6.3.: Satellite Portals - Detection Rates (SAT_MAIN_TRUCK Data Set)

Approach	**Classification**	**Moved**	**Static**	**Overall**
Tag-Occurrence Level	Accuracy	**99.30%**	96.74%	97.77%
	Error	**0.70%**	3.26%	2.23%
Tag-Event Level	Accuracy	97.50%	97.42%	97.45%
	Error	2.50%	2.58%	2.55%
Combined Approach	Accuracy	98.28%	**98.68%**	**98.52%**
	Error	1.72%	**1.32%**	**1.48%**

Table 6.3 shows the detection rates and the corresponding error rates that were achieved for moved, static and the entirety of the tags in the SAT_MAIN_TRUCK data set. Among the three classification models the *combined approach* is the most suitable for detecting false-positive RFID tag reads, with an error rate of only 1.32%. This holds true also for the overall detection rate, with an error rate of only 1.48%. However, the *tag-occurrence* approach outperforms the *combined approach* in terms of the moved detection rate.

Like the SAT_MAIN_ONLY data set the combined approach has the best static and overall detection rate but cannot reach the moved detection rate of the *tag-occurrence approach*.

Table 6.4.: Satellite Portals - Detection Rates (SAT_COMPLETE Data Set)

Approach	**Classification**	**Moved**	**Static**	**Overall**
Tag-Occurrence Level	Accuracy	**97.41%**	99.29%	99.04%
	Error	**2.59%**	0.71%	0.96%
Tag-Event Level	Accuracy	96.14%	99.45%	99.01%
	Error	3.86%	0.55%	0.99%
Combined Approach	Accuracy	96.70%	**99.66%**	**99.26%**
	Error	3.30%	**0.34%**	**0.74%**

Table 6.4 shows the detection rates and the corresponding error rates that were achieved for moved, static and the entirety of the tags on the SAT_COMPLETE data set. Of the three classification models the *combined approach* is the most suitable for detecting false-positive RFID tag reads - with an error rate of only 0.34%. This also holds true for the overall detection rate, with an error rate of only 0.74%. Once again though, the *tag-occurrence approach* outperforms the *combined approach* in terms of the moved detection rate.

6.1.3. Transition Portals

As with the Satellite Portals, the data collected at the Transition Portals was divided into 3 disjunctive data sets (see Section 4.3.3) corresponding to the respective antennas that read a tag. It was stated that only for the case where a tag was read by both DC- and Truck Antennas was the development of a classification model necessary. In the case where only DC- or only Truck Antennas read a tag this affected hardly any moved tags. Consequently, the detection rates achieved at the TRA_BOTH data sets and the detection rates for the whole data set TRA_COMPLETE are presented.

Table 6.5.: Transition Portals - Detection Rates (TRA_BOTH Data Set)

Approach	**Classification**	**Moved**	**Static**	**Overall**
Tag-Occurrence Level	Accuracy	93.69%	**97.49%**	**96.15%**
	Error	6.31%	**2.51%**	**3.85%**
Tag-Event Level	Accuracy	**94.69%**	95.23%	95.04%
	Error	**5.31%**	4.77%	4.96%
Combined Approach	Accuracy	94.38%	93.73%	93.96%
	Error	5.62%	6.27%	6.04%

Table 6.5 shows the detection rates and the corresponding error rates that were achieved for moved, static and the entirety of the tags in the TRA_BOTH data set. Of the three classification models the *tag-occurrence approach* is the most suitable for detecting false-positive RFID tag reads - with an error rate of 2.51%. This also holds true for the overall detection rate, with an error rate of only 3.85%. However, in terms of the moved detection rate the *tag-event-* and the

combined approach do a better job.

Surprisingly, the *combined approach* is not able to reach the performance of the other two approaches and in fact the error rates are quite high compared to the other portal types anyway.

Table 6.6.: Transition Portals - Detection Rates (`TRA_COMPLETE` Data Set)

Approach	**Classification**	**Moved**	**Static**	**Overall**
Tag-Occurrence Level	Accuracy	89.62%	**99.52%**	**98.55%**
	Error	10.38%	**0.48%**	**1.45%**
Tag-Event Level	Accuracy	**90.57%**	99.09%	98.25%
	Error	**9.43%**	0.91%	1.75%
Combined Approach	Accuracy	90.28%	98.80%	97.96%
	Error	9.72%	1.20%	2.04%

Table 6.6 shows the detection rates and the corresponding error rates that were achieved for moved, static and the entirety of the tags in the `TRA_COMPLETE` data set. Of the three classification models the *tag-occurrence approach* is the most suitable for detecting false-positive RFID tag reads - with an error rate of only 0.48%. This also holds true for the overall detection rate, with an error rate of only 1.45%. However, in terms of the moved detection rate the *tag-event-* and the *combined approach* do a better job.

Like the other Transition Portal data set, the *combined approach* is not able to reach the classification performance of the other two approaches. It is interesting too, to note that the static detection rate is significantly higher than the moved detection rate.

6.1.4. Summary

Table 6.7 shows a comparison of the best approaches for the three portal types. In regard to the Standard- and Satellite Portals, it was found that a combination of the classification model working on the tag-occurrence and tag-event level leads to the best results. This implies that the two approaches work together very well and are able to support each other's classifications.

In terms of the false-positive detection rate it is encouraging to see that values well above the desired 99% can be reached. Using the Satellite Portals leads to a static detection error

rate of only 0.34% and an overall detection rate of only 0.74%. The Standard Portals have the best moved detection rates but the worst static and overall detection rates. Furthermore, it is notable that when using the Transition Portals it is significantly more difficult to detect moved pallets correctly.

Table 6.7.: Comparison of Portal Type Detection Rates

Portal Type	**Best Approach**	**Moved**	**Static**	**Overall**
Standard Portals	Combined	97.49%	98.17%	98.00%
Satellite Portals	Combined	96.70%	99.66%	99.26%
Transition Portals	Tag-Occurrence	89.62%	99.52%	98.55%

However, this does not mean, that, for example, 2 out of 100 read static pallets are wrongly billed to the retail store. In Section 1.3.3 different cases of false-positives were identified but not all of them proved really crucial. For example, if a static pallet located *inside the container* is incorrectly classified as moved, it is not that much of a problem, because it is apparently truly shipped with that loading. In contrast, if a static pallet located *in the staging area* is incorrectly classified as moved and it is not going to be shipped with that loading then this is critical.

Table 6.8 shows how many pallets were monitored at each portal type and how many of these were actually *critical false-positives*. It can be seen that in this context the Transition Portals show the best results. The data in this table can be interpreted as follows: Using Standard Portals there is 1 critical false-positive per 885 pallets. This ratio improves to 1 per 2955 pallets using the Satellite Portals and even to 1 per 4615 using the Transition Portals. However, the quality assurance steps in the workflow operate to ensure that these pallets are not going to be shipped. For example, after the loading of a container the warehouseman cross-checks the list of pallets classified as loaded with the list of order pallets. At the latest, any erroneous loading is detected at this point.

Table 6.8.: Detection of Critical False-Positives

Portal Type	Pallets	Critical FP	Accuracy
Standard Portals	53,988	61	99.887%
Satellite Portals	14,777	5	99.966%
Transition Portals	13,845	3	99.978%

The overall goal of this thesis was to create a classification model able to minimize the number of incorrect loadings by maximizing the classification accuracy. It was shown that using the approaches presented it is possible to reduce the number of erroneous pallet loadings to a ratio of less than 1 per 4500 pallets. Hence, the target of $> 99\%$ was very definitely achieved.

6.2. Evaluation of Performance Reliability

In order to evaluate the performance of the classification models over time the best algorithms, i.e., the combined approach for Standard- and Satellite Portals and the tag-occurrence approach for the Transition Portals are applied to the collected data. The performances (abbreviated as CA) are then averaged over the individual days. In addition, the number of pallets monitored on each particular day is depicted, thus providing additional information.

Unfortunately this method cannot serve as a completely independent evaluation source, because part of the data was already used for the actual classification model building. Although the data was split into a training and a test set several times during the model building phase, it is not possible to identify the exact sub set of tags that were used as the training set. However, because no other data is available there is no other choice, but that said it is actually not that much of a problem, because the *constancy* of the performance can also be shown using the known data.

6.2.1. Standard Portals

The classification performances achieved each day at the Standard Portals are shown in Table 6.9 and Figure 6.1. From April 24th, the classification performance was always around 97%. Before that though, there were two particularly notable drops - to 94% on April 4th, and March 12th. However, in each case where the performance achieved was quite low the number

of pallets monitored that day was also significantly below the average of 642 pallets. Bearing this in mind, the performances on these days could therefore be considered as *outliers* because the sample size is too small.

