energies

MDPI

Article

Machine-Learning Methods to Select Potential Depot Locations for the Supply Chain of Biomass Co-Firing

Diana Goettsch, Krystel K. Castillo-Villar * and Maria Aranguren

Department of Mechanical Engineering, Texas Sustainable Energy Research Institute, University of Texas at San Antonio, San Antonio, TX 78249, USA; diana.goettschmelendez@my.utsa.edu (D.G.); maria.aranguren@utsa.edu (M.A.)

* Correspondence: Krystel.Castillo@utsa.edu

Received: 20 November 2020; Accepted: 8 December 2020; Published: 11 December 2020

Abstract: Coal is the second-largest source for electricity generation in the United States. However, the burning of coal produces dangerous gas emissions, such as carbon dioxide and Green House Gas (GHG) emissions. One alternative to decrease these emissions is biomass co-firing. To establish biomass as a viable option, the optimization of the biomass supply chain (BSC) is essential. Although most of the research conducted has focused on optimization models, the purpose of this paper is to incorporate machine-learning (ML) algorithms into a stochastic Mixed-Integer Linear Programming (MILP) model to select potential storage depot locations and improve the solution in two ways: by decreasing the total cost of the BSC and the computational burden. We consider the level of moisture and level of ash in the biomass from each parcel location, the average expected biomass yield, and the distance from each parcel to the closest power plant. The training labels (whether a potential depot location is beneficial or not) are obtained through the stochastic MILP model. Multiple ML algorithms are applied to a case study in the northeast area of the United States: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Multi-Layer Perceptron (MLP) Neural Network. After applying the hybrid methodology combining ML and optimization, it is found that the MLP outperforms the other algorithms in terms of selecting potential depots that decrease the total cost of the BSC and the computational burden of the stochastic MILP model. The LR and the DT also perform well in terms of decreasing total cost.

Keywords: machine learning; neural networks; logistics; biomass; mathematical programming; optimization

1. Introduction

In 2019, coal was the second-largest energy source for electricity generation in the United States, with coal-fueled power plants representing 23% of all electricity sources in the country [1], and 38% worldwide [2]. The abundance of the resource and its lower cost have made coal a popular choice for electricity generation around the world. However, coal combustion creates negative effects on the environment and human health. In fact, the burning of coal generates dangerous gas emissions, such as carbon dioxide, which is a contributor to Green House Gas (GHG) emissions; sulfur dioxide, which increases acid rain and respiratory illnesses; nitrogen oxides, which also cause respiratory illnesses and smog; and mercury, which is connected to neurological damage in humans and other animals [3].

An alternative to coal combustion is the use of biomass. Generally, biomass includes any natural renewable fuel, such as wood, agricultural residues, food waste, and industrial waste [4], and it is expected to be the largest source of renewable energy in the next years. Biomass co-firing involves converting biomass to electricity by adding biomass as a partial substitute fuel in high-efficiency coal boilers at relatively low biomass to coal ratios [4]. An advantage of biomass co-firing is the reduction of emissions of both carbon dioxide and sulfur dioxide caused by coal combustion [5].

On the other hand, biomass co-firing poses some challenges for the energy plants. Although fuel costs may be low, transportation, preparation, and handling costs for biomass can exceed total fuel costs for other fossil options [4]. Therefore, to establish biomass co-firing as an alternative energy source, the optimization of the biomass supply chain (BSC) is essential. Most authors investigating BSC's have focused on applying models that rely on optimization techniques, especially Mixed-Integer Linear Programming (MILP) models. In fact, authors such as Roni et al. [6], Park et al. [7], Aranguren et al. [8], and Poudel et al. [9] have implemented Hub-and-Spoke MILP models to solve case scenarios involving BSC's. The application of their studies has revealed that MILP models are an efficient method to optimize the BSC and minimize cost.

The purpose of this paper is to incorporate machine-learning (ML) techniques to enhance a Hub-and-Spoke two-stage stochastic MILP model originally developed by Aranguren et al. [10] to minimize the total cost of a BSC. In fact, the hypothesis that is tested in this paper is that the hybrid methodology will improve the optimization of the BSC model in two ways: by increasing its solution quality (i.e., minimizing total cost) and by decreasing the computational burden of solving the stochastic MILP. To accomplish these goals, the ML algorithms are used to select potential storage depot locations to be used to solve the optimization model and that are beneficial enough to decrease the cost of the BSC. In addition, limiting the number of potential depots considered to solve the optimization model is capable of decreasing the computational burden. The ML algorithms consider the distance from the parcels to the coal-powered plants, the average biomass yield, and the levels of biomass moisture and ash in the parcels. On the other hand, while the problem of applying ML into optimization problems is that the optimal training labels are not easily accessible in most cases [11], the training labels for the ML algorithms are obtained from the optimization model solved with CPLEX.

The contributions of this paper are two-fold. From the methodological point of view, this paper illustrates the ability of ML to improve the optimization of a large-scale Hub-and-Spoke stochastic MILP model, including a quantitative performance comparison among well-established ML algorithms, such as Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Multi-Layer Perceptron (MLP) Neural Network. From the practical point of view, the incorporation of ML is tested on a realistic case study of a BSC developed by Aranguren et al. [10] in the northeast area of the United States.

This paper is organized as follows: Section 2 reviews relevant works related to the topic of combining ML and optimization methods to enhance the solutions. Section 3 provides a brief overview of the ML algorithms used in this paper to improve the optimization model. Section 4 reviews the BSC case study that is used to evaluate the performance of ML. Section 5 presents the results of the incorporation of ML. Section 6 provides a conclusion and suggestions for future work.

2. Literature Review

This section reviews relevant articles related to the topic of integrating ML methods into optimization problems in the area of Operations Research (OR). In fact, the application of ML to optimization problems has been the focus of computer science research since the 1980s. In 1999, Smith [12] stated that researchers had been attempting for over a decade to make neural networks competitive with meta-heuristics models. The author concluded that more research was needed on the hybridization of neural networks with meta-heuristics, such as genetic algorithms and simulated annealing, to take advantage of each technique, along with more applications using neural networks, since most of the research conducted at the time was focused on the solution of the traveling salesman problem (TSP).

More recently, Bengio et al. [13] surveyed different attempts to use ML algorithms to solve optimization problems, where the algorithms usually rely on hand-crafted heuristics to make decisions that are otherwise too expensive to compute or not well defined mathematically. Therefore, Bengio et al. [13] propose the integration of ML and optimization to obtain solutions and make decisions without the computational burden of meta-heuristics.

A method to incorporate ML into discrete optimization problems is to train the ML algorithm to output direct solutions to solve the problem. For instance, Bello et al. [11] present a combinatorial optimization problem to train a Recurrent Neural Network (RNN) to solve the traveling salesman problem (TSP), where the parameters of the RNN are optimized through reinforced learning and the policy gradient method. Given a set of city coordinates, the RNN predicts a distribution over different city permutations using negative tour length as the reward signal. Despite the computational expense, the RNN achieves close to optimal results on 2D Euclidean graphs with up to 100 nodes.

On the other hand, in most OR applications, obtaining direct solutions from ML algorithms without the aid of optimization models is not the most suitable way to solve the problem. For example, Larsen et al. [14] propose a methodology that combines ML and OR in a way that resembles this paper. The authors set up the problem as a two-stage optimal prediction stochastic program whose solution they predict with a Multi-Layer Perceptron (MLP) Neural Network and a Logistic Regression (LR) model. To generate the training data for the MLP and LR, Larsen et al. [14] sample operational problem instances with probabilistic sampling, which are solved independently through an existing CPLEX solver. After applying their methodology to a train load planning problem, the authors conclude that the regression MLP neural network model has the best performance to predict the solutions with high accuracy and in a shorter time than with the use of an ILP solver. The ILP CPLEX solver yields slightly better results, but the computational time of the ILP is significantly larger. Although the focus of this study is the train load planning problem model, the study is a great example of how an existing ILP CPLEX solver can be used to obtain the training labels for a ML algorithm.

