2020-05

System-of-Systems Acquisition Analytics Using Machine Learning Techniques

Raz, Ali; Balasubramani, Prajwal; Harrington, Stephanie Marie; Guariniello, Cesare; DeLaurentis, Daniel

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/65986

SYM-AM-20-083

Proceedings
of the
Seventeenth Annual Acquisition Research Symposium

Acquisition Research:
Creating Synergy for Informed Change

May 13–14, 2020

Published: April 20, 2020

Approved for public release; distribution is unlimited.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

System-of-Systems Acquisition Analytics Using Machine Learning Techniques

Ali Raz—is a Visiting Assistant Professor at Purdue University School of Aeronautics and Astronautics. His research interests are in systems engineering, system-of-systems, and information fusion. He also holds a temporary faculty appointment with the U.S. Navy's Naval Surface Warfare Center in Crane, IN. He has worked with the Johns Hopkins University Applied Physics Laboratory on fusion systems and, prior to joining Purdue University, he was a flight controls and flight management systems engineer at Honeywell Aerospace. He is a Certified Systems Engineering Professional from the International Council on Systems Engineering (INCOSE), where he is a co-chair of the Complex Systems Working Group and the Assistant Director of early career professionals. He is a senior member of the American Institute for Aeronautics and Astronautics (AIAA) and a senior member of the Institute of Electrical and Electronic Engineers (IEEE). He holds a Bachelor and Master of Science in electrical engineering from Iowa State University and a PhD in Aeronautics and Astronautics from Purdue University.

Prajwal Balasubramani—is a Graduate Research Assistant at Purdue's Center for Integrated Systems in Aerospace and is currently pursuing a master's degree in aerospace engineering. He graduated with a BTech in aerospace engineering from Jain University, India, in 2018. His primary research interests are in the areas of systems engineering, multi-agent systems, mechanism design, and gamification. He has also worked on various projects involving operations research, agent-based modeling, and system-of-systems.

Stephanie Harrington—is a first-year master's student in the Department of Statistics at Purdue University. She is particularly interested in the application of statistical methods and techniques to the defense industry. Prior to graduate school, she worked at the Federal Reserve Board of Governors in Washington, DC, assisting economists in quantitative research. Harrington holds a Bachelor of Arts in quantitative economics and a minor in statistics from the University of California–Irvine.

Cesare Guariniello— is a Research Scientist at Purdue's School of Aeronautics and Astronautics. He received his PhD in Aeronautics and Astronautics from Purdue University in 2016 and his master's degrees in astronautical engineering and computer and automation engineering from the University of Rome "La Sapienza." Guariniello works as part of the Center for Integrated Systems in Aerospace led by Dr. DeLaurentis and is currently engaged in projects funded by NASA, the DoD Systems Engineering Research Center (SERC), Northrop Grumman, and Crane NWSC. His main research interests include modeling and analysis of complex systems and SoS architectures—with particular focus on space mission architectures—aerospace technologies, and robotics. Guariniello is a senior member of IEEE and AIAA.

Daniel A. DeLaurentis—is a Professor at Purdue University's School of Aeronautics and Astronautics. He leads Purdue's Center for Integrated Systems in Aerospace, and he is also the director of Purdue University's Institute for Global Security and Defense Innovation (iGSDI). His primary research interests are in the areas of problem formulation, modeling, and robust system design and control methods for aerospace systems and systems-of-systems. This includes agent-based modeling, network theory, optimization, and aerospace vehicle modeling. His research is conducted under grants from NASA, the FAA, the Navy, the DoD Systems Engineering Research Center (SERC) UARC, and the Missile Defense Agency.

Abstract

System-of-Systems capability emerges from the collaboration of multiple systems, which are acquired from independent organizations. The systems within an SoS serve two purposes: one is to meet their own independent objectives, and the second is to contribute some capability to the SoS from which all constituents can benefit. In recent decades, the fields of machine learning and data analytics have found widespread application in system design and acquisitions. It is unanimously understood that any organization acquiring a complex system employs some form of data analytics to assess a system's independent objectives. Even

though the systems contribute to and benefit from the larger SoS, the data analytics and decision-making about the independent system is rarely shared across the SoS stakeholders. The objective of this work is to identify how the sharing of datasets and the corresponding analytics among SoS stakeholders can lead to an improved SoS capability. We propose to utilize machine learning techniques to predict the SoS capability by sharing pertinent datasets and prescribe the information links between systems to enable this sharing. This paper is an interim update on the work in progress towards the above research effort and focuses on quantifying the value of sharing information across the SoS stakeholders.

Introduction

System-of-Systems (SoS) are comprised of multiple heterogeneous distributed systems that are independently acquired and maintain their operational and managerial independence (Maier, 1998). Although the systems are independent, the system-of-systems capability depends on effective collaboration between the systems. Given this collaborative nature of SoS, when considering acquisition decisions, it becomes important to recognize the stakeholders, resources, operations, policy, and economics of not only one system but the entire SoS. Considering the SoS capability as a multifaceted enterprise, in this paper, we develop research towards an information-centric framework that helps inform early stage decisions on an enterprise level.

Important context for our work comes from the ambitious goals put forth in both defense and commercial sectors for Digital Engineering (DE) and its related components in various engineering functions, such as Model-Based Systems Engineering (MBSE) for the SE domain. DE and MBSE pursue the use of digital models at every phase of acquisition. However, much of the focus right now is on the "how" of DE/MBSE and the desire to have models interoperate rather than the degree to which the extended enterprise (some say the "acquisition ecosystem") has awareness of and belief in the various datasets that underly the models and the development processes that use them.

Within this context, our goal is to identify data management and analytic deployment strategies that create synergies between different enterprise entities and link the stakeholders, resources, policy, and economics between different systems. The framework focuses on examining the impact that data features (e.g., survey categories, types of variables, ownership/privacy of data, etc.) have on the type and effectiveness of predictive and prescriptive analytics that can be employed and how the outcome can be shaped differently by examining the connectivity of data sets. This is particularly important for SoS acquisition where these data sets exist at the local system level but may not be shared at the SoS/enterprise level or vice versa. Our objective is to characterize how the sharing and the connectivity of data sets may lead to deployment of different predictive and prescriptive analytics (due to data access) and lead to better outcomes at the SoS level. We do so with a mapping of useful techniques from the data science/machine learning literature as well as a small, illustrative example. As this is a new research direction only recently underway, our attention is on learning which are the most important deeper research questions that should be pursued.