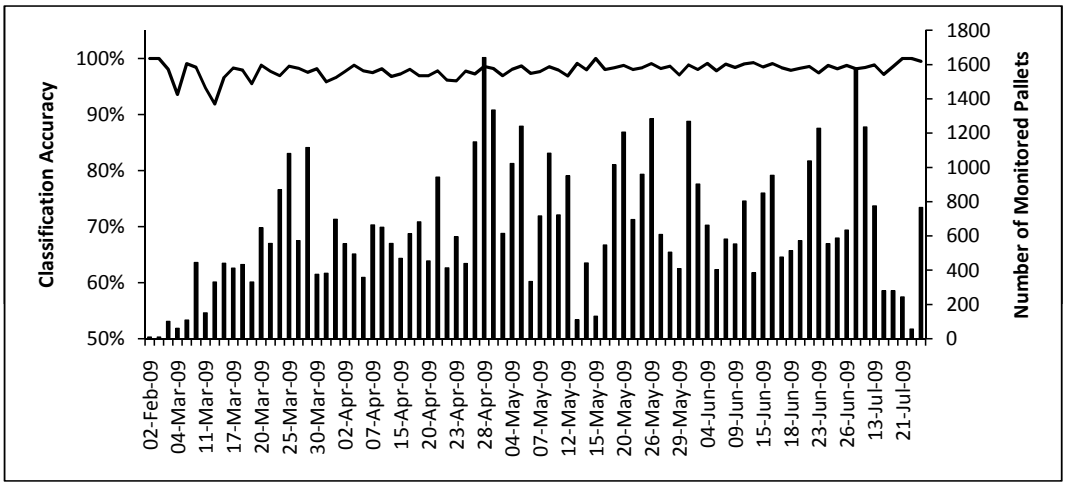


Figure 6.1.: Performance over Time at Standard Portals

Table 6.9.: Classification Accuracy per Day at the Standard Portals

Date	Tags	CA	Date	Tags	CA	Date	Tags	CA
2-Feb	10	100.0%	16-Apr	614	98.0%	28-May	506	98.6%
10-Feb	10	100.0%	17-Apr	683	96.9%	29-May	409	97.1%
2-Mar	102	98.0%	20-Apr	455	96.9%	2-Jun	1269	98.8%
4-Mar	62	93.5%	21-Apr	944	97.8%	3-Jun	903	98.0%
9-Mar	109	99.1%	22-Apr	414	96.1%	4-Jun	664	99.1%
10-Mar	446	98.4%	23-Apr	596	96.0%	5-Jun	403	97.8%
11-Mar	151	94.7%	24-Apr	440	97.7%	8-Jun	583	99.0%
12-Mar	331	91.8%	27-Apr	1149	97.2%	9-Jun	554	98.4%
16-Mar	441	96.6%	28-Apr	1642	98.5%	10-Jun	804	99.0%
17-Mar	412	98.3%	29-Apr	1335	98.2%	12-Jun	386	99.2%
18-Mar	433	97.9%	30-Apr	616	96.9%	15-Jun	851	98.5%
19-Mar	332	95.5%	4-May	1024	98.0%	16-Jun	955	99.1%
20-Mar	648	98.8%	5-May	1241	98.6%	17-Jun	477	98.3%
23-Mar	557	97.7%	6-May	335	97.3%	18-Jun	514	97.9%
24-Mar	870	96.9%	7-May	718	97.6%	19-Jun	573	98.3%
25-Mar	1082	98.6%	8-May	1083	98.5%	22-Jun	1038	98.6%
26-Mar	574	98.3%	11-May	723	97.9%	23-Jun	1229	97.4%
27-Mar	1117	97.5%	12-May	952	96.8%	24-Jun	556	98.7%
30-Mar	377	98.1%	13-May	111	99.1%	25-Jun	589	98.1%
31-Mar	383	95.8%	14-May	443	98.0%	26-Jun	635	98.7%
1-Apr	698	96.6%	15-May	132	100.0%	29-Jun	1581	98.2%
2-Apr	555	97.7%	18-May	548	98.0%	30-Jun	1237	98.4%
3-Apr	495	98.8%	19-May	1016	98.3%	13-Jul	776	98.8%
6-Apr	359	97.8%	20-May	1206	98.8%	14-Jul	281	97.2%
7-Apr	665	97.4%	22-May	696	98.0%	20-Jul	280	98.6%
9-Apr	652	98.2%	25-May	961	98.3%	21-Jul	244	100.0%
14-Apr	557	96.8%	26-May	1284	99.1%	28-Jul	57	100.0%
15-Apr	469	97.2%	27-May	609	98.2%	25-Aug	767	99.5%

6.2.2. Satellite Portals

The classification performances achieved per day at the Satellite Portals are shown in Table 6.10 and Figure 6.2. Except for two days, March 4th and March 19th, the classification performance was always at least around 98%. Like the cases with the Standard Portals, on these two days the number of monitored pallets, at 93 and 112 respectively, was well below the average of 410 pallets. Therefore, these performances too could be considered as outliers because of the small sample size. Note that the evaluation period is significantly shorter compared with the Standard Portals because only for this period of time was monitored data available.

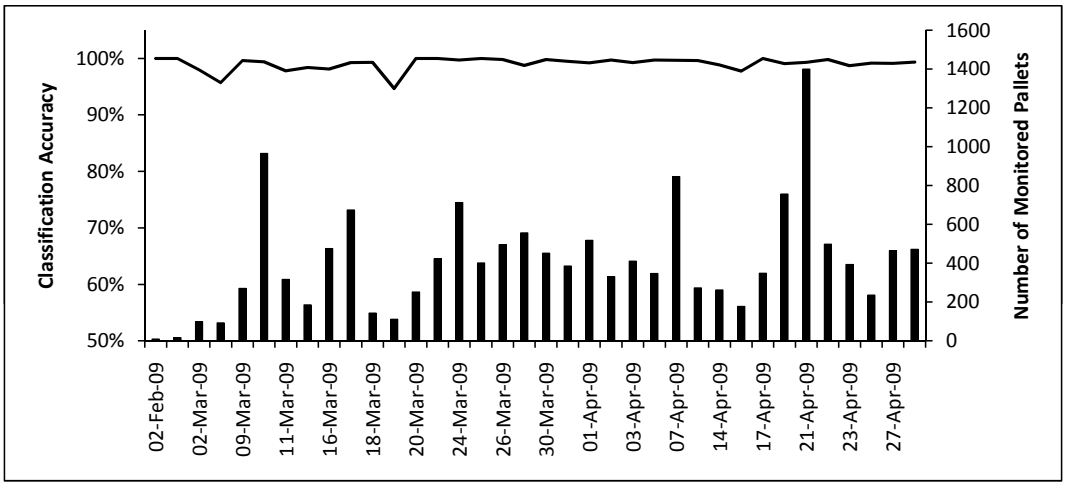


Figure 6.2.: Performance over Time at Satellite Portals

6.2.3. Transition Portals

The classification performances achieved per day at the Transition Portals are shown in Table 6.11 and Figure 6.3. Except for on March 11th, the classification rate was always at least around 97%. However, in contrast to the other two portals, the number of monitored pallets that day was significantly above the average of 512 pallets.

Table 6.10.: Classification Accuracy per Day at the Satellite Portals

Date	Tags	CA	Date	Tags	CA	Date	Tags	CA
2-Feb	10	100.0%	20-Mar	252	100.0%	7-Apr	846	99.6%
10-Feb	16	100.0%	23-Mar	423	100.0%	9-Apr	273	99.6%
2-Mar	99	98.0%	24-Mar	713	99.7%	14-Apr	263	98.9%
4-Mar	93	95.7%	25-Mar	402	100.0%	15-Apr	178	97.8%
9-Mar	270	99.6%	26-Mar	496	99.8%	17-Apr	349	100.0%
10-Mar	965	99.4%	27-Mar	556	98.7%	20-Apr	756	99.1%
11-Mar	317	97.8%	30-Mar	452	99.8%	21-Apr	1400	99.3%
12-Mar	186	98.4%	31-Mar	386	99.5%	22-Apr	498	99.8%
16-Mar	475	98.1%	1-Apr	518	99.2%	23-Apr	394	98.7%
17-Mar	674	99.3%	2-Apr	331	99.7%	24-Apr	236	99.2%
18-Mar	143	99.3%	3-Apr	411	99.3%	27-Apr	465	99.1%
19-Mar	112	94.6%	6-Apr	347	99.7%	28-Apr	472	99.4%

Table 6.11.: Classification Accuracy per Day at the Transition Portals

Date	Tags	CA	Date	Tags	CA	Date	Tags	CA
05-Mar-09	668	98.2%	19-Mar-09	173	97.7%	02-Apr-09	526	99.6%
09-Mar-09	404	96.5%	20-Mar-09	748	99.2%	03-Apr-09	134	97.8%
10-Mar-09	737	96.1%	23-Mar-09	222	98.2%	07-Apr-09	200	100.0%
11-Mar-09	794	97.0%	24-Mar-09	279	97.1%	09-Apr-09	287	99.0%
12-Mar-09	874	98.1%	25-Mar-09	1052	99.8%	14-Apr-09	203	100.0%
13-Mar-09	771	98.6%	26-Mar-09	377	98.7%	15-Apr-09	410	98.8%
16-Mar-09	215	99.5%	27-Mar-09	1010	99.3%	16-Apr-09	365	99.2%
17-Mar-09	1424	98.2%	31-Mar-09	561	99.1%	17-Apr-09	134	98.5%
18-Mar-09	140	100.0%	01-Apr-09	513	99.4%	22-Apr-09	624	99.2%

