# Linear Regression Model for Predictive Service Provider Selection

Tamerlan Aghayev*, Salviano Diamond*, Sathish Kumar*, Rachel Dudukovich† and Janette Briones†
* Electrical Engineering & Computer Science
Cleveland State University, Cleveland, OH USA
Email: aghayevtamerlan@gmail.com, s.a.diamond86@vikes.csuohio.edu, s.kumar13@csuohio.edu
†NASA Glenn Research Center, Cleveland, OH USA
Email: rachel.m.dudukovich@nasa.gov, janette.c.briones@nasa.gov

***Abstract*—The increasing number of satellites in orbit has led to a growing reliance on third-party service providers for data transfer between Earth and space. Traditional approaches to managing satellite communications require human intervention, which becomes more burdensome with the escalating number of satellites. This research addresses the need for an efficient and automated system to optimize service provider selection for NASA space communication. Previous research has utilized human-operated approaches for service provider management. Our study fills a gap by developing a cognitive algorithm that automates and optimizes the selection process based on various parameters, such as data volume, priority, quality of service and cost. This novel solution reduces user burden, facilitates service management, and contributes to the development of cognitive spaceflight missions, ultimately supporting NASA's research into Cognitive Communications technology. The algorithm design consists of three major steps: modeling data, developing a Link Selection Algorithm (LSA) based on a grading system, and applying machine learning using linear regression. The LSA evaluates providers based on user-defined constraints, considering factors such as delivery time, cost, and quality of service. We define a suitability metric which allows our algorithm to make a recommendation to a user regarding which commercial service providers to select. The addition of Linear Regression predicts the future suitability value. Our main findings demonstrate that the resulting algorithm can autonomously manage connections between satellites and providers, maximizing communication channel efficiency. This research has significant implications, as it not only addresses a pressing issue in satellite communication management but also advances the field of cognitive spaceflight missions.**

***Index Terms*—Linear regression, commercial service providers, link selection algorithms, data modeling, recommender systems**

## I. Introduction

As NASA's Tracking and Data Relay Satellite (TDRS) constellation begins to age, the agency has selected six potential commercial service providers to demonstrate more modern and flexible satellite communications capabilities [1]. Interoperability with a variety of communication systems, automation of resource scheduling and data management are essential to successful integration with these providers. Research into machine learning and artificial intelligence may be used to help guide service provider selection and scheduling, and account for unexpected schedule changes. Future science missions may operate in a way where the transfer of data is unscheduled, a-periodic, and ad-hoc [2]. As network complexity increases, there will be a greater need to manage connections between satellites and providers, so that communication channels are utilized efficiently between any possible combination of satellites and service providers. Such a system should be able to take multiple parameters into account and predict the availability of communications channels in an automated manner.

Currently, NASA uses the traditional approach where "the mission designer typically allocates a single frequency and bandwidth to the radio, applies for the corresponding spectrum license, and negotiates service with a communications service provider" [3]. While this approach is robust and proven, it requires human intervention and the increasing number of satellites in orbit over time leads to increasing demands in managing satellite connections. It would be beneficial to solve these problems to reduce user burden and thus "make it easier for missions to perform service management" [4].

This project proposes a data model and prediction algorithm to enable service provider selection and data transfer decisions without human interaction. Our basic model can be used to develop a knowledge base of previous system parameters, selection decisions, and performance measurements. The algorithm will be able to decide upon a service provider to transfer data to the ground team based on parameters such as data quantity, priority, and the cost of sending through each available provider. Once several selections have been evaluated and the outcomes stored in the knowledge base, the system can be used to make improved decisions based on prior performance.

### A. Recommender Systems

One approach to developing a service provider selection algorithm is to model it as a recommender system. Recommender systems are frequently used in e-commerce applica-

This manuscript is a work of the United States Government authored as part of the official duties of employee(s) of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All other rights are reserved by the United States Government. Any publisher accepting this manuscript for publication acknowledges that the United States Government retains a non-exclusive, irrevocable, worldwide license to prepare derivative works, publish, or reproduce the published form of this manuscript, or allow others to do so, for United States government purposes.

tions to recommend specific products to users or potential customers [5]. A recommender system may be "any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options" [6].

A basic model for a recommender system will take user data and a set of possible products as inputs to a recommender function and output a predicted user rating [7]. Our service provider selection model follows the same structure. The system takes a user spacecraft object and a set of service provider objects as inputs into a link selection function which outputs the top ranked provider for the given user. Section II details the model development, data generation process, and the link selection algorithm.