Other works have also applied different combinations of ML and optimization models to enhance the solution. For instance, Marjani et al. [15] use a coupled Genetic Algorithm (GA) and a Particle Swarm Optimization (PSO) technique to supervise a feed forward neural network, where the connections of layers and topology of the initial neural network are tracked by the GA, and numerical values of biases and weights are examined by the PSO to modify the optimal network topology. Their initial neural network converges to optimal topology in seven iterations, proving that the combination of both ML and optimization algorithms to solve a problem yields efficient solutions. Another study that combines ML and optimization models is a study conducted by Mahmood et al. [16], where the authors implement a Generative Adversarial Network (GAN) approach to predict a 3D dose distribution for desirable treatment plans concerning radiation therapy. Sixty percent of total CT images from patients with cancer who had undergone radiation therapy are used to train the GAN model, which is then used to predict high-quality dose distributions for out-of-sample patients. The predictions obtained from the GAN model are used as inputs for optimization models producing deliverable plans. Therefore, in the study by Mahmood et al. [16], ML is used to provide valuable information that is incorporated into optimization models to improve the solution.

Although the implementation of neural networks is the most common approach while using ML for optimization, Lin et al. [17] applied a Classification and Regression Tree (CART) model and a Random Forest (RF) model to obtain an approximation close to the MILP solution without actually solving the complete NP-hard MILP problem, which centers around the electricity generation schedule. Lin et al. [17] use the Linear Programming Relaxation (LPR) solutions as input parameters to implement the ML algorithms, and they find that applying a Regression Decision Tree allows them to obtain highly accurate approximations, which is an efficient alternative to solving the MILP directly because of its computational burden.

Perhaps more relevant to the topic of supply chains is a study conducted by Gumus et al. [18], where the authors develop a supply chain model for a beverage company by implementing an integrated Neuro-Fuzzy and MILP approach. The Neuro-Fuzzy algorithm is used for demand forecasting, and its outputs serve as inputs for the MILP, which solves the problem and designs the supply chain. The output of the Neuro-Fuzzy model is also used in an Artificial Neural Network (ANN) to study the applicability of solving a supply chain problem with ML. In their case study, Gumus et al. [18] implement an MLP Neural Network, and according to their results, while the MILP performs better than the MLP, the MILP is time consuming, so the authors conclude that the MLP can be used to obtain reliable results without the computational burden. Table 1 summarizes the closely related previous works and situates our paper with respect to the literature.

Table 1. Machine Learning and Operations Research Literature Review.

References	Methods Used	Notes
Bello et al. (2017)	RNN	Traveling salesman problem
Larsen et al. (2019)	MLP, LR, ILP CPLEX	CPLEX used to obtain training labels for MLP and LR
Marjani et al. (2016)	MLP, GA, PSO	GA and PSO used to supervise the MLP
Mahmood et al. (2018)	GAN, IO pipeline	GAN results used as inputs for optimization
Lin et al. (2019)	MILP, LPR, DT, RF	LPR solution as input parameters to solve MILP with ML
Gumus et al. (2009)	Neuro-Fuzzy, MILP, MLP	Neuro-Fuzzy used to create inputs for MILP and MLP
This paper	LR, DT, MLP, MILP CPLEX	CPLEX used to create training labels for ML

3. Background on Selected Machine-Learning Methods

According to Bengio et al. [13], some advantages of incorporating ML techniques into optimization is that ML can improve current solution methods, and it is also able to replace some heavy computations by a fast approximation. Even though ML is approximate, this does not mean that incorporating ML into optimization problems will compromise overall theoretical guarantees [13]. This work proposes the application of four ML algorithms to select potential depot locations for the BSC, which are then used to solve the Hub-and-Spoke optimization model originally developed by Aranguren et al. [10]. The first ML algorithm applied to the problem is Logistic Regression (LR), which is a classification algorithm that relies on the implementation of a function relating the independent (predicting) variables to the dependent (outcome) variable [19]. Once the function is obtained, the first step to classify a new unlabeled observation is to calculate its probability of belonging to each of the two classes [19]. The estimate obtained is denoted as P(Y = 1), which refers to the probability that the new observation belongs to the class label 1. Since the objective is to calculate probabilities, we use the Sigmoid function 1 to build the relationship between the inputs and the output, which yields values between [0, 1].

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n)}} \quad (1)$$

Once the probability of P(Y = 1) is calculated for each new observation, the second step is to establish a cutoff value that will serve as a threshold to classify each observation into one of the two classes. For instance, a popular initial cutoff value for binary cases is 0.5, where an observation with an estimated P(Y = 1) > 0.5 will be classified as belonging to class label 1, while another observation with a P(Y = 1) < 0.5 will be classified as belonging to class label 0 [19]. However, during the application of LR, the value of the threshold needs to be adjusted because the most optimal cutoff value depends on the nature of the problem at hand.

The second ML algorithm applied is a Decision Tree (DT), which consists of partitioning the input training space into distinct and non-overlapping regions after following certain test conditions that identify regions that have the most homogeneous response to the predictor [20]. More specifically, a DT is a flowchart structure containing the following components: root node, internal nodes, and leaf (also called terminal) nodes. The root node, which is the topmost node in the DT flowchart, and the internal nodes contain attribute test conditions to separate sample points until they belong to the

same class label [21]. Furthermore, to build one tree, all features in the training set are considered to split the data into branches, which means that the algorithm automatically performs feature selection. Moreover, another important aspect of building an effective DT classifier is selecting the method to determine impurity. To measure the performance of the test conditions in the nodes of the tree, we need to compare the impurity of the node before and after the split. In this work, we use the Gini index. However, one issue with trees is that pure subsets tend to overfit the training data [21], so in some cases, it is beneficial to stop the algorithm early.

A Random Forest (RF) is also included in this work, which usually consists of 50–100 individual trees, and the final classification decision of the RF for each new observation is the decision of most of the existing trees [22]. In addition, the RF uses the concept of bagging, which means that only a sample of the total training points are used to train each tree, and only a subset of the total input variables are considered to build each tree [22]. However, for this exact reason, a disadvantage is that if a variable is significantly more important than the others to be able to classify an observation with the correct class label, a RF will include trees that do not contain the most significant variable.

The last ML algorithm applied in this paper is a Multi-Layer Perceptron (MLP) Neural Network. The MLP consists of neurons arranged in three or more layers: an input layer that includes the data set, at least one hidden layer, and the final layer containing the outputs [23]. The neurons in the MLP (also called perceptrons in ANN applications) are responsible for providing a path for the information to flow through and to learn from examples to make predictions based on the trends present in the available data [24]. In the hidden layers, each input signal is multiplied by the initial weight of each input. If the final sum ($\sum WiXi$) is greater than a certain threshold value, the neuron activates and sends information to the next layer [24]. The MLP classifier uses a technique called backpropagation to compute the gradient descent of the cost function to update and define all weight parameters during the learning step [25]. Additionally, the threshold that determines if a neuron should be activated is established by the activation function, which introduces non-linearity into the algorithm. In this paper, the activation function used to train the MLP is the hyperbolic tangent function, which is similar to the Sigmoid function, except that the hyperbolic tangent yields values between $[-1, 1]$. Hyperbolic tangent functions often converge faster than Sigmoid functions because of their zero-centered nature [24]. Additionally, since the interval is $[-1, 1]$, any strongly negative inputs will result in actual negative outputs during training [24].