Background and Literature Review

Overview of Data Analytics Waves

Predictive data analytics provides an ability to anticipate and predict outcomes by collecting and utilizing prior information (Joseph & Johnson, 2013; Rehman, Chang, Batool, & Wah, 2016; Waller & Fawcett, 2013). Although using data to guide decision-making has been around since the Babylonian times, where data was recorded on tablets to predict harvest (Lo & Hasanhodzic, 2011), a major shift in the ability to reason over large amounts of data emerged in the 1940s with the advent of computer development, storage, and machine learning techniques. For application in complex systems, early usage of analytics can be traced back to the 1940s and 1950s, when data analytics models were used to predict outcomes for the behavior of nuclear chain reactions in the Manhattan Project and weather forecasting using the ENIAC computer (Lynch, 2008).

Prescriptive data analytics, on the other hand, aims to provide an ability to generate/prescribe the best courses of action based on given information which may be obtained from a predictive data analytic outcome. Starting around World War II, the need to optimize courses of actions stimulated the development of the operations research field (INFORMS, n.d.) which in the proceeding decades led to “Analytics 1.0” for introducing data-based decision-making in organizations. As the capabilities of computing and machine learning evolved to handle structured and unstructured large data sets (also known as Big Data), Analytics 2.0 became the new paradigm across most large enterprises such as Google and Amazon (Davenport, 2013). Today, the Big Data landscape is shaped by the volume, variety, velocity, and veracity of data (known as the big four Vs of data science), and organizations’ ability to include this “Analytics 3.0” in the decision-making process has become fundamental to its success and profitability. It will not be a generalization to state that most successful organizations employ some form of Analytics 3.0 for business and product development.

For SoS acquisition and capability development, deployment of Analytics 3.0 provides a unique challenge where the individual organizations contributing the constituent systems individually employ a suite of predictive and prescriptive analytics tools (the Literature Review of Machine Learning Techniques and Applications in the DoDprovides details on predictive machine learning techniques as applied primarily in the DoD application space). However, these analytics and the underlying data sets are rarely shared across the SoS stakeholders. Given that the SoS capability emerges from the collaboration of otherwise independent systems and considering the ever-increasing need of interoperability between systems for transitioning towards DE and MBSE, there is an imperative to connect the data sets across SoS for holistic Analytic 3.0 capability deployment. In previous work (summarized in Integrating Predictive and Prescriptive Analytics for Acquisition), we have established the significance of connecting data sets across an enterprise, and our objectives with this work in progress is to develop this capability for SoS acquisition.

Literature Review of Machine Learning Techniques and Applications in the DoD

To get a sense of how predictive analytics and machine learning models are used in the literature, we examine the most popular algorithms and their application. The main goal of statistical evaluation of data is to explain relationships between variables and use them to make predictive and prescriptive recommendations. Relationships between the response variable (output/target) and the independent variables (inputs/features/predictors) can be modeled using both supervised and unsupervised learning techniques. Supervised learning algorithms use predictors and a target variable to learn a function that maps the predictors to the target. It consists of regression and classification models depending on if the response variable is quantitative or categorical respectively. Unsupervised learning algorithms model the underlying structure of a data set with a set of predictors and no response variable (“Supervised Learning vs Unsupervised Learning,” 2018).

The simplest of the supervised learning models is linear regression. This type of algorithm is used to examine the linear relationship between one or more categorical and/or quantitative predictors and a continuous response variable. Linear regression uses an optimization method called “Ordinary Least Squares” (OLS), which minimizes the sum of squared error between the observed and predicted values to estimate the model parameters. Moore and White III (2005) combine a multivariate linear and logistic regression model to identify the root causes of procurement cost growth in engineering and manufacturing development in the DoD procurement process. They use a binary variable representing if a program will have cost growth in procurement dollars and a continuous variable of the percentage of procurement cost growth for the logistic regression and multivariate linear regression model, respectively. Moore and White III's two step-process involves the prediction of the amount of cost growth a program will have using the multivariate regression model results in which those programs are identified based on how likely a program will have procurement cost growth using the results from the logistic regression in the initial step.

Linear regression is not without its disadvantages: the OLS model becomes more complex when more variables are added to the model, introducing multicollinearity and overfitting. Modifications of the linear regression model, ridge and lasso, are used to address this. The parameter estimates are obtained similarly to the linear regression model with the difference being the addition of a penalization term to the loss function. Ridge regression adds the sum of squared magnitude of the coefficients (L2 norm), while lasso adds the sum of absolute value of magnitude (L1 norm). Both models include a tuning parameter, λ, in the penalization term to control the amount of shrinkage to the coefficients. The larger (smaller) the tuning parameter, the model runs the risk of under (over) fitting. If $\lambda=0$, the loss function is equivalent to OLS used in linear regression. Ridge regression is best used when the multivariate linear regression model suffers from multicollinearity, while lasso is best used as a variable reduction or feature selection technique as it shrinks unnecessary coefficients to zero. To address the multicollinearity issue in defense spending, Huang and Mintz (1990) use ridge regression to model the relationship between military expenditures and economic growth. Wang and Yang (2016) used lasso regression as a variable reduction technique to select variables most relevant to supply and demand of airline tickets.

Binary logistic regression is a classification algorithm that models the relationship between a dichotomous response variable, usually denoted as "success" or "failure," and a set of categorical and/or quantitative predictors. This model commonly uses a logit link function where the purpose is to transform the linear combination of the predictor variables, which can take on any value from the real line, and convert the values between zero and one, transforming them on a probabilistic scale (MacKenzie et al., 2017). The logit link function is defined as modeling the log odds of the "success" of the outcome variable as a linear combination of the input variables. In a univariate logistic regression model, the odds increase multiplicatively by the exponential of the coefficient per every unit increase in the predictor variable. Apte, Rendon, and Dixon (2016) explore how the DoD can use information on contractor performance to identify variables that drive the success in service acquisition by using logistic regression and other big data techniques. Success or failure of a contract was used as the response variable, and the authors found that type of contract, awarded dollar value, workload (actions) by filled billets, and percentage of 1102 billets filled by the contracting office had the largest impact on a contract's success. An additional workload of 10 actions per billet is more likely to have a failed contract by 13%, and cost plus award fee (CPAF) and cost plus fixed fee (CPFF) contracts are more likely to fail than firm fixed price (FPP) contracts (Apte et al., 2016).