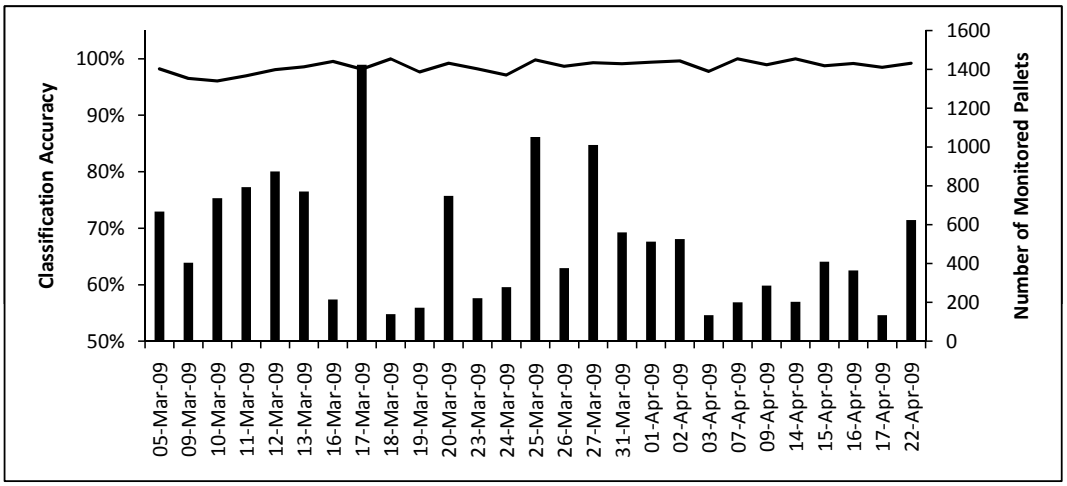


Figure 6.3.: Performance over Time at Transition Portals

6.2.4. Summary

It can be seen that implementing the approaches presented here on the different portal types leads in each case to a reliable and robust classification performance over time. In only a very few cases did the performances show a notable drop of more than 2 or 3 percentage points. However, in almost all of these cases the sample size collected on these days was very small so they can therefore be considered as outliers. Table 6.12 shows the averages and the standard deviations of the daily classification performances achieved at the individual portal types.

Table 6.12.: Classification Performances over Time

Portal Type	**Average CA**	**Std. Dev of CA**
Standard Portals	97.30%	1.30%
Satellite Portals	99.06%	1.14%
Transition Portals	98.62%	1.05%

6.3. Evaluation of Business Objectives

In Section 3.1.2 the business objectives were defined and afterwards mapped against several data mining goals. These will be evaluated in the following.

6.3.1. Knowledge Generation

Because the problem of false-positive tag reads exists in many RFID related processes, METRO representatives requested that the classification model allow them to easily understand why and how it decides the way it does. It was for this reason in particular that decision trees were chosen in favor of other classification models, as because decision trees can easily be transformed into a visual representation it is particularly easy to track which decisions (i.e., attribute tests) lead to the final classification of a tag as being a false-positive read or not. For example, if at the root of the tree a test on the maximum RSSI value of a tag is performed, then it can be concluded that this characteristic plays a major role in the classification procedure.

The following sections discuss some of the major insights derived from the results of this study.

6.3.1.1. Applicability to other Processes

Considering the achieved classification rates that could be reached it soon became apparent that the low-level reader data constitutes a valuable source of information. Because the class of false-positive reads was mapped to the class of pallets that did not move through a portal this means that the approaches presented here are particularly useful for the detection of movement of RFID tags. Identifying moving or non moving tags is naturally an important issue in various processes, especially similar processes such as an incoming goods process or the identification of pallets passing through some check point. For example an RFID portal to register goods moving from the back- to the front of a store. In the following section some other processes are presented where the framework used for this study could be successfully utilized. Note that for example in the pallet retrieval scenario, the pallet that is static relative to the RFID reader installed on the forklift is the one of interest.

Electronic Article Surveillance In this scenario a shoplifter carrying stolen RFID tagged products is leaving a store through a so-called EAS Gate. The task here is to distinguish between the stolen objects moving through the gate and several "static" tags - for example those present in the window display near the store exit.

Automatic Supermarket Check-Out In this scenario customers move through an RFID checkout portal so that all RFID tagged products are automatically detected and billed to them. These products need to be distinguished from products near this portal, for example items dropped by preceding customers or present in the window display.

Pallet Retrieval A common process in distribution centers is the retrieval of pallets from high-rack storage areas. In this context an RFID reader along with the corresponding antennas is usually attached directly to the fork lift. In contrast to the other applications presented above, in this case only the retrieved pallet is "static" and all others appear to move while the warehouseman navigates the forklift through the distribution center.

Replenishment: Backroom to Shelf In this scenario it is necessary to detect the items leaving the backroom and distinguish them both from the items left in there and the items already present in the sales area.

Pick-by-Voice RFID technology can also improve the picking process by automatically detecting the selected articles. In this scenario it is important to automatically detect the retrieved items in order to prevent incorrect removals.

Because of the similarities between the above scenarios and the scenario in our METRO group study, it seems reasonable to assume that the approaches presented here can also be applied to these scenarios as well.

6.3.1.2. Use of Alternative Antenna Configurations

There were three different portal types in use in the distribution center. The Satellite Portals and Transition Portals both used an additional reader with four more antennas. Because both advancements lead to improved classification results it is reasonable to assume that the number and direction of installed antennas are a useful enhancement in the scenario. This insight might also be helpful in some of the alternative scenarios presented above.

6.3.1.3. Examination of Low-Level Reader Data

Most of the approaches presented to detect false-positives in Chapter 2 make use only of the number of tag-events read during a predefined time window and entirely ignore the signal strength. However, it was found that the received signal strength indication (RSSI) is the most important movement indicator. It can be observed for example in Figure 5.4 how well the maximum RSSI value measured during a gathering-cycle can be used to distinguish between moved and static tags.

6.3.2. Avoidance of additional Costs

The underlying idea of all the approaches here is to make use only of the data generated by the reader anyway. Consequently, no additional investments in physical hardware are required if one of these classification approaches is adopted at an existing or planned RFID installation. However, if monetary savings are not the number one priority then it has been shown that additional hardware - in terms of an additional reader - can improve the classification performance even further. However, in addition to any hardware costs it is also necessary to spend some time implementing and parameterizing whichever approach is chosen.

6.4. Deployment

The first experiments at METRO Group using the low-level reader data to detect false-positive RFID tag reads go back to mid-2008, i.e. one year prior to the data collection considered in this thesis. The first ever, though very primitive, classification model in productive use was implemented in the distribution center in Unna, Germany, also in the middle of 2008. Around November 2008, it was succeeded by the first more advanced algorithm that involved artificial attribute generation and decision trees. Since then, the quality of the models has been constantly improved by using ever more advanced techniques such as the time-series analysis approach.

The classification model currently in use is called $TK-8B$ (i.e., the 8th algorithm, revision B) and has been in productive use since the middle of 2009 deciding the operative classification of pallets leaving the central distribution center of the METRO Group in Unna, Germany, all without undergoing any changes. It has been generated completely based on the attributes and techniques presented in this thesis and is a result of the underlying framework.

Although it was shown that using Satellite- or Transition Portal gives the best classification results it has been decided to use only Standard Portals in the future. Two major reasons were

given for this decision. First of all, Satellite- and Transition Portals require the installation of two readers instead of one single reader, and additionally a total of eight instead of only four antennas. Because readers and antennas make up the largest part of the total cost of installing the technology, the price of these two types of portals is almost twice as high as that of the Standard Portals. Furthermore, more hardware means more ongoing maintenance requirements, which in turn means even more costs. The second reason was that the classification rates achieved with the Standard Portals already significantly exceeded the overall target and thus the additional improvement did not justify the additional costs.

As with most IT installations, whether they be on a hardware or a software basis, it still requires the acceptance of the people working with it. If this support is missing, then any solution, no matter how good it is, is going to fail. The people working with the solution on an operative basis are the warehousemen who actually load the pallets into the containers. If they had felt that the RFID solution had significant draw backs compared with the previously used bar-code solution, they would have refused to give their cooperation. However, this was not the case and the warehousemen actively participated in the implementation of the RFID technology and also in our study, even during the initial evaluation phase.

7. Summary and Outlook

7.1. Summary

7.1.1. Background

Currently, radio frequency identification (RFID) technology has been implemented in a variety of different applications and in particular in the field of supply chain management and logistics. The ability to identify hundreds of RFID tagged objects at the same time and without the need for a line-of-sight is only one of the many advantages over the traditional bar-code. However, along with the enhanced functionality come new and previously unknown difficulties.

The ability to detect multiple RFID tags within the range of the antennas is generally considered desirable, but in many cases not all of the scanned transponders are really of interest. For example, when trying to automatically register incoming or outgoing RFID tagged pallets in a distribution center, not only are these detected but also additional pallets that are in range only by accident. The latter are denoted as *false-positive RFID tag reads* and they can be the result of two different circumstances. In the first case, the range of the radio waves used to scan for tags is unexpectedly extended, for example due to reflections on metal surfaces, and any pallets are detected that were actually assumed to be out of the antennas' coverage. In the second case, the pallets are indeed within the range of the antennas but may be considered as *unwanted reads*, for example, because they were buffered near the dock door by a warehouseman. In either case the false-positive tag reads need to be detected and distinguished from the pallets that were really coming in or going out.

Few approaches were presented in the literature to deal with this problem and of those that were none of them were well founded. Either the approaches made unrealistic assumptions because they made no reference to a real world scenario, or there was no meaningful data available to validate any their propositions. In most cases, the idea was to simply consider the number of tag detections during a predefined time window to distinguish between tags of interest and false-positives. However, in conjunction with any individual tag detection there is also additional data generated by the reader, this is known as low-level reader data. Besides a

timestamp and the identifier of the receiving antenna, the received signal strength indication (RSSI) as a measure of "how loud the tag was read" is also available. Based on this insight, it was the aim of this thesis to answer the question of how this low-level reader data can be used to detect false-positive RFID tag reads.