## II. Methodology

This section details the development, design choices, their justifications, and the features of the Link Selection Algorithm (LSA). Our proposed solution involves three objectives.

1) *Model data* - Datasets were generated for six different providers, each having a separate status and parameters at any given time of day, on which to process the algorithm and provide training data. This step was broken down into three main subtasks:
   a) Define provider, spacecraft, and user requirements
   b) Preprocess the data.
   c) Cache the data into comma separated values (CSV) format.
2) *Evaluate Providers* - Develop a Link Selection Algorithm (LSA). The LSA will select from an array of service providers based on their cost models, prior service provider performance, anticipated availability, mission data requirements, and "user" provided constraints. The best option for the data transmission will be ordered based on a scalar grading value. This was done by:
   a) Assigning a grading system to each user constraint.
   b) Develop a script for multi-objective decision making and grade accordingly.
   c) Populate and update CSV files.
3) *Predict rankings* - Implement the machine learning model Linear Regression to allow the algorithm to use cached performances to make decisions without relying on the link selection algorithm. The system must also incorporate user input into the algorithm's decision making. The basic tasks for this objective were:
   a) Ingest all CSV files from the LSA.
   b) Process all rows and columns to validate data.
   c) Generate linear regression models to see provider `suitability` change over time.

### *Objective 1. Model Data*

The first objective of data generation comes from the need to work around a lack of real world provider datasets. Data is needed to simulate various situations and allow the algorithm to use previously made selections as a contributing factor to deciding the next best options (ie. predicting the best provider). In early stages of the project, a thorough search was conducted to attempt to find relevant datasets, however none were found. As a solution to this problem, a custom data model and file format were developed. Fig. 1 demonstrates the inputs and outputs of our algorithm.

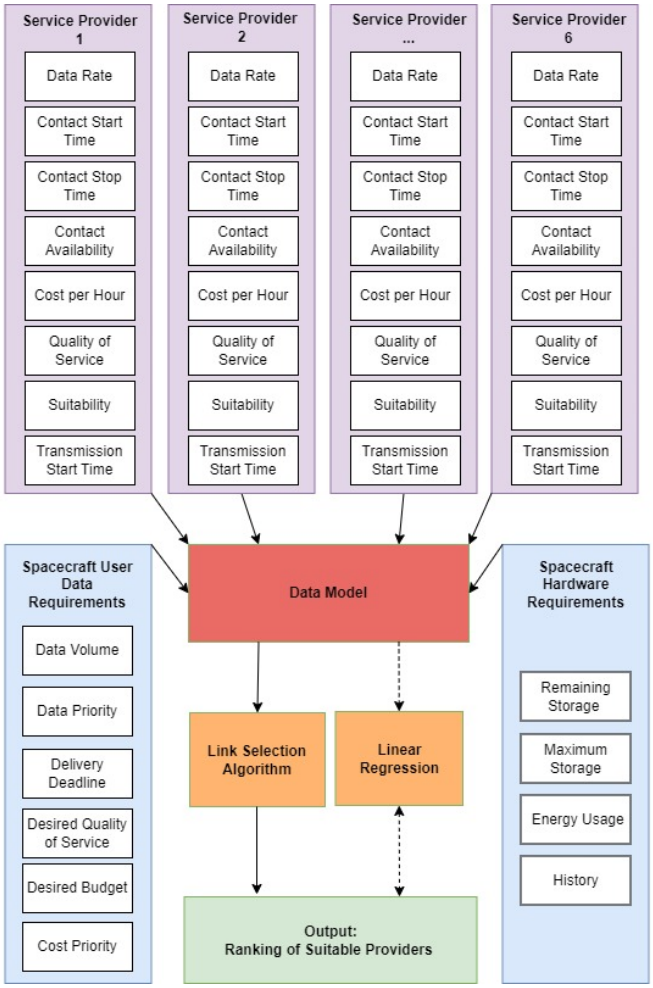


Fig. 1. Data Model and Link Selection Decision Process

A provider will have various parameters, such as data rate, cost per hour, quality of service, which all change through time, as well as a select period of availability to the satellite. Each provider was given parameter values to simulate conflicting environments for the algorithm to make decisions on. Some providers, for instance, were chosen to have strictly better data rates, with a cost to match. Others were given lower data rates, with lower prices to match. During some hours of availability, multiple providers are available at once, with there being many potential providers to link with that each have their own benefits in doing so. The algorithm needs data that features such differences in providers in order to compute a best match for a mission objective. Figures 3 and 4 show how the costs and contact availability are modeled.

