Optimization of Loading Operations and Palletized Goods

Nelson Capote
Krannert School of Management
Purdue University
West Lafayette, USA
ncapote@purdue.edu

Xindi Liu
Krannert School of Management
Purdue University
West Lafayette, USA
liu3507@purdue.edu

Udyog Pati
Krannert School of Management
Purdue University
West Lafayette, USA
upati@purdue.edu

Tommy Starnes
Krannert School of Management
Purdue University
West Lafayette, USA
tstarn00@purdue.edu

Matthew A. Lanham
Krannert School of Management
Purdue University
West Lafayette, USA
lanhamm@purdue.edu

***Abstract*—There are several facets within supply chain operations that are pivotal to the success of minimizing time between order placed and order delivered. This research focuses on the optimization of truck loading.**

The motivation for this research is that delays in the packing and loading of pallets into trucks can lead to delays in arrivals which lead to customer dissatisfaction. A common scenario for freight packers is there exists orders having multiple items. They must estimate the number of pallets needed to fulfill an order, estimate the size and weight of each pallet. Lastly, there is an optimization step to decide if pallets can be stacked, and how to load the truck to allocate space efficiently.

We construct a two-stage design where in the first stage prediction is used to estimate pallets, then in the second stage a linear program optimization model is employed to effectively provide a truck loading recommendation. Our model integrates proper truck identification, truck constraints, pallet dimensions, and weight.

Keywords*—*Container Loading, Freight, Linear Programming, Optimization

I. Introduction

The wise leader understands that proper planning prevents poor performance. This adage stands true in every facet of operations, especially in the supply chain operations. If a firm maintains centralized control of supply chain operations, more specifically in packaging and loading of orders onto their own trucks, that firm assumes the responsibility of perfecting its operations. Accountability of orders palletized and loaded onto trucks is a source of frustration in the supply chain. To streamline loading operations, predictive and prescriptive analytics (i.e., optimization modeling) are used in the palletization of orders, and loading of trucks, to maximize space use within the transportation assets. Fig. 1 illustrates this process.

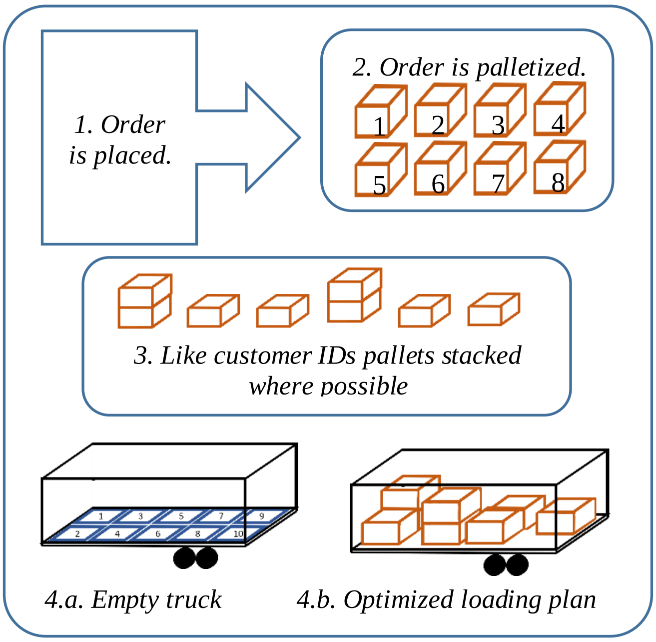

Fig. 1. Conceptualization

Failure to optimize truck loading leads to wasted time, resources, and increased operational costs. Given known orders, dimensions, weight of delivery-ready pallets, available trucks for delivery, and timelines for each order, there exists an opportunity to use this available data to drive better decision-making. Beginning with a prediction model which outputs the number of pallets needed to fulfill a given order, leaning the warehouse truck loading operation is incumbent on an optimized loading plan.

To formulate an optimization model for maximizing capacity and minimizing wasted space, we followed these steps:

1. Defined the objective function: The objective function should represent the goal of minimizing wasted space, which can be represented as a combination of the volume of the truck and the volume of the cargo.