7.1.2. Business Objectives

The research methodology, as well as the structure of this thesis, was mainly organized according to the well known data mining process model known as *CRISP-DM*. At first the unit of analysis, i.e., the RFID enabled outgoing goods process in the METRO Group central distribution center in Unna, Germany was described in detail. The unit of analysis, i.e., the RFID enabled outgoing goods process in the METRO Group central distribution center in Unna, Germany was first described in detail before the *business objectives* were identified in order to more exactly define the problem and the characteristics of the requested solution. The occurrence of false-positive tag reads at an outgoing goods portal in a distribution center can possibly lead to incorrectly billed pallets because they have been erroneously assumed to be loaded onto a container heading to a customer. Consequently, the *minimization of false-positive reads* was defined as the overall target, with the wider aim being to minimize the number of incorrectly billed pallets. However, the problem described here is not limited to the scenario under consideration, it can also emerge in various similar processes along a retailer's supply chain. Accordingly, it was required that the solution should lead to the *generation of fundamental insights and knowledge* that would allow potential application to other processes suffering from similar difficulties.

In data mining or machine learning terminology, the solution to be created is called a classification model because its task is to classify pallets as either *moved* if they were loaded or *static* if they were false-positive reads. Therefore, the business objectives have been mapped to measurable data mining goals which will ultimately be used to evaluate the quality and the success or failure of the developed classification model. Consequently, minimization of false-positive reads was mapped to a so-called *classification accuracy* of at least 99%. This means that the model should not classify more than 1 per 100 pallets incorrectly. It should be noted though that the misclassification of a pallet does not automatically lead to an incorrectly billed pallet; this is because the responsible warehouseman is given immediate visual feedback about the detected pallets and is therefore able to immediately respond to any contradictions.

The business objective to generate fundamental insights and knowledge was interpreted in the following way: classification models can be separated into two disjunctive classes, some using black-box models and some others using white-box models. The difference between these

is that a black-box model is normally very difficult if not impossible for a human being to understand. In contrast, white-box models are usually very transparent and allow a manual reconstruction of the decisions that lead to the final classification of a pallet. Based on this requirement and due to other favorable characteristics, decision trees were identified as the classification model of choice.

7.1.3. Data Basis

The quality of the classification model heavily depends upon two different factors: firstly, it must be a good representative of the real-world environment; and secondly, it must be possible to extract meaningful characteristics typical to both moved and static pallets.

The data used as the basis for the classification model was collected over a period of several months in the central distribution center of the METRO Group in Unna, Germany. This kind of real-world data allows for greater insights than any other dataset possibly acquired under lab conditions or created by computer simulations as it was presented in the literature by other researchers. 2,840,571 individual RFID tag detections were monitored by students, representing 92,857 pallets in total. The students furthermore labeled the individual pallet data as belonging to either a moved or a static pallet, depending on whether it was being loaded or not in the moment it was observed.

Three different RFID portal types were used in the distribution center to register outgoing pallets. *Standard Portals* have two antennas to the left and to the right aligned towards each other so a pallet passing trough is scanned from both sides at the same time. *Satellite Portals* and *Transition Portals* are more advanced versions as they each have an additional reader with four more antennas and an alternative antenna configuration. The data was mainly collected at the Standard Portals because they are the most commonly used. However, the data collected at the other two portal types was also sufficient enough to be used as a training basis.

The collected data at the portals was mainly of some numerical type but additional nominal characteristics were derived as well. An example of numerical data is the received signal strength indication, measured in dBm, that also constitutes the most important information source. Examples of nominal characteristics include information about the order in which tags were detected or about which RFID reader antennas detected the tag. Because of the inexhaustible possibilities available in attempting to extract meaningful characteristics describing both moved and static pallets, a major part of this thesis was spent on identifying and evaluating these so called *attributes*.

7.1.4. Classification Model Building

An initial evaluation of the available data led to the conclusion that two different model building approaches are reasonable. Although they significantly differ in the level of data granularity used and the way they describe moved and static pallets, it is however possible to treat them both simply as a preprocessing step before they are ultimately used as input into the decision tree learner. The general idea behind this procedure was to create a common ground such that an additional third approach, able to make use of both the approaches, could be generated.

The first approach presented in this thesis was called the *tag-occurrence level approach*. Based on the data collected during a 10 second time window around the loading of a pallet, meaningful attributes were derived by calculating various aggregations of the low-level reader data. For example, a pallet can be described by the maximum, minimum, or average RSSI value observed during that period, the number of answers it gave to the reader or the time since the beginning of the loading that has passed before it was first scanned. These attributes are then used as input to a decision tree to generate rules similar to

> "If the maximum signal strength of a tag is less than X and it has been detected less than Y times then it is a false-positive else it is a moved pallet."

The second approach was called the *tag-event level approach*. Based on the same data collected during the 10-second time window around the loading of a pallet the development of the signal strength during that period was evaluated. The insight that the received signal strength varies depending on the distance between RFID tag and antennas was used to discriminate between moved and static pallets. The idea next was to derive *time-series* representing the typical development of RSSI values for moved and static pallets over time. These could then be used to generate rules similar to

> "If the RSSI values time-series of a pallet is more similar to the typical time-series of a static pallet then it is a false-positive else it is a moved pallet."

The resulting rules and attributes derived using the two approaches were then unified in a third approach. Finally, a fourth approach called the *exclusive approach* was proposed, although it cannot be used in the scenario of an RFID enabled outgoing goods process because it assumes that there is always exactly one single RFID tag moving through the portal, no more and no less. However, it is highly conceivable that this approach will perform even better in other processes where that assumption does hold true.

7.1.5. Evaluation

In the evaluation chapter it was shown that the classification models significantly exceeded any initial expectations. It can be seen that the idea of using low-level reader information for the detection of false-positive RFID tag reads was a good one. *Tag-occurrence level* and *tag-event level* approaches were found to be of comparable quality with their respective strengths and weaknesses because the unification of both approaches led to even better results.

Implementation of the classification models at the different portal types showed that *Satellite Portals* and *Transition Portals* outperformed the detection rates achieved at the *Standard Portals*. The number of pallets that could possibly be incorrectly billed now only ranges from 1 in 900 pallets up to only 1 in 4500 pallets.

Although the advanced portal types yielded the best results, METRO Group made a decision in favor of the *Standard Portals*, especially since these still exceeded the required detection rates to the point where any further improvement in quality did not justify doubling the investments by installing twice as many readers and antennas.

7.2. Outlook

Both practitioners and researchers in the field of RFID were identified as the intended audience for this piece of work. In the conclusion of this thesis possible implications for both groups are discussed.

7.2.1. Implications for the Researcher

From a researcher's perspective it will be interesting to see how far the detection rates can be pushed. There are currently a great number of machine learning techniques available (e.g., Neural Networks, Support Vector Machines) that have been proven to lead to better classifications than decision trees can make, and in many different scenarios too. These could be applied to the available data and the results then compared to those presented in this thesis.

Every classification model relies on the quality of the attributes to describe the objects under consideration. Although a large number of attributes were identified and evaluated in this thesis it is worth spending time and effort to explore even more attributes that might help to distinguish between moved and static tags even more effectively.

Furthermore, there is a lot of research suggestive of the value of improving the time-series analysis approach. The application of cluster analysis to identify different sub groups of moved and static tags was a first step, but it is the opinion of the author that it is very likely that some kind of sub-sequence matching will lead to best results in this context.

A drawback of classification models is that they require some kind of class labeled training data. Because a large example data set was available for our framework this was not a problem. However, it seems evident that the insights of the thesis could also be used for the generation of a self-learning and / or self-adapting classification algorithm. For example, such an algorithm would be able to automatically recognize process changes like new RFID tags with improved readability and to adapt itself to them. However, the research issues associated with these concepts are beyond the scope of this thesis.

7.2.2. Implications for the Practitioner

From a practitioner's perspective it will be interesting to evaluate the application of the framework to other processes. As it was stated in the beginning, the problem of false-positive tag reads exists in many different domains involving radio frequency identification technology. It is expected that the framework presented in this thesis can be successfully applied to these domains and in particular to any processes where it is crucial to detect the movement of RFID tags. Several examples of such processes were briefly described in the thesis.

Currently, it is being evaluated to what extent the framework and the corresponding knowledge can be used in the scenario of pallet retrieval from high rack storage areas. Although the scenario is a little different, initial tests were promising. In this case, an RFID reader is installed on the fork lift and it is the pallet removed from the storage area that needs to be identified. As soon as the forklift moves, the retrieved pallet becomes static relative to the reader and any other pallets appear to move relative to the reader. Consequently, in this case the moving pallets are considered to be false-positives.

The next step is to evaluate the application of the framework in an electronic article surveillance process. A research scenario under laboratory conditions has already been set up and initial tests in this direction are going to be performed in the medium-term.

A. Glossary

Antennas, DC A group of antennas directed towards the distribution center. Installed at the Satellite- and the Transition Portals.

Antennas, Main A group of antennas used to detect anything passing through a portal. Installed at the Standard- and the Satellite Portals.

Antennas, Truck A group of antennas directed towards the truck. Installed at the Satellite- and Transition Portals.

Attribute A characteristic to describe an object (e.g., height, depth, weight).

Attribute, Antenna One of the Domain Attribute types. A characteristic that describes a pallet on the basis of the respective reader antennas where the individual tag detections took place. Example: Number of detections by antenna #1.

Attribute, Artificial Attributes derived from the Domain Attributes by applying a sequence of unary and binary mathematical operators to them. In contrast to their latter they do not have an intuitive semantic.