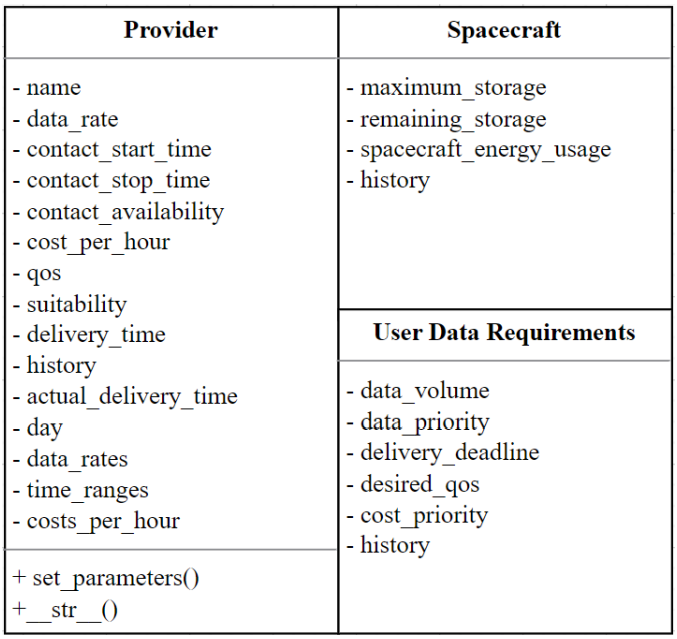


Fig. 2. Data Objects

The Spacecraft class is utilized to represent the attributes of a spacecraft that may affect data scheduling decisions. These attributes include maximum and remaining storage, spacecraft energy usage, and a running history of previous parameters, providing a rich dataset to simulate varying spacecraft conditions. The maximum storage attribute, for instance, fluctuates between 1,000,000 Mb and 1,500,000 Mb, while the remaining storage attribute varies between 100,000 Mb and 1,000,000 Mb, offering a dynamic environment for the algorithm.

The User Data Requirements class is utilized to represent user constraints, such as data volume, data priority, delivery deadline, desired Quality of Service (QoS), desired budget, and cost priority. Similar to the Spacecraft class, we used a range of values for each parameter to mimic real-world variability. For instance, data volume varies as per user requirements, while data priority is set on a scale of 1-3, reflecting the differing urgency of data.

*Objective 2. Evaluate Providers*

The second objective is to develop the algorithm which will interpret the data and evaluate the providers. The algorithm will take a number of parameters that will be configurable by the user, depending on how real-time data will be collected (at this moment, a user's input for the data fields simulates a satellite providing the algorithm the necessary parameters to evaluate the best provider, which will occur at an assigned time of day). Fig. 2 shows the object-oriented design that was used in the Python prototype. The parameters listed in the objective above are the main parameters that are expected to be common and relevant to all circumstances and configurations. Generally, variable weights are assigned to parameters in order to modify the influence each one has over the final decision. The decision being made will be showcased in an output of service providers available for a transmission, ranked in accordance to a scalar value `suitability`.

The LSA itself is based on a grading system and consists of several functions that perform different calculations to evaluate and grade each provider. The main goal is to find the best provider by maximizing the `suitability` score based on the specified constraints by the user.

*1) Implementation:* These are all functions and variables used in the grading system to evaluate the `suitability` metric.

- `leeway`: the amount of time in hours past the delivery deadline. Leeway helps differentiate and prioritize the data priority. If data priority is 1, then the data must be delivered by the delivery deadline, so `leeway` is 0. If data priority is 2, `leeway` increments by an hour, meaning that the spacecraft may deliver later than the delivery deadline. If data priority is 3 or not provided, the `leeway` is 3.
- `calculate_anticipated_budget(provider)`: This function calculates the anticipated budget required to use the given provider, based on their historical data and considering the remaining data load, data rate, cost per hour and the delivery deadline (with some `leeway`).
- `calculate_delivery_hour()`: This function calculates the provider's delivery time and the starting hour for transmission, considering the delivery deadline, data volume, data rate, and provider history. The function first checks the provider's contact availability (1 or 0) at the `delivery_deadline + leeway` hour, and will then calculate how much of the data volume would be calculated each hour using each hour's data rate, until no data would be left. The hour which no data remains is referred to as `provider_delivery_hour`, and providers which are available at both `provider_delivery_hour` and `delivery_deadline + leeway` are considered available, so the function appends these available providers to a list. This is the most significant evaluation since it does not select the providers to be evaluated in the first place, since they are unavailable.
- `cost_evaluation(available_providers)`: This function evaluates the suitability of each available provider based on the cost factor. Depending on the data priority, it increases the suitability score of a provider if its mission budget is equal to or above the desired budget. The current function is `(suitability + (cost priority - 3) * x)`. For data priorities 1, 2, 3 the corresponding `x` are 5, 10, 15 respectively. Fig. 5 shows a summary of the `suitability` calculation.
- `qos_evaluation(available_providers)`: This function evaluates the suitability of each available provider based on the QoS factor. If the actual QoS is less than the desired QoS, it calculates the `qos_evaluation` value and subtracts it from the provider's suitability score. The current function is `qos_evaluation = (desired_qos`