Support Vector Machines (SVM) is a classification technique that plots each data point in an N-dimensional space where the goal is to identify a linearly separable hyperplane that maximizes the distance (margin) between the data points of a dichotomous response variable. This algorithm uses only the set of data points, called support vectors, closest to the margin to classify the data. If the data is linearly inseparable, a kernel function is used to map the non-linear data into a high dimensional space to become linearly separable (Berwick, n.d.). Wei, Wu, Ma, and Li (2019) use SVMs to estimate the state of charge of lithium-ion batteries for unmanned aerial vehicles (UAVs).

Artificial Neural Networks (ANN) is one of the most powerful machine learning algorithms. A neural network consists of a set of inputs (input layer), interconnected nodes, and an output layer. Data from each node in the input layer is passed to a node in the hidden layer (interconnected nodes) that calculates a weighted sum (Hardetsy, 2017). The hidden layer uses an activation function (e.g., sigmoid function) to determine if the weighted sum of the inputs is passed to the next hidden layer based on if the weighted sum is greater than a threshold/bias until the data reaches the output layer. ANNs are best used when relationships are not constricted to linearity or normality assumptions, when relationships between the variables are difficult to model using traditional approaches, and to discover patterns in the data (*Burger*, n.d.). Brotherton and Johnson (2001) use a neural network to detect anomalies or unexpected faulty conditions in engine operations of advanced military aircraft.

K-Nearest Neighbors (KNN) is used to classify a data point based on the known class of its neighbors. To classify an observation in the test set, the distance between the observation and all the data points in the training set must be calculated using a distance metric (e.g., Euclidean distance) to identify the k-nearest points. Classification of a data point is assigned to one of the categories that appears the most among an observation's k-closest points if the response is categorical. If the response is quantitative, KNN becomes a regression problem, and the assigned output value for an observation is calculated using the arithmetic mean of its k-nearest points. Xiao, Cai, and Chen (2006) use KNN and SVMs to classify types of military vehicles based on the acoustic and seismic signals generated.

K-means is a clustering algorithm that is used to identify K homogenous clusters in the data such that the points in each cluster are similar to each other. The algorithm estimates initial values of centroids (the average of the data points in a cluster) as a first step, and then iteratively assigns each data point to the closest centroid based on a distance metric and takes the mean of all the data points in the cluster to calculate a new centroid. The iteration of the algorithm stops when cluster centroids are stabilized. K-means ensures that data points are homogenous within and heterogeneous between clusters. The final result is the assignment of each data point to a single cluster. Zainol et al. (2018) use K-means to uncover text patterns in military peacekeeping documents.

Naive Bayes Classifier is a probabilistic model that uses Bayes' theorem with the assumption that each pair of input variables is conditionally independent given a value of the response variable where the classification rule is defined as $P(Y|X_1, \ldots, X_n) \propto P(Y)\prod_{i=1}^{n} P(X_i|Y)$ (Naive Bayes, n.d.). The classifier selects the class of the response variable with the highest conditional probability as the target outcome. Freeman (2013) combines multinomial Naive Bayes and multivariate classification to identify DoD acquisition programs with elevated levels of cost risk.

Decision tree analysis is a supervised learning technique that can be used for both regression and classification to visually display decisions as a tree-like diagram represented as homogeneous partitions of the data that lead to a target outcome. The structure has a root node, internal nodes, which represent a split on a predictor variable, and leaf nodes, which represent a target outcome. Every decision tree can be represented using binary decisions at each internal node (Gales, 2013). Decision trees are most commonly built using a top-down approach, which is an iterative and recursive process that selects the best predictor variable for splitting the data into disjoint subgroups based on a splitting criterion (e.g., Information gain, Gini gain) applied to each descendant node (Hand, Mannila, & Smyth, 2001). Apte et al. (2016) also used a decision tree to analyze success or failure of a contract. From top-down, the decision tree first split on awarded dollar value of contract, workload (actions) by filled billets, then finally percentage of filled billets.

A summary of the previously mentioned ML techniques and their application to the DoD related problem is provided in Table 1.

Table 1. Summary of ML methodologies

Method	Key Features	Assumptions	DoD Reference
Supervised Learning			
Linear Regression	Fits quantitative/categorical predictors and continuous response to regression line using OLS	Linear parameters, constant error variance, independent error terms, errors are normally distributed, random sample of observations, no multi-collinearity	Moore and White III (2005)
Ridge Regression	Modification of linear regression that uses L2 norm when multi-collinearity assumption in linear regression is broken	Standardization of predictors, linear parameters, constant error variance, independent errors ("Regression Analysis Software | NCSS Software," n.d.)	Huang and Mintz (1990)
Lasso Regression	Used as a variable reduction or feature selection technique that shrinks some predictor coefficients to exactly zero to reduce overfitting from the linear regression model	Model has sparsity, irrepresentable conditions (Zhao & Yu, 2006)	Wang and Yang (2016)
Binary Logistic Regression	Models the log odds (using logit link) of a categorical binary outcome variable as a linear combination of quantitative/categorical predictors	Independent observations and errors, binomial distribution of response variable, linearity between logit of response and predictors ("Summary Points for Logistic Regression," n.d.)	Apte et al. (2016)
Support Vector Machine	Uses a linearly separable hyperplane to classify data into two classes	Independent and identically distributed observations, margin is as large as possible, support vectors are most useful data points	Wei et al. (2019)
Artificial Neural Networks	Model consisting of interconnected nodes that receive inputs and return outputs based on an activation function	Independence of inputs	Brotherton and Johnson (2001)
K-Nearest Neighbors	Used to classify data points based on class that appears the most among neighboring points (classification) or average of classes (regression)	Similar inputs have similar outputs (Weinberger, 2018)	Xiao et al. (2006)
Naive Bayes Classifier	Uses Bayes theorem to calculate probabilities of a class response and selects the class with highest probability as the output	Predictors are conditionally independent of each other given the response	Freeman (2013)
Decision Tree	Algorithm that recursively and iteratively partitions the data into homogeneous subsets to identify a target outcome	Entire training set is at root node, quantitative predictors must be discretized	Apte et al. (2016)
Unsupervised Learning			
K-means	Use to identify homogeneous clusters in a data set	Cluster sizes are similar and spherical in form	Zainol et al. (2018)

The objectives of the literature review are two-fold: first is to identify the various ML methods which can be applied to SoS acquisition problems and map their input, output, and data requirements, and second is to assess how these methods are applied for different DoD problems. An emerging thread in the literature review is isolated application of these methods where the outcomes along with the data sets are rarely shared across the different hierarchical levels of an organization and SoS. Moore and White III (2005) use the combination of multiple algorithms on a single data set at a local system to identify programs with increased cost growth. The objectives for our research differ in that we aim to analyze how multiple systems run their own individual predictive analytics at the local level and how to best share different data sets across SoS hierarchy to prescribe the SoS capability.