Attribute, Domain A group of attributes (i.e., RSSI-, SinceStart and Antenna Attributes) derived from the domain knowledge of people working in the scenario under consideration that have an intuitive semantic. Example: Maximum signal strength a tag has been detected with.

Attribute, Logical A characteristic to describe a tag read by a portal with more than one reader. Example: where a tag was read by reader #1 before it was read by reader #2.

Attribute, RSSI One of the Domain Attribute types. A characteristic used to describe a pallet on the basis of the signal strength the corresponding tag was read with.

Attribute, SinceStart One of the Domain Attribute types. A characteristic that describes a pallet on the basis of the timestamps of the individual tag detections. Example: Time that has passed before the first tag detection.

Attribute, Time-Series A characteristic that describes a tag on the basis of the development of RSSI values over the period of a gathering-cycle. Example: Degree of similarity to a typical moved signal strength development.

Attribute, TO-Count An attribute indicating the number of times a tag has been seen in previous gathering-cycles. If the attribute value is greater than 1 then it is known that the tag has been seen before.

C4.5 A decision tree learning algorithm.

CART Another decision tree learning algorithm.

Class Precision A measure of how confident a classification is, i.e., the ratio of all samples classified as a specific class that were correctly classified.

Class Recall A measure used to determine the ratio of tag of a specific class that were classified correctly. It is often also called *Class Detection Rate*.

Classification Accuracy A measure often simply called *Accuracy*, that corresponds to the ratio of correctly classified samples in a data set.

Completeness A measure of generality, i.e., how many of the samples of a specific class are covered by that rule and are classified correctly.

Confidence A measure of how confident a rule is, i.e., how many of the samples covered by that rule are classified correctly.

CRISP-DM The Cross Industry Standard Process for Data Mining is commonly used to structure a data mining project.

Decision Tree A machine learning technique to classify objects based on their describing attributes.

Electronic Product Code (EPC) A coding scheme to uniquely identify products.

False-Negative RFID Tag Read An RFID tag that has not been read by an RFID reader device although it was present within the range of the reader antennas.

False-Positive RFID Tag Read An RFID tag that has been read unexpectedly or undesirably by an RFID reader device.

Gathering-Cycle Denomination of the time period of a pallet loading where the low-level reader data is collected. A gathering-cycle is usually less than 10 seconds long.

Knowledge Discovery The process of deriving knowledge from data.

Neighbor, Furthest The object in a data set that is least similar to a query object.

Neighbor, Nearest The object in a data set that is most similar to a query object.

Pallet, Mob-Ware A shipping unit consisting of furniture.

Pallet, Stacked At least two low-height pallets that are stacked on top of each other. Note that each of the pallets still has its own RFID tag attached.

Portal, Satellite An RFID portal similar to the Standard Portal, but with an additional reader. In addition to the 4 Main Antennas, 2 more are directed towards the distribution center (DC Antennas) and 2 more are directed towards the truck (Truck Antennas).

Portal, Standard An RFID portal consisting of 4 antennas, called Main Antennas, intended to detect any tag passing through.

Portal, Transition An RFID portal consisting of 2 different readers. The first reader has 4 antennas directed towards the distribution center (DC Antennas) and the latter has 4 antennas directed towards the truck (Truck Antennas).

Received Signal Strength Indication (RSSI) This denotes the power of a tag's radio signal measured in dBm: this can intuitively be interpreted as how "loud" the tag was heard by the antennas.

SinceStart A measurement of how many microseconds passed since the time the pallet loading at that portal started and the time the tag was read.

Support A measure of generality, i.e., how many of the total number of samples are covered by a rule and are classified correctly.

Tag-Event A single tag detection during a gathering-cycle mainly represented by the measured signal strength, a timestamp and the identifier of the involved antenna.

Tag-Occurrence The entirety of all the tag-events associated with a specific EPC during a single gathering-cycle.

B. Monitored Data per Portal

Table B.1.: Monitored Pallets at the Satellite Portals

Portal	Moved Tags		Static Tags		Total Tags	
	Quantity	[%]	Quantity	[%]	Quantity	[%] of all
Portal 23	464	11.4%	3,605	88.6%	4,069	27.5%
Portal 24	519	20.5%	2,009	79.5%	2,528	17.1%
Portal 25	509	10.4%	4,366	89.6%	4,875	33.0%
Portal 26	479	14.5%	2,826	85.5%	3,305	22.4%
Total	1,971	13.3%	12,806	86.7%	14,777	100.0%

Table B.2.: Monitored Pallets at the Transition Portals

Portal	Moved Tags		Static Tags		Total Tags	
	Quantity	[%]	Quantity	[%]	Quantity	[%] of all
Portal 1	236	12.9%	1,591	87.1%	1,827	13.2%
Portal 2	173	16.6%	869	83.4%	1,042	7.5%
Portal 3	202	10.3%	1,762	89.7%	1,964	14.2%
Portal 4	219	9.7%	2,037	90.3%	2,256	16.3%
Portal 5	93	7.2%	1,199	92.8%	1,292	9.3%
Portal 6	131	7.4%	1,635	92.6%	1,766	12.8%
Portal 7	141	8.0%	1,611	92.0%	1,752	12.7%
Portal 8	163	8.4%	1,783	91.6%	1,946	14.1%
Total	1,358	9.8%	12,487	90.2%	13,845	100.0%

Table B.3.: Monitored Pallets at the Standard Portals (Part 1)

Portal	Moved Tags		Static Tags		Total Tags	
	Quantity	[%]	Quantity	[%]	Quantity	[%] of all
Portal 9	226	46.4%	261	53.6%	487	0.9%
Portal 10	266	26.4%	740	73.6%	1,006	1.9%
Portal 11	195	39.2%	303	60.8%	498	0.9%
Portal 12	319	27.7%	831	72.3%	1,150	2.1%
Portal 13	306	26.0%	870	74.0%	1,176	2.2%
Portal 14	314	36.5%	546	63.5%	860	1.6%
Portal 15	313	28.0%	803	72.0%	1,116	2.1%
Portal 16	418	21.6%	1,521	78.4%	1,939	3.6%
Portal 17	509	44.8%	628	55.2%	1,137	2.1%
Portal 18	500	34.1%	968	65.9%	1,468	2.7%
Portal 19	549	21.3%	2,026	78.7%	2,575	4.8%
Portal 20	662	20.3%	2,592	79.7%	3,254	6.0%
Portal 21	696	13.5%	4,458	86.5%	5,154	9.5%
Portal 23	964	19.5%	3,984	80.5%	4,948	9.2%
Portal 24	972	22.8%	3,283	77.2%	4,255	7.9%
Portal 25	932	31.1%	2,065	68.9%	2,997	5.6%
Portal 26	1,002	25.0%	3,008	75.0%	4,010	7.4%
Portal 27	776	32.5%	1,613	67.5%	2,389	4.4%
Total (Part 1)	9,919	24.5%	30,500	75.5%	40,419	74.9%

Table B.4.: Monitored Pallets at the Standard Portals (Part 2)

Portal	Moved Tags		Static Tags		Total Tags	
	Quantity	[%]	Quantity	[%]	Quantity	[%] of all
Portal 28	734	21.9%	2,625	78.1%	3,359	6.2%
Portal 29	485	29.6%	1,151	70.4%	1,636	3.0%
Portal 30	604	27.7%	1,574	72.3%	2,178	4.0%
Portal 31	428	21.0%	1,607	79.0%	2,035	3.8%
Portal 32	315	26.1%	892	73.9%	1,207	2.2%
Portal 33	359	21.3%	1,324	78.7%	1,683	3.1%
Portal 34	160	20.0%	640	80.0%	800	1.5%
Portal 35	52	19.4%	216	80.6%	268	0.5%
Portal 58	20	48.8%	21	51.2%	41	0.1%
Portal 59	16	42.1%	22	57.9%	38	0.1%
Portal 61	19	100.0%	0	0.0%	19	0.0%
Portal 64	4	66.7%	2	33.3%	6	0.0%
Portal 65	12	85.7%	2	14.3%	14	0.0%
Portal 66	16	32.7%	33	67.3%	49	0.1%
Portal 71	28	47.5%	31	52.5%	59	0.1%
Portal 72	39	33.3%	78	66.7%	117	0.2%
Portal 75	19	73.1%	7	26.9%	26	0.0%
Portal 76	16	47.1%	18	52.9%	34	0.1%
Total (Part 1)	9,919	24.5%	30,500	75.5%	40,419	74.9%
Total (Part 2)	3,326	24.5%	10,243	75.5%	13,569	25.1%
Total	13,245	24.5%	40,743	75.5%	53,988	100.0%

C. Time Series Attribute Values

Table C.1.: Detailed Time-Series Attribute Values (`STD_COMPLETE` Data Set)