| | Start Time | End Time |
|---|---|---|
| Provider 1 | 4 | 17 |
| Provider 2 | 7 | 20 |
| Provider 3 | 16 | 23 |
| Provider 4 | 0 | 6 |
| Provider 5 | 8 | 15 |
| Provider 6 | 11 | 16 |

Fig. 3. Contact Availability

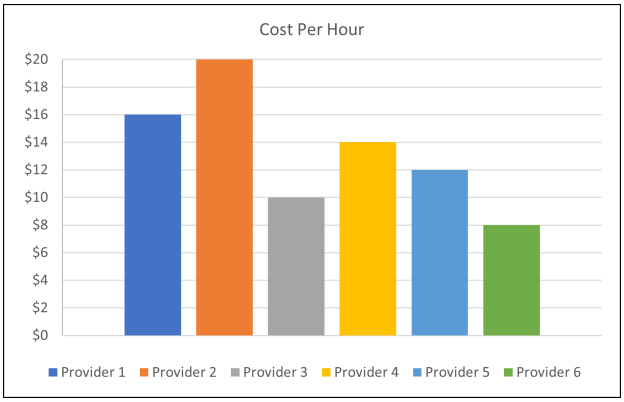


Fig. 4. Cost per Hour

– `dd_actual_qos) * 2`, where `dd_actual_qos` is provider's QoS value at that hour.

- `determine_available_providers()`: This is the main function that calls the other functions to calculate the delivery hour, perform cost evaluation, and perform QoS evaluation. It then resets the suitability scores for providers not in the `available_providers` list.

The code goes through each provider and assigns them a suitability score based on the given criteria. In the end, the algorithm returns a list of available providers with their updated suitability scores. The provider with the highest suitability score can be considered the best provider according to the specified parameters.

*Objective 3. Predict Rankings*

The third objective focuses on the machine learning nature of the task, namely through a Linear Regression implementation, automation of data processing, and allowing user input on the importance of a particular parameter (cost, deadline, priority) to influence the algorithm's output. Every run from the algorithm will store the resulting providers available for the job, the determined hour of the transfer time, and their evaluation rating (`suitability` parameter). Apart from the algorithm learning from previous decisions, the algorithm will also support influence in decision making by user defined parameter values. This will be useful for users that wish to, for instance, save budget, or give data different prioritization than previously declared.

The machine learning model decided upon was the linear regression model. The Linear Regression model is a machine learning model which predicts a "future" value of a linear relationship defined by pairs of data values. It uses the slope of a generated graph of these data pairs, typically a measured data quantity on the y-axis versus another quantity, for instance of a unit of time, on the x-axis of a dataset. A line of best fit (see Equation 1) is generated for this data set, and that line of best fit is used to determine the next data value at the next value of time.

$$Y_i = \beta_0 + \beta_1 X_i \tag{1}$$

The model was chosen as it was seen to easiest fit into the LSA's functionalities, especially as the addition of the suitability scalar in the algorithm's functionalities. The purpose of `suitability` in the algorithm allows it to fit into the aforementioned plot of measured data values versus time in a linear regression model.

A day in the algorithm's terms is one execution of the entire program; the algorithm theoretically runs at 12:00 AM every day and obtains the best delivery hour for available providers of that day. The condition for using Linear Regression instead of the LSA is 7 days of a link selection hour being used, this is determined by calling `check_rerun()`. If `check_rerun()` returns true, we use the LSA, as we do not have enough days of data to use. Otherwise, we can use Linear Regression as an alternative to the LSA. The current approach is to use conditional statements to determine if the user provided `delivery_deadline` has 7 days of data generated, or in other words, the LSA has run 7 times for this particular `delivery_deadline`. Some challenges arise when it comes to using this method, as we would need to use 24 linear regression models for each of the 6 providers, one for each hour of the day. With only 6 predetermined providers in our problem scale, it is not predicted to be an issue for execution times, nor are any efficiency concerns present. Of course, if the algorithm is scaled upwards and made dynamic, problems could occur using this method.