2. Define the constraints: The constraints should represent the limitations on the cargo that can be loaded, such as the maximum weight and size restrictions.
3. Formulate the mathematical model: The mathematical model can be formulated as a linear programming problem, depending on the specific requirements of the problem.
4. Solve the optimization model: The optimization model can be solved using off-the-shelf mathematical programming algorithm solvers.
5. Evaluate the solution: the solution obtained from the optimization model should be evaluated against the constraints and the objective function to determine its feasibility and optimality.
6. Implement the solution: The solution obtained from the optimization model should be implemented in practice to fill the truck cargo space and minimize wasted space.

A wide range of products are packaged onto pallets and loaded onto vehicles for transportation. For each pallet, we use two attributes. First, the unit weight (in lb./pallet) of a pallet with given products. Second, the number of pallets that are loaded onto a truck to fulfill a given order, linked by order as to prevent multiple trucks used to fulfill a single order. Products were grouped into two categories (i.e., palletized, slab) based on the type of product being shipped. Palletized products requiring weight information and stacking constraints. Slab products tied to orders, which may have palletized goods, require non-box truck vehicles called flatbeds.

Most shipping firms have policies that ensure a minimum order of products must be delivered in a single shipment. This is to ensure the fleet is being effectively used with minimal small deliveries. Such policies introduce time as a potential constraint. Some products are packaged and loaded onto pallets within hours of an order being placed while others may take days or weeks. Instead of delivering the first pallet of product immediately and delaying the second pallet of the product, the first pallet will be stored while it waits for the second pallet to be packaged.

This paper will address this project in totality, beginning with a review on the literature used to frame the problem and known solutions. Next, this paper will address the proposed methodology for our solution and the criteria. We then outline the results from our model and address their performance. We conclude with recommendations for further study, lessons learned, and final remarks.

II. LITERATURE REVIEW

The baseline issue behind our combinatorial optimization problem is how to load a truck with a given number of stacked and unstacked pallets. Additionally, how is this to be done without exceeding the limitations of the trucks, minimizing wasted space within the truck, all while leaning the loading procedures.

This truck loading problem has been investigated from different perspectives including layering of items [2] in containers such as box trucks. This study brings to light the importance of some constraints over others. For example, a constraint in loading optimization will likely be volume when the weight of a palletized order is low. However, when pallet weights are large, the constraint of maximum weight allowed on a truck becomes more of a concern than the volume of the container. To address the concept of maximizing volume, we begin with Multi Container Loading Problems (MCLPs).

A. Multi Container Loading Problems

Multi Container Loading Problems (MCLPs) are the increase in complexity of the optimization of packing and trucking [1]. The foundation of MCLP is optimization of space utilization with items that are single stacked, making it a two-dimensional problem. Once pallets are identified as stackable, the problem evolves to the third dimension. An additional constraint must be added to the model which is the maximum allowable cargo load, or the weight of a pallet that another pallet can support on top of it. Another constraint observed is the maximum height of the stacked pallets which prevents tipping of the stacked goods during transportation and loading/unloading.

B. Operational models in use

Operational models in research are a tried and tested domain for study. When considering multiple criteria for decision analysis, stochastic process modeling, and neural networks, operational models are the go-to [7]. Deterministic and stochastic models are the foundation upon which our model was influenced by.

Within the umbrella of deterministic models lies optimization and numerical methods. Since we are deciding the optimization of packing and palletization, our focus in this study is on optimization techniques and numerical methods.

C. LAFF (Largest Area Fit First) Algorithm

Largest Area Fit First (LAFF) models help decide load order of palletized orders. LAFF has applications to different dimensions and can be considered for this model.

The bin-packing problem (BP) is a one-dimensional problem common in optimization of combinations. In the context of the given problem, the combination of packaged orders on pallets and loading of pallets onto a truck presents a problem beyond the first dimension. To expand this concept, using BP to address the filling of trucks with completed pallets grouped by orders is appropriate. First-fit (FF), best-fit (BF), and worst-fit (WF) are examples of heuristics for BP [4].

FF is an approach where the first free spatial partition in a container (a truck for example) that can accommodate a pallet is filled with the first available pallet. The advantage of this method is rapid filling of space within the truck at the risk of order mismatch on a truck and wasted space within the truck.

