Journal of Computer Science

Original Research Paper

An Adaptive Data Preprocessing Framework for Improved Learning: A Case Study of Tangier Container Terminal

[1]Mostafa Al Uahabi, [1]Hicham Attariuas, [1]Mohammed Saleh and [2]Mohamed Chentouf

[1]Department of System Engineering, Faculty of Sciences, Abdelmalek Essaadi University, Tetouan, Morocco
[2]Calypto Synthesis Solutions, Siemens Digital Industries Software, Rabat, Morocco

Article history
Received: 19-06-2023
Revised: 15-10-2023
Accepted: 28-11-2023

Corresponding Author:
Mostafa AL Uahabi
Department of System Engineering, Faculty of Sciences, Abdelmalek Essaadi University, Tetouan, Morocco
Email: Mostafa.aluahabi@etu.uae.ac.ma

Abstract: Container terminals are critical nodes within the maritime transportation system that have a vital function in global merchandise trade, handling a significant volume of cargo through the use of various equipment and personnel. Thus, the efficiency of container terminal operations relies heavily on the ability to collect, analyze, and utilize operational data. However, such data can be corrupted by noise, missing points, outliers, and incomplete or inconsistent information, making subsequent analysis or modeling challenging. This study proposes an adaptive data preprocessing framework tailored to the context of container terminal operations, using data from tangier container terminal as a case study, the leading container port in the Mediterranean and Africa, and also ranked 4th in the CPPI 2022. This framework includes techniques for data integration, cleaning, transformation, and encoding to acquire high-quality data. In addition, the RFE feature selection method is employed to identify the most discriminative feature subset. Finally, the proposed approach, assessed using an extra tree regressor model, demonstrates strong prediction capabilities with an R-squared score of 95.4% based on the selected features for predicting the duration of vessels at port, highlighting that its integration into the terminal operating system can improve management efficiency.

Keywords: Data Preprocessing, Container Terminal Operation, Extra-Trees Regressor, Duration at Port Prediction, Feature Selection

Introduction

The ever-expanding reach and refinement of maritime transport have given rise to a surge in the amount of data generated by container terminal traffic. Achieving optimal predictive capabilities and crafting reliable decision support systems and models necessitates a thorough analysis of this data, which, in turn, requires the implementation of fitting preprocessing techniques and methods.

Data preprocessing is a crucial and necessary phase in Machine Learning (ML) that must precede any modeling and analysis. Poor data quality is a common reason for failure in many ML and AI projects. According to the white paper (Bowes, 2015), 77% of companies assume that low-quality data leads to poor results, which may include incomplete, noisy, inconsistent, inaccurate, missing, and high-dimensional data. To ensure high-quality data, proper data preparation is necessary and as such, data scientists often dedicate a considerable portion of their time (about 80%) to cleaning and preparing data before analysis (García *et al*., 2015; Press, 2016).

The process of data preprocessing involves employing techniques to transform raw data into a format suitable for building and training ML models. The primary aims of data preprocessing are to enhance data quality and make the dataset more appropriate for giving the learning models the ability to learn accurately and independently from unbiased data, thereby producing accurate results that may affect the performance of the models (Karagiannidis and Themelis, 2021).

To the best of our knowledge, after conducting an exhaustive search across multiple electronic databases, this article is the first to present a comprehensive and adaptive framework for data preprocessing specifically tailored to the container terminal operation context. Additionally, we employ feature selection to identify the most relevant attributes for a predictive module that forecasts the duration of vessels' stay in the port. Our dataset is sourced from the tangier container terminal, the Mediterranean and Africa's leading container port, ranked 4th in the CPPI 2022.

Science Publications

© 2024 Mostafa Al Uahabi, Hicham Attariuas, Mohammed Saleh and Mohamed Chentouf. This open-access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

Data preprocessing is an inevitable phase in Machine Learning (ML), data mining, and other data-driven applications since the input data quality directly affects the accuracy and reliability of the output results. It encompasses a group of techniques that aim to enhance the quality of raw data and transform it into a more manageable and helpful format. Various research studies have been conducted on data preprocessing techniques in diverse fields, where it has been shown that choosing the appropriate techniques for a particular application can significantly improve the quality and efficiency of data analysis. This section examines several studies that investigate preprocessing techniques used in various fields, including the logistics and port industries, as well as studies of several prediction models.

A complete collection of data preprocessing techniques was provided by García *et al*. (2015), highlighting the gaps in real data caused by various factors, along with the most relevant proposed solutions. In addition, García *et al*. (2016); Prakash *et al*. (2019) introduced detailed data preprocessing methods for data mining in the context of big data. The selection methodology of techniques has been extensively discussed by Han *et al*. (2012); Subasi (2020) to help researchers choose the appropriate techniques for data analysis.

Alasadi and Bhaya (2017) provide a thorough review of various data preprocessing methods employed for data mining, which can be a useful resource for those seeking to select the most appropriate data preprocessing techniques for their specific application. In their study, Alexandropoulos *et al*. (2019) present an overview of different data preprocessing techniques in predictive data mining and their impact on the accuracy of the results.

Frye and Schmitt (2020) present a framework for a structured and reusable data preprocessing pipeline tailored for ML applications within a production context. In their subsequent study Frye *et al*. (2021), they present a structured data preprocessing approach designed for production use case requirements, assessing these methods based on their influence on ML model performance.

Muresan *et al*. (2015) applied several data cleaning and selection methods in the medical field, which significantly improved classification performance. Mohd *et al*. (2013) proposed a data preparation methodology to transform raw clinical data from diverse sources into a well-prepared clinical dataset. Pérez *et al*. (2015) presented a methodology for data preparation that consists of a general part and a specific part oriented for an epidemiological domain, followed by a data mining system performed on real mortality databases.