It is important to note that our methodology does not account for factors like varying elevation angles, distances, or the procession of the satellite's orbit, which can impact the quality of service (QoS) and data rate. One way to improve the model might be to use machine learning techniques to learn from historical data on the relationship between the satellite's orbital characteristics and the communication conditions. This could include training a regression model to predict QoS and data rate based on variables such as elevation angle, distance, and orbital position. This approach would also require a significant amount of data, but could potentially provide a more accurate and adaptable model.

| Data Priority | Leeway | Cost | QoS |
|---|---|---|---|
| 1 | 0 | **if** mission_budget >= desired_budget:<br>suitability -= (cost_priority - 3) * 5 | **if** (mission_qos < desired_qos:<br>suitability -= (desired_qos - mission_qos) * 2 |
| 2 | 1 | **if** mission_budget >= desired_budget:<br>suitability -= (cost_priority - 3) * 10 | **if** (mission_qos < desired_qos:<br>suitability -= (desired_qos - mission_qos) * 2 |
| 3 | 3 | **if** mission_budget >= desired_budget:<br>**remove** provider<br><br>**else:**<br>suitability -= (cost_priority - 3) * 15 | **if** (mission_qos < desired_qos:<br>suitability -= (desired_qos - mission_qos) * 2 |

Fig. 5. Suitability Calculation

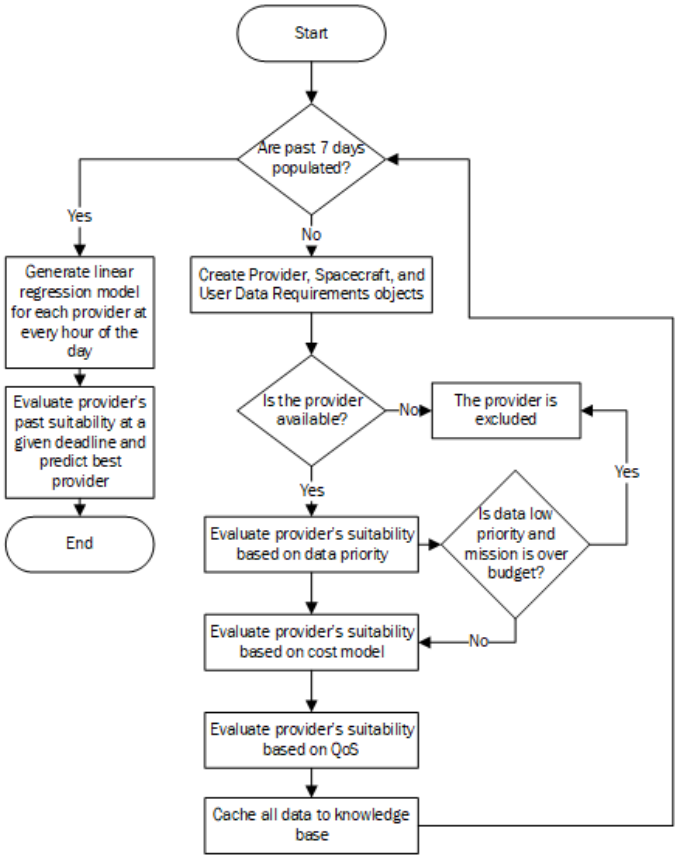


Fig. 6. Flowchart of Data Collection and Algorithm Selection

## III. Results

The final algorithm can operate in two modes. The first mode calls the standard LSA based on data for the current time. The second mode calls a Linear Regression model based on cached `suitability` values from previous runs of the LSA. This will allow the system to predict which provider should be the most suitable based on previous selections. Figures 7 and 8 show the previous `suitability` scores for Providers 1 and 2, and their predicted values based on the linear regression trend line.

The system's initial mode will be to start by using the standard LSA, which only requires knowledge of the current parameters and provider information. The suitability metric will be calculated by using all of the created provider, spacecraft and user requirements models. Once there are at least 7 days of suitability collected at the required times, the system can switch to Linear Regression to make future suitability predictions. Fig. 6 demonstrates a more detailed flowchart of the data collection and algorithm selection process. The advantage of using the Linear Regression mode is that the suitability metric will now be calculated based on a larger knowledge base of historic data, which can also incorporate the actual measured QoS from previous selections. The data model we have proposed is relatively small and Linear Regression itself is rather simple, so there is not a significant level of computational overhead associated with the Linear Regression mode versus the LSA mode.

