Moataz Mansour
Senior Database Architect
Swiss Post
Moataz.mansour@bluewin.ch

Tobias Böni
Wissenschaftlicher Assistant
Berner Fachhochschule
Tobias.boeni@bluewin.ch

CAS ADS Uni Bern Data Science Final Project

Swiss Post Sorting Center Package Sorting Performance Analysis and Prediction

Final Project Report

15 June 2025

Abstract

Swiss Post processes a substantial volume of packages each year, delivering around 200 million parcels in 2023 alone. This highlights its crucial role in Switzerland's logistics network, covering both domestic and international shipments. However, in 2023, Swiss Post encountered performance challenges, driven by external factors such as inflation, weakened consumer confidence, and geopolitical events. These factors led to a 4.7% decline in parcel volumes compared to 2022. Despite this, Swiss Post has made considerable investments in its logistics network to address efficiency issues, including the introduction of new regional parcel centers and upgraded sorting technology. These improvements have eased the pressure on processing times in high-demand areas like Zurich and Basel.

The surge in package deliveries that began during the COVID-19 crisis overwhelmed the existing infrastructure, prompting Swiss Post to invest millions in new sorting centers to meet the growing demand. By enhancing the efficiency of the existing sorting centers, Swiss Post could fully optimize their usage, reducing the need for further costly investments in additional facilities.

This project focuses on analyzing sorting center operations performance to identify congestion points and their causes, while also developing predictive models to foresee and prevent performance issues before they occur. This proactive strategy will help improve the overall performance of Swiss Post's sorting centers.

Table of Contents

1. Introduction

The goal of this project is to analyze the postal sorting center's performance by identifying the most influential factors contributing to sorting issues, including shipment attributes (e.g., dimensions, weight, coding stations, and timestamps) as well as chute congestion which we believe have a major impact on overall performance of the center and finally using a model to determine whether overburdened chutes or certain features create bottlenecks that reduce overall system efficiency.

The sorting process consists of three key stages:

A: Shipments **arrive** and are delivered to the sorting center, using designated units

B: Shipments are **scanned and transferred** to the conveyor belts, where automatic and manual sorting machines determine their route and send them to the appropriate chute based on the destination.

C: The **chute serves as the output** of the sorting machine, directing the parcels to different destinations depending on the ZIP code., one chute can serve several ZIP codes

The center's performance is measured by the number of parcels processed per time frame. Some centers have shown up to a 15% increase in processing efficiency compared to others with similar setups. Further investigation revealed that traffic bottlenecks at certain chutes, which handle significantly higher package volumes, cause an imbalance. This results in a non-normal distribution of packages across the chutes, leading to a noticeable reduction in overall performance.

Our goal is to achieve a balanced, normally distributed flow of packages across all available chutes, which would enhance the sorting rate and improve overall center performance.

Specific Objectives:

1.1. **Determine Feature Importance**: Rank the shipment features by their importance and identify which shipment attributes (e.g., dimensions, weight, coding station) are most influential in causing sorting issues at postal centers. As well Correlation between the features and the impact on performance

1.2. **Determine chute congestion impact:** Determine whether chutes are handling disproportionately large volumes of packages can create bottlenecks that reduce overall system efficiency

1.3. **Predict Sorting Performance and Issues**: Develop a predictive model capable of forecasting sorting issues based on historical shipment data.

1.4. **Generate Actionable Insights**: Provide data-driven recommendations for improving chute utilization and enhance overall performance

2. Infrastructure and Tools

Swiss Post's IT infrastructure consists of over 5,300 databases and more than 800 custom-built applications, distributed across two highly secure data centers supported by a fiber network backbone. These data centers are designed for disaster recovery, ensuring operational continuity. At the core of this system is the Oracle Exadata Cloud at Customer (EXACC) platform, a high-performance database server valued at 5 million CHF, which hosts many of Swiss Post's databases and is mirrored across both data centers for redundancy and high availability in case of system failures.

A- Databases are used to store shipment data and provide real-time updates, ensuring accurate and up-to-date information flow. These systems are built with high availability and redundancy for continuous operation, and they support advanced data analytics and reporting for performance monitoring and decision-making.

B- Applications, including parcel tracking systems, sorting system management, and predictive analytics for forecasting performance issues. Middleware and API integration enable seamless communication between different systems. These applications are hosted on virtualized or containerized platforms (e.g., Docker, Kubernetes), allowing Swiss Post to easily scale its IT resources based on demand.

C- network infrastructure, consisting of high-speed fiber optic networks, redundant architecture with failover mechanisms, VPNs, and multi-cloud integration, ensures reliable connectivity between sorting centers and offices, while stringent security measures, including firewalls, encryption, and regular audits, protect data and maintain compliance.

D- advanced control and monitoring systems for its sorting centers, such as SCADA (Supervisory Control and Data Acquisition) systems to oversee sorting operations, and centralized command centers that manage and coordinate the activities of the sorting centers.

E- sorting centers in Härkingen, Frauenfeld, Daillens, Wallisellen (new), and Pratteln (new) are equipped with automated sorting machines connected to the IT infrastructure through IoT sensors. These sensors provide continuous data streams to monitor sorting accuracy, operational efficiency, and package flow in real time.

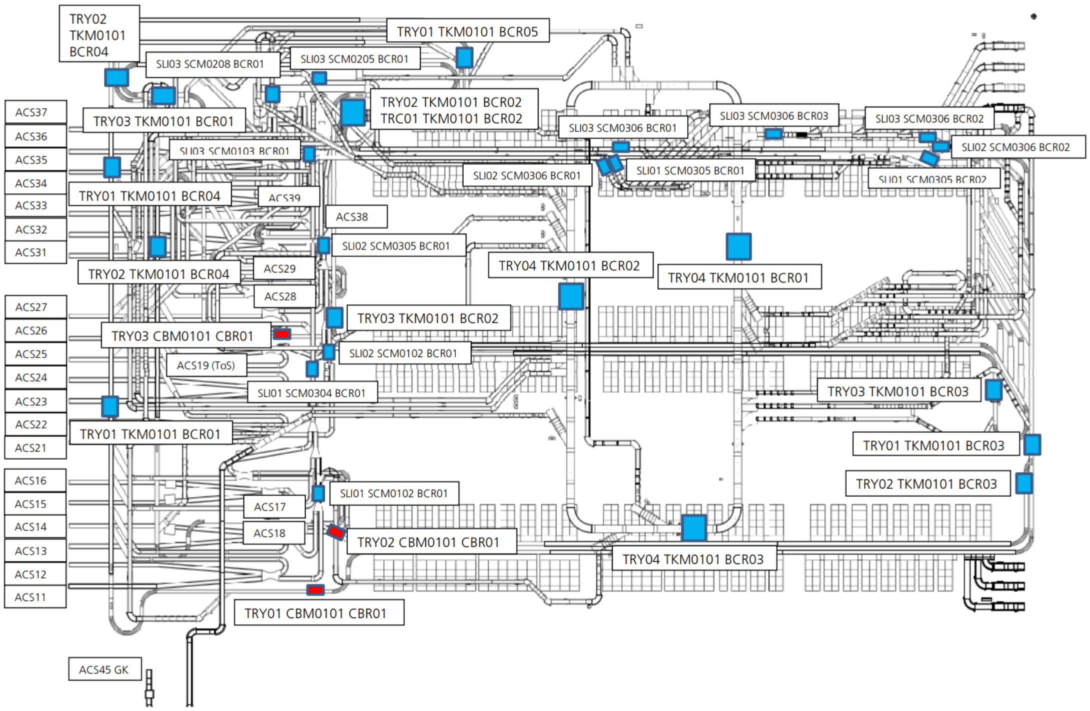
Swiss Post's comprehensive IT infrastructure, featuring powerful databases, application servers, and a robust network architecture, serves as the backbone of its logistics operations, enabling seamless parcel processing and tracking nationwide. With its scalability, security, and redundancy, the system ensures reliable and efficient performance, even during peak demand periods.

- **infrastructure and tools used for this project:**

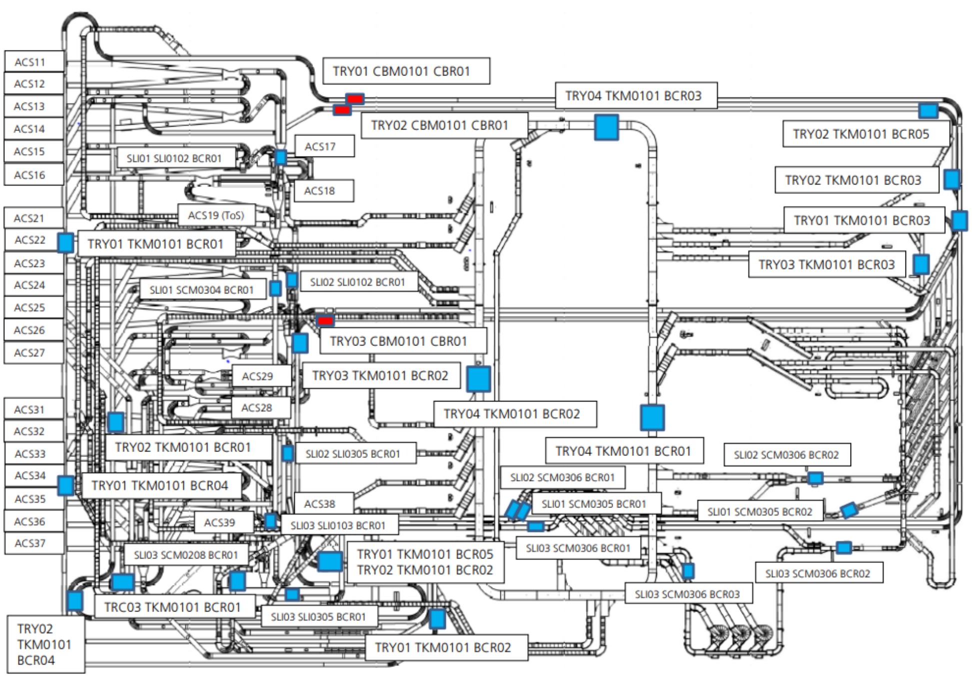
We have limited our study to focus on two parcel centers: **Härkingen** and **Frauenfeld**, and are analyzing data from a **single major database** that holds parcel processing information. This database contains massive tables, with an average of **5 billion records**, providing extensive data for our analysis of parcel operations.