BF seeks to account for the dimensions of the pallet and space available within the truck. It is similar to FF, but it chooses truck in a decreasing space order.

The WF approach searches the available space available within the truck, chooses the largest unallocated space, and places the pallet into this space. The problem is the assumption of no follow-on pallet for loading, and it may prevent a larger pallet from being placed into this space. A smaller pallet may be placed into the largest space available within the truck, preventing larger pallets from later being added to the truck.

D. Heuristics and genetic algorithms

Reference [5] discusses a genetic algorithm (GA) addressing the stacking of pallets to maximize the loading of containers for shipment. Using their GA approach, they identified three factors: 1) the subset of cargo packaged onto pallets for shipment, 2) date on which cargo is delivered and unloaded (in our case, order of unloading of truck based on delivery route), and 3) the pallets themselves. In their approach, the dimensions of the pallets used for shipment varied based on the product being delivered. For this project, the model selected should consider the first two factors because of the assumption that all pallets for shipment are of the same dimension (base of the pallet is 40" x 48" and no taller than 48"). This alludes to the final segment of the optimization model, which ensures pallets are loaded onto the truck in a way that prevents repacking of the truck at a given destination after an order is unloaded.

E. Machine Learning Models

To deal with the missing values, many machine learning models could be applied, such as Random Forest [3], Decision Tree [6], Boosting [8] and Linear Regression.

III. DATA

Data was collected from an industry-leading floor tile company which contained their inventory, loading trucks, and historical shipping data. The data was from one of their shipping facilities and allowed us to correlate weight and volume to optimize the loading of their shipments.

The study data consisted of three primary datasets: item master dataset, truck master dataset, and operation dataset. Table I provides a brief description of data used in this study.

The item master data table was composed of 31,497 records and 21 columns. The item master data describes the item number, description, product lines, weight, and different dimensions of the items that need to be shipped. Some products that the company no longer carries, redundant item data items, and some with missing values necessitated some data cleaning.

The truck master table was composed of the different loading trucks the firm uses to ship their products. There is a unique asset id of each truck by branch, VIN number, type, and brand. Truck dimensions and loading capacities were also listed. There were four main truck sizes for the company to deliver: 18", 22", 26", and 29" box trucks.

Lastly, an operation table was provided that detailed the order list composed of 1,558 records for the period from 1st December 2022 to 31st December 2022. This data set focuses on the items being purchased with attributes like item number, item description, quantity, weight, pieces per crate, invoice number, customer, and transition date. We used this table to predict the number of crates needed for each customer and the number of crates to be loaded in the truck.

TABLE I. DATA DESCRIPTION

No.	Data	Dataset	Description
1	ItemNumber	Item, Operation	Unique product ID
2	ItemDescription	Item, Operation	Item descriptions
3	ProductLine	Item, Operation	Product line ID
4	PcsPerCrate	Item, Operation	Pieces for each crate, used to calculate the actual order percentage for each crate
5	WtPerCrate	Item	Weight per Crate, used to calculate number of crates
6	CrateSizeLength	Item	Length of Crate Size, used for dimension prediction
7	CrateSizeWidth	Item	Width of Crate Size, used for dimension prediction
8	CrateSizeHeight	Item	Height of Crate Size, used for dimension prediction
9	TruckLength	Truck	Length of Truck Size, used as a prediction constraint
10	TruckWidth	Truck	Width of Truck Size, used as a prediction constraint
11	TruckHeight	Truck	Height of Truck Size, used as a prediction constraint
12	TruckLoad	Truck	Loading capacity of Truck
13	Transaction Date	Operation	Date of transaction
14	Qty	Operation	Quantity of item(pcs)
15	Weight	Operation	Total weight for each order items (Qty*weight/pcs)
16	SO#	Operation	Sales Order ID
17	Customer#	Operation	Customer ID

IV. METHODOLOGY

The methodological process we followed is depicted in Fig. 2. Before we solve the problem, we need to first understand the data to get a sense of the data. Data joining and preprocessing was performed on the provided datasets. Exploratory Data Analysis allowed us to identify and address missing and noisy data, which led to a clean, reduced data set for modeling. We drew some insights from the EDA (Exploratory Data Analysis) and data preparing:

- Most of the orders have between 1 to 4 unique item types.
- PT and LVT account for a considerable proportion of the product lines.
- There were around 3,000 unique items in February's orders and about 25% of those have missing values.