Attribute	Moved Tags				Static Tags			
	Min	Max	Avg	SD	Min	Max	Avg	SD
D_M	1.1	6.7	3.5	1.2	1.1	11.2	5.2	2.4
$D_{M,1}$	0.9	7.3	3.5	1.3	0.9	9.9	4.3	2.1
$D_{M,2}$	1.1	5.7	2.8	0.9	1.4	11.7	5.5	2.5
$D_{M,3}$	1.2	8.7	4.5	1.6	0.8	8.5	3.4	1.8
$D_{M,4}$	1.4	6.6	3.7	1.1	1.6	12.4	6.3	2.6
$D_{M,5}$	1.1	7.2	3.7	1.2	2.3	14.8	8.3	2.8
D_S	1.5	12.3	6.7	2.2	0.3	6.5	2.0	1.3
$D_{S,1}$	1.3	10.9	5.7	2.0	0.3	7.4	2.7	1.5
$D_{S,2}$	2.2	14.7	8.2	2.5	0.2	7.5	2.1	1.6
$D_{M,Min}$	0.7	4.7	2.6	0.9	0.7	8.5	3.3	1.8
$D_{M,Max}$	2.5	9.2	5.0	1.4	2.6	14.8	8.3	2.8
$D_{M,Mean}$	1.6	6.4	3.6	1.0	1.6	11.4	5.5	2.3
$D_{M,StDev}$	0.2	1.8	0.8	0.4	0.4	2.4	1.6	0.4
$D_{M,CoV}$	0.068	0.547	0.231	0.099	0.151	0.582	0.327	0.101
$D_{S,Min}$	0.9	10.9	5.7	2.1	0.2	4.9	1.3	1.0
$D_{S,Max}$	2.7	14.7	8.2	2.5	1.5	8.2	3.4	1.5
$D_{S,Mean}$	1.8	12.6	6.8	2.3	1.1	6.5	2.3	1.2
$D_{S,StDev}$	0.3	1.8	1.0	0.3	0.2	1.7	0.9	0.4
$D_{S,CoV}$	0.074	0.488	0.164	0.071	0.085	0.754	0.420	0.175

Table C.2.: Detailed Time-Series Attribute Values (`SAT_MAIN_ONLY` Data Set)

Attribute	Moved Tags				Static Tags			
	Min	Max	Avg	SD	Min	Max	Avg	SD
D_M	1.2	7.3	3.9	1.3	1.4	10.0	5.3	2.2
$D_{M,1}$	1.8	6.8	4.0	1.1	2.0	11.3	6.5	2.4
$D_{M,2}$	1.3	6.3	3.3	1.0	2.7	13.7	8.3	2.7
$D_{M,3}$	0.9	8.9	4.4	1.7	0.8	8.2	4.0	1.8
$D_{M,4}$	1.2	9.2	4.6	1.8	0.8	7.0	3.2	1.5
$D_{M,5}$	1.3	5.5	2.9	0.9	1.6	10.9	5.8	2.4
D_S	1.8	13.0	6.9	2.5	0.3	5.0	1.7	1.0
$D_{S,1}$	1.5	12.3	6.3	2.3	0.3	5.6	2.1	1.2
$D_{S,2}$	2.5	14.6	8.1	2.6	0.2	5.9	1.7	1.2
$D_{M,Min}$	0.8	4.5	2.6	0.8	0.7	7.0	3.1	1.5
$D_{M,Max}$	2.7	9.6	5.2	1.5	2.9	13.7	8.3	2.7
$D_{M,Mean}$	2.0	6.9	3.9	1.1	1.8	10.2	5.5	2.1
$D_{M,StDev}$	0.2	2.3	0.9	0.5	0.5	2.5	1.7	0.5
$D_{M,CoV}$	0.070	0.520	0.239	0.101	0.182	0.553	0.329	0.084
$D_{S,Min}$	1.5	12.3	6.3	2.4	0.2	3.8	1.1	0.8
$D_{S,Max}$	2.5	14.6	8.1	2.6	1.1	6.2	2.7	1.1
$D_{S,Mean}$	1.9	13.2	7.1	2.5	0.8	5.0	1.9	0.9
$D_{S,StDev}$	0.3	1.3	0.8	0.2	0.1	1.3	0.7	0.3
$D_{S,CoV}$	0.050	0.298	0.116	0.049	0.061	0.775	0.385	0.178

Table C.3.: Detailed Time-Series Attribute Values (SAT_MAIN_TRUCK Data Set)

Attribute	Moved Tags				Static Tags			
	Min	Max	Avg	SD	Min	Max	Avg	SD
$DIST_{S,Main}$	1.5	12.9	6.9	2.4	0.2	7.1	2.1	1.5
$DIST_{S,Truck}$	0.9	9.4	4.5	2.0	0.5	7.6	2.8	1.8
$DIST_{M,Main}$	1.2	6.6	3.7	1.2	1.3	11.0	5.8	2.4
$DIST_{M,Truck}$	0.8	7.2	3.0	1.3	0.6	9.2	3.2	2.2
$D_{M,Min}$	0.7	5.1	2.6	0.9	0.6	8.1	2.9	1.9
$D_{M,Max}$	1.8	7.8	4.0	1.2	1.6	11.0	6.1	2.3
$D_{M,Mean}$	1.5	6.2	3.3	0.9	1.3	8.8	4.5	1.8
$D_{M,StDev}$	0.0	2.5	0.7	0.6	0.0	4.4	1.6	1.1
$D_{M,CoV}$	0.005	0.653	0.219	0.157	0.006	0.860	0.375	0.246
$D_{S,Min}$	0.7	8.6	4.2	1.8	0.2	4.6	1.6	1.0
$D_{S,Max}$	2.6	12.9	7.2	2.2	0.8	8.3	3.3	1.7
$D_{S,Mean}$	2.1	9.7	5.7	1.7	0.6	6.4	2.4	1.2
$D_{S,StDev}$	0.0	5.0	1.5	1.1	0.0	3.1	0.8	0.7
$D_{S,CoV}$	0.008	0.790	0.273	0.187	0.006	0.863	0.340	0.226

Table C.4.: Detailed Time-Series Attribute Values (TRA_BOTH Data Set)

Attribute	Moved Tags				Static Tags			
	Min	Max	Avg	SD	Min	Max	Avg	SD
$D_{S,DC}$	1.0	11.3	6.4	2.5	0.4	8.5	2.8	1.8
$D_{S,Truck}$	0.8	9.2	4.9	1.9	0.4	8.4	3.3	1.9
$D_{M,DC}$	1.1	7.1	3.8	1.3	1.5	11.7	6.3	2.6
$D_{M,Truck}$	1.0	7.2	3.4	1.2	0.7	11.5	4.1	2.8
$D_{M,Min}$	0.8	5.3	3.0	1.0	0.7	9.8	3.4	2.0
$D_{M,Max}$	1.7	8.5	4.2	1.2	2.1	12.0	7.0	2.5
$D_{M,Mean}$	1.5	6.3	3.6	1.0	1.8	10.3	5.2	1.9
$D_{M,StDev}$	0.0	2.5	0.6	0.5	0.0	4.7	1.8	1.2
$D_{M,CoV}$	0.004	0.651	0.173	0.143	0.007	0.829	0.361	0.224
$D_{S,Min}$	0.6	8.7	4.6	1.9	0.4	6.2	2.1	1.2
$D_{S,Max}$	2.1	11.3	6.7	2.3	1.1	9.0	4.0	1.9
$D_{S,Mean}$	1.6	9.5	5.7	1.9	0.9	7.2	3.1	1.4
$D_{S,StDev}$	0.0	3.6	1.1	0.8	0.0	3.3	1.0	0.8
$D_{S,CoV}$	0.003	0.755	0.206	0.164	0.007	0.832	0.326	0.215

Bibliography

[AL05] Glen Allmendinger and Ralph Lombreglia, *Four Strategies for the Age of Smart Services*, Harvard Business Review **83** (2005), no. 10, 131–145.

[AM05] Zaheeruddin Asif and Munir Mandviwalla, *Integrating the Supply Chain with RFID: A Technical and Business Analysis*, Communications of the Association for Information Systems **15** (2005), no. 12, 393–427.

[Ang05] Rebecca Angeles, *RFID Technologies: Supply-Chain Applications and Implementation Issues*, Information Systems Management **22** (2005), no. 1, 51–65.

[Ash09] Kevin Ashton, *That "Internet of Things" Thing*, RFID Journal **6** (2009), no. 3, 4.

[Bel61] Richard E. Bellman, *Adaptive Control Processes - A guided Tour*, Princeton University Press, Princeton, NJ, USA, 1961.

[BFHF03] James Brusey, Christian Floerkemeier, Mark Harrison, and Martyn Fletcher, *Reasoning about Uncertainty in Address Identification with RFID*, Workshop on Reasoning with Uncertainty in Robotics at International Joint Conferences on Artificial Intelligence (IJCAC) (Acapulco, Mexico), August 2003, pp. 23–30.

[BFSO84] Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard Olshen, *Classification and Regression Trees*, Wadsworth Mathematics Series, Wadsworth International Group, Belmont, CA, 1984.

[BP05] Indranil Bose and Raktim Pal, *Auto-ID: Managing Anything, Anywhere, Anytime in the Supply Chain*, Communications of the ACM **48** (2005), no. 8, 100–106.

[Bra02] Max Bramer, *Using J-Pruning to reduce Overfitting in Classification Trees*, Knowledge Based Systems **15** (2002), no. 5, 301–308.

[Bra07a] ———, *Principles of Data Mining*, Springer, London, UK, 2007.

[Bra07b] ———, *Principles of Data Mining*, ch. Decision Tree Induction: Using Entropy for Attribute Selection, pp. 51–64, Springer, London, UK, 2007.

[Bra07c] ———, *Principles of Data Mining*, ch. More about Entropy, pp. 135–154, Springer, London, UK, 2007.

[Bro07] Dennis E. Brown, *RFID Implementation*, ch. Bar Codes and RFID Tags, pp. 115–132, McGraw-Hill, New York, NY, USA, 2007.