Integrating Predictive and Prescriptive Analytics for Acquisition

In previous work, we have used a conceptual problem to demonstrate the impact that even small, intuitive changes in how data is collected and shared can result in different predictive and prescriptive analytic implementations and lead to a different outcome for SoS decision-making (Davendralingam, Maheshwari, Raz, & DeLaurentis, 2018). Let us take a simple and conceptual example of an enterprise where the objective is to maximize profit by selling a product for which multiple independent entities such as the dealer/distributors of the product, the corporate headquarters, and market research organization must work together. Each of these entities have their own independent objectives for which they use data analytics for decision-making. Consider a scenario where the Market Research Team performs a market study to understand the consumer's opinion on the product design. Now, this information can be used to support the *future* product design at the Corporate HQ by providing insights on what aspects of the design are the most crucial for the consumer. With the better understanding of the information flow, the same set of collected data can also provide insights to the Dealer/Distributor on what features of the *current* product design dictate the demand and, thus, lead to higher profits. In this simplified example, this link of information flow might seem trivial, but when looking at real-world system-of-systems, identifying this important link remains a challenge. In this research, we are pursuing development of a framework that will facilitate identification of such links and quantify how the SoS level capability could evolve by sharing data sets across the systems.

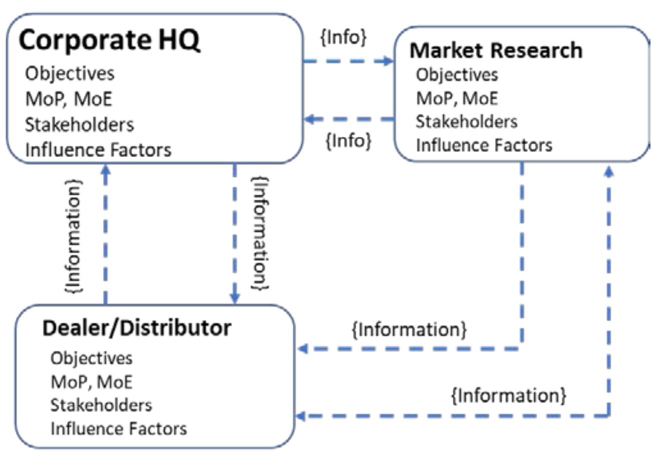

Figure 1. Conceptual problem to identify impact of data-set connectivity

System-of-Systems Acquisition Conceptual Problem Formulation

Introduction to Definition, Acquisition, and Implementation Framework

We build our approach from a conceptual model of SoS that provides a lexicon and taxonomy for representing the various SoS constructs and utilize it to examine data needs and their respective connectivity (Davendralingam et al., 2018). The framework is envisioned to assist in orchestration of analytics and data architecture components across an organization for improved enterprise level performance. The framework is comprised of three phases: Definition, Abstraction, and Implementation. The purpose of the Definition phase is to holistically identify the stakeholders, resources, policy, and economics at different hierarchal levels within an SoS. The Abstraction phase, then, develops representation of the artifacts identified in the Definition phase and recognizes the networks, and hence, interconnection of stakeholders, resources, policy, and economics. This is where the opportunity lies to identify what new connections between the artifacts could be established. Finally, in the Implementation phase, the solution to the SoS problem as defined and abstracted is investigated. Here, the focus is on identifying the right solution methods which are tailored to the SoS problem.

SoS Acquisition Problem Formulation

Consider, for example, a system-of-systems capability as illustrated in Figure 2. The Definition phase identifies elements comprising the system-of-systems at different hierarchy levels, while the abstraction phase identifies the links between these elements. In this case, sub-systems α_1 and α_2 form the system β_1, whereas α_3 relates to the system, β_2. At the higher level, β_1, and $\beta_2$form the system-of-systems, γ_1. Now, each of the sub-system suppliers, system manager, and SoS managers have independent goals of employing data analytics to improve their figures of merit. At the sub-system level, the supplier 1 and supplier 2 may not foresee a need for data set connectivity between α_1 and α_2. However, the potential need for such connectivity becomes evident only at the β_1 system. Since supplier 1, supplier 2, and sys 1 manager all become part of the same system, identifying the right information pathways and connecting data sets for predictive and prescriptive analytics becomes necessary. Similarly, the same logical formulation can be applied to the SoS-level which may demand connectivity of data sets between system β_1 and system β_2, and subsequently imply connectivity between supplier 1, supplier 2, and supplier 3. However, it may not be pragmatic to achieve a full connectivity between all constituent systems and elements of the SoS. Therefore, identifying which datasets need to be connected by characterizing how their connectivity impacts the SoS and the figures of merit becomes a pertinent question.