- **Database Tools: Oracle 23ai** with vector search and other DB tools and Toad is used to manage the Data extraction and query and cleanup
- **Hardware:** on prem Exacc Database server with huge database power where package databases (PADASA and X) with 50 TB storage usage and 5 billion records tables runs and the Swiss post AWS server cloud infrastructure to run the developed predictive models
- **Python Libraries**:
    - **OracleDB:** using Python SQL To extract relevant parcel data from databases.
    - **Core Libraries:** pandas & numpy: data loading and manipulation, operations and arrays
    - **Visualization:** matplotlib & seaborn : data visualization & Advanced visualizations
    - **Deep Learning : PyTorch :** torch – PyTorch base torch.nn – for defining neural networks (e.g., LSTM, MLP) and torch.utils.data – for datasets and data loaders
    - **Machine Learning & Preprocessing:** sklearn.preprocessing.MinMaxScaler – feature scaling
    - sklearn.metrics – evaluation metrics, mean_absolute_error, accuracy_score, precision_score
    - recall_score, f1_score
    - **Reinforcement Learning:** gym – environment wrapper for RL agents
    - stable_baselines3 – RL agent training (PPO, DQN, etc.)
    - shimmy – compatibility layer for gym + stable_baselines3

Übersicht alle Lesesysteme PZ Härkingen Stand 2022



Übersicht alle Lesesysteme PZ Frauenfeld Stand 2022



3. DATA

The data for this project comes from Swiss Post's shipment sorting system and includes:

- **Shipment number** (anonymized) SND_IDENTCODE
- **Shipment dimensions**: Length, width, and height (in millimeters) SND_CODS_DIM1, SND_CODS_DIM2, SND_CODS_DIM3
- **Shipment weight**: (n grams) SND_GEW
- **Scanning timestamps** when the item first scanned in the sorting center CODS_COD_DAT
- **Scanner station**: Sorting station identifier CODS_CO_STATION
- **Sorting center Number** CODS_ZENT_NR_x
- **leaving timestamps** when the item left the sorting center chute CODS_LERE_DAT
- **chute station** where the item is sent CODS_SD_RUTSCHE

3.1. Sample Data Example

	CODS_IDENTCODE_AN	CODS_DIM1	CODS_DIM2	CODS_DIM3	CODS_GEW	CODS_ADR_PLZZZ_AN	CODS_CO_STATION	CODS_ZENT_NR	CODS_SD_RUTSCHE	DAY_15MIN	CODS_LERE_DAT	MINUTE_COUNTER	PROCESSING_TIME_MINUTES	SORTING_PERF_ISSUE
0	242455661	700	454	54	1480	449561	FRA-ACS18	3	R0109	202411061015	06/11/2024 10:26:25	447026	1	0
1	482169492	704	454	48	1520	568172	FRA-ACS18	3	R0109	202411061015	06/11/2024 10:26:15	447026	2	0
2	407572193	400	380	235	4700	23223	FRA-ACS21	3	R2515	202411061015	06/11/2024 10:26:16	447026	4	0
3	999133381	470	450	120	1060	629595	FRA-ACS21	3	R0109	202411061015	06/11/2024 10:26:04	447026	7	0
4	748472889	360	290	110	1480	421619	FRA-ACS21	3	R2319	202411061015	06/11/2024 10:26:37	447027	15	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
170012	302468774	490	390	90	1280	712077	DAI-ACS13	1	R2419	202411061430	06/11/2024 14:38:26	447278	4	0
170013	225629123	170	110	55	120	869319	DAI-ACS13	1	R1223	202411061430	06/11/2024 14:38:22	447278	2	0
170014	822331195	410	360	55	820	49468	DAI-ACS13	1	R2704	202411061430	06/11/2024 14:38:25	447278	6	0
170015	804061373	400	310	110	1640	32705	DAI-ACS13	1	R1709	202411061430	06/11/2024 14:38:19	447278	6	0
170016	486028131	410	300	110	2900	608088	DAI-ACS13	1	R2207	202411061430	06/11/2024 14:38:32	447279	2	0

809022 rows × 14 columns

3.2. Security Considerations

- **Anonymization**: All shipment numbers are anonymized to protect sensitive data.
- **Data Storage**: Data is stored on secure servers and access is controlled according to Swiss Post's privacy and security guidelines.

3.3. Required Metadata

Metadata such as the coding station, shipment size, and coding timestamps are critical for reproducing the analysis. These attributes allow for the recreation of sorting scenarios and the identification of problematic shipments.

Field Name	Description	Data Type	Example
SND_IDENTCODE	Unique identifier for each shipment (anonymized for privacy)	String	A12345
SND_CODS_DIM1	Length of the shipment in millimeters	Integer	300
SND_CODS_DIM2	Width of the shipment in millimeters	Integer	150
SND_CODS_DIM3	Height of the shipment in millimeters	Integer	50
SND_GEW	Weight of the shipment in grams	Integer	1000
CODS_COD_DAT	Timestamp indicating when the shipment was scanned into the sorting center	Datetime	15.01.2023 08:32
CODS_LERE_DAT	Timestamp indicating when the shipment left the sorting center	Datetime	15.01.2023 09:45
CODS_CO_STATION	Station or scanner ID at which the shipment was processed	String	STATION01
CODS_SD_RUTSCHE	Chute identifier where the package was routed for further processing	String	CHUTE10
processing_time_minutes	Calculated field representing the time taken to process a shipment in minutes	Float	73.5

- Dataset Structure: The dataset consists of shipment records, where each row represents a unique shipment with detailed metadata about its dimensions, weight, timestamps, processing station, and chute.
- Calculated Field: processing_time_minutes is derived from CODS_COD_DAT and CODS_LERE_DAT to measure the time a shipment spends in the sorting center.

3.4. Metadata Storage

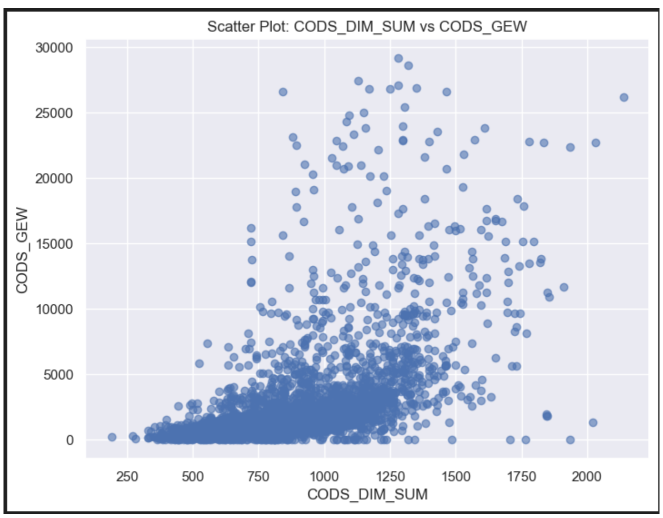
- Metadata will be stored alongside the shipment data in a secure database, with access controlled by Swiss Post.
- Authorized users can access the metadata through SQL queries and data dumps exported to CSV for further analysis.

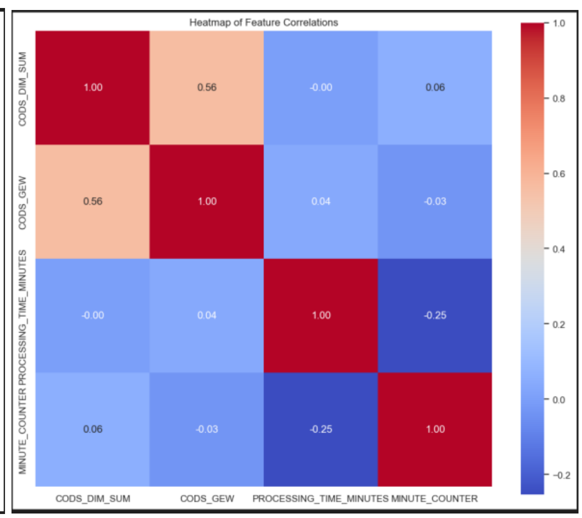
4. Exploratory data analysis (Statistical Descriptive Analysis)

- **Chute Utilization**: The number of packages processed per chute and the average processing time per chute were calculated. This analysis helped identify chutes that were handling a disproportionately large number of shipments and thus were more likely to experience congestion.

- **Performance Metrics**: Center-wide performance metrics were computed, including average processing time per package, total number of packages processed, and individual chute performance.

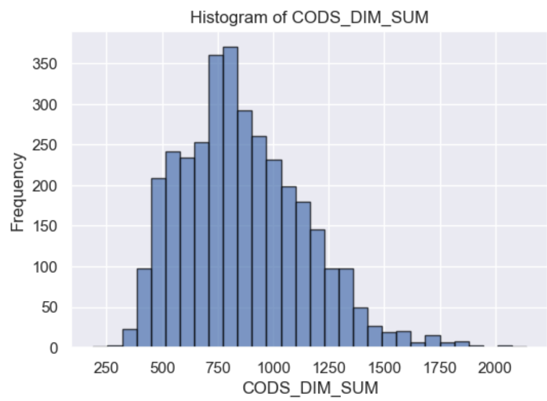
- **Data Correlation:** The correlation between specific shipments or packages and performance bottlenecks will be analyzed to determine if certain supplier lots arriving at the center are contributing to the overutilization of particular chutes. By identifying these patterns, proactive redistribution measures can be implemented to prevent bottlenecks and optimize center efficiency.

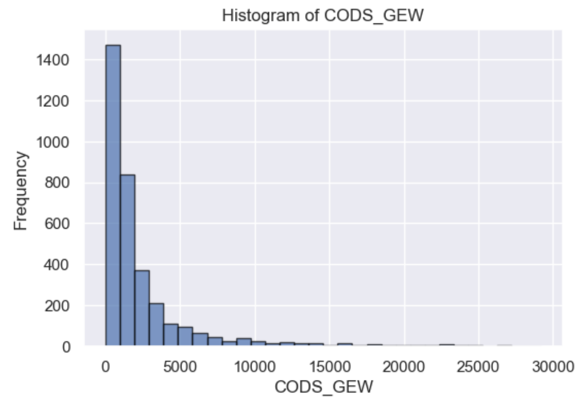




(Scatter Plot) & (Correlation Heatmap):
There is a positive correlation between total package dimensions and weight, with larger parcels generally weighing more, but a substantial spread indicates other factors also influence package weight.

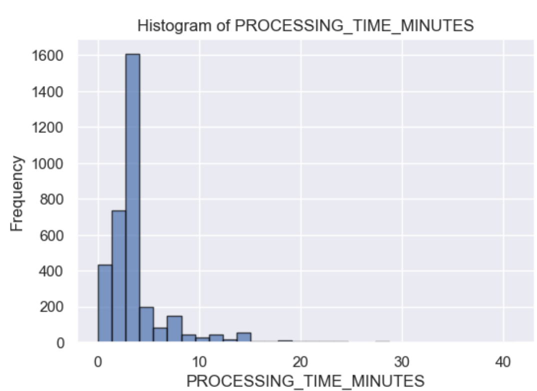
The heatmap confirms that while dimensions and weight are strongly correlated, both have weak correlation with processing time, suggesting operational delays are not primarily driven by package size or weight.

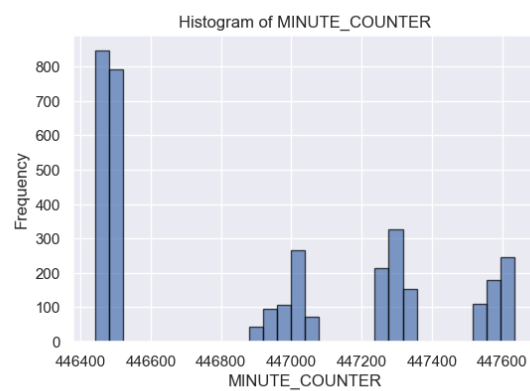




? Histogram of CODS_DIM_SUM:
This histogram illustrates the distribution of total package dimensions (sum of length, width, height) processed at the sorting center. The data is approximately normally distributed, with most parcels having dimensions between 500 and 1500 millimeters. This suggests a consistent range of parcel sizes handled by the system.

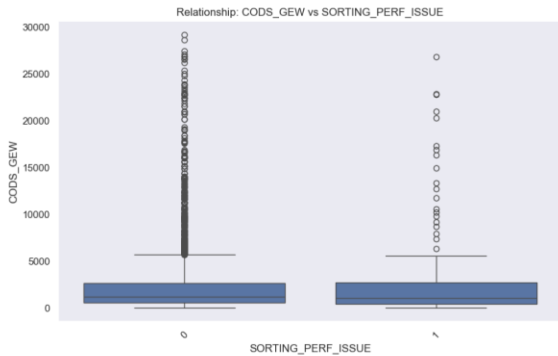
? Histogram of CODS_GEW:
The shipment weight distribution is highly right-skewed, indicating that the majority of parcels are relatively lightweight (under 5000 grams), while a small number of packages are much heavier. These heavy outliers may require special handling and could contribute to processing delays.