[BS09] Henning Baars and Xuanpu Sun, *Multidimensional Analysis of RFID Data in Logistics*, Proceedings of the 42nd Hawaii International Conference on System Science (HICSS) (Waikoloa, Big Island, HI, USA), January 2009, pp. 1–10.

[Bur98] Christopher J.C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery **2** (1998), 121–167.

[BWL06] Yijian Bai, Fusheng Wang, and Peiya Liu, *Efficiently filtering RFID Data Streams*, Proceedings of the 1st International Very Large Data Base Workshop on Clean Databases (CleanDB) (Seoul, South Korea), September 2006, pp. 50–57.

[CCK+00] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth, *CRISP-DM 1.0: Step-By-Step Data Mining Guide*, Tech. report, The CRISP-DM consortium, 2000.

[CFZ09] Bertrand Clarke, Ernest Fokoue, and Hao Helen Zhang, *Principles and Theory for Data Mining and Machine Learning*, Springer Series in Statistics, Springer, New York, NY, USA, 2009.

[CKRS04] Sudarshan S. Chawathe, Venkat Krishnamurthy, Sridhar Ramachandran, and Sanjay Sarma, *Managing RFID Data*, Proceedings of the 30th International Conference on Very Large Data Bases (VLDB) (Toronto, Canada), 2004, pp. 1189–1195.

[Coh95] Wiliam W. Cohen, *Fast Effective Rule Induction*, Proceedings of the 12th International Conference on Machine Learning (ICML) (Tahoe City, CA, USA), July 1995, pp. 115–123.

[CPB+05] David D. Clark, Craig Partridge, Robert T. Braden, Bruce Davie, Sally Floyd, Van Jacobson, Dina Katabi, Greg Minshall, K.K. Ramakrishnan, Timothy Roscoe, Ion Stoica, John Wroclawski, and Lixia Zhang, *Making the World (of Communications)*

a different Place, ACM SIGCOMM Computer Communication Review **35** (2005), no. 3, 91–96.

[CPSK07] Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski, and Lukasz A. Kurgan, *Data Mining: A Knowledge Discovery Approach*, ch. The Knowledge Discovery Process, pp. 9–24, Springer, New York, NY, USA, 2007.

[DB79] David L. Davies and Donald W. Bouldin, *A Cluster Separation Measure*, IEEE Transactions on Pattern Analysis and Machine Intelligence **1** (1979), no. 2, 224–227.

[DHS07] Dursun Delen, Bill C. Hardgrave, and Ramesh Sharda, *RFID for Better Supply-Chain Management through Enhanced Information Visibility*, Production and Operations Management **16** (2007), no. 5, 613–624.

[DW05] Daniel M. Dobkin and Steven M. Weigand, *Environmental Effects on RFID Tag Antennas*, Proceedings of the MTT-S International Microwave Symposium Digest (Long Beach, CA, USA), June 2005, pp. 135–138.

[ER08] Waiyawuth Euachongprasit and Chotirat Ann Ratanamahatana, *Efficient Multimedia Time Series Data Retrieval Under Uniform Scaling and Normalisation*, Lecture Notes in Computer Science **4956** (2008), 506–513.

[Fin03] Klaus Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards Identification*, John Wiles & Sons Inc., Hoboken, NJ, USA, 2003.

[FKL+05] Ada Wai-Chee Fu, Eamonn Keogh, Leo Yung Hang Lau, Chotirat Ann Ratanamahatana, and Raymond Chi-Wing Wong, *Scaling and Time Warping in Time Series Querying*, Proceedings of the 31st International Conference on Very Large Data Bases (VLDB) (Trondheim, Norway), August 2005, pp. 649–660.

[FL04] Christian Floerkemeier and Matthias Lampe, *Issues with RFID Usage in Ubiquitous Computing Applications*, Proceedings of the 2nd International Conference on Pervasive Computing (Vienna, Austria), April 2004, pp. 188–193.

[FL05] ———, *RFID Middleware Design - Addressing Application Requirements and RFID Constraints*, Proceedings of the Joint Conference on Smart Objects and Ambient Intelligence (soC-EUASI) (Grenoble, France), October 2005, pp. 219–224.

[FPSM92] William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus, *Knowledge Discovery in Databases: An Overview*, AI Magazine **13** (1992), no. 3, 57–70.

[FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth, *The KDD Process for Extracting Useful Knowledge from Volumes of Data*, Communications of the ACM **39** (1996), no. 11, 27–34.

[GDHK06] Joshua D. Griffin, Gregory D. Durgin, Andreas Haldi, and Bernard Kippelen, *RF Tag Antenna Performance on Various Materials Using Radio Link Budgets*, IEEE Antennas and Wireless Propagation Letters **5** (2006), no. 1, 247–250.

[Ger] GS1 Germany, *RFID / EPC - Transportetikett Empfehlung*, http://www.epcglobal.de.

[Ger99] Neil Gershenfeld, *When Things Start to Think*, Henry Holt and Co., Inc., New York, NY, USA, 1999.

[GH07] Fabrice Guillet and Howard J. Hamilton, *Quality Measures in Data Mining*, Studies in Computational Intelligence, Springer, New York, NY, USA, 2007.

[GK95] Dina Q. Goldin and Paris C. Kanellakis, *On Similarity Queries for Time-Series Data: Constraint Specification and Implementation*, Proceedings of the 1st International Conference on Principles and Practice of Constraint Programming (CP) (Cassis, France), September 1995, pp. 137–153.

[Hec95] David Heckerman, *A Tuturial on Learning with Bayesian Networks*, Tech. report, Microsoft Research, 1995.

[Hei05] Claus Heinrich, *RFID and Beyond: Growing Your Business Through Real World Awareness*, John Wiley & Sons Inc., Hoboken, NJ, USA, 2005.

[HM08] Bill C. Hardgrave and Robert Miller, *RFID Technology and Applications*, ch. RFID in the Retail Supply Chain: Issues and Opportunities, Cambridge University Press, Cambridge, UK, 2008.

[HMS66] Earl B. Hunt, Janet Marin, and Phlip J. Stone, *Experiments in Induction*, Academic Press, New York, NY, USA, 1966.

[HT06] Michael Hugos and Chris Thomas, *Supply Chain Management in the Retail Industry*, ch. An Introduction to Supply Chain Management, pp. 1–30, John Wiley & Sons Inc., Hoboken, NJ, USA, 2006.

[JAF+06] Shawn R. Jeffery, Gustavo Alonso, Michael J. Franklin, Wei Hong, and Jennifer Widom, *Declarative Support for Sensor Data Cleaning*, Proceedings of the 4th International Conference on Pervasive Computing (Dublin, Ireland), May 2006, pp. 83–100.

[JC08] Erick C. Jones and Christopher A. Chung, *RFID in Logistics: A practical Introduction*, ch. RFID System Design, pp. 63–84, CRC Press, Boca Raton, FL, USA, 2008.

[JFRP06] Bing Jiang, K.P. Fishkin, S. Roy, and M. Philipose, *I Sense a Disturbance in the Force: Unobtrusive long-range Detection of passive RFID Tag Motion*, IEEE Transactions on Instrumentation and Measurement **55** (2006), no. 1, 187–196.

[JGF06] Shawn R. Jeffery, Minos Garofalakis, and Michael Franklin, *Adaptive Cleaning for RFID Data Streams*, Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB) (Seoul, South Korea), September 2006, pp. 163–174.

[Kam09] Chandrika Kamath, *Scientific Data Mining: A Practical Perspective*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009.

[Keo03] Eamonn Keogh, *Knowledge Discovery in Databases*, Lecture Notes in Computer Science, ch. Efficiently Finding Arbitrarily Scaled Patterns in Massive Time Series Databases, pp. 253–265, Springer, Berlin, Germany, 2003.

[KH02] Mikko Karkkainen and Jan Holmstrom, *Wireless Product Identification: Enabler for Handling Efficiency, Customisation and Information Sharing*, Supply Chain Management: An International Journal **7** (2002), no. 4, 242–252.

[Kot07] Sotiris B. Kotsiantis, *Supervised Machine Learning: A Review of Classification Techniques*, Informatica **31** (2007), no. 3, 249–268.

[KPZ+04] Eamonn Keogh, Themistoklis Palpanas, Victor B. Zordan, Dimitrios Gunopulos, and Marc Cardle, *Indexing Large Human-Motion Databases*, Proceedings of the 30th International Conference on Very Large Data Bases (VLDB) (Toronto, Canada), August 2004, pp. 780–791.

[KR90] Leonard Kaufman and Peter J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons Inc., Hoboken, NJ, USA, 1990.

[LAR02] Weiyang Lin, Sergio A. Alvarez, and Carolina Ruiz, *Efficient Adaptive-Support Association Rule Mining for Recommender Systems*, Data Mining and Knowledge Discovery **6** (2002), no. 1, 83–105.

[Lar05a] Daniel T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*, Wiley Interscience, John Wiley & Sons Inc., Hoboken, NJ, USA, 2005.

[Lar05b] ———, *Discovering Knowledge in Data: An Introduction to Data Mining*, Wiley Interscience, ch. Data Preprocessing, pp. 27–40, John Wiley & Sons Inc., Hoboken, NJ, USA, 2005.

[Lar05c] ———, *Discovering Knowledge in Data: An Introduction to Data Mining*, Wiley Interscience, ch. Decision Trees, pp. 107–127, John Wiley & Sons Inc., Hoboken, NJ, USA, 2005.

[LM68] Peter A. Lachenbruch and M. Ray Mickey, *Estimation of Error Rates in Discriminant Analysis*, Technometrics **10** (1968), no. 1, 1–11.