The modeling phase begins upon cleaned data and ends with a finalized model identified for prediction. Python is the primary coding platform used for the analysis and the Numpy, Pandas and Sklearn packages are used. To build our optimization model, we need to fill in the missing dimensions data. Due to the limited number of variables available, we chose to use the numerical information of the item numbers and product lines as our predictors to predict the crates' dimensions, weight, and pieces per crate for each crate with missing values. With the data cleaned and partitioned (80% train, 20% test), we applied different machine learning models to conduct cross-validation.

Guided by the findings during the EDA, the modeling phase is driven by these insights. Level algorithm and base area models are considered during this phase. We conduct war-gaming on each of the models in the model comparison segment of this phase. Once a model is decided upon, it is used in the third and final phase.

The prediction phase begins with the best model for prediction being selected and ends with implementation in the prediction of the number of pallets required to meet an order as well as the optimal loading of a truck to minimize space for shipment. The team was divided into two groups, each formulating their own models. Finalized models were compared to determine the best model, and this model was carried over to building a User Interface (UI) for the client.

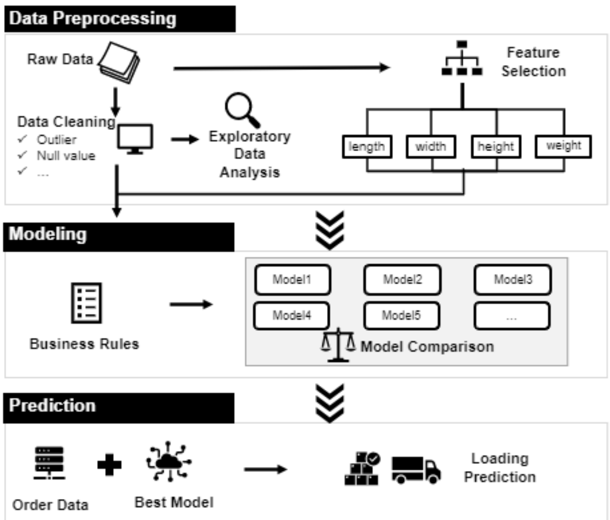

Fig. 2. Methodology diagram

V. MODELS

Before building the optimization models, we standardized the parameters and decision variables to get data needed for our models.

A. Inputs to the models

The inputs are the dimensions of the trucks and the dimensions of the crates.
The truck dimensions are shown in (1).

$$Truck = (T_L, T_W, T_H, T_{WT}) \quad (1)$$

T_L: Length of Truck
T_W: Width of Truck
T_H: Height of Truck
T_{WT}: Weight Capacity of Truck
The truck dimensions are shown in (2).

$$Crate = (C_L, C_W, C_H, C_{WT}, C_P) \quad (2)$$

C_L: Length of Crate
C_W: Width of Crate
C_H: Height of Crate
C_{WT}: Weight of Crate
C_P: Pieces of Item per Crate

B. Standardization

The most frequently used crate sizes are 32*32 and 48*40, so in table II we divided all different crate sizes into 3 types:

TABLE II. STANDARD CRATE SIZES

Types	Original Size	Standard Size
1	C_L <32 & C_W <32	32*32
2	C_L <48 & C_W <40	48*40
3	Other dimensions	Keep original size

C. Calculation

To design the space utilized according to the business rules, four fields were calculated. The number of crates for each order item is calculated by (3) and the stackable create number got from (4). The number of partial crates for each client is calculated by (5) and the position number for each order item is shown in (6).

$$C_N = I_Q / C_P \quad (3)$$

Where
C_N: Crate Number
I_Q: Item Quantity

$$S_N = (I_P \in \{\text{“LVT”}, \text{“PT”}\}) \ \& \ \text{Min}\ \{(T_H / C_H), (F_{WT} / C_{WT})\} \quad (4)$$

Where
S_N: Stackable Crate Number
I_P: Item Product Line
F_{WT}: Floor Jack Capacity

$$C_{PN} = \sum_{n=1}^{n}(\text{CN} - \text{FloorCN}) \quad (5)$$

Where
C_{PN}: Partial Crate Number

$$P_N = \text{Abs}(C_N / S_N) \mid \text{Abs}(C_{PN}) \quad (6)$$

Where
P_N: Crate Position Number

D. Algorithm

1. Level Algorithm

Fig.3 is an example of crates loading applying level algorithm. This algorithm considers constraints shown as (7) to help check whether the crates given can or cannot be loaded in all truck types and choose the best fitted truck type to efficiently save the loading space and operation cost.