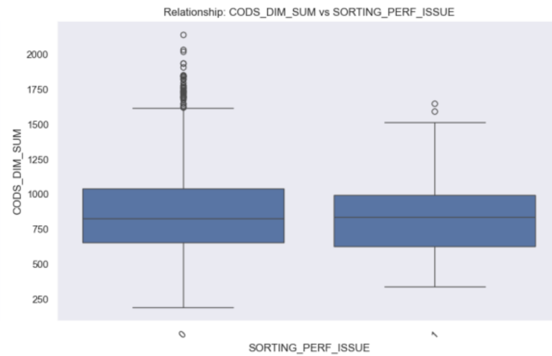




? Histogram of PROCESSING_TIME_MINUTES:
Most parcels are processed in less than 5 minutes, but there is a long tail of packages with significantly higher processing times. This skewed distribution highlights generally efficient operations but with occasional significant delays.

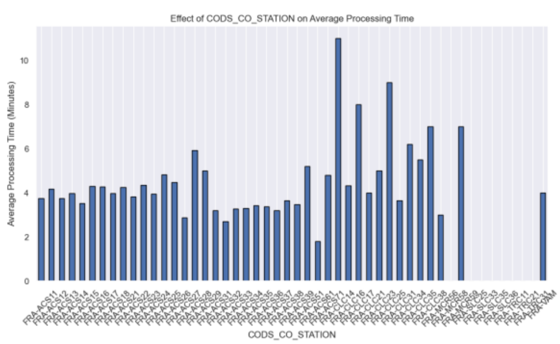
? Histogram of MINUTE_COUNTER:
This plot shows the distribution of processing times across different minute intervals, revealing periodic spikes that may correspond to operational schedules, batch arrivals, or system resets within the sorting center.

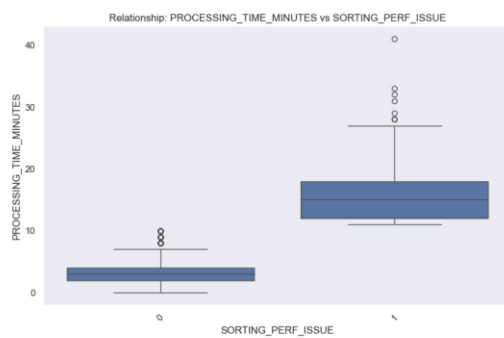




Boxplot: CODS_GEW vs. SORTING_PERF_ISSUE:
This boxplot compares package weights for shipments with and without sorting performance issues. There is a slight tendency for heavier packages to be associated with performance issues, though the effect appears marginal and most weights cluster below 5,000 grams.

Boxplot: CODS_DIM_SUM vs. SORTING_PERF_ISSUE:
Here, the sum of package dimensions is compared for problematic vs. non-problematic shipments. The distributions are similar, indicating that package size alone does not strongly predict sorting issues.



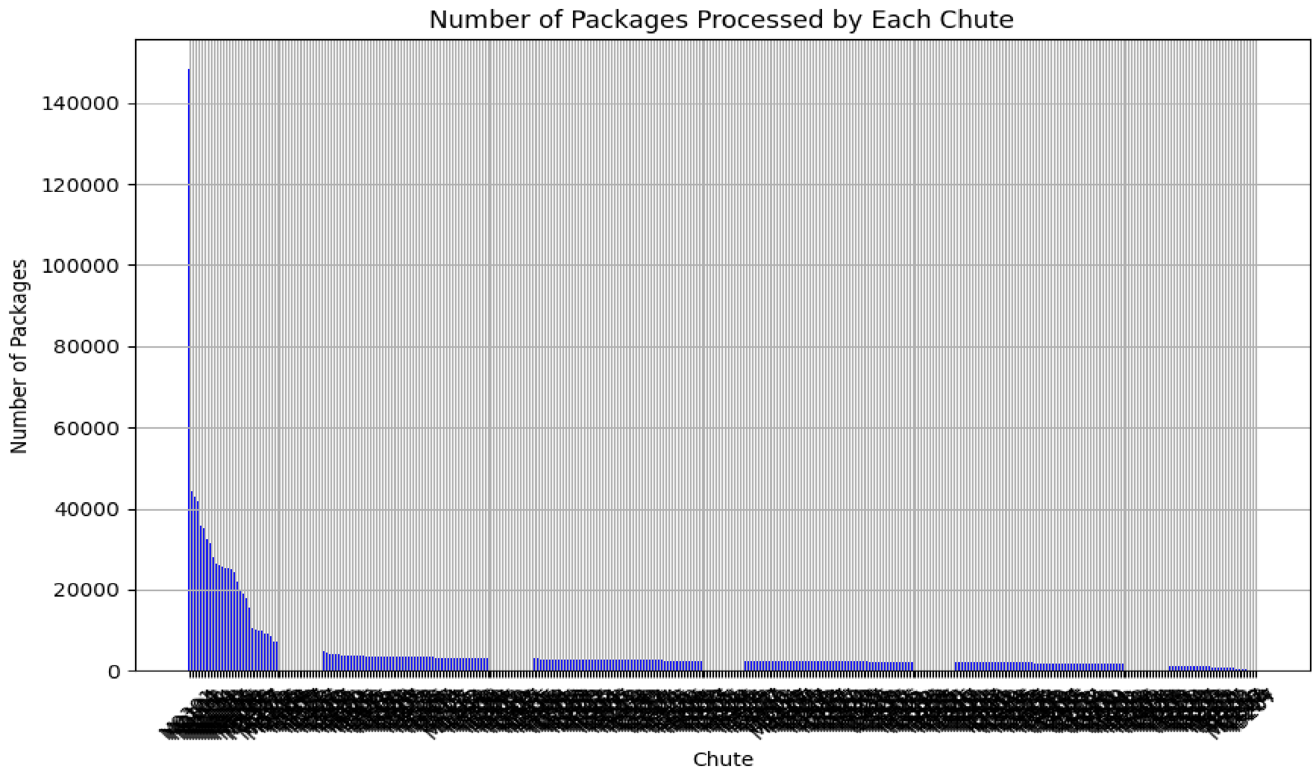


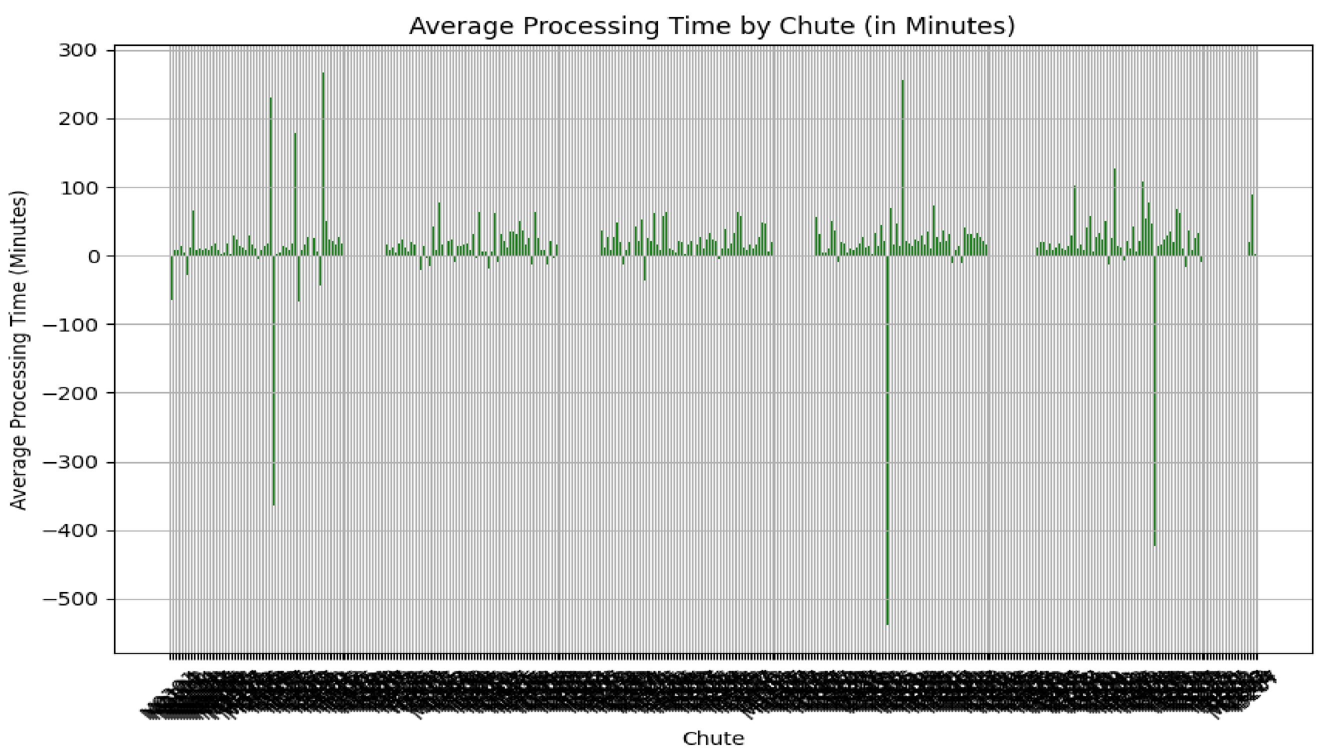
Bar Plot: Average Processing Time by CODS_CO_STATION:
This bar plot displays the average processing time for each coding station. There is notable variation, suggesting that some stations consistently process parcels more quickly than others, possibly due to differences in workload, staffing, or equipment.

Boxplot: PROCESSING_TIME_MINUTES vs. SORTING_PERF_ISSUE:
Processing times are compared for shipments with and without performance issues. Parcels associated with sorting performance problems have a much wider spread and higher median processing times, confirming that this metric is a good indicator of operational delays.

Most parcels are processed quickly and efficiently, but rare outliers—often heavier or handled at certain times or stations—lead to significant delays. Targeting these bottlenecks can further optimize overall sorting center performance

Histograms and Plots to visualize the distribution of shipment across the Chutes





4.1. Calculating Performance (processing_time_minutes)

To evaluate the performance of the sorting center, we introduced a new column that captures processing time, which is a key performance metric. Processing time refers to the duration a package spends in the sorting process, from the moment it is scanned until it leaves the sorting center, added to the dataset

processing_time_minutes (i) = CODS_COD_DAT(i) - CODS_LERE_DAT(i)

Purpose of the Performance Columns:

a. Identify Delays: The processing_time_minutes column allows us to detect delays and inefficiencies in the sorting process. A high value indicates a potential issue, such as congestion or slow processing, while a lower value suggests efficient performance.

b. Compare Across Stations/Chutes: By grouping the data by sorting stations or chutes, we can compare average processing times and detect whether certain stations or chutes are causing bottlenecks.

c. Overall Center Performance can be calculated on time period bases like daily or hourly and then can be compared with other centers to identify which is more performant and validate the predictions

Performance (Center) = SUM (processing_time_minutes (i)) / Count

4.2. Data Preprocessing

The raw dataset was first preprocessed to ensure that it was suitable for analysis and modeling. Key steps in the preprocessing pipeline included:

- **Data Cleaning**: Missing values in key fields such as CODS_COD_DAT and CODS_LERE_DAT (timestamps) were filtered out. Inconsistent data, such as negative processing times, were also removed.
- **Feature Creation**:The processing_time_minutes column was derived by calculating the difference between the entry and exit timestamps for each shipment.
- Additional features, such as the number of packages processed per chute and average processing time per chute, can be generated to assess congestion at the chute level.

- **Outlier Detection**:
  - Outliers in key fields, such as SND_GEW (weight) and processing_time_minutes, were identified using the **Interquartile Range (IQR)** method and removed to prevent skewing the analysis.