[LO07] Hau Lee and Ozalp Ozer, *Unlocking the Value of RFID*, Production and Operations Management **16** (2007), no. 1, 40–46.

[Mac67] James B. MacQueen, *Some Methods of Classification and Analysis of Multivariate Observations*, Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability (Berkeley, CA, USA), December 1967, pp. 281–297.

[Mar97] J. Kent Martin, *An Exact Probability Metric for Decision Tree Splitting and Stopping*, Machine Learning **28** (1997), no. 2, 257–291.

[MB10] David Martens and Bart Baesens, *Data Mining*, 8th ed., Special Issue in Annals of Information Systems, ch. Building Acceptable Classification Systems, pp. 53–74, Springer, New York, NY, USA, 2010.

[MET07a] METRO Group, *METRO Group RFID Innovation Center - An Information and Development Platform for the Future of Commerce*, 2007.

[MET07b] ———, *METRO Group rolls out largest RFID project in the European retail sector*, Press Release, November 2007.

[MET07c] ———, *RFID Ready for Action - Technical Analysis of the Use of RFID at Case Level in Retail Logistics*, 2007.

[MF10] Friedemann Mattern and Christian Floerkemeier, *From Active Data Management to Event-Based Systems and More*, Lecture Notes in Computer Science, vol. 6462, ch. From the Internet of Computers to the Internet of Things, pp. 242–259, Springer, Berlin, Germany, 2010.

[MM05] Katina Michael and Luke McCathie, *The Pros and Cons of RFID in Supply Chain Management*, Proceedings of the 4th International Conference on Mobile Business (ICMB) (Sydney, Australia), July 2005, pp. 623–629.

[MS03] Duncan McFarlane and Yossi Sheffi, *The Impact of Automatic Identification on Supply Chain Operations*, International Journal of Logistics Management **14** (2003), no. 1, 1–18.

[MSW08] Stephen B. Miles, Sanjay E. Sarma, and John R. Williams, *RFID Technology and Applications*, Cambridge University Press, Cambridge, UK, 2008.

[MTS07] Adam Melski, Lars Thoroe, and Matthias Schumann, *Managing RFID Data in Supply Chains*, International Journal of Internet Protocol Technology **2** (2007), no. 3, 176–189.

[MV01] V. Manthou and M. Vlachopoulo, *Bar-Code Technology for Inventory and Marketing Management Systems: A Model for its Development and Implementation*, International Journal of Production Economics **71** (2001), no. 1, 157–164.

[NMRY08] E.W.T. Ngai, Karen K.L. Moon, Frederick J. Riggins, and Candace Y. Yi, *RFID Research: An academic Literature Review (1995-2005) and future Research Directions*, International Journal of Production Economics **112** (2008), no. 2, 510–520.

[NSML09] Pavel V. Nikitin, K.V. Seshagiri, Rene Martinez, and Sander F. Lam, *Sensitivity and Impedance Measurements of UHF RFID Chips*, IEEE Transactions on Microwave Theory and Techniques **57** (2009), no. 5, 1297–1302.

[PB02] Katherine Pullen and Christoph Berger, *Motion Capture assisted Animation: Texturing and Synthesis*, ACM Transaction on Graphics **21** (2002), no. 3, 501–508.

[PDG06] Joel T. Prothro, Gregory D. Durgin, and Joshua D. Griffin, *The Effects of a Metal Ground Plane on RFID Tag Antennas*, Proceedings of the IEEE Antennas and Propagation Society International Symposium (APS) (Albuquerque, NM, USA), July 2006.

[PKSK06] Katariina Penttila, Mikko Keskilammi, Lauri Sydanheimo, and Markku Kivikoski, *Radio Frequency Technology for automated Manufacturing and Logistics Control. Part 2: RFID Antenna Utilisation in Industrial Applications*, The International Journal of Advanced Manufacturing Technology **31** (2006), no. 2, 116–124.

[Pyl99] Dorian Pyle, *Data Preparation for Data Mining*, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, San Francisco, CA, USA, 1999.

[Qui86] John R. Quinlan, *Induction of Decision Trees*, Machine Learning **1** (1986), 81–106.

[Qui93] ———, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, CA, USA, 1993.

[Sat01] Mahadev Satyanarayanan, *Pervasive Computing: Vision and Challenges*, IEEE Personal Communications **8** (2001), no. 4, 10–17.

[SBA00] Sanjay Sarma, David L. Brock, and Kevin Ashton, *The Networked Physical World: Proposals for Engineering the Next Generation of Computing, Commerce & Automatic-Identification*, Tech. report, Auto-ID Center, MIT, Cambridge, USA, 2000.

[SC78] Hiroaki Sakoe and Seibi Chiba, *Dynamic Programming Algorithm Optimization for spoken Word Recognition*, IEEE Transactions on Acoustics, Speech, and Signal Processing **26** (1978), no. 1, 43–49.

[SON95] Ashoka Savasere, Edward Omiecinski, and Shamkant B. Navathe, *An Efficient Algorithm for Mining Association Rules in Large Databases*, Proceedings of the 21th International Conference on Very Large Data Bases (VLDB) (Zurich, Switzerland), September 1995, pp. 432–444.

[SOVT09] J. Singh, E. Olsen, K. Vorst, and K. Tripp, *RFID Tag Readability Issues with Palletized Loads of Consumer Goods*, Packaging Technology and Science **22** (2009), no. 8, 431–441.

[SP06] Riyaz T. Sikora and Selwyn Piramuthu, *Genetic Algorithm Based Learning Using Feature Construction*, Institute for Operations Resarch and Management Sciences Workshop on Artificial Intelligence and Data Mining (Pittsburgh, PA, USA), November 2006.

[Sri04] Bharatendu Srivastava, *Radio Frequency ID Technology: The next Revolution in SCM*, Business Horizons **47** (2004), no. 6, 60–68.

[Sto74] Mervyn Stone, *Cross-Validatory Choice and Assessment of Statistical Predictions*, Journal of the Royal Statistical Society **36** (1974), no. 1, 111–147.

[TAKF09] Frederic Thiesse, Jasser Al-Kassab, and Elgar Fleisch, *Understanding the Value of Integrated RFID Systems: A Case Study from Apparel Retail*, European Journal of Information Systems **18** (2009), no. 6, 592–614.

[TFH+09] Frederic Thiesse, Christian Floerkemeier, Mark Harrison, Florian Michahelles, and Christof Roduner, *Technology, Standards, and Real-World Deployments of the EPC Network*, IEEE Internet Computing **13** (2009), no. 2, 36–43.

[TP08a] Yu-Ju Tu and Selwyn Piramuthu, *A Decision Support Model for Filtering RFID Read Data*, Proceedings of the 16th International Conference on Advanced Computing and Communications (ADCOM) (Chennai, India), December 2008, pp. 221–224.

[TP08b] ———, *Reducing False Reads in RFID-Embedded Supply Chains*, Journal of Theoretical and Applied Electronic Commerce Research **3** (2008), no. 2, 60–70.

[TZP09] Yu-Ju Tu, Wei Zhou, and Selwyn Piramuthu, *Identifying RFID-embedded Objects in pervasive Healthcare Applications*, Decision Support Systems **46** (2009), no. 2, 586–593.

[Vog02] Harald Vogt, *Efficient Object Identification with Passive RFID Tags*, Proceedings of the 1st International Conference on Pervasive Computing (Zurich, Switzerland), August 2002, pp. 98–113.

[Wan04] Roy Want, *The Magic of RFID*, ACM Queue **2** (2004), no. 7, 41–48.

[Wei91] Mark Weiser, *The Computer for the 21st Century*, Scientific American **265** (1991), no. 3, 66–75.

[WGPR07] Gareth R.T. White, Georgina Gardiner, Guru Prabhakar, and Azley Abd Razak, *A Comparison of Barcoding and RFID Technologies in Practice*, Journal of Information, Information Technology and Organizations **2** (2007), 119–132.

[Wyl06] David C. Wyland, *RFID 101: The next big Thing for Management*, Management Research News **29** (2006), no. 4, 154–173.

[Ye08] Lu Ye, *A Research on General Software Architecture on RFID*, Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT) (Busan, South Korea), November 2008, pp. 1054–1057.

[ZADB06] Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko, *Similarity Search - The Metric Space Approach*, Advances in Database Systems, Springer, New York, NY, USA, 2006.

[Zha00] Guoqiang P. Zhang, *Neural Networks for Classification: A Survey*, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **20** (2000), no. 4, 451–462.

Curriculum Vitae

Personal Data

Birthplace	Wuppertal, Germany
Birthdate	December 16, 1981
Citizenship	German

Education

1981 - 1991	Grundschule Kohlstrasse (Wuppertal, Germany)
1997 - 1998	Myrtle Beach High School (Myrtle Beach, SC, USA)
1991 - 2000	Altsprachliches Wilhem-Dörpfeld Gymnasium (Wuppertal, Germany) Abitur
2001 - 2007	RWTH Aachen University (Aachen, Germany) Master of Science in Computer Science
2009 - 2011	University of St. Gallen (St. Gallen, Switzerland) Ph.D. in Management

Civil Service

2000 - 2001	German Air Force (Budel, Netherlands and Cologne, Germany) Private First Class

Work Experience

2002 - 2007	Institute of Plastics Processing (IKV) (Aachen, Germany) Student Assistant - Software Development
2008 - 2011	IBM Deutschland GmbH (Frankfurt, Germany) Senior Consultant - RFID & Sensor Solutions