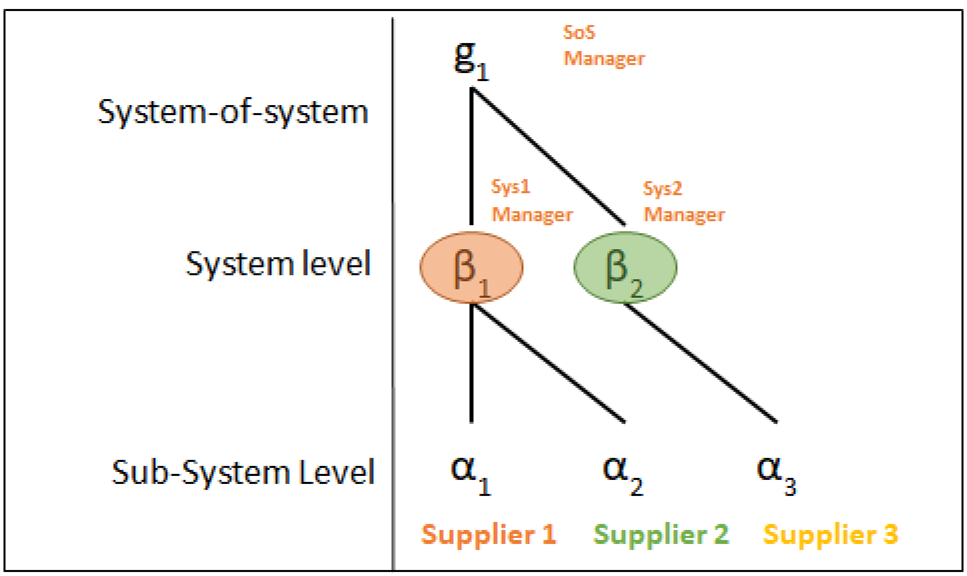

Figure 2. SoS conceptual example

The complexity and scale of this problem for any real-world implementation prohibits an analytical solution. In this research, we address this problem by first formulating the SoS capability measure based on acquiring multiple systems within the DoD application domain and demonstrate how the SoS capability evolves due to sharing preferences between sub-hierarchical systems while maintaining the independent system objectives. Second, we aim to investigate deployment of machine learning techniques (reviewed in the Literature Review section) to predict and prescribe the connectivity of data sets across the different hierarchical levels. Since this paper is an interim update on the ongoing research work, our focus here is on demonstration of SoS acquisition problem formulation, whereas the latter objective of identifying ML techniques will be discussed in a future update.

SoS Acquisition Implementation in Defense Using Decision Support Framework

The Decision Support Framework (DSF) is a tool that includes various SoS analytical tools. The primary function of the DSF is to perform quantitative Analysis of Alternatives (AoA) by generating portfolios of systems that provide both the SoS capabilities of interest and the necessary logistical support for the systems included in the portfolio. This capability is accomplished by integrating a Robust Portfolio Optimization (RPO; Davendralingam & DeLaurentis, 2015) analysis tool for SoS which evaluates not only system- and SoS-level capabilities but also the constraints imposed by interactions between systems (i.e., via support capability requirements). The DSF also performs quantitative and qualitative analysis of each architecture by generating analysis of disruptions via Systems Operational Dependency Analysis (SODA; Guariniello & DeLaurentis, 2017), network representation of the systems and their dependencies through biographs, and cascading matrices that show how systems contribute to SoS capabilities. These methods will be described in more detail in the following sections.

A synthetic problem was created to run simulations using the DSF software and interface with other existing System-of-Systems (SoS) analytical tools. The synthetic problem is an Amphibious Warfare Scenario, which was chosen since it is a multi-domain problem involving air, ground, naval, and space systems. The systems in Amphibious Warfare interact to provide logistical support and system-level capabilities to achieve certain SoS-level capabilities. A case study was developed for the synthetic problem that specifically defined systems from a World War II Amphibious Warfare Scenario. Use of World War II systems and context was chosen since many measures of system capabilities from this time period are public knowledge, which allowed the research team to create a case study with adequate detail. The Mission System Library (MSL) is the key means to pass user inputs into the DSF. The MSL is created in an Excel workbook, where a series of eight sheets provide specific information on the problem:

1. Main Sheet: System names, support capabilities (i.e., internal logistic requirement), system capabilities, and capability uncertainties
2. SoS Capabilities: SoS capability names and sets of indices of the system capabilities that contribute to each SoS capability
3. Compatibility Constraints: matrices containing information on compatibility between systems, specification of maximum amount of specific systems allowed in a portfolio
4. Must Have Systems: to indicate any mandatory systems in a portfolio
5. Conditional Must Have Systems: establishing system interdependencies for operations

Each of these sheets can be read automatically by the DSF software to run the SoS analysis tools and create the outputs. The user is expected to create their own Mission

System Library for their specific problem. An example problem was created for the Amphibious Warfare Scenario to evaluate the portfolios generated under different scenarios and user-input parameters (case scenario and experiments are discussed in detail in the section entitled Problem Description and Setup in Decision Support Framework).

In this paper, we are not using all the tools included in the DSF, but we are focusing on those that provide results suitable for the objectives of this research on SoS and ML tools. The tool that we used to generate different portfolios based on various scenarios and input parameters is the Robust Portfolio Optimization (RPO). This is a methodology to maximize the expected performance of SoS and keep within acceptable levels of developmental risk and cost, while at the same time deal with uncertain information. Implementation of RPO for a certain SoS design problem yields a set of Pareto optimal portfolios of cost versus SoS performance, corresponding to a user-defined risk aversion factor. The optimization is based on a mixed integer programming technique, and all the interdependency between component systems are depicted as constraints.

In the DSF, RPO has been improved with an additional layer that includes not only support capabilities and systems capabilities, but also multiple SoS capabilities that can be included in a weighted function for multi-objective optimization. Initial quantitative architecture analysis of alternatives is performed in the DSF using the RPO method. RPO generates optimized portfolios of systems, and it creates Pareto graphs to display results for SoS-level performance versus portfolio cost. Other tools can be added for further quantitative evaluation.

The DSF runs the RPO tool using as input the system information from the MSL. The user can modify the parameters of the analysis in the DSF Main GUI. Based on the scenario loaded from the opening screen, the GUI will display the user's list of SoS capabilities that can be selected for optimization, as well as a list of support capabilities, from which the user can select whether uncertainty needs to be considered or not. These options implement concepts of Mission-Based design, where even the same set of available systems will generate different portfolios based on different mission requirements. On the right side of the GUI, the user can define levels of risk aversion and levels of available budget, which are used later for generating Pareto frontiers. Other inputs include the importance weights for the selected SoS capabilities and the option to set the requirement to include modular systems.

Simulation and Results

Problem Description and Setup in Decision Support Framework

Considering a realistic setting, where multiple officers/designers/managers are involved, in a multi-objective SoS acquisition problem, a common occurrence is differences in interpretation of the mission requirements either due to lack of communication or judgement. This study investigates how such dissimilarities in the definition of the mission requirements of one contributing individual from another affects the final SoS performance and cost.

In the previously discussed Amphibious Warfare case study, multiple systems were defined in each domain, including air, ground, naval, and space, as well as human systems (e.g., operators). In the MSL, 26 systems were defined, though only an excerpt is provided in Figure 3 and Figure 4, and then evaluated for their support and system capabilities. Five support capabilities were defined for this case study: Transport Range (measured by range in miles), Transport Capacity (measured by capacity in pounds), Refuel (measured by fuel capacity in pounds), Communication Relay (measured using a constructed rating), and

Operator (measured by number of operators). Each system might have one or more support input requirements, which must be fulfilled by a system that has a matching support output capability. Therefore, two sets of columns were defined in the MSL for support capabilities: Support Input Requirement and Support Output Capability. Some systems might be only “support systems” if they only provide support output but do not provide system capabilities. Though the quantified SoS capabilities are evaluated using only the system capabilities, the Robust Portfolio Optimization tool is able to consider the support inputs and outputs by creating constraints that must be satisfied for any portfolio, making these interdependencies still critical to the architecture results.