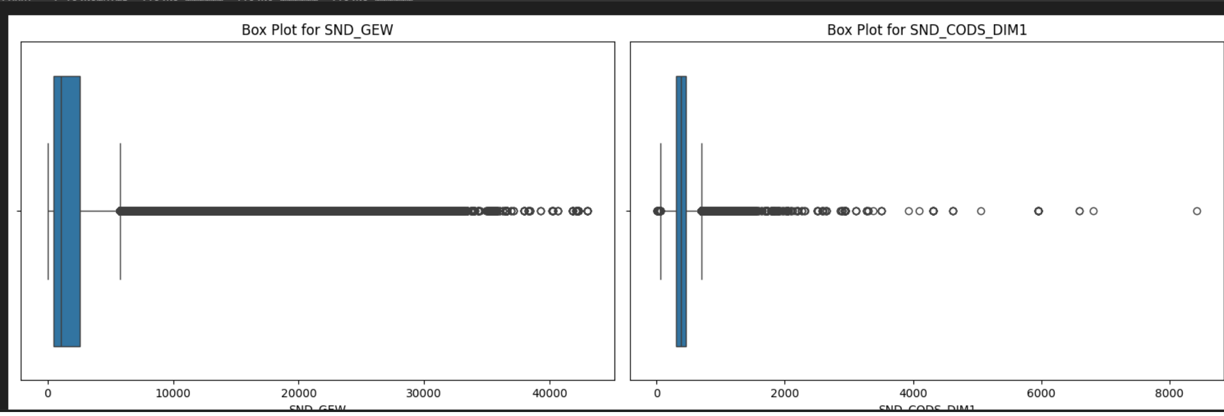
For example The processing time outfitters and the negative time values should be cleaned

a. **Calculate IQR** = Q3 - Q1 and Define the bounds for outliers lower_bound = Q1 - 1.5 * IQR upper_bound = Q3 + 1.5 * IQR
b. **Identify outliers** = data[(data['processing_time_minutes'] < lower_bound) | (data['processing_time_minutes'] > upper_bound)]
c. **Remove the outliers by filtering the data cleaned_data** = data[(data['processing_time_minutes'] >= lower_bound) & (data['processing_time_minutes'] <= upper_bound)]

```
Summary of outliers in SND_GEW:
       SND_IDENTCODE  SND_CODS_DIM1  SND_CODS_DIM2  SND_CODS_DIM3  \
count   3.984580e+05  398385.000000  398385.000000  398385.000000
mean    6.670258e+09     565.754303     407.914206     292.407407
std     3.317503e+10     157.241636      95.041501      91.422043
min     4.946300e+04      10.000000      10.000000      10.000000
25%     2.482289e+09     450.000000     340.000000     225.000000
50%     5.006381e+09     590.000000     400.000000     295.000000
75%     7.516051e+09     620.000000     460.000000     350.000000
max     9.981578e+11    5960.000000     912.000000     740.000000

             SND_GEW  CODS_ZENT_NR_x  CODS_ZENT_NR_y  processing_time_minutes
count  398458.000000   398458.000000   398458.000000            398458.000000
mean    12179.256396        2.372365        2.372365                 0.261907
std      5848.953016        0.483436        0.483436               348.457689
min      5785.000000        2.000000        2.000000             -6413.600000
25%      7560.000000        2.000000        2.000000                 1.620000
50%     10360.000000        2.000000        2.000000                 3.170000
75%     15160.000000        3.000000        3.000000                 5.000000
max     42980.000000        3.000000        3.000000              6421.780000


Summary of outliers in SND_CODS_DIM1:
       SND_IDENTCODE  SND_CODS_DIM1  SND_CODS_DIM2  SND_CODS_DIM3  \
```



Descriptive Statistics (Top):
The summary statistics reveal that both package weights and dimensions exhibit high variability, with mean values much lower than their respective maximums, indicating the presence of extreme outliers in the dataset.

Box Plots (Bottom):
The box plots for weight (SND_GEW) and length (SND_CODS_DIM1) confirm a significant number of outliers, highlighting that while most parcels are within a typical range, a small number of unusually large or heavy items could impact operational consistency.

5. Data Quality

Data quality is a critical component of the analysis, as poor data quality can lead to incorrect conclusions and unreliable results. In this section, we evaluate the data for **completeness**, **consistency**, **accuracy**, and **timeliness**.

5.1. Completeness

Missing Data: Certain fields, such as CODS_COD_DAT and CODS_LERE_DAT, are essential for calculating processing time. Rows with missing or incorrect timestamps result in missing processing_time_minutes values.

Handling Missing Data: Missing timestamp values are filtered out, as they would disrupt the calculation of performance metrics. For other fields (e.g., dimensions or weight), imputation or removal may be necessary if the data is critical to the analysis.

5.2. Consistency

Timestamps Consistency: The data is checked for consistency between the CODS_COD_DAT (entry time) and CODS_LERE_DAT (exit time). Any cases where the exit time is earlier than the entry time (resulting in negative processing times) are flagged as inconsistent and removed from the dataset.

Data Formatting: All date and time fields are standardized to UTC to avoid issues arising from different time zones or formats. Other fields, such as SND_CODS_DIM1, SND_CODS_DIM2, and SND_GEW, are checked to ensure consistent units (millimeters for dimensions, grams for weight).

5.3. Accuracy

Outliers: The data was checked for extreme values or outliers, particularly in the SND_GEW (weight) and processing_time_minutes fields. Outliers may indicate potential data entry errors or operational inefficiencies. The Interquartile Range (IQR) method was used to detect and remove outliers from the dataset.

Anomalies: Anomalies in the timestamps, such as extremely short or long processing times, are investigated. While extremely short times could indicate system issues, extremely long times might signal congestion or inefficiencies within the sorting center.

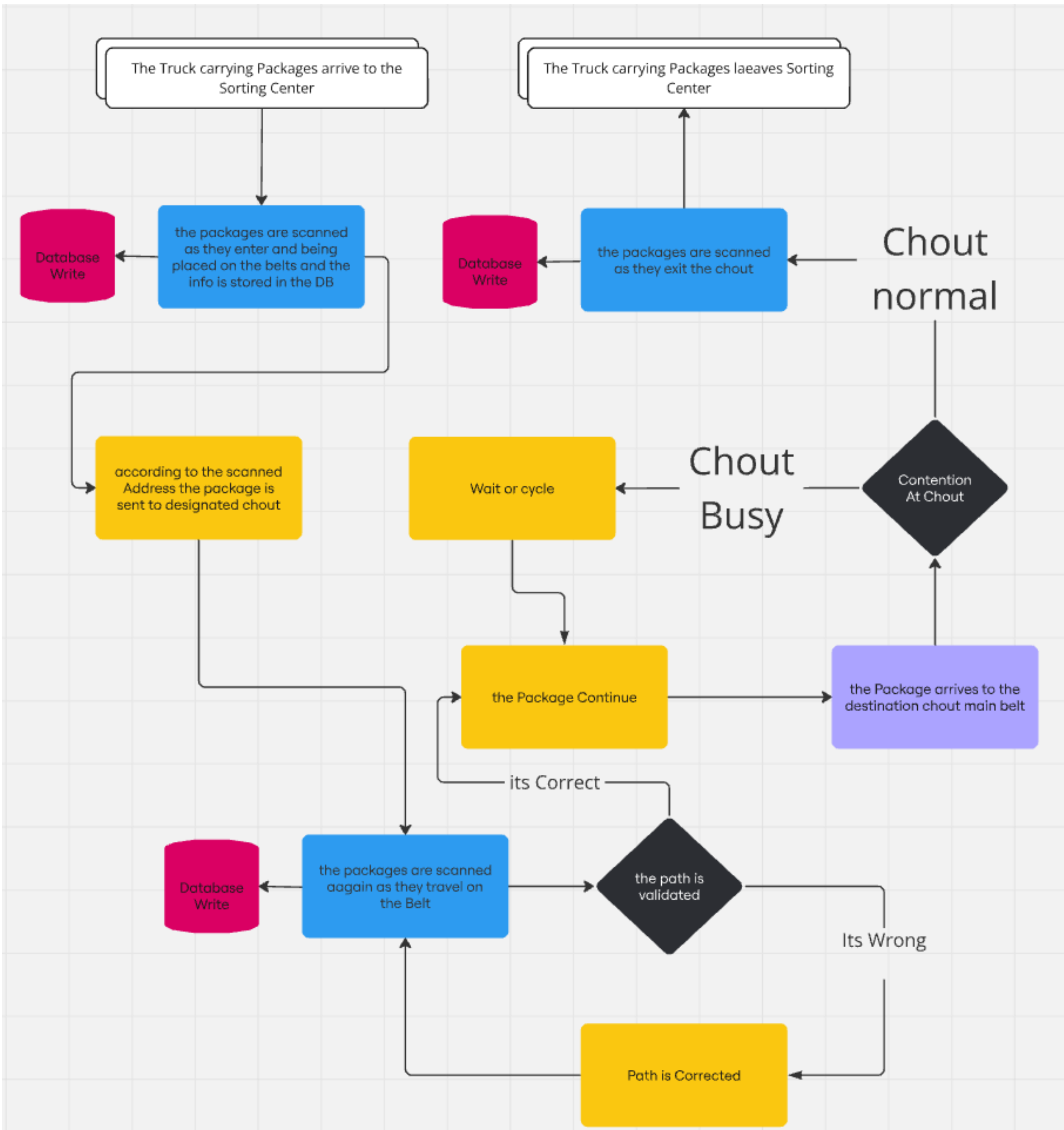
5.4. Data Integrity

Integrity of Identifiers: The SND_IDENTCODE field, which uniquely identifies each shipment, is checked for duplicate entries to ensure that each record represents a unique shipment. This field is also anonymized using a hashing function to protect sensitive information.

Chute and Station Integrity: Validations are in place to ensure that CODS_CO_STATION and CODS_SD_RUTSCHE match with known station and chute identifiers to avoid mismatches or routing errors.

6. Data Flow

The shipment physical flow in the Sorting Center and its corresponding DATA Flow

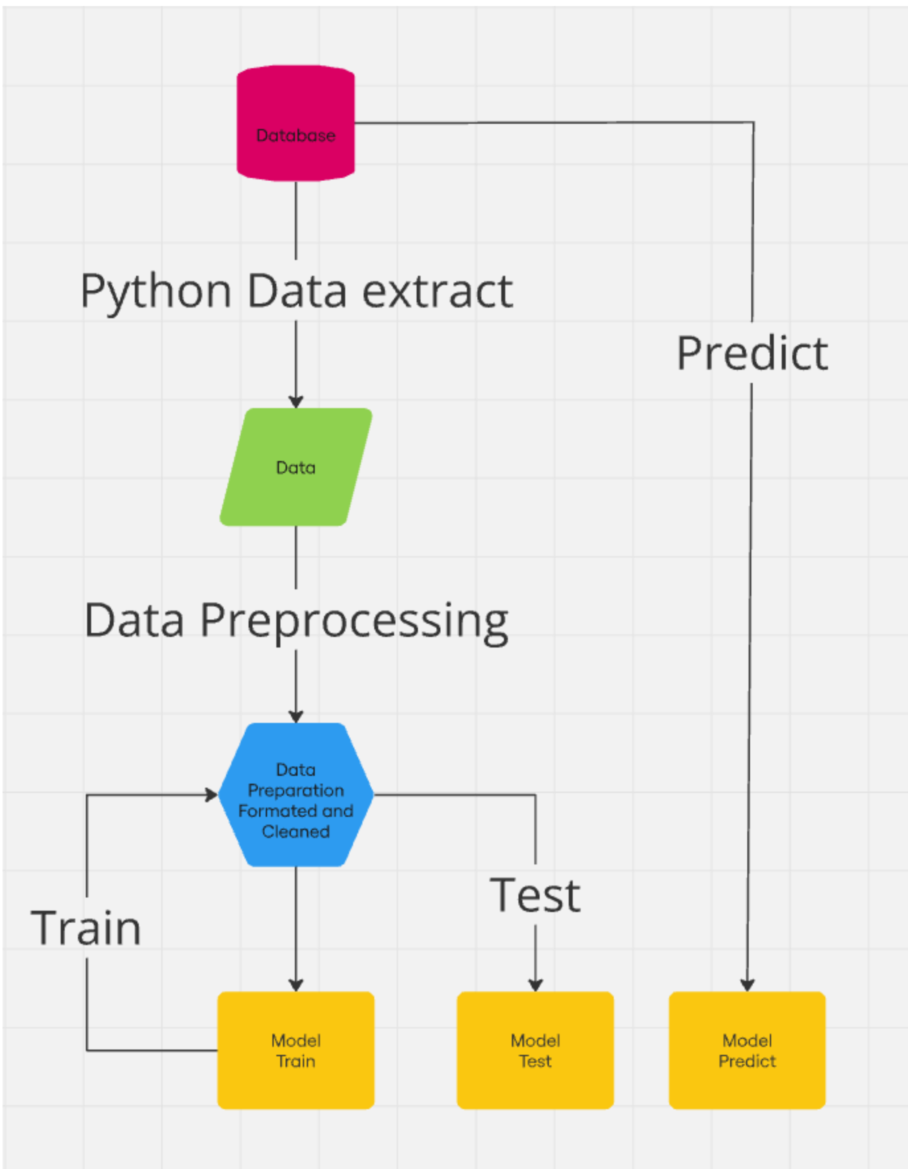


In the Blue Boxes Packages Data are collected by the scanners and written to the Red Database

Data Flow for the project study:

- **Data Source**: Shipment data is extracted from Swiss Post's databases.
- **Data Preprocessing**: Data is cleaned and prepared, with missing values handled appropriately and Invalid or inconsistent timestamps (e.g., negative processing times) are filtered out.
- **Feature Engineering**: Key features such as shipment size, weight, and timestamps are used and Derived metrics like processing_time_minutes are created to evaluate performance.
- **Model Training**: Data is passed to machine learning models to Train and Test
- **Model Outputs**: Data is passed to machine learning models to predict and analyze sorting performance.