4. Hub-and-Spoke Stochastic MILP Optimization Model

This section presents the mathematical formulation of the Hub-and-Spoke two-stage stochastic MILP model and describes the data inputs for the realistic case study in the northeastern area of the United States developed by Aranguren et al. [10]. The biomass network consists of three sets of nodes, with the first set of nodes representing the parcels, the second set representing the depots, and the third set representing the coal-powered plants. A visual representation of a simplified Hub-and-Spoke model can be seen in Figure 1.

In Figure 1, the arcs T1 represent the possible connections between the parcels and the depots, the arcs T2 represent the possible connections between the depots and the power plants, and the arcs T3 represent the possible connections between the parcels and the power plants. The Hub-and-Spoke model aims to find the parcels to be used to grow the biomass, the depots to be used to store the biomass, and the distribution network that minimizes the total BSC cost [10]. The design variables, the problem parameters, the objective function, and the constraints can be seen below.

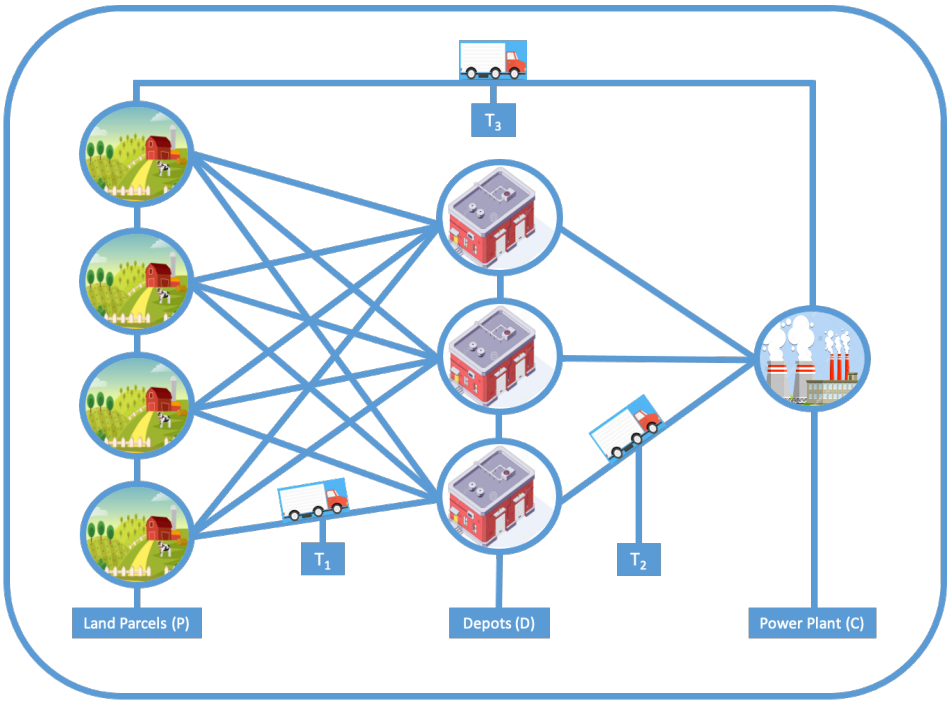

Figure 1. Hub-and-Spoke Model.

Design Variables:

- $p(o)$: scenario probability for $o \in \Omega$
- $X_{ij}(o)$: flow along arc $(i,j) \in T_1$ from parcel (P) to depot facility (D)
- $Y_{jk}(o)$: flow along arc $(j,k) \in T_2$ from depot facility (D) to coal power plant (C)
- $Z_{ik}(o)$: flow along arc $(i,k) \in T_3$ from parcel (P) to coal power plant (C)
- $\pi(o)$: third-party biomass supply
- W_j: binary variable-1 if $j \in D$ is used, and 0 otherwise

Problem Parameters:

- $c_{ij}^{T_1}$: cost charged per metric ton shipped along $(i,j) \in T_1$
- $c_{jk}^{T_2}$: cost charged per metric ton shipped along $(j,k) \in T_2$
- $c_{ik}^{T_3}$: cost charged per metric ton shipped along $(i,k) \in T_3$
- ξ_j: fixed investment cost to install a depot at node $j \in D$
- ζ: cost per metric ton of biomass from third-party
- u_j: storage capacity of depot facility $j \in D$
- m_i: moisture level of biomass supply from parcel $i \in P$
- s_i: supply of biomass at parcel location $i \in P$
- d_k: biomass demand at coal power plant $k \in C$

$$Min: \sum_{j \in D} \xi_j W_j + \sum_{o \in \Omega} p(o)[\sum_{i \in P} \sum_{j \in D} c_{ij}^{T_1} X_{ij}(o) + \sum_{j \in D} \sum_{k \in C} c_{jk}^{T_2} Y_{jk}(o) + \sum_{i \in P} \sum_{k \in C} c_{ik}^{T_3} Z_{ik}(o) + \zeta \pi(o)] \quad (2)$$

Subject to:

$$\sum_{j \in D} X_{ij}(o) + \sum_{k \in C} Z_{ik}(o) \leq s_i(o) \quad \forall i \in P \quad o \in \Omega \quad (3)$$

$$\sum_{i \in P} (1 - m_i) X_{ij}(o) = \sum_{k \in C} Y_{jk}(o) \quad \forall j \in D \quad o \in \Omega \quad (4)$$

$$\sum_{i \in P} X_{ij}(o) \leq u_j W_j \quad \forall j \in D \quad o \in \Omega \quad (5)$$

$$\sum_{j \in D} Y_{jk}(o) + \sum_{i \in P} (1 - m_i) Z_{ik}(o) + \pi(o) = d_k \quad \forall k \in C \quad o \in \Omega \tag{6}$$

$$(i, j) \in T_1 \quad (j, k) \in T_2 \quad (i, k) \in T_3 \tag{7}$$

$$X_{ij}(o), Y_{jk}(o), Z_{ik}(o), \pi(o) \in R^+ \quad o \in \Omega \tag{8}$$

$$W_j \in \{0, 1\} \quad \forall j \in D \tag{9}$$

This paper uses two scenarios involving future climate, which affect the biomass yield. Therefore, the stochastic variables are related to the flow of biomass: $X_{ij}(o)$, $Y_{jk}(o)$, and $Z_{ik}(o)$. The decision variables include biomass flow and depot location. The objective function (2) serves to minimize the cost for harvesting, processing and transporting the biomass, and the depot investment cost. Constraint (3) restricts the supply from parcels to the maximum yield, constraint (4) guarantees a mass balance in depots, (5) limits depot capacity, (6) satisfies the demand of the coal-powered plants, (7) sets arc definitions, (8) guarantees a non-negative flow, and (9) sets the binary limitations for depot selection variables.

5. Machine-Learning Methods

This section reviews the process of applying the proposed hybrid method of ML and optimization to the BSC case study, and it evaluates the improvements accomplished with the use of the methodology.

5.1. Algorithm Training

The data for the four input variables in this study was obtained from the case study developed by Aranguren et al. [10]. There are a total of 3750 parcel locations (P) available, and 11 coal-powered plants (C) are considered. For the training data, the distance between each of the parcel locations (P) and the closest coal plant (C) is taken into consideration. To calculate the potential biomass yield (s_i) in each parcel location, Aranguren et al. [10] used the Agricultural Land Management and Numerical Assessment Criteria (ALMANAC) simulation model. In their work, Miscanthus was used in the design of the BSC due to its high yield density. The level of moisture and the level of ash in the Miscanthus supply was calculated from field sample data from the Idaho National Lab. Because moisture increases in coastal areas and decreases toward in-land, a linear regression model was used to obtain moisture levels for the rest of the locations. The costs related to harvesting, collection, and transportation are obtained from the work of Aranguren et al. [8].