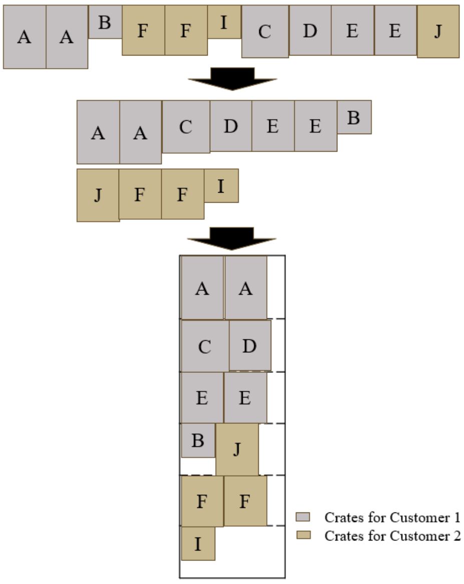

Fig. 3. Example of Loading

Constraints

(7)

Max $\{(C_{1W} + C_{2W}), (C_{3W} + C_{4W}), \ldots, (C_{n-1W} + C_{nW})\} <= T_{iW}$

Max $\{C_{1L}, C_{2L}\}$ + Max $\{C_{3L}, C_{4L}\}$ + … + Max $\{C_{n-1L}, C_{nL}\} <= T_{iL}$

Max $\{\sum_{j=1}^{k} C_{jH}\} <= T_{iH}$

$\sum_{j=1}^{n} C_{jWT} <= T_{iWT}$

Algorithm 1: Level Algorithm

Input: $TruckLength(T_{iL})$=$T_{1L}, T_{2L}, \ldots, T_{nL}$
$TruckWidth(T_{iW})$=$T_{1W}, T_{2W}, \ldots, T_{nW}$
$TruckHeight(T_{iH})$=$T_{1H}, T_{2H}, \ldots, T_{nH}$
$TruckWeight(T_{iWT})$=$T_{1WT}, T_{2WT}, \ldots, T_{nWT}$
$CrateLength(C_{jL})$=$C_{1L}, C_{2L}, \ldots, C_{mL}$
$CrateWidth(C_{jW})$=$C_{1W}, C_{2W}, \ldots, C_{mW}$
$CrateHeight(C_{jH})$=$C_{1H}, C_{2H}, \ldots, C_{mH}$
$CrateWeight(C_{jWT})$=$C_{1WT}, C_{2WT}, \ldots, C_{mWT}$
Output: Whether all crates can be loaded in the Truck and T_i

1 Standardize crate sizes C_{jL} and C_{jW};
2 Calculate C_N, S_N, C_{PN}, P_N based on (3)(4)(5)(6) for each order record;
3 Sort records according to customer, C_{jL} and C_{jW};
4 **for** $i = 1; i \leq n$ **do**
5 　**for** $j = 1; j \leq m$ **do**
6 　　**if** $C_{jWT} \leq remainingT_{iWT}$ **then**
7 　　　**if** $C_{jL} \leq remainingT_{iL}$ **then**
8 　　　　**if** $C_{jW} \leq remainingT_{iW}$ **then**
9 　　　　　$remainingT_{iW} - = C_{jW}$;
10 　　　　　$remainingT_{iWT} - = C_{jWT}$;
11 　　　　else
12 　　　　　$remainingT_{iL} - = C_{jL}$;
13 　　　　　$remainingT_{iW} = T_{iW}$;
14 　　　　　jump to step 7;
15 　　　　**end**
16 　　　**end**
17 　　**end**
18 　**end**
19 **end**
20 **if** $j \neq m$ **then**
21 　Cannot load all crates
22 **else**
23 　return Min(T_i)
24 **end**

2. Base Area Model

This base area model defines a function that takes three arguments: crates available, crate routes, and containers.