			Support Input Requirement				
No.	System Type	System Name	Transport Range	Transport Capacity	Refuel	Communication Relay	Operator
-	-	-	Range (mi)	Capacity (lb)	Fuel capacity (lb)	Rating (n.d.)	Number of Operators
1	Air Systems	P-51 Mustang	0	2000	2795	0	1
2		B-17 Flying Fortress	0	6000	18500	0	10
3		C-47	0	0	5369	0	4
4		B-52H Stratofortress	0	60000	321000	1	5
5		B-2 Spirit	0	40000	167000	1	2
6	Ground Systems	Infantry Platoon	10	1845	0	0	42
7		M114 155mm Howitzer	12480	12480	0	0	4
8		M-4 Sherman	150	1251	869	0	5
9		M8 Greyhound	175	274	353	0	4
10		Jeep Willis	0	0	95	0	1
11		"Deuce and a half" (supply truck)	0	0	378	0	1
12		Advanced Targeting Pod	0	0	0	0	0
13		TARDEC Chassis	0	0	378	0	1
14		TARDEC Anti Air Module	100	879	0	0	4
15		TARDEC Artillery Module	100	1750	0	0	4
16		TARDEC Personal Module	100	0	0	0	0
17		Bofors 40 mm gun (L60)	100	4800	0	0	4
18		Refuel Depot	0	0	0	0	0
19		Resupply Depot	0	0	0	0	0
20	Naval Systems	Allen M. Sumner Destroyer	0	0	0	0	336
21		Higgins Boat (LCVP)	0	0	0	0	3
22		Landing Ship, Tank (LST)	0	0	0	0	140
23		Battleship	0	0	0	0	2,220
24	Space Systems	Ultrahigh Frequency Follow-on (UFO) Communication Satellite	0	0	0	0	100
25		Wideband Global Satellite Communication Satellite (WGS)	0	0	0	0	100
26	Human	General Personnel	0		0	0	0

1 Main Sheet | 2 SoS Capabilities | 3 Compatibility Constraints | 4 "Must Have" Systems | 5 Conditional Must Ha

Figure 3. List of available systems and support requirements

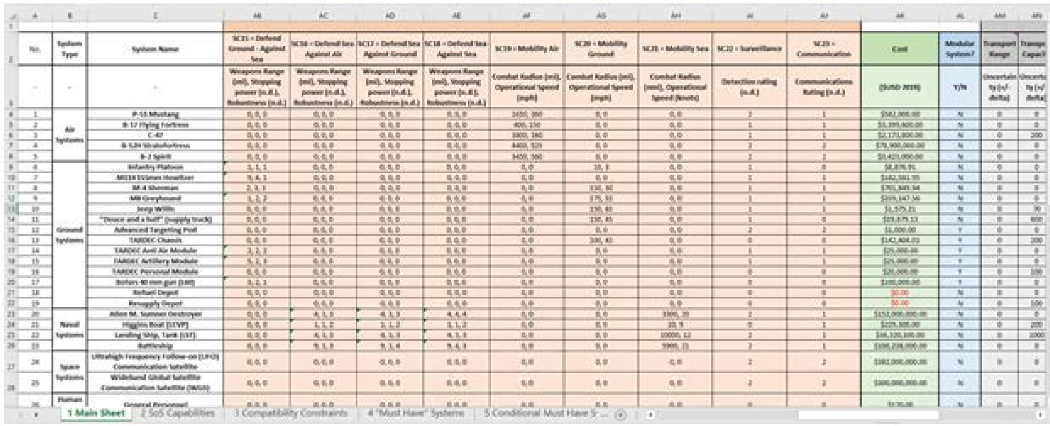

Figure 4. List of the same systems shown in Figure 3. Orange columns are the provided system capabilities; green column is cost; blue column indicates modularity; grey columns are uncertainties.

Next, all desired SoS Capabilities and the indices of the system capabilities that contribute to the SoS capability are defined. Each SoS capability is computed using a normalized sum of individual system capabilities in their respective domain. For this experiment, the focus will be on three SoS capabilities related to an amphibious warfare scenario—Air Superiority, Naval Superiority, and Reconnaissance (shown in Table 2). DSF is also equipped to handle several more SoS capabilities in an effort to extend its usability to a larger spectrum of acquisition problems going forward to leverage the use of machine learning techniques.

Table 2. SoS Capabilities for Amphibious Warfare case study, with system capability contributions

No.	SoS-Capability	System-Capability Indices					
1	Air Superiority	1-6	19-27	46	47	52	53
2	Naval Superiority	13-18	37-45	50-53			
3	Reconnaissance	46-53					

A main feature of the DSF, and the one that is used to investigate our problem statement, is the ability to assign weights to the SoS capability based on the preference for the mission requirement. For example, an acquisition manager who believes that the final portfolio of systems needs to be oriented more towards Air Superiority capability over the others would assign weights accordingly (example: Air Superiority = 0.8, Naval Superiority = 0.1, and Reconnaissance = 0.1). This then leads to the issue of conflicting objectives among the team of acquisition managers or SoS designers. In order to learn the impact of different acquisition strategies (characterized by manager expectations concerning the relative importance of SoS capabilities) on final portfolios, we run 30 cases of varying weights among the team of acquisition managers to understand the variance in portfolios, performance, and cost of the SoS. Figure 5 shows the weight distribution for each of the SoS capabilities.