    - **the data flow for this project, starting from data collection to model outputs.**



7. Data Model

7.1. Conceptual Data Model

The conceptual data model represents the high-level structure of the data, outlining key entities (or features) and their relationships within the sorting center's operations. For this project, the key entities involve shipment attributes, chute performance, and processing times.

Shipments: The core entity that includes information such as package identifiers, dimensions, weight, and timestamps (arrival and departure times).

Sorting Stations: The stations (or scanners) that process shipments and direct them to chutes.

Chutes: The output of the sorting machine that directs parcels to different destinations based on ZIP codes. Each chute may serve multiple ZIP codes.

Performance Metrics: Data points used to evaluate performance, such as processing time, number of packages handled, and overall throughput for stations and chutes.

7.2. Logical Data Model

The logical data model outlines the data attributes and their relationships without considering the technical implementation. In this project, the dataset consists of the following attributes:

Shipment Attributes:

SND_IDENTCODE: Anonymized unique identifier for each shipment.

SND_CODS_DIM1, SND_CODS_DIM2, SND_CODS_DIM3: Dimensions of the shipment (length, width, height in millimeters).

SND_GEW: Weight of the shipment (in grams).

Timestamps:

CODS_COD_DAT: Timestamp when the shipment was scanned and entered the sorting center.

CODS_LERE_DAT: Timestamp when the shipment left the sorting center.

Sorting Details:

CODS_CO_STATION: The station that scanned and processed the shipment.

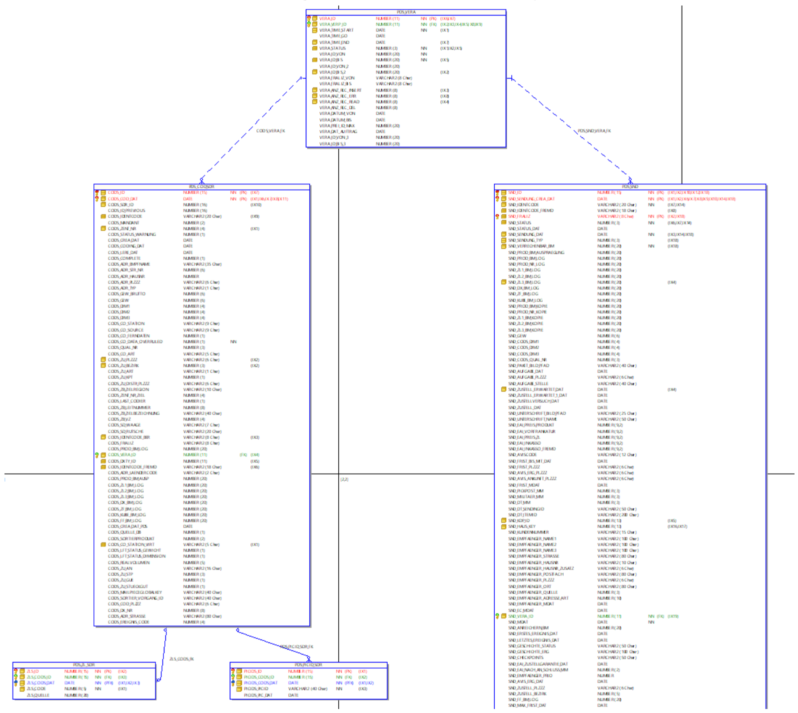
CODS_SD_RUTSCHE: The chute the shipment was sent to for further processing.

Performance Metrics:

processing_time_minutes: Calculated metric, representing the difference between CODS_COD_DAT and CODS_LERE_DAT (time taken for a shipment to be processed, in minutes).

Derived metrics like average processing time per chute, number of packages handled by each chute, and performance statistics for each sorting station.

Database Model showing Tables relationships of the used tables



7.3. Physical Data Model

The physical data model defines how the data is stored and processed. The data is typically managed in tabular form, using tools like pandas (for data manipulation), SQL databases and SQL plus (for querying), and local storage systems for persistent storage.

Storage Infrastructure: The dataset is stored in database Tables , with each row representing a shipment and the associated attributes. The data can be loaded into python data frame and manipulated in memory using Python libraries (e.g., pandas).

7.4. Relationships

Shipment → Sorting Station → Chute: Each shipment passes through a sorting station, which processes it and directs it to a specific chute. The performance of the sorting center is evaluated based on how quickly and efficiently shipments are processed through these entities.

Chute Performance → Sorting Center Performance: The collective performance of the chutes contributes to the overall efficiency of the sorting center. Chutes that are overburdened or congested may affect the performance of the entire center.

8. ML Predictive Model for Chute Congestion (Random Forest)

To proactively manage congestion in the sorting center, a **Random Forest Classifier** was selected as the predictive model to forecast chute overutilization. This model was chosen for its ability to handle non-linear relationships between features and provide feature importance insights.

8.1. Model Selection:

- **Random Forest Classifier**: This model is an ensemble of decision trees that is well-suited for binary classification problems like detecting whether a chute is congested or not.
  - **Advantages**: The Random Forest model is robust to overfitting and handles large datasets efficiently. It can also output feature importance, helping identify which factors contribute most to chute congestion.
  - **Target Variable**: The target for the model was defined as whether a chute was congested based on a threshold of average processing time and package volume.
    -

8.2. Model Training:

- **Data Split**: The dataset was split into training and test sets using an 80/20 ratio, ensuring that the model was trained on a portion of the data and evaluated on unseen data.
- **Model Hyperparameters**:
  - The Random Forest model's hyperparameters, such as the number of decision trees (n_estimators) and maximum tree depth (max_depth), were tuned using grid search cross-validation.
- **Evaluation Metrics**:
  - The model was evaluated using precision, recall, and F1-score, with a particular focus on minimizing false negatives (i.e., instances where a congested chute was not flagged).
- **Feature Importance**:
  - The model provided insights into the most important features driving chute congestion. Features such as the number of packages processed and processing time had the highest importance scores, indicating they played a key role in predicting congestion.

8.3. Model Deployment and Monitoring

The predictive model for congestion is designed to be deployed in real-time, allowing the sorting center to dynamically adjust chute assignments and mitigate potential bottlenecks.

- **Real-Time Monitoring**:

- A real-time dashboard can be built to monitor chute utilization and processing time. The predictive model will flag chutes that are at risk of congestion, triggering proactive operational interventions.

- **Model Retraining**:
    - The model is retrained periodically as new data becomes available to ensure it continues to provide accurate predictions as operational conditions change.

8.4. Result of Statistical Tests

Understanding which features most affect sorting issues provides insights that can be used to improve sorting operations at Swiss Post. Focusing on key factors like shipment size, weight, and station performance will help optimize sorting machine performance and reduce errors.

Key Findings

Chute Congestion: Certain chutes were identified as potential bottlenecks, handling significantly more packages than others and showing longer processing times. Managing chute congestion is critical to improving overall efficiency.

Processing Time Variability: There was substantial variability in processing times across shipments. Factors such as shipment dimensions, weight, and chute assignment contributed to this variability.

Data Quality Issues: Several data quality issues, such as missing or inconsistent timestamps, were identified. These issues were addressed to ensure accurate analysis, but continued data quality monitoring is recommended.

Model Insights

The Random Forest model provided insights into the factors most influencing sorting performance, with shipment weight and chute utilization being significant contributors. However, additional factors not captured in the dataset may also play a role in performance variations.

Recommendations
Chute Balancing: Implement dynamic chute load balancing to distribute shipments more evenly across available chutes. This would reduce bottlenecks and improve throughput.
Real-Time Monitoring: Introduce real-time monitoring to detect and address chute congestion before it affects overall performance.
Further Data Collection: Collect additional data on shipment characteristics and operational factors to refine the performance models and improve accuracy.

Next Steps

Continue refining the performance models with updated data and explore additional machine learning techniques to predict sorting center performance under different conditions. Implement operational changes based on the findings and monitor their impact on sorting efficiency.