Ramírez-Gallego *et al*. (2017) reviewed data preprocessing techniques in stream data mining, covering existing methods and open challenges. A method for automating and simplifying data preprocessing tasks is presented by Bilal *et al*. (2022) This approach provides interactive, data-driven support by identifying data issues, recommending suitable preprocessing techniques, and offering valuable insights. The evaluation confirms its effectiveness in streamlining preprocessing and enhancing model performance. Al-Taie *et al*. (2019) evaluate the effectiveness of online data preprocessing techniques on data quality and machine learning performance, providing insights for selecting appropriate methods.

Furthermore, Marco *et al*. (2021) focus on improving data preprocessing for software effort estimation, addressing challenges related to missing data and irrelevant features in categorical variables. The research highlights the efficacy of a novel approach for improved accuracy in this domain. Karagiannidis and Themelis (2021) discuss the application of data-driven modeling to predict fuel consumption and speed loss in the port industry. It also investigates the impact of preprocessing techniques on the accuracy of prediction models for these variables. Their study revealed that implementing appropriate techniques can significantly enhance the accuracy of the models. Wang *et al*. (2020) used ML and deep learning algorithms to predict the number of vessel arrivals and duration at port for container barges exclusively. They also used forward pass search of wrapper methods to select the best set of features.

As data continues to grow in volume, complexity, and heterogeneity, there is an increasing interest in data preprocessing. Despite this interest, the literature reveals some gaps. Studies in the logistics and the port industry have been primarily fixated on predictive modeling, often sidelining the realm of data preprocessing. There's also a noticeable absence of research that uniquely adapts preprocessing techniques to the specificities of logistics and port operations, which is pivotal for elevating data quality and machine learning outcomes.

In line with this, our study focuses on tailoring data preprocessing techniques to suit the requirements of container terminals. Our objective is to create a context-specific preprocessing guide to enhance data quality. Additionally, we will identify relevant features and develop a model to predict the duration at the port of the vessels using the preprocessed data to validate our approach.

Materials and Methods

Research Area and Data

The focus of this study is the Tangier Med Port, strategically located at the convergence of the Atlantic Ocean and the Mediterranean Sea on the Strait of Gibraltar. Boasting a handling capacity of 9 million Twenty-foot Equivalent Unit (TEU) containers, the port managed 7.6 million TEU containers in 2022. This remarkable achievement ranks it as the leading container port in both Africa and the Mediterranean. Furthermore, according to the "Container Port Performance Index" (CPPI) report from 2022, it stands as the fourth-largest globally (Bank, 2023).

The data utilized in this study originates from TC2, which became operational in August 2008. This data, housed within a relational database, is distributed across various tables and can be accessed only with management's authorization. To delve deeper into the data's quality and content, an analysis phase was undertaken. Additionally, planners were interviewed to gather insights into the challenges they encountered.

Data Analysis

The dataset under scrutiny offers a detailed view of vessel movements at TC2 over twelve years. Each record captures specifics related to a vessel and its journey. The original dataset comprises 13,796 instances, each endowed with 115 attributes.

Transforming this comprehensive data into a more refined format primarily enables efficient quality control and error analysis. This transformation can potentially shed light on existing constraints within the system.

What sets our study apart from preceding research is the vastness and depth of the dataset. This not only highlights the scalability of our preprocessing methods but also underscores their practical application.

Adaptive Data Preprocessing Framework

Effectively managing and analyzing data is a substantial task, particularly in the context of container terminal operations, where the quality of data is of paramount importance (Karagiannidis and Themelis, 2021). Therefore, the meticulous choice of preprocessing methods is a critical consideration to ensure the resulting dataset's quality.

The initial step in preprocessing is to identify the specific requirements of the use case and establish the criteria for performing an initial data quality check. Subsequently, various techniques are used to convert raw data into a refined, high-quality format for further analysis or modeling (Alasadi and Bhaya, 2017). A flowchart of the data preprocessing procedure is depicted in Fig. 1.

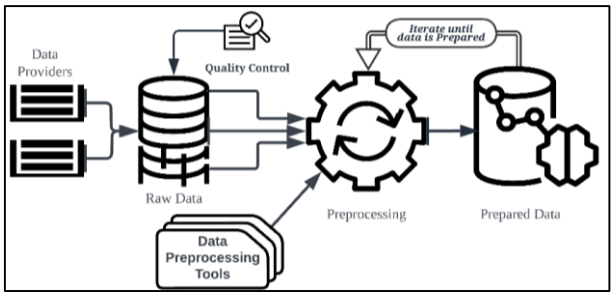

Fig. 1: Data preprocessing in the ML process

Our framework tailors its preprocessing approach to the specific context of container terminal operations and highlights its role in enhancing prediction quality through a predictive model. It is designed to adapt to the distinctive data characteristics of container terminals. This adaptive framework can identify specific data patterns, inconsistencies, and gaps and recommend the most suitable preprocessing methods, including their sequence and parameterization, ensuring adaptability throughout the model's lifecycle.

In the following subsections, we provide a detailed examination of various data problems encountered and their potential solutions.

Data Integration

This process involves combining data from multiple sources and merging them into a single dataset (Al-Taie *et al.*, 2019). The initial step is schema integration and object matching, which integrate data from distinct tables into a single dataset while maintaining entity identification through the common attribute key. In addition, data integration includes two other facets.

Redundant attribute detection: This step identifies attributes that can be derived from other attributes (Han *et al.*, 2012). Measures like correlation and covariance coefficients are used to assess the strength of implication between attributes. For categorical variables, the χ^2 (chi-square) test is commonly applied (Subasi, 2020). In our analysis, we identified three redundant attributes with different names using a correlation coefficient and deleted one of them.

Duplication tuple detection: Redundant instances can appear in the data, leading to inconsistencies between duplicates (García *et al.*, 2015). To avoid this, full data scanning and scrubbing tools in the Pandas library are used to identify and eliminate duplicates.