To build the training set for the ML algorithms and generate the training labels, the optimization model was solved three times using the IBM CPLEX solver considering 100 random parcel locations per run to obtain a total of 300 training sample points. The randomness in the selection of sample points for the training set makes the experimentation portion more robust and decreases the chance of overfitting the training data. The locations that CPLEX selected as storage depots in each solution were labeled as potential depot location (1), while the rest of the locations in the run were labeled as not beneficial as potential depot (0).

As opposed to standard ML applications, where the available data is divided into training (80%) and test (20%) sets, the training set in this work consists of the 300 random locations with the training labels. If we were to divide the available data 20:80, 60 sample points would be lost from the training set, resulting in a smaller number of samples to train. Moreover, solving the optimization model with more parcel locations and including those solutions to train the ML algorithms would increase their dependency on the optimization model. Additionally, the test set in this work consists of the total 3750 parcel locations. Therefore, after the ML algorithms were trained using the 300 random sample points, the algorithms were applied to the 3750 parcel locations to select potential depot locations from all the available locations. Then, each set of potential depots obtained from ML was used to optimize the optimization model with CPLEX, and the total cost yielded by each set of depots was obtained.

Considering the fact that in this type of optimization problem the labels of the points are not available, the main metric to assess the performance of the ML algorithms was the decrease in total investment.

It is important to note that after obtaining the training labels for the 300 random parcel locations through CPLEX, only 35 locations were selected as optimal. These results create an imbalance in the training labels, with the locations classified as beneficial being underrepresented in the data set. Therefore, to avoid issues during the training of the ML algorithms, the technique of Synthetic Minority Over-Sampling (SMOTE) was applied to the data. This technique generates synthetic minority examples to over-sample the minority class, and for every minority sample, its five nearest neighbors of the same class are calculated to generate synthetic samples between the minority sample and its nearest neighbors [26].

The baseline comparison used in this paper is the total cost of a set of 203 potential depot locations obtained from a heuristic developed for the optimization model by Aranguren et al. [10], which yielded a total cost of $113,000,000. In addition, the computational burden of solving the Hub-and-Spoke model with CPLEX while considering these 203 depots was 168.51 s. To assess the performance of the incorporation of ML into the MILP model, the total cost and computational burden obtained from each solution were compared to the values in the baseline. The use of the depots selected by CPLEX in each solution was also considered to compare the performance of the algorithms.

The four ML algorithms were coded in Python 3 using the Scikit-learn (sklearn) library for ML and statistical modeling. The experimentation portion of this paper was completed using a personal computer with an Intel Core i5 with a processor of 2.4 GHz and 8 GB of memory.

5.2. Logistic Regression (LR)

The first ML technique applied to the problem of selecting potential depot locations was a LR algorithm. After training the algorithm using the set of 300 random locations, the following logistic Sigmoid function (10) was built.

$$y = \frac{1}{1 + e^{-(0.00474 + 0.0359x_1 + 0.1209x_2 + 0.0027x_3 - 0.0079x_4)}} \tag{10}$$

- x_1: Level of Moisture
- x_2: Level of Ash
- x_3: Average Biomass Yield
- x_4: Distance to Closest Plant

The *p*-values were also calculated to determine which variables were the most significant to select potential depot locations that are beneficial. The results can be seen in Table 2.

Table 2. *p*-Values of Each Variable.

Variable	*p*-Value
Moisture Level	0.0093
Ash Level	0.1747
Average Biomass Yield	0.8891
Distance to Plant	0.0000

The *p*-values of the moisture level in the biomass and the distance from the parcel location to the closest coal-powered plant are both less than 0.05, which is the standard value of alpha. Therefore, we can conclude that the level of moisture and the distance to the plants are significant to the selection of potential depot locations. On the other hand, the average expected biomass yield is the least significant variable, followed by the level of ash in the biomass.

To limit the number of potential depots selected by the LR, the threshold value had to be tuned. When the threshold was set to an initial standard value of 0.5, the algorithm selected a total of 1582 locations as potential depot locations. Since the purpose of the paper is to decrease

the computational burden of the initial baseline model, we want the number of potential depots to be close to the potential depots considered in the initial baseline study-203 depots. Solving the optimization model considering 1582 potential depots would increase the computational time significantly. Therefore, three different threshold values were tested: 0.65, 0.675, and 0.70. After the three sets of potential depots were obtained by the LR, each set was used to solve the optimization model with CPLEX. A summary of the results of each threshold value, the amount of potential depot locations selected by the LR, the number of final depots selected by CPLEX, the total cost of each set, and its computational burden can be seen in Table 3.

Table 3. Cost of Each Solution Obtained from Logistic Regression.

Threshold	Num. of Potential Depots	Num. of Depots CPLEX	Total Cost $	Burden (s)
0.65	257	14	108,810,000	266.80
0.675	153	13	110,120,000	84.00
0.70	72	10	112,400,000	22.12

According to the results and the total cost of each set of potential depot locations, a threshold value of 0.65, which selects 257 potential depots, creates the largest decrease in the total cost of the BSC. Considering that the baseline cost obtained by solving the optimization model without ML was $113,000,000 [10], we can conclude that the application of LR to obtain potential depot locations has improved the optimization of the BSC and has decreased the total cost by 3.71%, which proves the first part of the hypothesis of this work (increase in solution quality). It is important to note that the total cost decreases as the amount of potential depot locations taken into consideration increases. In fact, the larger the amount of potential depots that the optimization model considers, the more likely it is that the CPLEX solver will find beneficial depots that lower the cost. However, the disadvantage of this behavior is that the computational burden of optimizing the BSC increases significantly as the amount of potential depot locations increases. Nevertheless, it is important to note that the LR has yielded an acceptable solution, where the total cost has decreased by only 0.23%, but the computational burden has decreased by 86.87% compared to the baseline results.

To better understand the results and the performance of the LR algorithm, the total cost of the BSC of each set was broken down into two elements: the investment depot cost and the transportation cost to take the biomass supply to the plants, which is further broken down into the cost of taking the biomass supply from the parcels to the depots and then from the depots to the plants. If there is not a depot that is beneficial enough for a parcel location, then the biomass is taken directly to the plant. The details can be seen in Tables 4–6.

Table 4. Detailed Cost of Solution Obtained with 257 Potential LR.

14 Depots Selected by CPLEX	Transportation Cost	Depot Cost	Total Cost
Parcel to Depot	86,142,337.71		
Depot to Plant	5,702,266.95		
Parcel to Plant	12,304,351.22		
Total $	104,148,955.89	4,662,000.00	108,810,955.89

Table 5. Detailed Cost of Solution Obtained with 153 Potential LR.

13 Depots Selected by CPLEX	Transportation Cost	Depot Cost	Total Cost
Parcel to Depot	86,438,450.73		
Depot to Plant	6,085,995.22		
Parcel to Plant	13,602,134.49		
Total $	106,126,580.45	4,329,000.000	110,455,580.45

Table 6. Detailed Cost of Solution Obtained with 72 Potential LR.

10 Depots Selected by CPLEX	Transportation Cost	Depot Cost	Total Cost
Parcel to Depot	81,744,437.78		
Depot to Plant	4,322,836.18		
Parcel to Plant	23,340,497.23		
Total $	109,407,771.19	3,330,000.00	112,737,771.19

The increase in depots selected by CPLEX increases depot investment cost. However, the increase in potential depots allows CPLEX to select the most beneficial depot locations that lower the transportation cost. For instance, in the set of 72 potential depots, the transportation cost of taking the biomass from parcels to plants directly without the use of a depot is higher than the rest, which means that the 10 depots selected by the Hub-and-Spoke optimization model in this case are not efficient enough to be used for certain parcel locations. On the other hand, in the set of 257 potential depots, which yields the lowest total cost obtained through LR, the cost of transportation from parcel to plant is lower because there are fewer parcel locations without a beneficial depot location, which proves that the depots selected from the 257 potential depots are more efficient than the depots selected from the 72 potential depots. Therefore, even though the amount of depots selected by CPLEX has increased, the 14 depots obtained from the set of 257 potential depot locations are the most beneficial in terms of lowering the transportation cost. In fact, the decrease in transportation cost is so significant that even with the increase in the depot investment cost, this set still yields the lowest total cost.