The function takes the input data and tries to fit the available crates into containers, considering various constraints in (8) such as weight capacity and area capacity of the container. It iterates through the routes and containers to find the best combination of crates to fit in each container.

The function uses several data structures to keep track of the loaded and stacked crates, the amount of space and weight capacity used per container, and the remaining crates that were not loaded.

The function returns several output data frames, including loaded and stacked crates, constraints, and the remaining crates.

Constraints

(8)

$\sum_{j=1}^{n} C_{jL} * C_{jW} <= T_{iL} * T_{iW}$

Max $\{\sum_{j=1}^{k} C_{jH}\} <= T_{iH}$

$\sum_{j=1}^{n} C_{jWT} <= T_{iWT}$

Algorithm 2: Base Area Algorithm

Input: $TruckLength(T_{iL})$=$T_{1L}, T_{2L}, \ldots, T_{nL}$
$TruckWidth(T_{iW})$=$T_{1W}, T_{2W}, \ldots, T_{nW}$
$TruckHeight(T_{iH})$=$T_{1H}, T_{2H}, \ldots, T_{nH}$
$TruckWeight(T_{iWT})$=$T_{1WT}, T_{2WT}, \ldots, T_{nWT}$
$CrateLength(C_{jL})$=$C_{1L}, C_{2L}, \ldots, C_{mL}$
$CrateWidth(C_{jW})$=$C_{1W}, C_{2W}, \ldots, C_{mW}$
$CrateHeight(C_{jH})$=$C_{1H}, C_{2H}, \ldots, C_{mH}$
$CrateWeight(C_{jWT})$=$C_{1WT}, C_{2WT}, \ldots, C_{mWT}$ and threshold θ_r
Output: T_i

1 Standardize crate sizes C_{jL} and C_{jW};
2 Calculate C_N, S_N, C_{PN}, P_N based on (3)(4)(5)(6) for each order record;
3 **for** $i = 1; i \leq n$ **do**
4 　$T_{iArea} = T_{iL} \times T_{iW} \times \theta_r$;
5 　**for** $j = 1; j \leq m$ **do**
6 　　$C_{jArea} = C_{jL} \times C_{jW}$;
7 　　**if** $C_{jWT} \leq remainingT_{iWT}$ **then**
8 　　　**if** $C_{jArea} \leq remainingT_{iArea}$ **then**
9 　　　　$remainingT_{iWT} - = C_{jWT}$;
10 　　　　$remainingT_{jArea} - = C_{iArea}$;
11 　　　**end**
12 　　**end**
13 　**end**
14 **end**
15 return (T_i)

E. Complexity of models

Compared to previous loading methods that the staff manually planned crates to be packaged, loaded, and chose trucks size based solely on weight limitations, our models consider both the dimensions and weight capacity of crates and trucks. These algorithms manage various constraints by comparing the weight, length, width, and staked height of crates with different truck types. This is done to efficiently optimize the utilization of available space within each truck.

Our models offer several advantages:

- Space utilization optimization: By standardizing crates' dimensions and arranging crates based on customers, stackable status, length and width, the model can check whether there is enough space with each truck and decide which size of truck could be used to minimize wasted space.

- Time and cost efficiency: Using models to select truck types and arrange packaging and loading is faster than manual processing and minimizes reloading mistakes. As a result, manual time-consuming processes can be eliminated and staff costs reduced, allowing those employees to be shifted to other critical business positions and improving operational efficiency.

  Loading stability improvement: Considering crate dimensions and stackable status for each route can reduce the risk of crates being damaged or sliding when loading and transporting. Besides, it can help the company plan ahead and eliminate last-minute changes or addition requests.

VI. RESULTS

For the crate dimensions prediction, we experimented with different machine learning models to get the model with the highest accuracy score. The results of the validation are seen in Table III.