Data Cleaning

This process involves correcting inaccurate data, filtering incorrect data, and reducing unnecessary details (Al-Taie *et al.*, 2019; García *et al.*, 2015). It aims to ensure the data's consistency, validity, and accuracy (Han *et al.*, 2012). While some researchers include missing data and anomalies in this step, we'll address them separately for a more detailed analysis.

At this stage, some features may initially appear irrelevant but could be crucial for the company's operations. We begin by eliminating constant features (zero-variance variables) that offer no useful information. Subsequently, quasi-constant features with high similarity among most observations are removed using a variance threshold of 0.05, as they add little or no value to the analysis. Moreover, empty columns that have no values for any of their rows are removed (Fig. 2).

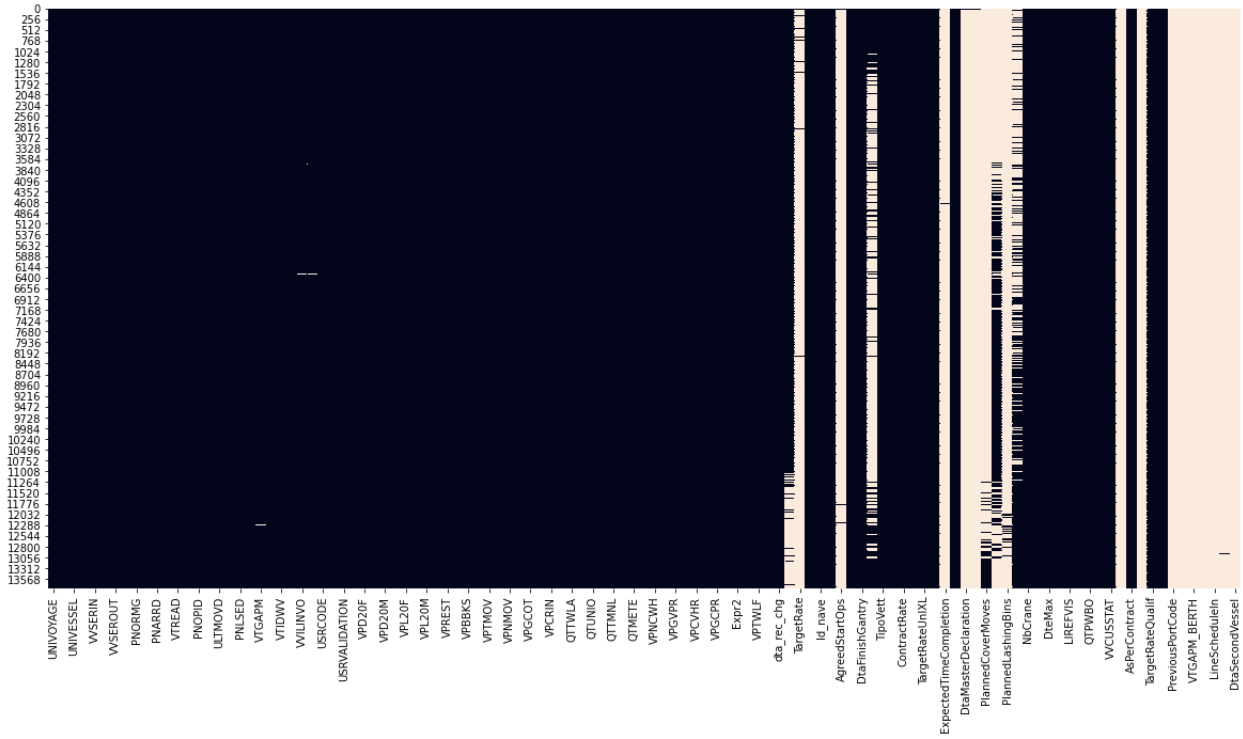

Fig. 2: Representation of missing values in the dataset

Table 1: Types of missing data mechanisms

Label	Description
Missing completely at random-MCAR	There is no relationship between missing data and other values in the dataset; the probability of missingness is the same for all cases
Missing at random-MAR	Missingness can be explained by variables with full information; the probability of missingness is the same only within groups defined by observed data
Missing not at random-MNAR	Missingness is specifically related to what is missing, with the probability of missingness varying for unknown reasons

Primary key and unique fields, crucial for data integrity, may not be needed for analysis or prediction and can be excluded from the feature set used for modeling.

Handle the Missing Data

Missing data refers to a value for an attribute that was not entered or was lost during the recording process, often caused by manual data entry procedures, equipment errors, or incorrect measurements (García *et al*., 2016). It's classified into three mechanisms: MCAR, MAR, and MNAR (Table 1). The approach for dealing with missing data depends on the assumptions made about the mechanics (Little and Rubin, 2002).

Dealing with missing data is a complex task and while there is no perfect solution, several strategies are available (Farhangfar *et al*., 2007):

- Discarding instances containing missing values
- Using maximum likelihood to estimate the parameters of a model for the full portion of data and then using the model for imputation by means of sampling
- Imputing missing values by identifying relationships between attributes

Although removing instances with missing values is the simplest method, it can lead to biased results or a loss of information (Alexandropoulos *et al*., 2019; Little and Rubin, 2002). Therefore, imputation is often used to establish a statistical relationship between the missing data and the other instances (tuples) in the dataset (Prakash *et al*., 2019).

Imputation methods are generally more suitable for randomly occurring missing values (Farhangfar *et al*., 2007). For missing values generated by an NMAR mechanism, additional information or expert knowledge is needed for imputation. The following two subsections describe how missing values were handled in our study.

Deletion of missing values: In this process, instances and attributes with high levels of missing data are removed. Deletion is advisable when the level of missing data is substantial (Frye and Schmitt, 2020), especially when there are enough instances or attributes to prevent significant information loss. Figure 2, missing values are indicated as empty space, and attributes exceeding a 90% missing data threshold are removed, as they can distort data and lead to inaccurate results. Some experts may opt for a threshold value of 60 or 70%.