Ultimately, the choice of which set is more adequate depends on several factors-the decision-maker, the particular industry, and the actual investment cost of the depots. A set with fewer depot locations is more beneficial if the depot cost is considerably higher, while a set with more depots is beneficial when the depot cost is lower. The advantage of applying LR to select potential depot locations is that the algorithm can yield different amount of potential depots, all of which result in high-quality solutions.

Furthermore, as seen in Figure 2, a map was created to illustrate the location of the 14 depots selected by CPLEX in relation to the power plants considered in this study.

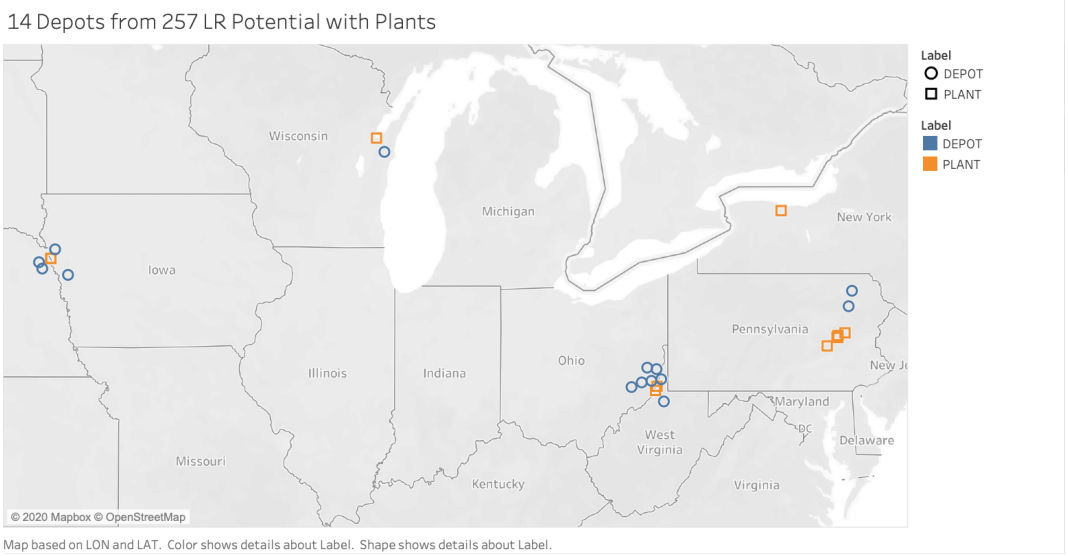

Figure 2. 14 Depots Selected from 257 Potential Depots with Plants.

5.3. Decision Tree

The second ML algorithm that was used to select beneficial potential depot locations was a DT. After training the algorithm with the same set of 300 random locations, the first DT classifier selected 813 parcel locations as potential depots. As discussed in Section 3, an efficient way to limit the amount of depot locations selected by the DT and avoid overfitting the training data into completely pure subsets is to stop the algorithm early. To accomplish this goal, the parameter that was tuned during the training phase was the minimum impurity decrease. To get a clear idea of how the tuning of this parameter affects the amount of potential depot locations selected by the DT, the plot in Figure 3 was created to test threshold values in the range of [0.001, 0.1].

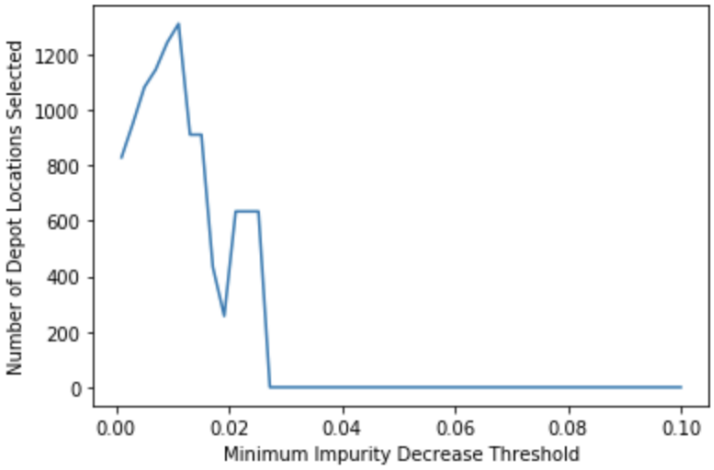

Figure 3. Minimum Impurity Decrease Threshold vs. Number of Depots Selected.

As seen in Figure 3, the most adequate threshold value in this application was 0.02, which selected 257 potential depot locations. The DT created with the minimum impurity decrease threshold value set to 0.02 is seen in Figure 4.

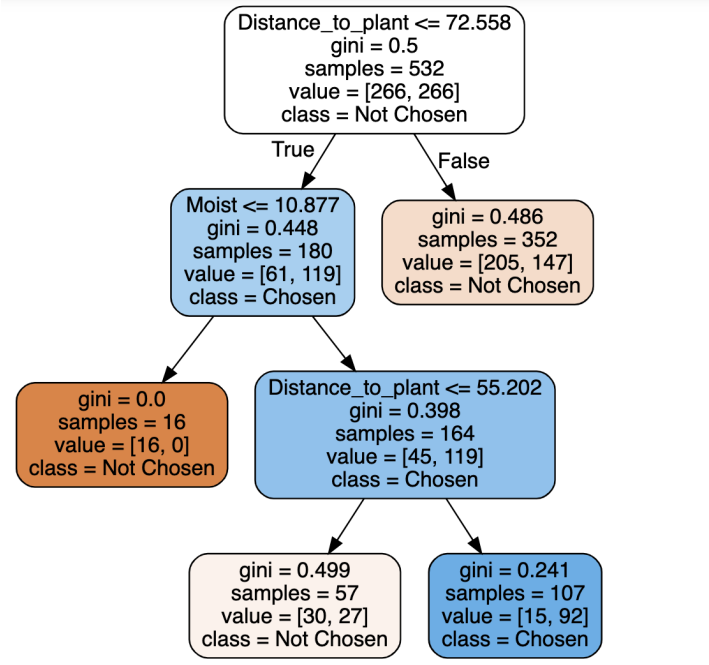

Figure 4. Decision Tree to Select Potential Depots.

As seen in Figure 4, two variables were considered to build the tree and classify new points: the distance from the parcel location to the closest coal-powered plant and the level of moisture in the biomass. The other two variables-level of ash and expected biomass yield-were not taken into consideration to classify new locations. This conclusion is the same as the one obtained from the *p*-values calculated through LR.

Furthermore, the set of 257 potential depots obtained from the DT was used to solve the optimization model with CPLEX. The results are shown in Table 7.

Table 7. Cost of Solution Obtained from Decision Tree.

Min Impurity	Num. of Potential Depots	Num. of Depots CPLEX	Total Cost $	Burden (s)
0.02	257	11	108,980,000	222.83

According to the results, we can conclude that the application of the DT algorithm to select potential depot locations has also improved the Hub-and-Spoke model by decreasing the total cost of the BSC by 3.55% when compared to the baseline cost. Therefore, the DT has also proven the first part of the hypothesis. The resulting total cost is similar to the lowest total cost obtained from the LR. In fact, the total cost has increased by only 0.156% when comparing the potential depots obtained from DT to the potential depots obtained from LR. Therefore, we can conclude that both ML techniques perform well to select potential depot locations that lower the total cost of the BSC to a similar value. However, the same disadvantage remains-the computational burden of solving the stochastic MILP model increases as the number of potential depot locations increases. Since the DT yielded only one set of potential depots and its computational time was high when compared to the ones obtained from LR, we can conclude that the DT has not proven the second part of the initial hypothesis addressing a decrease in computational burden.