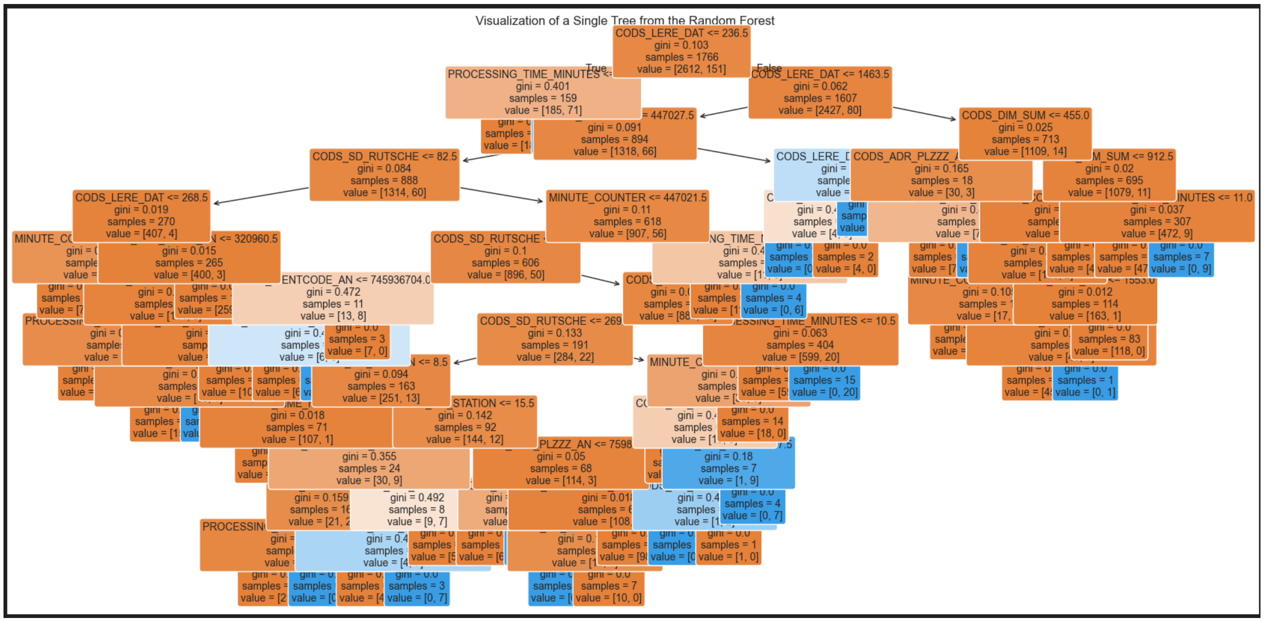
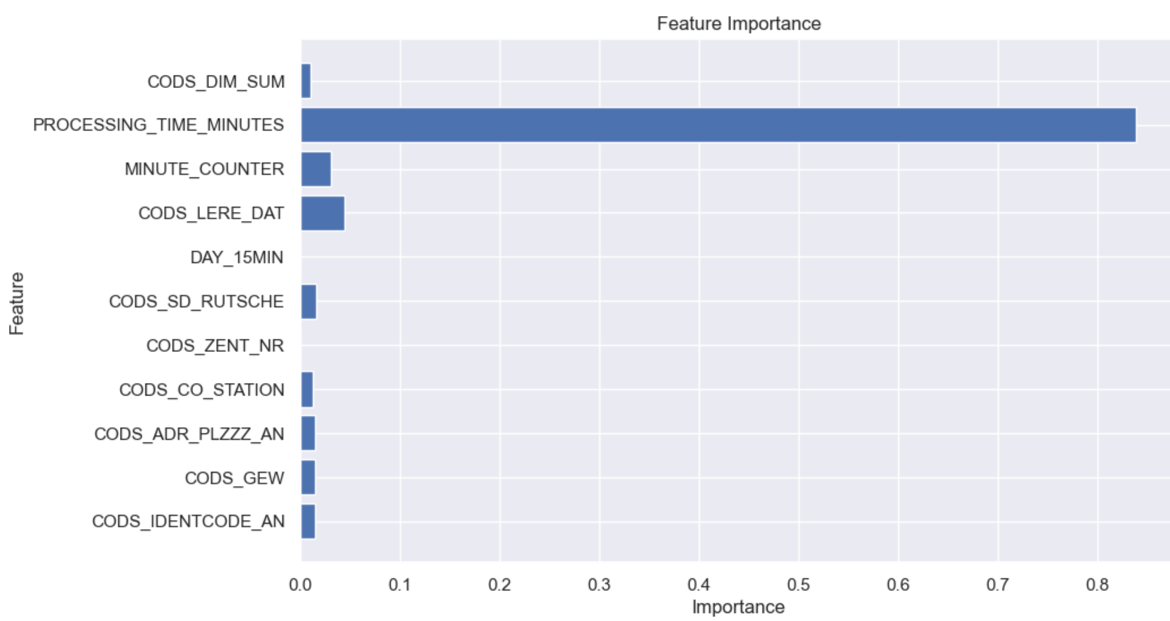


Fig. 10 :An attempt to visualize the decision tree made by our Random Forest



9. LSTM Deep Learning Approach for Predicting Sorting Center Performance Issues

The Random Forest model treated each time step as an independent observation, ignoring the sequential nature of package processing data. As a result, it struggled with forecasting tasks and could not detect leading indicators of congestion. To address this, we adopted Long Short-Term Memory (LSTM) modelling capable of learning from historical sequences and capturing temporal dependencies. As we are dealing with rather large datasets coming from the sorting centers, LSTM seems like a good fit as the model can store and retrieve information even over long sequences.

9.1. Modelling LSTM

An important change that had to be done from the previous Random Forest model was that we now needed to define congestion via the average processing time which we chose to be 10 minutes. Due to the high availability to data from the sorting centers we decided to take different approaches with LSTM. Another learning from the feature importance we applied was that the dimensions including weight and size of the packages only play a minor role for chute congestion and thus should be dropped from the data set to reduce possible overfitting.
Additionally, we decided to use two different preprocessed data sets for the models. A first data set that included data for 30 days of sorting packages and another one that included only 4 days' worth of sorting data.

LSTM model 1:
Included data of 30 days from all chutes which was aggregated on an hourly basis. With this model we aimed to predict the average processing time of the next 6 hours in a 10-minute frequency.

LSTM model 2:
Was based on LSTM model 1 but added two data engineering steps beforehand as it looked at only the top 20 most congested chutes during those 30 days of data collection and the granularity of time was increased from hourly intervals to 10-minute intervals to better reflect real-time dynamics. For the output was a binary classification chosen which would either be congested or a uncongested chute.

LSTM model 3:
This model used data of 4 days over all chutes. The goal of the model was to predict chutes that were going to experience a congestion and predict their expected average sorting times over the next 6 hours. To increase the number of features for this data set we "enriched" it with simulated time and count lags and rolling means. These additional features are based on the previous counts and times 1 and 3 hours ago (lags) and the averages of both 3 and 6 hours ago (rolling means). This brings the number of features from 5 to 15.

9.2. Results and Findings

The transition from the tree-based Random Forest model to the sequential deep learning LSTM model greatly improved the prediction quality. By narrowing the data scope and enhancing temporal resolution and increasing the number of features, the refined LSTM models offer actionable insights for detecting and mitigating performance issues.

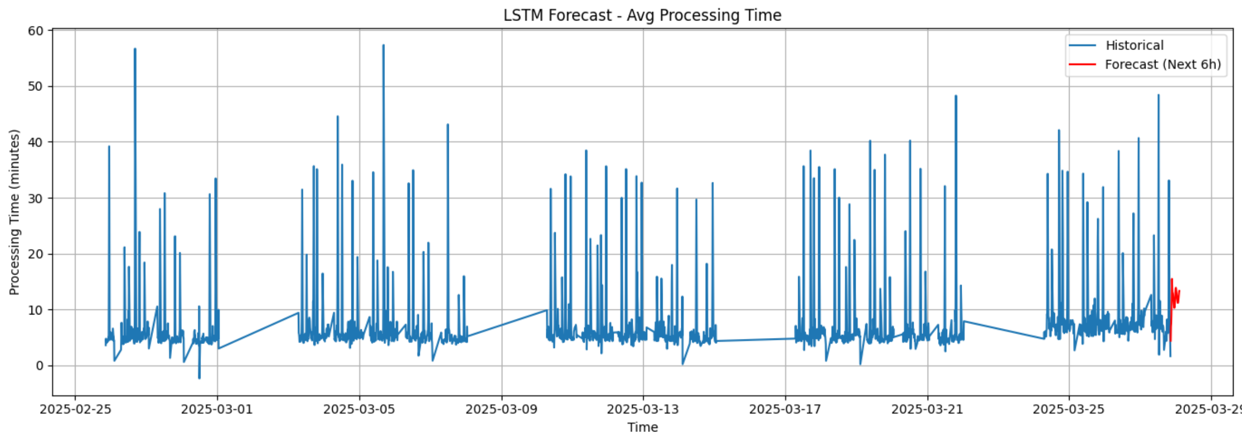


Fig. 13: LSTM model 1 predicting in red the average processing time for the next 6 hours in 10-minute intervals based on the data of 30 days prior.

LSTM model 1 (Fig. 13) predicts an average processing time which seems to vary a bit initially but then does remain around the 10-to-12-minute mark quite consistently. This would not be what we would expect for a full 6 hours, and neither is it something that was previously observed during the 30 days of data collection. As it would mean that for almost the full duration of the prediction congestions for all chutes would be expected. Looking at LSTM model 2, we could see an improvement for peak prediction and less of an 'average' prediction as the previous model did. However, the actual average processing times in this model were incredibly high and these predictions should still be taken with a grain of salt (2LSTM.ipynb in appendix). For our third model, LSTM model 3, we produced a heatmap (Fig. 14) which shows the hourly predicted congestion times for chutes that experience at least one congestion during the prediction period of 6 hours. The result here seems somewhat reasonable but the enrichment of the data with more features through the lag and rolling means could have influenced the outcome of the prediction in a way we are yet unaware.

One major problem while dealing with the LSTM model or modelling in general is that most data shows no issues, meaning that the average processing time was under 10 minutes and thus no congestion can be considered. As a consequence the model tends to overfit to majority class and in turn will most often predict normal behavior or simply predicting the average values, missing performance spikes and produce only limited forecasting accuracy.