Imputation of missing values: Dataset attributes are often interdependent, making imputation a useful technique to estimate missing values based on available data. Little and Rubin (2002) distinguished between 2 imputation methods: (i) Simple imputation, which fills in the missing value with a single value (mean, median, mode, or using an ML algorithm), and (ii) Multiple imputation, which generates multiple plausible values for each missing data point based on a statistical model. The values are drawn from a probability distribution representing the uncertainty around the true value of the missing data point.

For our study, different imputation scenarios were used depending on the situation of each feature. The 'lashingcare' feature had 70% missing values and four possible categories ('egt', 'vsl', 'fcc', 'vegt'). It was convenient to fill these values by creating a new category 'unknown'. Similarly, we filled two features, 'is general cargo only' and 'is spot call', which had almost 50% missing values, with 'N', assuming that the missing values should be 'Y'. Next, features 'PNORMG' and 'USRCODE' with less than 1% missing values were replaced with the most frequent categories.

Lastly, the widely-used KNNImputer model was applied to the numerical feature 'QTCOVE,' which had about 1% missing values. This imputation technique predicts missing values by observing trends in related features, making it a reliable choice (Muresan *et al*., 2015; Marco *et al*., 2021).

Anomaly Data Detection (Outlier)

Outliers are data points that deviate significantly from the norm in a dataset. Anomaly detection aims to identify patterns in the data that do not conform to expected behavior (Chandola *et al*., 2009). Outlier treatment and noise removal, although distinct, are related methods for handling undesirable noise in data, with outlier removal potentially resulting in decreased noise in a dataset (Prakash *et al*., 2019).

Outliers might emerge from various sources, such as measurement variability and misinterpretation of data inputs (García *et al*., 2015). Their presence can affect the measures of central tendency and variability, which can impact the results of the analyses. Generally, outliers can be handled in similar ways to missing data, i.e., ignored, removed, or imputed.

In investigating anomaly detection, we experimentally appraise four common techniques to conduct a comparative analysis of their effectiveness.

Box plot: A simple parametric statistical technique used to detect outliers in univariate and multivariate data assumed to be generated from a Gaussian distribution (Chandola *et al*., 2009). It divides the data into quartiles and uses an interquartile range to define outliers. Outliers fall outside the range:

$$x \notin [Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR] \quad (1)$$

Z-score: A parametric technique used to detect outliers in one-dimensional or low-dimensional feature spaces, assuming the data is Gaussian distributed. Outliers are data points in the tails of the distribution far from the mean and a threshold (typically set to 2.5, 3.0, and 3.5) is set for the normalized data points z_i to determine if they are outliers. The z-score is calculated as follows:

$$z_i = \frac{|x_i - \mu|}{\sigma} \quad (2)$$

Any z-score greater than 3 is considered an outlier since most data lies within 3σ above or below the mean.

One-class-SVM: A one-class classification method used for anomaly detection. It learns a boundary that contains the instances of the training data and identifies the smallest hypersphere in kernel space that includes all the training instances (Chandola *et al*., 2009). Any test instance outside the hypersphere is considered anomalous. To learn complex regions, special kernels such as the Radial Basis Function (RBF) kernel may be utilized.

DBSCAN: Is a powerful density-based clustering algorithm for detecting outliers, capable of finding an optimal number of clusters with arbitrary shapes in a dataset (Ester *et al*., 1996). It only needs two user-defined parameters: Epsilon (neighborhood distance) and minpts (minimum number of points) and distinguishes data points as core points, border points, or outliers (Çelik *et al*., 2011).

Compared to other methods, DBSCAN makes minimal assumptions about clusters and does not need to know the expected number of clusters in advance. In addition, it is efficient and robust in the presence of noisy data.

Outlier treatment predominantly relies on the analytical aim and data context. In extensive datasets, removing outliers is common due to their potential to skew analyses and affect results' precision.

Through a comparison of the different techniques, statistical techniques are used to detect outliers that fall above or below a specified threshold, while outliers can also include infrequent data. Additionally, these techniques assume that the data conforms to a particular distribution and they are incapable of detecting interactions between different attributes in the case of multivariate data.

Table 2: Comparison of the results of the four techniques

Technique	Hyper parameter	Number of outliers
IQR	1.5× IQR	8499
Z-Score	Threshold = 3.5	1788
OCSVM	kernel = 'rbf', gamm a = 0.01, nu = 0. 03475	3563
DBSCAN	eps = 3, min_samples = 25	1329

Table 3: Categorical attributes of the dataset

Attributes	Nb categories	Example
PNORMG	4	['A3' 'A2' 'A1' 'SH']
USRCODE	15	['AZAOUDI' 'IESSAIDI' 'DANIELEC' 'STAT_']
SAILDIR	5	['STD' 'NB' 'WB' 'EB']
LashingCare	5	['Egt' 'vsl' 'unknown' 'vegt']
InSchedule	2	[False True]
TipoVet	2	['OV' 'FD']
IsSpotCall	2	['Y' 'N']
IsGeneralCargoOnly	2	['Y' 'N']

The DBSCAN algorithm stands out as it can identify outliers that are not necessarily extreme values, with fewer parameters, while having the ability to find clusters of arbitrary shapes. On the other hand, OCSVM is efficient in high-dimensional space but requires careful hyperparameter tuning and may not be suitable for analyzing large datasets.

Table 2 contrasts outlier detection outcomes across four techniques, revealing varied efficacy with IQR identifying the most outliers (8499). Z-score and OCSVM identified 1788 and 3563 outliers, respectively. Conversely, DBSCAN identified the fewest outliers (1329), implying a tendency to incorporate more points into clusters. These significant variations among techniques underline the necessity of choosing techniques that align with the dataset's characteristics and the overarching research objectives.

Data Normalization

Also known as standardization, this process aims to scale data, ensuring all attributes have equal weight and utilize a common scale or range (Alasadi and Bhaya, 2017; Mohd *et al*., 2013). Three principal techniques are widely used for data normalization (Alexandropoulos *et al*., 2019; Patro and Sahu, 2015).