The two elements of the total cost of the BSC-the transportation cost and the depot investment cost-are seen in more detail in Table 8.

Table 8. Detailed Cost of Solution Obtained with 257 Potential DT.

11 Depots Selected by CPLEX	Transportation Cost	Depot Cost	Total Cost
Parcel to Depot	74,494,070.36		
Depot to Plant	8,246,198.74		
Parcel to Plant	22,585,219.95		
Total $	105,325,489.04	3,663,000.00	108,988,489.04

Even though the solution from the LR and the DT selected the same amount of potential depots and resulted in a similar total cost, the DT result is different when the details are analyzed. For instance, the transportation cost from parcels to plants in the DT result has increased by 83.55% when compared to the one from LR, which indicates that there are more parcel locations that could not find an efficient depot in this case. However, the total cost is almost as low as the one from LR because CPLEX has selected a fewer amount of depots, which causes a decrease of 21.34% in the investment depot cost. This decrease, along with the decrease of 13.52% in transportation cost from parcels to depots and the decrease of 30.85% in transportation cost from depots to plants, create a balance and lower the total cost to around the same value, which indicates that this solution is also beneficial. In fact, the solution is specifically advantageous when it is beneficial for a particular industry or decision-maker to have fewer depots, or when the depot cost is higher. Therefore, we can conclude that both the LR and the DT algorithms perform well to select potential depot locations that decrease the total cost of the BSC, with the final choice between the two depending on the particular supply chain.

The location of the 11 depots selected by CPLEX from the 257 potential depots obtained by the DT, along with the location of the plants, can be seen in Figure 5.

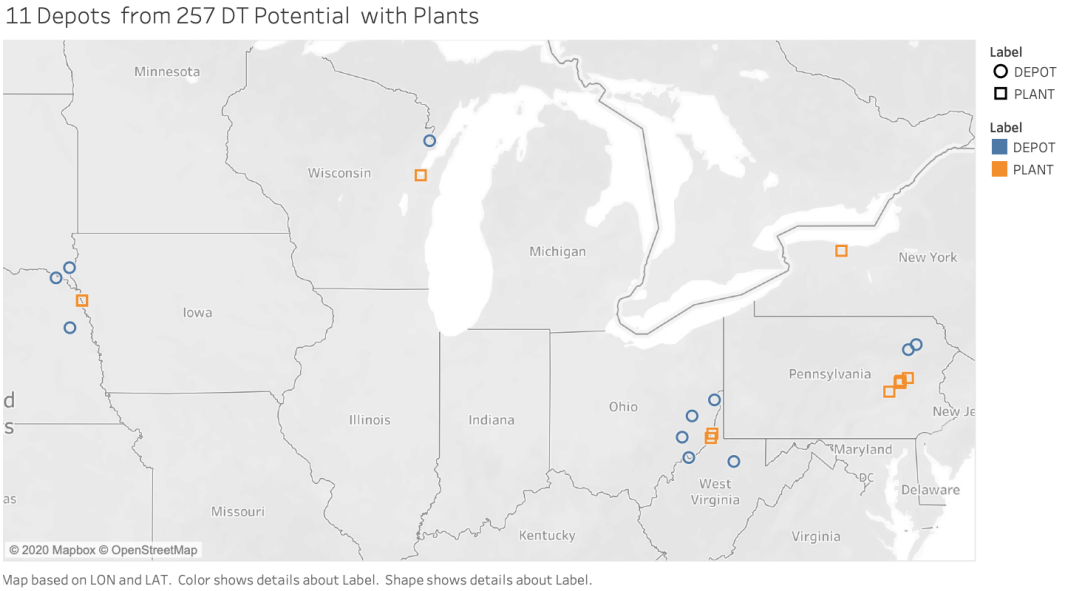

Figure 5. 11 Depots Selected from 257 Potential Depots with Plants.

A Random Forest (RF), which is an extension of a DT, was also applied in this work. However, after exploring different parameters, such as the minimum impurity decrease, the number of trees, and the maximum number of leaf nodes, the number of potential depots did not decrease to an amount similar to the baseline. The lowest number of potential depots obtained from the RF was 450, which would have been too computationally intensive to be solved in the optimization model and beat the baseline result. Therefore, we conclude that the RF algorithm is not an appropriate method to select potential depot locations for this Hub-and-Spoke case study. By building multiple trees, the RF algorithm can consider different selections of observations and features, but as the results of the LR and the DT indicate, only two variables are significant for accurate classification. Since each tree is built based on a smaller subset of variables instead of considering all variables as a whole, the algorithm did not distinguish the importance of the level of moisture and the distance to the plants. In an attempt to solve this issue and reduce the number of potential depots, another RF algorithm was built considering only the level of moisture and the distance. However, during the tuning phase of the parameters, similar results to the ones previously described were obtained, and the number of potential depots selected did not decrease. Another possible reason the RF might not be an appropriate method for the problem at hand is that RF is commonly used for problems where large training sets are available [22]. However, because of the nature of this problem, obtaining more training labels would increase the dependency of the algorithm on the stochastic MILP model. Therefore, according to our findings, we can conclude that the RF is not an appropriate algorithm to be applied in this Hub-and-Spoke model, and a single tree works best.

5.4. Multi-Layer Perceptron Neural Network

The fourth ML algorithm that was applied was a Multi-Layer Perceptron (MLP) Neural Network. The main difference between the previous ML algorithms and the MLP is that only two features were considered during training: the level of moisture in the biomass of each parcel and the distance from each parcel to the closest coal-powered plant. According to previous results, these two variables are the most significant to select potential depot locations that are beneficial enough to decrease the cost. This decision was made because the MLP considering the four variables led to issues with convergence. Therefore, the output of the two previous ML algorithms was used, with both the LR and DT demonstrating that the most significant variables in the study were the level of moisture and distance. In addition, before training the MLP, the features in the training set were standardized through the standard scaler available in the Scikit-learn library, which normalizes the features of the

set individually. After building the MLP with and without standardizing the features, it was found that the MLP converged faster after applying the standardization technique.

The initial solution generated by the MLP using a multi-start method selected a large range of potential depot locations. Therefore, several parameters had to be tuned to limit the number of potential depots: the number of hidden layers, the number of neurons in each hidden layer, the solver selected for weight optimization, the penalty value alpha, and the activation function for the hidden layer. After using the three different solvers available in Sklearn ('sgd', 'adam', and 'lbfgs'), it was found that the MLP tended to select a large number of potential depots and had issues converging when either the 'sgd' or 'adam' solvers were used. Therefore, we decided to set the solver to 'lbfgs', which is a quasi-Newton optimizer. In addition, several activation functions were tested-the identity, logistic, hyperbolic tangent, and rectified linear unit function. It was found that while the MLP selected a large number of potential depots with most of the activation functions, the hyperbolic tangent function was capable of decreasing the number of potential depots to an amount close to our baseline.

Once those two parameters were set, different combinations of layers, number of neurons, and penalty values were tested to find a selection of potential depot locations that could be used in the Hub-and-Spoke model to improve the result. After conducting the experiment with different values, the most beneficial solutions were obtained with three hidden layers, the number of neurons per layer in the range [70, 90], and the penalty value alpha in the range [0.00001, 0.00009]. The best results obtained by the MLP are seen in Table 9.

Table 9. Cost of Each Solution Obtained from MLP Neural Network.