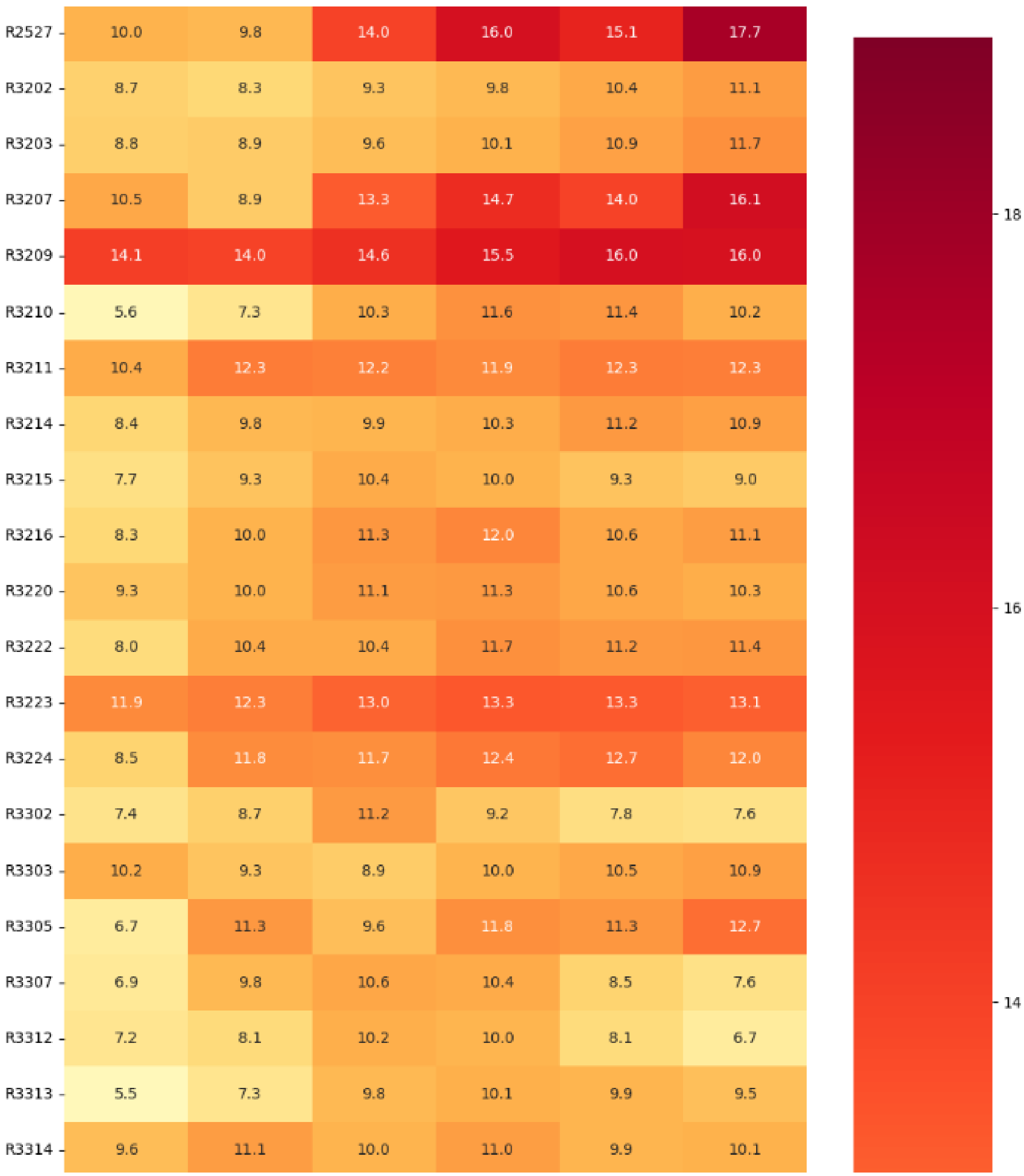
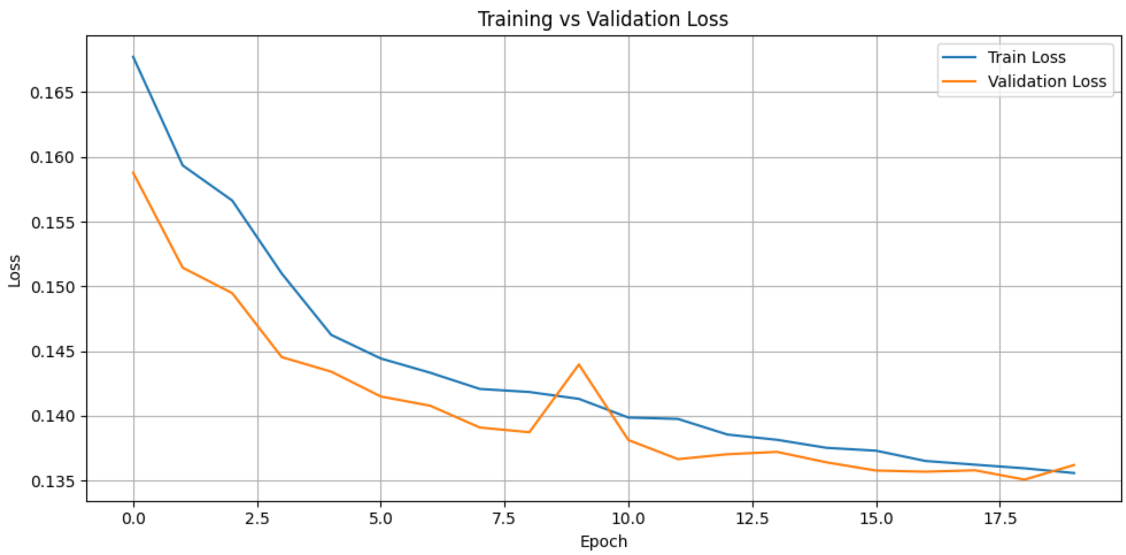
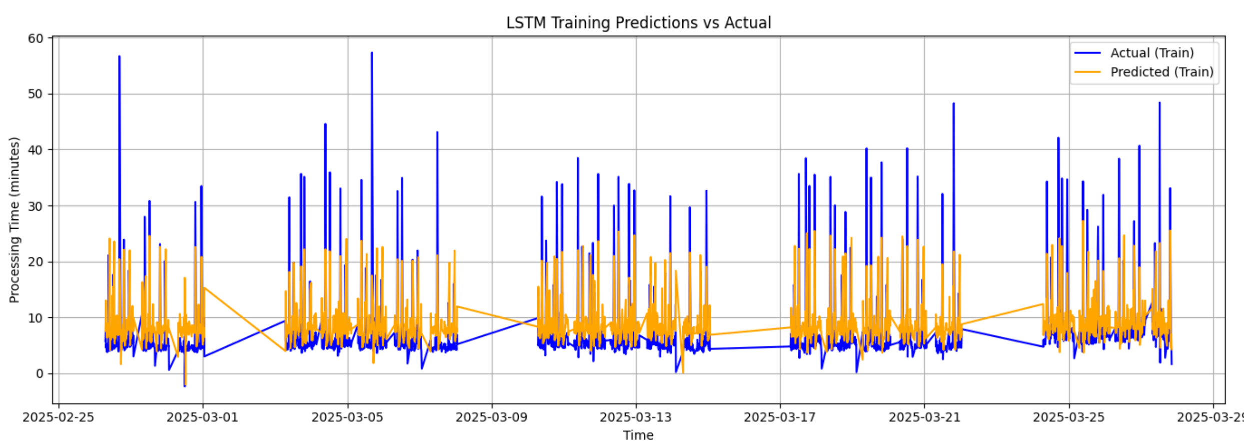


Fig. 14: LSTM model 3, a snippet of the heatmap predicting the average processing times of chutes that experience at least one congestion during the next 6 hours



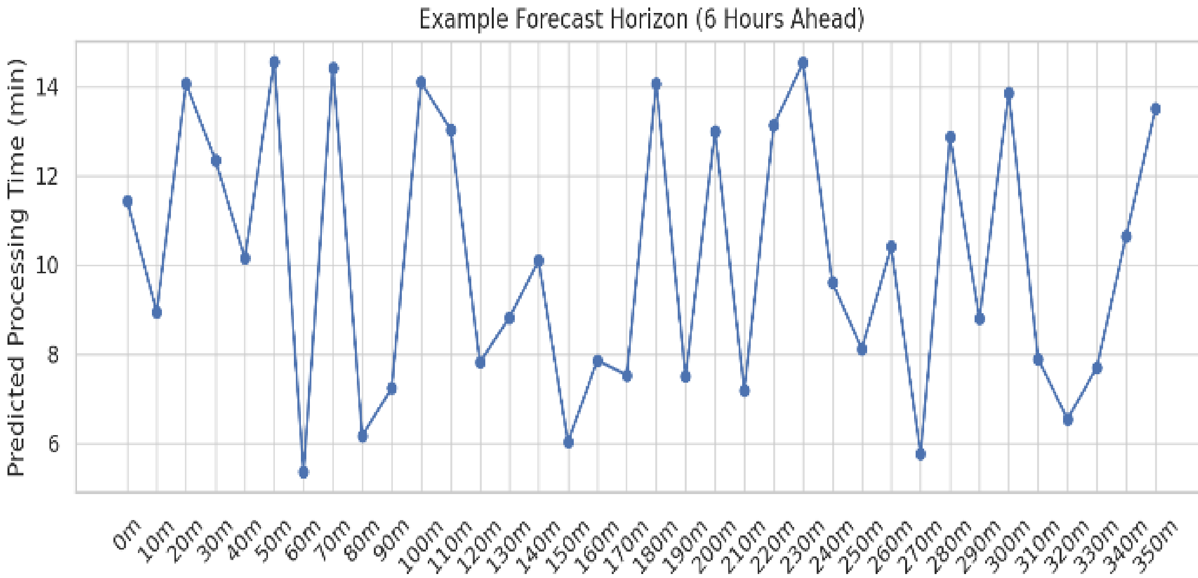
(Training vs Validation Loss):

Both training and validation loss decrease steadily over epochs, indicating the LSTM model learns effectively without significant overfitting, as the two curves remain close.



(LSTM Training Predictions vs Actual):

While the LSTM model captures the overall trend of actual processing times, it tends to underpredict the magnitude of extreme spikes, highlighting the challenge of forecasting rare congestion events.



(Forecast Horizon – 6 Hours Ahead):
The model's forecast for the next 6 hours shows substantial fluctuation in predicted processing times, reflecting the variability and complexity of short-term sorting center operations.

Results Interpretation:
The training versus validation loss plot demonstrates that the LSTM model converges well, with both losses steadily decreasing and remaining close throughout training, indicating minimal overfitting. In the comparison of training predictions versus actual processing times, the model succeeds in tracking general trends but often underestimates the scale of sudden congestion spikes, which reflects the inherent difficulty of predicting rare or extreme events in operational data. The 6-hour forecast horizon further highlights the highly variable nature of processing times, emphasizing both the strengths and limitations of the LSTM model in anticipating rapid changes within the sorting center.

10.Reinforcement Learning (RL) for Dynamic Chute Allocation

To overcome the limitations observed with classical Random Forest models and LSTM-based forecasting in predicting chute congestion at Swiss Post sorting centers. Despite decent performance in detecting delay trends, these models lacked the ability to adapt in real-time and proactively prevent overloads. To address this, we implemented a Reinforcement Learning (RL) approach that learns a dynamic policy to optimize ZIP-to-chute assignment decisions, thereby minimizing sorting delays.

10.1. Data Preparation and Feature Engineering

In a first step we decided to use the historical sorting center data which spanned over four days, based on the LSTM model which we did in the previous chapter. We opted for the enriched data set, which included the additional lag features and rolling average to capture recent trends. As a second step we further artificially enriched the dataset with more congestions to have a dataset that would present a problematic state of the sorting centers performance.

10.2. RL modelling

We developed a custom OpenAI Gym environment which we called 'ZipChuteEnv' which simulates the decision space for ZIP to chute assignment. The environment defines a state (current chute loads, ZIP Code and times), an action (rerouting to a different chute, delaying the sorting process or 'do nothing') and a reward function. A PPO (Proximal Policy Optimization) agent from the `stable-baselines3` library was trained to interact with this environment and learn a strategy to minimize delay. We chose to iterate over a maximum of 50 steps which gives the model to choose 50 opportunities to choose the most fitting action. For the dataset with additional congestions, we ran three different complexity models on the data to have an indicator on if the RL agent improves the average processing time or not. These model were a static approach where a package would always be assigned a default chut, a model with periodic rerouting and the model with our RL agent making a decision.

10.3. Results and Findings

Running our model on the enriched four-day dataset granted us some valuable insights, as the result of 2.88 minutes per step represents a reasonable average sorting time which correlates with the expected values from the dataset which tend to be between 1 to 3 minutes per package if there is no active congestion. Rerouting a package and doing nothing were the only two steps that the RL model chose, where the rerouting action was chosen 42 times out of 50 with an average of 3.41 minutes. The 'do nothing'-action was chosen only 8 times with an average of 2.01 minutes. (Fig. 15)

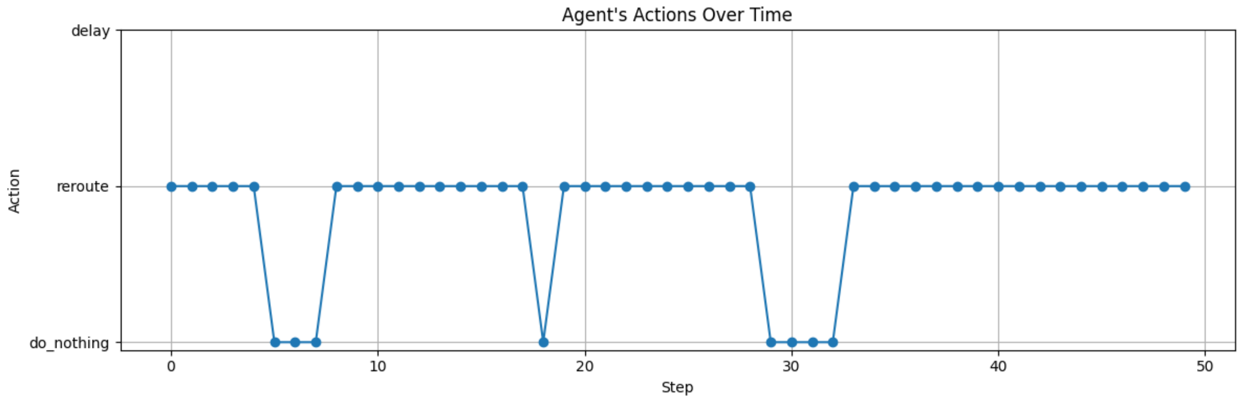


Fig. 15: Actions taken by the RL model for the four-day dataset. Rerouting, despite having a penalty for repeated usage, seems to be the best solution in case of congestions.

The results from the four-day data set enriched with more issues showed us also some interesting results. As shown in the table (Tab. 1) below, our model using the RL agent for decision making improved the average processing time by quite a margin and significantly reduced the number overload events.