Z-score normalization (or standardization): Converts the values of an attribute to a mean of zero and a standard deviation of one, so that the distribution of the attribute is centered around zero:

$$A_i' = \frac{A_i - \bar{A}}{\sigma_A} \quad (3)$$

where, A_i' is a standardized value of the original value A_i from attribute A and $\bar{A}$, σ_A are the mean and standard deviation of attribute A, respectively.

Min-max normalization: It scales the values of an attribute A to a specified range (C, D).

$$A_i' = \frac{A_i - minValue\ of\ A}{maxValue\ of\ A - minValue\ of\ A}(D - C) + C \quad (4)$$

Usually, "normalization" refers to a specific form of this technique, where the resulting range is [0,1].

Decimal scaling: This method scales numerical values in a common range by moving the decimal point of the values so that the maximum absolute value is always less than 1:

$$A_i' = \frac{A_i}{10^j} \quad (5)$$

where, j is the smallest integer such that max ($|A_i'|$)<1.

The choice of normalization method depends on the dataset being transformed. Standardization is advisable in the presence of outliers and heterogeneity, as it mitigates their effects through centralization (Han *et al*., 2012). Furthermore, many ML estimators require standardized data (Karagiannidis and Themelis, 2021; Subasi, 2020). Thus, we adopt z-score normalization for transforming our dataset.

Categorical Data Encoding

Qualitative attributes pose a challenge for ML algorithms that require numerical inputs. However, they often contain useful information that can lead to better performance (Potdar *et al*., 2017). Two types can be distinguished: Nominal and ordinal. Several encoding methods exist to deal with this problem, including "deterministic" methods such as ordinal coding, code counting, one-hot encoding, hash-encoding, target encoding, and leave-one-out encoding (Hancock and Khoshgoftaar, 2020).

One-hot encoding is quite popular and it has been shown to be the best encoding method with the lowest valuation errors, as supported by several studies (Dahouda and Joe, 2021; Gnat, 2021; Hancock and Khoshgoftaar, 2020; Melnykova, 2022; Micci-Barreca, 2001; Potdar *et al*., 2017). One of its key advantages is its simplicity and efficiency of implementation. This method transforms a variable with n observations and c distinct categories into c binary vectors, where each category is associated with a vector that contains 1 and 0 to indicate the presence or absence of the category (Fig. 3).

At this stage, Table 3 enumerates the qualitative attributes in the dataset. While implementing one-hot encoding did augment the feature count, this was mitigated by employing sparse representations, ensuring data compression.

Scikit-learn's one-hot encoder uses the "sparse" parameter by default, efficiently storing encoded values. The "drop" parameter encodes each attribute as n_categories-1 attributes instead of n_categories, dropping one category to avoid collinearity in the input matrix of some classifiers.

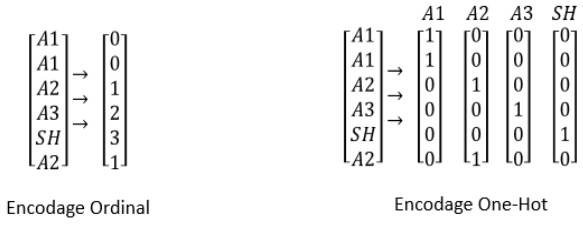

Fig. 3: Encoding of the "PNORMG" attribute

Feature Selection

The dimensionality of data is a serious obstacle for many learning algorithms due to their computational expense, which can make analysis challenging (García *et al*., 2016; Ramírez-Gallego *et al*., 2017; Subasi, 2020). Feature selection is an effective approach for addressing the curse of dimensionality by selecting a subset of features that can effectively describe the input data, reduce computation time, improve prediction performance, and enhance the understanding of the data (Chandrashekar and Sahin, 2014; García *et al*., 2015). Feature selection methods are classified into three categories: Filters, wrappers, and embedded methods (Frye and Schmitt, 2020; Guyon and De, 2003). Wrapper methods are known to be more accurate, utilizing the predictor as a black box and its performance as an objective function to evaluate a subset of features (Chandrashekar and Sahin, 2014; Muresan *et al*., 2015).

Recursive Feature Elimination (RFE) is a widely used algorithm in wrapper methods, based on the backward feature elimination procedure and allows for improving the performance of the ML process (Guyon and De, 2003) and amplifies model performance, whereby RFE starts with all available features and iteratively eliminates less relevant features until the most informative subset of features is identified. In pursuit of an optimal trade-off between accuracy and robustness, the cross-validation procedure is utilized to identify optimal features by eliminating insignificant ones that have no positive impact on the model's accuracy (Ossai *et al*., 2022; Subasi, 2020; Yang *et al*., 2021).

In our study, we utilized the Extra Tree Regressor (ETR). Previous studies have reported that the ETR achieves higher modeling accuracy than other algorithms, as indicated by Abebe *et al*. (2020); Ossai *et al*. (2022); Yothapakdee *et al*. (2022). This ensemble learning technique, serving as a noteworthy alternative to the random forest algorithm, excels in terms of computational efficiency due to its swifter execution (Geurts *et al*., 2006). It randomizes certain decisions and subsets the data to prevent overlearning, and overfitting and reduce variance in the data, outperforming other methods with weaker randomization.

Our methodology employed the Recursive Feature Elimination with Cross-Validation (RFECV) technique, utilizing tenfold cross-validation to evaluate the combinations of input features and select the most important ones for the best predictive accuracy. We selected the Extra Tree Regressor (ETR) as the base estimator to build a model for predicting the duration of a vessel's stay in port. In the process of training the model, the dataset was randomly divided into two sets: 70% for the training set and 30% for the test set.

In order to evaluate the performance of the developed model, we employed key metrics including R-squared (R^2), which assesses the accuracy of the prediction, and additional metrics like Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The R^2 value indicates the proportion of the variability in the predicted variable explained by the model, with higher values indicating a better fit between the model and the data.