## IV. Conclusion

A main focus of this work was the data modeling, however the basic prediction results show that our recommender system provides a basic framework for a predictive approach to link selection. Linear regression was chosen as the machine learning method, although there are numerous algorithms that could be used. There are a number of techniques for recommender systems that remain topics for future work.

The lack of data was a significant challenge and we plan to release our data model for other researchers to use. By addressing this challenge, we have contributed to several areas relevant to the artificial intelligence and machine learning community including data modeling and simulation, in addition to algorithm development. The lack of standardization for many terms such as "quality of service", as well as data for in depth cost modeling remain open topics for investigation as well.

Several of the metrics are defined as notional scalar values, such as the quality of service. The definition of an optimal quality of service metric for satellite service provider selection is a broad topic, which we leave a future work. In addition, we plan to further refine the equations for calculating the `suitability` metric. We believe that the concepts defined for the general data model and `suitability` grading system are novel contributions developed by this work.

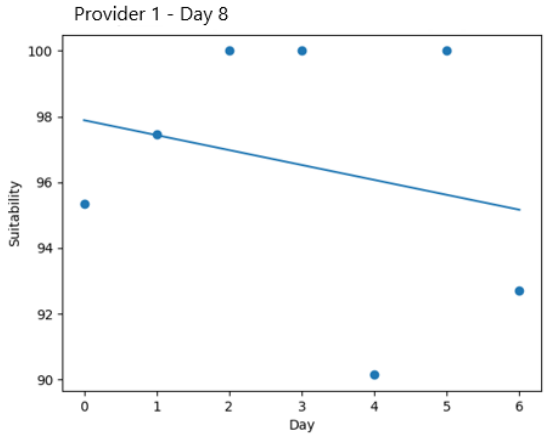


Fig. 7. Linear Regression Prediction for Provider 1

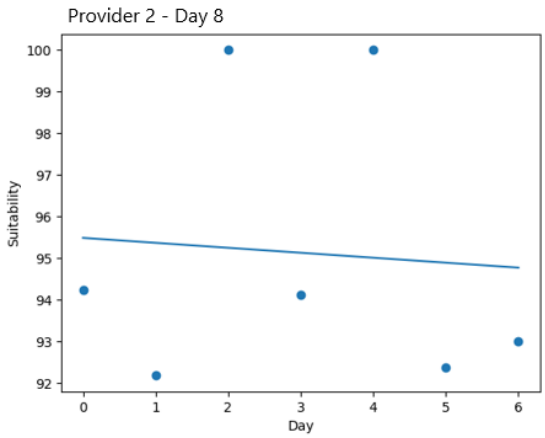


Fig. 8. Linear Regression Prediction for Provider 2

## REFERENCES


[1] J. Foust. *NASA Selects Six Companies to Demonstrate Commercial Successors to TDRS*. 2022. URL: https://spacenews.com/nasa-selects-six-companies-to-demonstrate-commercial-successors-to-tdrs/.

[2] S. Reddy. *Cognitive Communication*. 2020. URL: https://sbir.nasa.gov/content/cognitive-communication-1.

[3] David Chelmins et al. "Cognitive communications for NASA space systems". In: *Advances in Communications Satellite Systems. Proceedings of the 37th International Communications Satellite Systems Conference (ICSSC-2019)*. 2019, pp. 1–16. DOI: 10.1049/cp.2019.1222.

[4] Gilbert Clark et al. "Architecture for Cognitive Networking within NASA's Future Space Communications Infrastructure". In: Oct. 2016. DOI: 10.2514/6.2016-5725.

[5] Robin Burke, Alexander Felfernig, and Mehmet H. Göker. "Recommender Systems: An Overview". In: *AI Magazine* 32.3 (June 2011), pp. 13–18. DOI: 10.1609/aimag.v32i3.2361. URL: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2361.

[6] Robin Burke. "Hybrid Recommender Systems: Survey and Experiments". In: *User Modeling and User-Adapted Interaction* 12 (Nov. 2002). DOI: 10.1023/A:1021240730564.

[7] B. Shetty. *An In-Depth Guide to How Recommender Systems Work*. 2019. URL: https://builtin.com/data-science/recommender-systems.