TABLE III. MODELS HIGHEST ACCURACY SCORE

Target Variables	Model	Accuracy Score
CrateSizeLength	Extra Tree	90.27%
CrateSizeWidth	Random Forest	25.31%
CrateSizeHeight	Adaptive Boosting	43.98%
PcsPerCrate	Extra Tree	80.85%
WtPerCrate	Decision Tree	53.11%

In this first model, we have 417 rows of route data for February. After mapping the order table and removing exclusive product line such as “QZ” and “SL”, we get 187 routes. By using a level algorithm, we found that 71% route results are similar to the actual loading practice. The second model exhibits an 86% match rate with the current operations of our clients.

We found that there were 26 routes with orders that were unable to be loaded. However, with our algorithm, those orders could have been loaded by selecting appropriate truck type.

However, it is noteworthy that the accuracy is affected by certain factors, such as the truck assigned to a specific route. Due to the unavailability of information regarding the exact truck used for the delivery of goods, the dimensions utilized may deviate slightly from the actual ones. Consequently, the model generates three distinct outputs. The first output is comprised of the total number of crates loaded on a specific route, which includes 11 attributes such as the unique container identifier, which route it is on, unutilized area within the truck, the number of loaded crates, and makes a recommendation as to the optimal truck to complete each delivery. The second output encompasses the specific client and its corresponding shipping order number loaded onto a designated truck. Lastly, final output presents all the relevant information for orders that could not be loaded.

Digitalization of loading operations and the provision of accurate information to clients can offer numerous benefits to their business. Firstly, clients can leverage this data to optimize their logistics and supply chain management. Precise information regarding the number of crates loaded, the truck utilized, and the weight distribution can aid in improving the route planning, leading to reduced transportation costs and improved delivery timelines.

Moreover, this information can provide clients with a better understanding of their inventory levels and order fulfillment capabilities, enabling them to make informed decisions regarding their operations. Clients can use this data to improve their customer service by providing accurate delivery timelines and efficient order fulfillment.

Digitalization of loading operations can also enhance the overall efficiency of clients' operations by reducing manual errors and minimizing the need for manual intervention. This can lead to increased productivity, improved resource utilization, and streamlined operations. Additionally, this information can help clients monitor and optimize the performance of their logistics partners, leading to better collaboration and improved outcomes.

Overall, digitalization of loading operations and the provision of accurate information to clients can offer numerous benefits to their business, including improved logistics and supply chain management, better customer service, increased efficiency, and optimized operations.

VII. CONCLUSION

Our program and models remain accurate in predicting the requirements of a given order, but they can be improved. This is due to significant gaps in the dataset provided by the client. The client has since begun an initiative to fill in the gaps in their database, replacing all missing values with accurate measurements of the dimensions of the palletized goods. Upon completion of this initiative, it is expected that our model’s accuracy will improve significantly.

REFERENCES

[1] Alonso, M. T., Alvarez-Valdes, R., Iori, M., Parreño, F., & Tamarit, J. M. (2017). Mathematical models for multicontainer loading problems. *Omega*, *66*, 106–117. https://doi.org/10.1016/j.omega.2016.02.002

[2] Bortfeldt, A., & Gehring, H. (2001). A Hybrid Genetic Algorithm for the Container Loading Problem. *European Journal of Operational Research* 131: 143-161.

[3] Brown, D.E., 2016. Text mining the contributors to rail accidents. IEEE Trans. Intell.Transp. Syst. 17, 346–355.

[4] Coffman EG Jr, Csirik J, Galambos G, Martello S, Vigo D (2013) Bin packing approximation algorithms: survey and classification. In: {ardalos P, Du DZ, Graham R (eds) Handbook of combinatorial optimization. Springer, New York, pp 455-531

[5] Lau, H. C. W., T. M. Chan, W. T. Tsui, Ho, G. T. S., & Choy, K. L. (2009). An AI Approach for Optimizing Multi-pallet Loading Operations. *Expert Systems with Applications* 36: 4296-4312.

[6] Oztekin, A., Kong, Z.J., Delen, D., 2011. Development of a structural equation modeling-based decision tree methodology for the analysis of lung transplantations. Decis. Support Syst. 51, 155–166. doi:10.1016/j.dss.2010.12.004.

[7] Patil, J. T., & Patil, M. E. (2016). Cargo space optimization for Container. *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC).* https://doi.org/10.1109/icgtspicc.2016.7955271

[8] Y.Freund, R.Shapire. A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, 1999.