Results and Discussion

This study utilized the raw dataset from TC2 in the Tangier Med port. Although we implemented some basic data preprocessing and cleaning steps, the resultant data exhibited insufficient quality for meaningful learning, resulting in a modest $R^2 = 92.87\%$.

The experimental setup unfolded in three distinct phases. Initially, adaptive preprocessing steps, tailor-made for the context of container terminal operations, were deployed to establish a well-prepared dataset for analysis and modeling. Subsequently, the second phase used preprocessed data to identify the top-performing subset of features. Finally, the ETR algorithm was applied to validate the proposed strategy, providing a robust assessment of the preprocessing methodologies and feature selection deployed in the earlier stages.

Our adaptive framework meticulously enhances predictive quality by discerning optimal data preprocessing pipelines and ensuring robust data quality, all while facilitating smooth integration into machine learning applications. Detailed methodology, techniques, and findings will be articulated in the following sub-sections.

Data integration: Our study began with a comprehensive data quality check of data structures, which contained a dataset of 115 features and 13,796 instances. A preliminary evaluation revealed that the problem is a supervised regression problem based on the target variable 'duration at port'. This was followed by integrating and synchronizing the data, removing three redundant attributes, and identifying and removing duplicate tuples.

Data cleaning: A thorough cleaning process eliminated irrelevant and uninformative features. We removed ten textual features that did not contribute to the analysis. We also identified and eliminated eight empty features without any values and 13 features with zero or low variance. This step minimized missing data processing needs and ensured usable features for analysis.

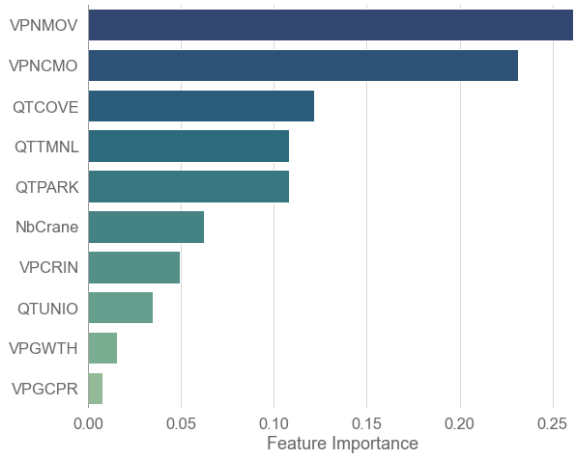

Fig. 4: Importance of optimal features

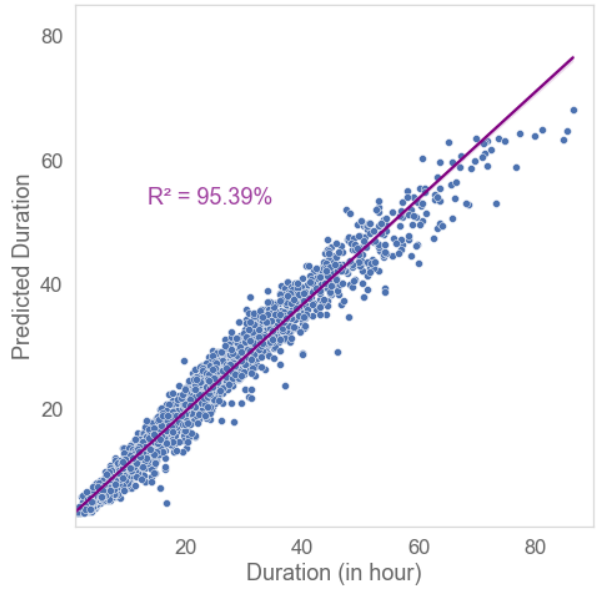

Fig. 5: Illustrative plot of observed and predicted data

Missing data: Based on data profiling, 16% of the values were missing in Fig. 2. We removed features with over 90% missing values. For the rest, we used three imputation methods based on the status of each feature: (i) Imputing with new values for three features, (ii) The most frequent category for two categorical features and (iii) Using the KnnImputer method for the numerical attribute QTCOVE. The imputed features were graphically checked using histograms and boxplots to ensure reliability.

Anomaly detection: Four techniques were implemented (IQR, z-score, OCSVM, and DBSCAN) and their limitations and advantages are discussed. The DBSCAN outperformed the other techniques in terms of efficiency, requiring fewer parameters and less complexity and being more suitable for large and complex datasets. The results and comparison of these techniques (Table 2) demonstrated that DBSCAN was the most effective in identifying outliers and appropriately adjudicating these values compared to other techniques.

Data normalization: The features were scaled using standardization, setting a zero mean and a unit standard deviation, resulting in a unified metric representation. This approach helps to address the issue of data heterogeneity while retaining the shape properties of the original dataset.

Categorical feature encoding: We used the one-hot encoding to convert categorical data. To manage feature expansion, we represented all categories of each feature by N-1 binary variables (N = the number of categories), resulting in a dimensionality of 29 columns (Table 3). Sparse representations were used to compress the data.

Feature selection: We used the RFECV technique to rank features according to their importance, discarding those deemed weak or irrelevant. The 10-fold cross-validation process identified the most important features for improved performance after eliminating irrelevant features that did not affect the model's accuracy positively. Moreover, the ETR algorithm was chosen as the core algorithm to be used in RFECV to determine the optimal features for predicting duration at the port.

The RFECV-ETR model showcased a striking accuracy, achieving an R^2 score of 95.4%, indicating that 95.4% of the variation in the vessel "duration at port" can be explained by the independent variables included model, suggesting a notably strong model fit, also demonstrating a tangible enhancement post-preprocessing adaptive. Furthermore, the model registered an RMSE of 2.38 and an MAE of 2.2, which is better than that given by Wang *et al*. (2020), MAE = 2.32, which relates to the length of stay at the port.