Weights			
Cases	Air Superiority	Naval Superiority	Reconnaissance
1	0.8	0.1	0.1
2	0.7	0.2	0.1
3	0.7	0.1	0.2
4	0.6	0.2	0.2
5	0.6	0.3	0.1
6	0.6	0.1	0.3
7	0.5	0.1	0.4
8	0.5	0.2	0.3
9	0.5	0.3	0.2
10	0.5	0.4	0.1
11	0.4	0.5	0.1
12	0.4	0.4	0.2
13	0.4	0.3	0.3
14	0.4	0.2	0.4
15	0.4	0.1	0.5
16	0.3	0.6	0.1
17	0.3	0.5	0.2
18	0.3	0.4	0.3
19	0.3	0.3	0.4
20	0.3	0.2	0.5
21	0.3	0.1	0.6
22	0.2	0.7	0.1
23	0.2	0.6	0.2
24	0.2	0.5	0.3
25	0.2	0.4	0.4
26	0.2	0.3	0.5
27	0.2	0.2	0.6
28	0.2	0.1	0.7
29	0.1	0.1	0.8
30	0.1	0.8	0.1

SoS capability	0	0.0624	0.0691	0.0737	0.0737
Cost	$0.00	$ 145.24M	$ 297.24M	$ 597.24M	$ 679.24M
P-51 Mustang	0	1	1	1	1
B-17 Flying Fortress	0	1	1	1	1
C-47	0	0	0	0	0
B-52H Stratofortress	0	0	0	0	0
B-2 Spirit	0	0	0	0	0
Infantry Platoon	0	1	1	1	1
M114 155mm Howitzer	0	0	0	0	0
M-4 Sherman	0	1	1	1	1
M8 Greyhound	0	1	1	1	1
Jeep Willis	0	1	1	1	1
"Deuce and a half" (supply truck)	0	0	0	0	0
Advanced Targeting Pod	0	1	1	1	1
TARDEC Chassis	0	0	0	0	0
TARDEC Anti Air Module	0	0	0	0	0
TARDEC Artillery Module	0	0	0	0	0
TARDEC Personal Module	0	0	0	0	0
Bofors 40 mm gun (L60)	0	0	0	0	0
Refuel Depot	0	1	1	1	1
Resupply Depot	0	0	0	0	0
Allen M. Sumner Destroyer	0	0	1	1	1
Higgins Boat (LCVP)	0	1	1	1	1
Landing Ship, Tank (LST)	0	1	1	1	1
Battleship	0	1	1	1	1
Ultrahigh Frequency Follow-on (UFO) Communication Satellite	0	0	0	0	1
Wideband Global Satellite Communication Satellite (WGS)	0	0	0	1	0
General Personnel	0	1	1	1	1

Figure 6. Test runs with variation in weight distribution

Running the DSF and using RPO, we collected the resulting SoS portfolios for all these cases. In each case, we obtain multiple portfolios that are feasible within given budget limits. From our runs, a total of four feasible instances for each case was produced, and these were used to form a pareto frontier to better understand the relation between SoS capability preferences, performance, and cost.

Results and Analysis

For each case, the DSF produces portfolios containing data about the various possible architectures and their associated SoS performance index and cost. A portfolio is a feasible combination of systems, which includes some that provide the required capabilities and others that provide the needed support. Figure 6 is one example (case 1) for a portfolio generated for a test case. Each column is a Pareto-optimal portfolio for a given budget limit. Zeros mean that the corresponding architecture does not utilize the system in question. Ones indicate systems that are part of the architecture. Looking into this data will give the user insight on how these suggested architectures differ amongst themselves and how they compare with other cases.

In this example, it is observable that when the architectures switch to a combination that includes one or more different systems, better performing yet expensive, the SoS capability improves. This possibility of various permutations of system architectures make a portfolio-based study more relevant and accurate for SoS acquisition problems.

The data from the portfolios generated in the 30 scenarios are then used to identify the space of solutions for all the cases individually. To do so, Portfolio Performance Frontiers where the SoS Performance Index is mapped with its corresponding costs are plotted. Figure 7 is a representation of this for two cases (Case 1 and Case 17 as examples).

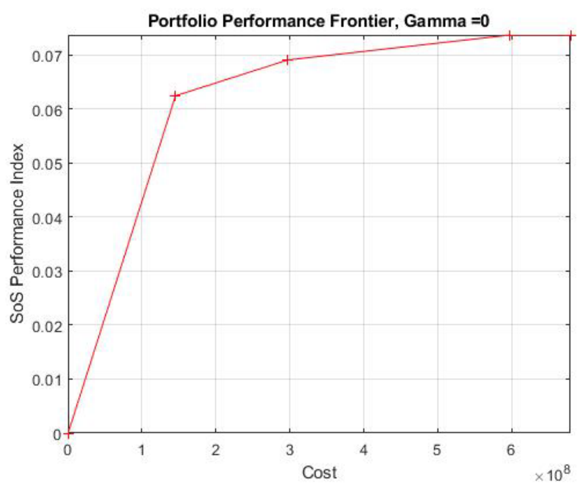

Figure 7. Architectures in a single scenario

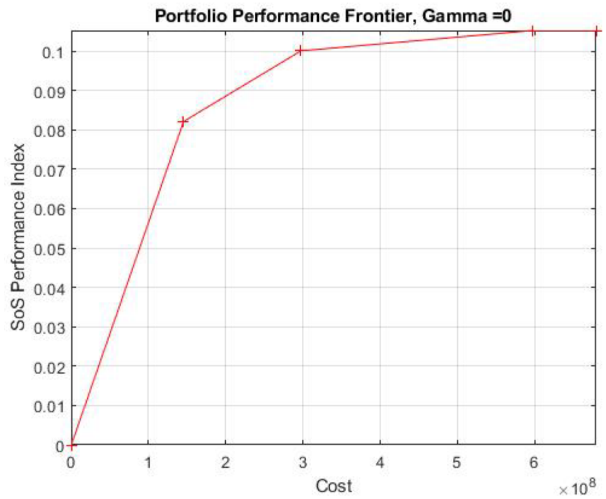

Figure 8. Test runs with variation in weight distribution

First, each of these portfolio performance frontiers identify the best possible solution (architecture) for a given cost. Every distinguishable point on the frontier is a feasible architecture (one column from Figure 6). With increase in budget, as expected, better performing systems are acquired to form the SoS architecture. This results in better performing SoS architectures within the same scenario. Second, upon closely inspecting and comparing the two pareto frontiers, it is evident that while the shape/form of the two is similar, the data points are not the same. This indicates that different weight preferences for the SoS capability produce portfolios that provide different performances. This is clearly visible when multiple pareto frontiers from various cases in the experiment are plotted in the same graph, as shown in Figure 8. We can notice that any uncertainty in SoS capability preferences (while setting up the acquisition problem) affects the resulting performance of the SoS portfolios. For example, Case 1 had a weight of 0.8 (out of 1) and Case 29 had 0.1 for Air Superiority and, as stated, the SoS performance index for their respective portfolios are inversely related to the value of the assigned weights, leading to two portfolios with a sizeable difference in their performance index. Another influencing factor in any acquisition problem is the restrictive nature of the proposed budget (i.e., cost). By using RPO, the accountability of cost-based comparisons are visible, too, such as in instances where the performance index of a portfolio for one case (case 26) is higher than the other (case 22) for a specified cost value. However, with an increase in cost to a higher value, the previous trend does not hold true.