Layers	Alpha	Num. of Potential Depots	Num. of Depots CPLEX	Total Cost $	CPLEX Burden (s)
(90,70,80)	0.00002	199	15	110,120,000	184.14
(90,80,80)	0.00008	227	15	108,218,000	252.66

The MLP has performed better than the best results obtained by the LR and the DT. In fact, the total cost has decreased by 4.23% in comparison to the baseline cost. Moreover, it is important to note that the number of potential depots selected by the MLP decreased by 11.67% when compared to the LR and the DT. Even if fewer potential depots were considered to solve the Hub-and-Spoke model, the potential depots selected by the MLP were so beneficial that the total cost decreased significantly. The transportation cost and the investment depot cost for the sets obtained from the MLP can be seen in more detail in Tables 10 and 11.

Table 10. Detailed Cost of Solution Obtained with 199 Potential MLP.

15 Depots Selected by CPLEX	Transportation Cost	Depot Cost	Total Cost
Parcel to Depot	89,986,439.15		
Depot to Plant	4,243,873.64		
Parcel to Plant	10,903,980.17		
Total $	105,134,292.95	4,995,000.00	110,129,292.95

Table 11. Detailed Cost of Solution Obtained with 227 Potential MLP.

15 Depots Selected by CPLEX	Transportation Cost	Depot Cost	Total Cost
Parcel to Depot	87,221,504.63		
Depot to Plant	5,173,258.38		
Parcel to Plant	10,828,603.09		
Total $	103,223,366.10	4,995,000.00	108,218,366.10

Despite the fact that solving the optimization model while considering the 227 potential depots yielded one of the largest amounts of selected depots (15) so far, the solution obtained with the MLP resulted in the lowest transportation cost in comparison to the other ML algorithms. Moreover, this result had the lowest cost of transportation to take the biomass directly from the parcels to the plants, which indicates that the depots obtained from the MLP are the most efficient of all the other sets of potential depots. In this case, more parcels find the depots efficient enough to be used rather than taking the supply directly to the plants. In addition, even though the investment depot cost is higher because of the larger amount of depots selected by CPLEX, the decrease in transportation cost is considerable thanks to the efficiency of the depots, which results in the lowest total cost. On the other hand, CPLEX selected the exact same amount of depots from the 199 potential depots, but the transportation cost of taking the biomass from the parcels to the depots is higher by 3.17%, which indicates that the locations selected in this case are not as cost-efficient as the 227 potential depots. This behavior is consistent with previous results, given the fact that it is more likely that CPLEX will find the most beneficial depots when a larger number of potential depots is considered. Therefore, we can conclude that the MLP is the most adequate ML algorithm in terms of selecting potential depot locations that decrease the total cost of the BSC.

In addition, a map illustrating the 15 depots selected by CPLEX from the 227 potential depots obtained from the MLP, along with the plants, can be seen in Figure 6.

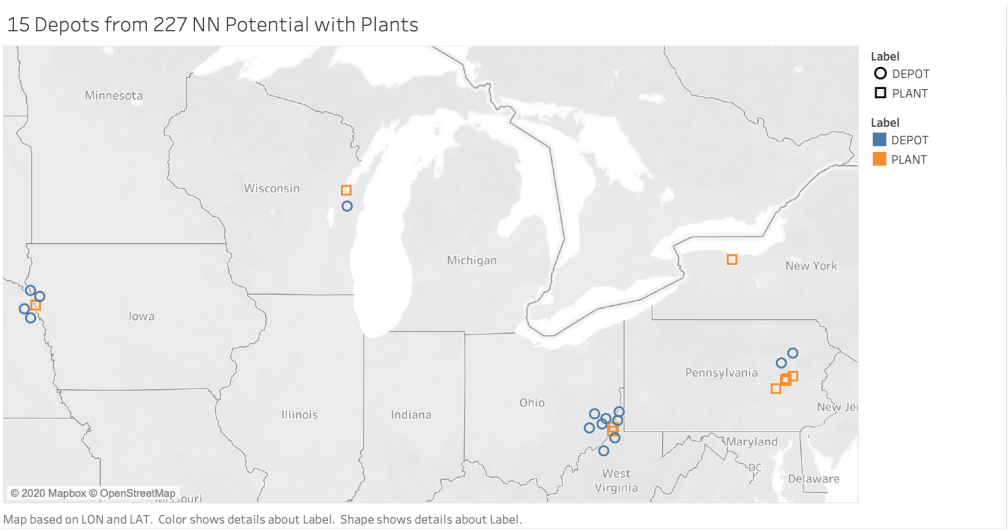

Figure 6. 15 Depots Selected from 227 Potential Depots with Plants.

5.5. Reducing the Number of Potential Depots Further

Despite the significant decrease in cost achieved from ML, it remains computationally intensive to obtain solutions from the optimization model due to the large amount of potential depots being considered. Therefore, the aim of this section is to prove the second part of the hypothesis—that the incorporation of ML to solve the Hub-and-Spoke stochastic MILP can reduce the computational burden while obtaining high-quality solutions.

After the tuning of the four ML algorithms, it was found that the MLP was the only algorithm capable of significantly reducing the amount of beneficial potential depots. During the tuning, most of the parameter remained the same: the weight optimizer solver was 'lbfgs,' the activation function used was the hyperbolic tangent function, the neurons in the hidden layers remained in the range [70, 90], and the penalty value in the range [0.00001, 0.00009]. However, it was precisely the last two parameters—the number of neurons per layer and the penalty value—that were further adjusted to reduce the number of potential depots selected. The best result obtained by the MLP with a significantly reduced number of potential depots is seen in Table 12.

Table 12. Cost of Each Solution Obtained from Neural Network.

Layers	Alpha	Num. of Potential Depots	Num. of Depots CPLEX	Total Cost $	Burden (s)
(90,70,80)	0.00008	13	5	111,090,000	2.36

This set of 13 potential depot locations has decreased the total cost by 1.69% when compared to the baseline cost. The previous results were able to lower the total cost in a more significant way, but the advantage of this particular set of 13 potential depot locations is that the computational burden while solving the optimization model has lowered significantly. Although the 227 potential depot locations selected by the previous MLP decreased the total cost to $108,218,000, the optimization problem took 252.66 s to optimize the BSC. Meanwhile, the total cost resulting from considering the 13 potential depots selected by MLP was $111,090,000, but the computational time decreased to 2.36 s, which means that the time has decreased by 99.07%. Therefore, this solution proves both parts of the hypothesis-this set of 13 potential depots obtained from the MLP is capable of both decreasing the total cost of the BSC and decreasing the computational burden of the Hub-and-Spoke problem at the same time. Although the computational burden is not extremely intense in this case study, this result demonstrates the ability of the MLP to limit the amount of potential depot locations while still obtaining high-quality solutions. This advantage is especially useful in cases where the minimum number of depots is desired by a particular industry, or in cases where the computational burden is extremely intensive.

After comparing the results of the ML algorithms applied in this paper, we can conclude that MLP is the ML algorithm that has the best performance when it comes to selecting beneficial potential depot locations that decrease the total cost of the presented Hub-and-Spoke BSC. The MLP was capable of selecting the potential depot locations that caused the largest decrease in total cost, and it was the only ML algorithm that reduced the amount of potential depots considered to decrease the computational burden of optimizing the problem with CPLEX. Therefore, the MLP has proven the hypothesis: the incorporation of ML into the optimization model has improved the solution by decreasing both the total cost of the BSC and the computational burden.

6. Conclusions and Future Work

We demonstrated that the application of ML techniques coupled with stochastic MILP optimization models enhances the solution quality and reduces the computational burden to tackle large-scale instances. This work included a quantitative performance comparison among four ML algorithms, and the hybrid method was tested using a realistic case study of a BSC. The variables considered to build the ML algorithms were the distance from the parcels to the closest power plant, the average biomass yield in the parcels, and the levels of moisture and ash of the biomass.