Our feature selection process started by testing 114 input features, eventually identifying ten optimal features that significantly bolstered the model's performance, while streamlining the model and mitigating complexity. The relative importance of the selected features ranged from 0.8-27%, with the top-ranked features being VPNMOV, VPNCMO, QTCOVE, QTTMNL, QTPARK, NbCrane, VPCRIN, QTUNIO, VPGWTH and VPGCPR, as illustrated in Fig. 4. Moreover, the analysis of the selected features revealed that workload attributes were the predominant factor in accurately predicting the target variable, emphasizing their indispensable role in not only enhancing operational efficiency but also in improving supply chain efficiency.

Figure 5 displays both observed and predicted durations alongside the regression estimate, affirming the aptitude of the ETR model in accurately encapsulating the target variable. This is evidenced by a fitting congruence between the estimated and the actual data points, which align closely with the model line.

The evaluation of our model's accuracy indicated high predictive performance, underscoring the efficiency and reliability of our approach, as well as the high quality of the preprocessed data. The results of this study provide valuable insights into the development of a framework that guarantees superior data quality and an efficient predictive model while ensuring that the model relies on key features pertinent to the context of container terminal operations.

In effect, the limited deviation between the actual and predicted timetables, as demonstrated by our model's validity through the experience of diverse handling operations with real-world TC2 data, confirms its role as a practical decision support tool for container terminal scheduling and operations. This success not only ensures enhanced performance but also contributes significantly to optimizing port operations, enhancing supply chain efficiency, and facilitating informed decision-making.

To conclude, future work will focus on exploring various ML algorithms to enhance the accuracy of predictions regarding the duration of stays at the port. Investigations into diverse feature selection methods will also be pursued to yield more nuanced insights. Additionally, we aim to broaden our predictive reach by devising models that forecast container volumes and anticipate the number of vessels arriving at the terminal. Ultimately, our ambition is to integrate these ML models with a robust platform, driving the terminal system towards real-time prediction automation.

Our research, while insightful, is not without limitations. In terms of data diversity, the study relied exclusively on data from tangier TC2. Furthermore, the current model does not account for variations in handling times and temporal data trends and it omits features related to vessel types and line operators. These considerations present points that warrant further exploration and validation in the future.

Conclusion

The approach presented in this study adeptly tackles various challenges linked to real port data. We have introduced a comprehensive adaptive data preprocessing framework, meticulously structured in six sequential steps, specifically designed for the context of the container terminal. Additionally, the utilization of the RFE feature selection method has effectively identified the most discriminative variables within our analysis. To enhance the robustness of the results, we have also employed a cross-validation process.

The quality of the preprocessed dataset has been validated through the implementation of the ETR model, which predicts vessel duration at the port. The performance evaluation of our model has been experimentally confirmed, achieving an impressive R^2 metric score of 95.4%. With these insights gained from our study, tangier TC2 is poised to embrace a structured approach to mitigate operational inefficiencies. By ensuring more efficient resource allocation and minimizing costs associated with delays and congestion, TC2 can solidify its competitive advantage, offering a more reliable service portfolio.

Acknowledgment

The authors extend their heartfelt gratitude to all those who contributed to this study by providing essential data and references. It is through their invaluable contribution that this research was made possible.

Funding Information

The authors have conducted this study without receiving any financial support or funding.

Author's Contributions

Mostafa Al Uahabi: Conceptualization, methodology, software, formal analysis, investigation, resources, data curation, written original drafted, preparation, visualization, and funded acquisition.

Hicham Attariuas: Methodology, validation, formal analysis, resources, supervision, and project administration.

Mohammed Saleh: Assisted in the data analysis, and drafted and reviewed the manuscript.

Mohamed Chentouf: Validation, formal analysis, and written, reviewed, and edited.

Ethics

The authors affirm that this article is an original work and has not been previously published elsewhere.

References

Abebe, M., Shin, Y., Noh, Y., Lee, S., & Lee, I. (2020). Machine learning approaches for ship speed prediction towards energy efficient shipping. *Applied Sciences*, *10*(7), 2325. https://doi.org/10.3390/app10072325

Alexandropoulos, S. A. N., Kotsiantis, S. B., & Vrahatis, M. N. (2019). Data preprocessing in predictive data mining. *The Knowledge Engineering Review*, *34*, e1. https://doi.org/10.1017/S026988891800036X

Al-Taie, M. Z., Kadry, S., & Lucas, J. P. (2019). Online data preprocessing: A case study approach. *International Journal of Electrical and Computer Engineering*, *9*(4), 2620. https://doi.org/10.11591/ijece.v9i4.pp2620-2626

Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, *12*(16), 4102-4107. https://www.researchgate.net/publication/319990923_Review_of_Data_Preprocessing_Techniques_in_Data_Mining

Bank, W. (2023). The Container Port Performance Index 2021: A Comparable Assessment of Container Port Performance. In *IOP Conf. Series: Earth and Environmental Science* (Vol. *1188*, p. 012023). http://documents.worldbank.org/curated/en/099051723134019182/P1758330d05f3607f09690076fedcf4e71a

Bilal, M., Ali, G., Iqbal, M. W., Anwar, M., Malik, M. S. A., & Kadir, R. A. (2022). Auto-prep: Efficient and automated data preprocessing pipeline. *IEEE Access*, *10*, 107764-107784. https://doi.org/10.1109/ACCESS.2022.3198662

Çelik, M., Dadaşer-Çelik, F., & Dokuz, A. Ş. (2011, June). Anomaly detection in temperature data using DBSCAN algorithm. *In 2011 International Symposium on Innovations in Intelligent Systems and Applications* (pp. 91-95). IEEE. https://doi.org/10.1109/INISTA.2011.5946052

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 1-58. https://doi.org/10.1145/1541880.1541882

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering*, *40*(1), 16-28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Dahouda, M. K., & Joe, I. (2021). A deep-learned embedding technique for categorical features encoding. *IEEE Access*, *9*, 114381-114391. https://doi.org/10.1109/ACCESS.2021.3104357