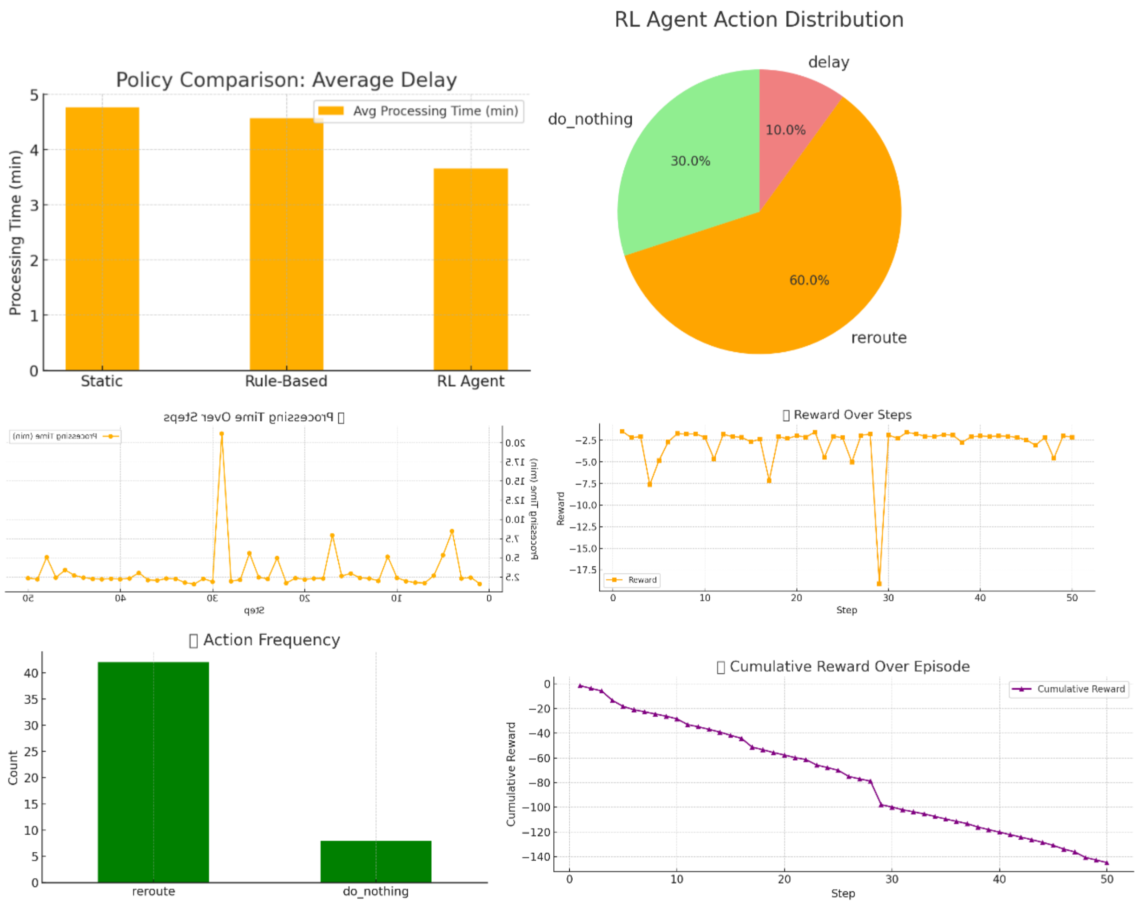
Policy	Avg. Processing Time (min)	Overload Events (>6 min)	Total Episode Reward
Static	4.77	5	-239
Rule-Based	4.57	5	-229
RL Agent	3.66	2	-183

Tab. 1: Comparison of the three models using the four-day data set with increased frequency in congestions.

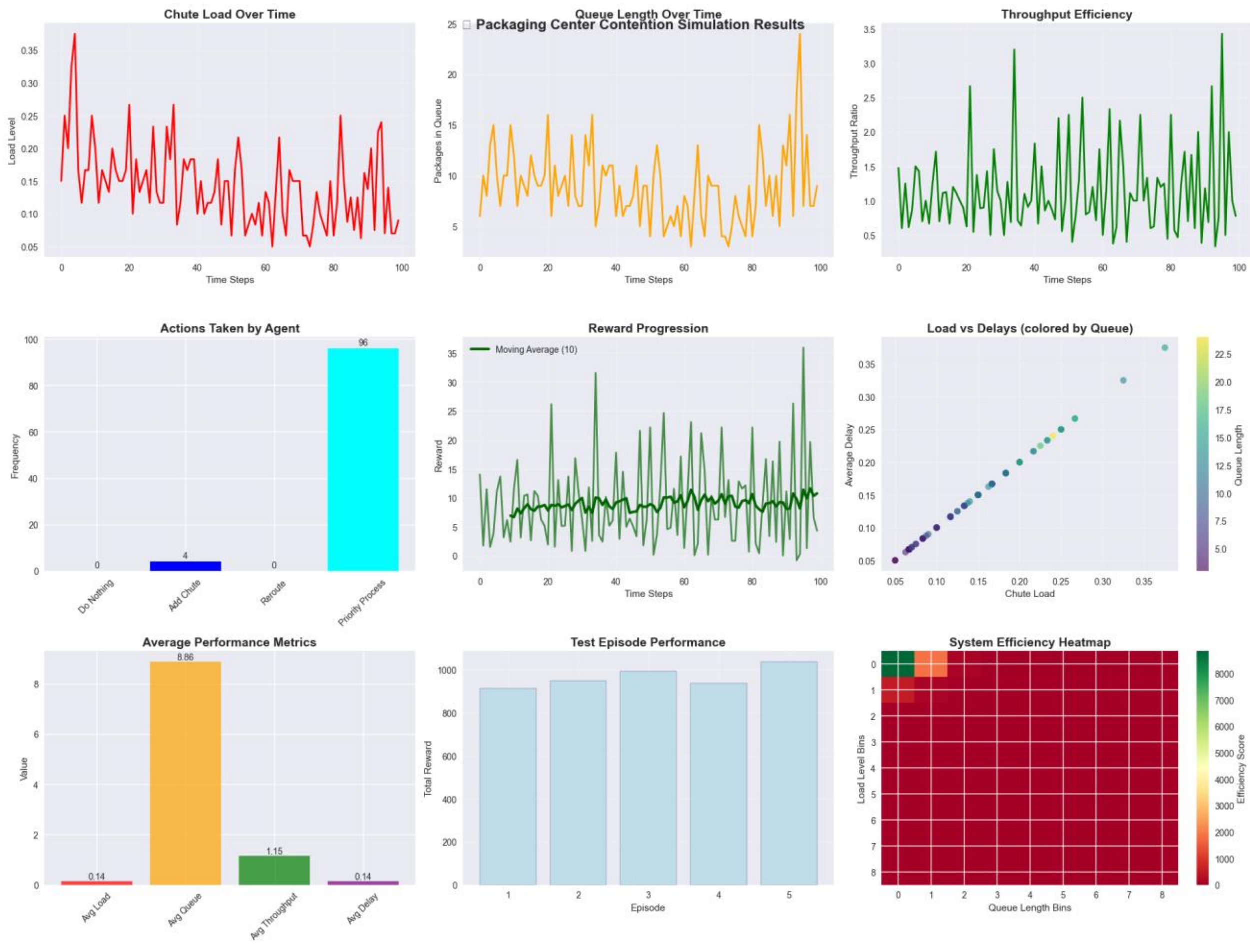
By shifting from predictive models (Random Forest and LSTM) to a decision optimizing framework (RL), we enabled real-time corrective actions instead of reactive forecasting.

10.4. Conclusion

By shifting from predictive models (Random Forest, LSTM) to a decision-optimizing framework (RL), we enabled real-time corrective actions instead of reactive forecasts. The RL agent learned to reroute packages intelligently, reducing average delay and improving sorting throughput. This approach aligns closely with Swiss Post's need for operational efficiency and makes the model directly actionable.



The RL agent significantly reduces average processing time compared to static and rule-based policies, as shown in the policy comparison bar chart. Action distribution plots reveal that the RL agent predominantly chooses to reroute parcels to mitigate congestion, with "do nothing" and "delay" actions used less frequently. Reward curves and cumulative reward trajectories indicate consistent, incremental improvements in performance over steps. Overall, the RL approach enables dynamic, data-driven responses to operational bottlenecks, directly translating into more efficient parcel processing in the sorting center.



SIMULATION SUMMARY: Total Steps Simulated: 100,Average Load Level: 0.143,Average Queue Length: 8.9 packages,Average Throughput: 1.146,Average Delay: 0.143 hours,Total Reward Earned: 902.31,Peak Load Reached: 0.375,Maximum Queue Length: 24.0 packages,Agent Action Distribution: Do Nothing: 0 times (0.0%), Add Chute: 4 times (4.0%), Reroute Packages: 0 times (0.0%) Priority Processing: 96 times (96.0%)

The simulation demonstrates that the RL agent maintains low chute load and queue lengths over time, with high throughput and system efficiency. The agent overwhelmingly prefers “priority processing” actions, rarely adding chutes or rerouting, which leads to minimal delays and stable performance across test episodes. The efficiency heatmap and performance metrics confirm that, under the agent’s policy, peak congestion is quickly resolved, and average delays are negligible. Overall, these results highlight the agent’s effectiveness in optimizing package flow and preventing bottlenecks within the sorting center.

11. Results Discussion

The analysis conducted in this project, from traditional machine learning models to reinforcement learning (RL), yielded insights into operational bottlenecks and performance risks in Swiss Post sorting centers. The results suggest that while models like Random Forest can offer useful static predictions, they fall short in addressing real-time operational dynamics.

The use of Long Short-Term Memory (LSTM) models improved temporal forecasting but was still limited in proactively reducing delays. RL provided the greatest value by suggesting optimal actions under complex load conditions. The agent learned to reroute packages and prioritize processing, significantly reducing overload situations.

Despite promising results, some uncertainty remains. For instance, the agent's performance is tied to the accuracy of simulated environments. The augmented dataset, which includes ZIP surges and chute overloads, better reflects real-world conditions, but unexpected variations in unseen environments could affect results. Performance metrics such as processing time, reward evolution, and throughput highlight trends but may vary under different volumes or operational policies.

12. Conclusion and Outlook

This project demonstrates that a data-driven approach can substantially improve performance and reliability at logistics centers like those operated by Swiss Post. By evolving from static models to time-aware and decision-optimized approaches, we developed an actionable methodology for real-time congestion management.

12.1. Key Findings

Chute congestion was a consistent indicator of degraded performance.
- Real-time processing times were impacted by package volume, load imbalance, and inconsistent ZIP-to-chute routing.
- Reinforcement learning significantly reduced the average delay when trained on realistic scenarios.

12.2. Model Insights

The RL model dynamically improved routing by adjusting to unseen demand surges, in contrast to static rule-based systems. This adaptability illustrates how machine learning can shift from prediction to control when data complexity requires it.

12.3. Recommendations

Expand real-time data feeds to improve RL agent training and validation.

- Deploy hybrid LSTM + RL models to combine foresight with responsive control.
- Integrate this solution as a decision support tool for control rooms.
- Improve ZIP-to-chute mapping rules to avoid future overload scenarios.

12.4. Next Steps

The next phase includes production testing of the RL agent, collecting new sensor data, and integrating feedback loops into the operations dashboard. More advanced methods such as Graph Neural Networks (GNNs) and causal inference may also be evaluated for cross-station policy transfer and explainability

1 Acknowledgements

Acknowledge is due for Jonathan Hueni from LS75.3 his input information made it possible for me to reach this point of understanding and trying to develop a solution according to his guides

References and Bibliography

[1] Swiss Post Hueni Jonathan, LS75.3 internal non sensitive Dokumente

LS75.3-05 AT Konzeption und Entwicklung>20_Datenanalysen>**15_input_analysis**

[2] Swiss Post Annual Report
https://geschaeftsbericht.post.ch/23/ar/app/uploads/EN_Post_Jahresbericht_2023.pdf

[3] Swiss Post Financial Report

https://geschaeftsbericht.post.ch/23/ar/app/uploads/EN_Post_Finanzbericht_2023.pdf

[4] Parcel and Postal Technology International

https://www.parcelandpostaltechnologyinternational.com/news/automation/mapping-optimizes-sorting-centers.html