After applying the four ML algorithms, the MLP proved to be the most effective in terms of both decreasing the total cost of the BSC and the computational burden of solving the stochastic MILP. In fact, through the adjustment of different parameters during the training phase, the MLP algorithm was able to select 227 potential depot locations that were beneficial to the BSC and decreased the total cost by 4.23%. In addition, the parameters of the MLP algorithm were further adjusted to obtain a smaller set of potential depots to decrease the computational burden of solving the optimization model while also decreasing the total cost of the BSC. A set of 13 potential depot locations selected by the MLP yielded a solution that decreased the total cost by 1.69% while decreasing the computational burden by 98.6%. This result illustrated the ability of the MLP to decrease the number of depots considered while getting high-quality solutions. On the other hand, the LR and the DT also performed well in terms of decreasing the total cost of the BSC. Additionally, both algorithms revealed that the most significant variables to select beneficial potential depots were the distance from the parcels to the closest plant and the level of moisture in the biomass. Meanwhile, the fourth ML algorithm that was applied was a RF. However, the RF was not capable of reducing the number of potential depots in a way that the set

could be used to optimize Hub-and-spoke model and beat the baseline result. Hence, the conclusion according to this paper's findings is that the RF is not an appropriate method to be incorporated into a Hub-and-Spoke model.

The results obtained from the MLP to select potential depot locations for the Hub-and-Spoke stochastic MILP are promising. Therefore, this result should encourage researchers to venture into neural networks to improve supply chains in different industries, especially supply chains involving location selection. The investigation of the performance of neural networks in instances addressing a national network (large-scale problems) can be a future line of inquiry.

Author Contributions: Conceptualization, K.K.C.-V.; methodology, D.G. and M.A.; data curation, M.A.; writing—original draft preparation, D.G.; writing—review and editing, K.K.C.-V.; supervision, K.K.C.-V.; funding acquisition, K.K.C.-V. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work that is supported by the National Institute of Food and Agriculture (NIFA), U.S. Department of Agriculture (USDA), under the Hispanic Serving Institutions Education Grants Program, award no. 2020-38422-32258.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. U.S. Energy Information Administration. Electricity Explained: Electricity in the US. Available online: https://www.eia.gov/energyexplained/electricity/electricity-in-the-us.php (accessed on 20 July 2020).
2. World Coal Association. Coal & Electricity. Available online: https://www.worldcoal.org/reducing-co2-emissions (accessed on 20 July 2020).
3. U.S. Energy Information Administration. Coal Explained: Coal and the Environment. Available online: https://www.eia.gov/energyexplained/coal/coal-and-the-environment.php (accessed on 20 July 2020).
4. American Coal Council. Biomass Co-Firing With Coal as an Emissions Reduction Strategy. Available online: https://www.americancoalcouncil.org/page/biomass (accessed on 20 July 2020).
5. National Renewable Energy Laboratory. Biomass Co-firing: A Renewable Alternative for Utilities. Available online: https://www.nrel.gov/docs/fy00osti/28009.pdf (accessed on 20 July 2020).
6. Roni, M.S.; Eksioglu, S.D.; Searcy, E.; Jha, K. A supply chain network design model for biomass co-firing in coal-fired power plants. *Transp. Res. Part E* **2014**, *61*, 115–134. [CrossRef]
7. Park, Y.S.; Szmerekovsky, J.; Osmani, A. Integrated Multimodal Transportation Model for a Switchgrass-Based Bioethanol Supply Chain. *Transp. Res. Rec. J. Transp. Res. Board* **2017**, *2628*, 32–41. [CrossRef]
8. Aranguren, M.F.; Castillo-Villar, K.K.; Aboytes-Ojeda, M.; Giacomoni, M.H. Simulation-Optimization Approach for the Logistics Network Design of Biomass Co-Firing with Coal at Power Plants. *Sustainability* **2018**, *10*, 4299. [CrossRef]
9. Poudel, S.R.; Marufuzzaman, M.; Bian, L. A hybrid decomposition algorithm for designing a multi-modal transportation network under biomass supply uncertainty. *Transp. Res. Part E* **2016**, *94*, 1–25. [CrossRef]
10. Aranguren, M.F.; Castillo-Villar, K.K.; Aboytes-Ojeda, M. A Two-Stage Stochastic Model for Co-Firing Biomass Supply Chain Network. *J. Clean. Prod.* **2020**, submitted.
11. Bello, I.; Pham, H.; Le, Q.V.; Norouzi, M.; Bengio, S. Neural Combinatorial Optimization with Reinforcement Learning. *arXiv* **2017**, arXiv:1611.09940.
12. Smith, K.A. Neural Networks for Combinatorial Optimization: A Review of More than a Decade in Research. *Informs J. Comput.* **1999**, *11*, 15–34. [CrossRef]
13. Bengio, Y.; Lodi, A.; Prouvost, A. Machine Learning for Combinatorial Optimization: A Methodological Tour d'Horizon. *arXiv* **2018**, arXiv:1811.06128.
14. Larsen, E.; Lachapelle, S.; Bengio, Y.; Frejinger, E.; Lacoste-Julien, S.; Lodi, A. Predicting Tactical Solutions to Operational Planning Problems under Imperfect Information. *arXiv* **2019**, arXiv:1807.11876.
15. Marjani, A.; Shirazian, S.; Asadollahzadeh, M. Topology optimization of neural networks based on a coupled genetic algorithm and particle swarm optimization techniques (c-GA-PSO-NN). *Neural Comput. Appl.* **2016**, *29*, 1073–1076. [CrossRef]
16. Mahmood, R.; Babier, A.; McNiven, A.; Diamant, A.; Chan, T.C.Y. Automated Treatment Planning in Radiation Therapy Using Generative Adversarial Networks. *Proc. Mach. Learn. Res.* **2018**, *85*, 1–15.

17. Lin, X.; Hou, Z.J.; Ren, H.; Pan, F. Approximate Mixed-Integer Programming Solution with Machine Learning Technique and Linear Programming Relaxation. In Proceedings of the 2019 3rd International Conference on Smart Grid and Smart Cities (ICSGSC), Berkeley, CA, USA, 25–28 June 2019; pp. 101–107.
18. Gumus, A.T.; Guneri, A.F.; Keles, S. Supply Chain Network Design Using an Integrated Neuro-Fuzzy and MILP Approach: A Comparative Design Study. *Expert Syst. Appl.* **2009**, *36*, 12570–12577. [CrossRef]
19. Shmueli, G.; Patel, N.R.; Bruce, P.C. *Data Mining for Business Intelligence*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
20. Touzani, S.; Granderson, J.; Fernandes, S. Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy Build.* **2018**, *158*, 1533–1543. [CrossRef]
21. Tan, P.N.; Steinbach, M.; Karpatne, A.; Kumar, V. *Introduction to Data Mining*, 2nd ed.; Pearson: London, UK, 2018.
22. Williams, G. *Data Mining with Rattle and R*; Springer: New York, NY, USA, 2011.
23. Zhai, X.; Ali, A.A.S.; Amira, A.; Bensaali, F. MLP Neural Network Based Gas Classification System on Zynq SoC. *IEEE Access* **2016**, *4*, 8138–8146. [CrossRef]
24. Goyal, M.; Goyal, R.; Reddy, P.V.; Lall, B. Activation Functions. *arXiv* **2020**, arXiv:1906.09529.
25. Sun, X.; Ren, X.; Ma, S.; Wei, B.; Li, W.; Xu, J.; Wang, H.; Zhang, Y. Training Simplification and Model Simplification for Deep Learning: A Minimal Effort Back Propagation Method. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 374–387. [CrossRef]
26. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *Advances in Intelligent Computing*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 878–887.