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. *In kdd* (Vol. *96*, No. 34, pp. 226-231). https://cdn.aaai.org/KDD/1996/KDD96-037.pdf

Farhangfar, A., Kurgan, L. A., & Pedrycz, W. (2007). A novel framework for imputation of missing values in databases. *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, *37*(5), 692-709. https://doi.org/10.1109/TSMCA.2007.902631

Frye, M., Mohren, J., & Schmitt, R. H. (2021). Benchmarking of data preprocessing methods for machine learning-applications in production. *Procedia CIRP*, *104*, 50-55. https://doi.org/10.1016/j.procir.2021.11.009

Frye, M., & Schmitt, R. H. (2020). Structured data preparation pipeline for machine learning-applications in pro-duction. *17th IMEKO TC*, *10*, 241-246. https://www.imeko.org/publications/tc10-2020/IMEKO-TC10-2020-034.pdf

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72, pp. 59-139). Cham, Switzerland: Springer International Publishing. https://doi.org/10.1007/978-3-319-10247-4

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: Methods and prospects. *Big Data Analytics*, *1*(1), 1-22. https://doi.org/10.1186/S41044-016-0014-0

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*, 3-42. https://doi.org/10.1007/s10994-006-6226-1

Gnat, S. (2021). Impact of categorical variables encoding on property mass valuation. *Procedia Computer Science*, *192*, 3542-3550. https://doi.org/10.1016/j.procs.2021.09.127

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(Mar), 1157-1182.

Han, J., Kamber, M., & Pei, J. (2012). Data mining concepts and techniques third edition. *University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University*. https://doi.org/10.1016/B978-0-12-381479-1.00003-4

Hancock, J. T., & Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. *Journal of Big Data*, *7*(1), 1-41. https://doi.org/10.1186/s40537-020-00305-w

Karagiannidis, P., & Themelis, N. (2021). Data-driven modelling of ship propulsion and the effect of data pre-processing on the prediction of ship fuel consumption and speed loss. *Ocean Engineering*, *222*, 108616. https://doi.org/10.1016/J.OCEANENG.2021.108616

Little, R. J., & Rubin, D. B. (2002). *Statistical Analysis with Missing Data* (Vol. *793*). John Wiley and Sons. https://doi.org/10.1002/9781119013563

Marco, R., Syed Ahmad, S. S., & Ahmad, S. (2021). Empirical Analysis of Software Effort Preprocessing Techniques Based on Machine Learning. *International Journal of Intelligent Engineering and Systems*, *14*(6). https://doi.org/10.22266/ijies2021.1231.49

Melnykova, N. (2022). A Novel Approach for the Automatic Detection of COVID in a Patient by Using a Categorization Methods. *Procedia Computer Science*, *198*, 712-717. https://doi.org/10.1016/j.procs.2021.12.311

Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, *3*(1), 27-32. https://doi.org/10.1145/507533.507538

Mohd, F., Bakar, Z. A., Noor, N. M. M., & Rajion, Z. A. (2013, October). Data preparation for pre-processing on oral cancer dataset. *In 2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)* (pp. 324-328). IEEE. https://doi.org/10.1109/ICCAS.2013.6703916

Muresan, S., Faloba, I., Lemnaru, C., & Potolea, R. (2015, September). Pre-processing flow for enhancing learning from medical data. In *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)* (pp. 27-34). IEEE. https://doi.org/10.1109/ICCP.2015.7312601

Ossai, C. I., Rankin, D., & Wickramasinghe, N. (2022). Preadmission assessment of extended length of hospital stay with RFECV-ETC and hospital-specific data. *European Journal of Medical Research*, *27*(1), 1-16. https://doi.org/10.1186/s40001-022-00754-4

Patro, S. G. O. P. A. L., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv Preprint arXiv:1503.06462*. https://doi.org/10.48550/arXiv.1503.06462

Pérez, J., Iturbide, E., Olivares, V., Hidalgo, M., Almanza, N., & Martínez, A. (2015). A data preparation methodology in data mining applied to mortality population databases. In *New Contributions in Information Systems and Technologies: Volume 1* (1173-1182). Springer International Publishing. https://doi.org/10.1007/s10916-015-0312-5

Potdar, K., Pardawala, T. S., & Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, *175*(4), 7-9. https://doi.org/10.5120/ijca2017915495

Press, G. (2016). Cleaning big data: Most time-consuming, least enjoyable data science task, survey says. *Forbes, March*, *23*, 15. Bowes, P. (2015). MapMarker 24.1 release notes.

Prakash, A., Navya, N., & Natarajan, J. (2019). Big data preprocessing for modern world: Opportunities and challenges. *In International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018* (335-343). Springer International Publishing. https://doi.org/10.1007/978-3-030-03146-6_37

Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., & Herrera, F. (2017). A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing*, *239*, 39-57. https://doi.org/10.1016/j.neucom.2017.01.078

Subasi, A. (2020). *Practical Machine Learning for Data Analysis Using Python*. Academic Press. https://doi.org/10.1016/B978-0-12-821379-7.00002-3

Wang, N., Shen, S., Cao, J., Ding, Y., & Xiao, Y. (2020, December). A system for container terminal operation prediction. *In 2020 IEEE International Conference on Progress in Informatics and Computing (PIC)* (pp. 407-411). IEEE. https://doi.org/10.1109/PIC50277.2020.9350770

Yang, B., Haghighat, F., Fung, B. C., & Panchabikesan, K. (2021). Season-based occupancy prediction in residential buildings using machine learning models. *E-Prime-Advances in Electrical Engineering, Electronics and Energy*, *1*, 100003. https://doi.org/10.1016/j.prime.2021.100003

Yothapakdee, K., Charoenkhum, S., & Boonnuk, T. (2022). Improving the efficiency of machine learning models for predicting blood glucose levels and diabetes risk. *Indonesian Journal of Electrical Engineering and Computer Science*, *27*(1), 555-562. https://doi.org/10.11591/ijeecs.v27.i1.pp555-562