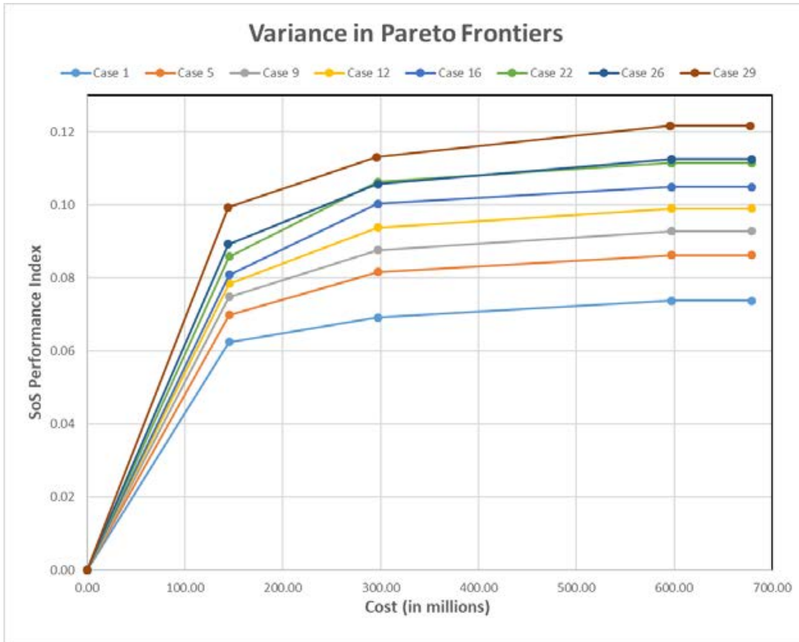

Figure 9. Variation in pareto frontiers across the cases

Conclusion and Future Work

Machine learning and data analytic techniques are increasingly being employed for decision-making regarding system acquisition and capability development which eventually become part of a larger SoS. These independent systems are both consumers and producers of the SoS capability which is shaped by the collaboration between systems. However, due to operational and managerial independence of these systems, the machine learning and data analytic techniques are often applied in isolation from the SoS. The literature survey of various machine learning applications in the DoD domain indicate a siloed treatment where only a few cases exhibit exchange of datasets and outcomes between different machine learning methods, let alone across systems and SoS. In the research, we aim to investigate how the SoS capability evolves from the individual system preferences and how we can leverage the datasets employed for siloed system-level decision-making for the SoS-level decision making. Our example results in this paper clearly establish the significance of sharing these datasets by demonstrating that differing preferences of SoS stakeholders—modeled as weights of independent capabilities provided by individual systems—lead to different sets of systems (i.e., portfolios) being selected in the SoS for a given budget.

As we continue the development of this research, a major challenge that we aim to address is identification of which datasets need to be connected across the SoS, since fully connected data enterprise is unlikely to be pragmatic in the real world. In ongoing developments, we are focused on investigating machine learning techniques that can predict the SoS capability based on having access to decision-making loops at the system level and prescribe a path forward for generating information flows between systems in the SoS.

References

Apte, U., Rendon, R., & Dixon, M. (2016). *Big data analysis of contractor performance information for services acquisition in DoD: A proof of concept.* Monterey, CA: Naval Postgraduate School.

A basic introduction to neural networks. (n.d.). Retrieved April 9, 2020, from http://pages.cs.wisc.edu/~bolo/shipyard/neural/neural.html

Berwick, R. (n.d.). *An idiot's guide to support vector machines (SVMs).* 28.

Brotherton, T., & Johnson, T. (2001). *Anomaly detection for advanced military aircraft using neural networks, 6*, 3113–3123.

Davendralingam, N., & DeLaurentis, D. A. (2015). A robust portfolio optimization approach to system of system architectures. *Systems Engineering, 18*(3), 269–283.

Davendralingam, N., Maheshwari, A., Raz, A. K., & DeLaurentis, D. A. (2018). *Predictive and prescriptive analytics via system of systems engineering: Informational determinants for enterprise-level effectiveness.*

Davenport, T. H. (2013, December 1). Analytics 3.0. *Harvard Business Review*. Retrieved from https://hbr.org/2013/12/analytics-30

Explained: Neural networks. (2017, April 14). *MIT News*. Retrieved from http://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

Freeman, C. E. (2013). *Multivariate and naïve Bayes text classification approach to cost growth risk in Department of Defense acquisition programs*.

Gales, M. (2013). *Handout 10: Decision trees*. Retrieved from http://mi.eng.cam.ac.uk/~mjfg/local/4F10/lect12.2up.pdf

Guariniello, C., & DeLaurentis, D. (2017). Supporting design via the system operational dependency analysis methodology. *Research in Engineering Design, 28*(1), 53–69.

Hand, D. J., Mannila, H., & Smyth, P. (2001)*. A systematic overeview of data mining algorithms.*

Huang, C., & Mintz, A. (1990). Ridge regression analysis of the defence‐growth tradeoff in the United States. *Defence and Peace Economics, 2*(1), 29–37.

INFORMS. (n.d.). *History of OR: Useful history of operations research.* INFORMS. Retrieved from https://www.informs.org/ORMS-Today/Public-Articles/June-Volume-42-Number-3/History-of-OR-Useful-history-of-operations-research

Joseph, R. C., & Johnson, N. A. (2013). Big data and transformational government*. IT Professional, 15*(6), 43–48. Retrieved from https://doi.org/10.1109/MITP.2013.61

Lo, A. W., & Hasanhodzic, J. (2011). *The evolution of technical analysis: Financial prediction from Babylonian tablets to Bloomberg terminals*. John Wiley & Sons.

Lynch, P. (2008). The ENIAC forecasts: A re-creation. *Bulletin of the American Meteorological Society, 89*(1), 45–56. Retrieved from https://doi.org/10.1175/BAMS-89-1-45

MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L., & Hines, J. E. (2017). *Occupancy estimation and modeling: Inferring patterns and dynamics of species occurrence*. Elsevier.

Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering, 1*(4), 267–284. Retrieved from https://doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D

Moore, G. W., & White III, E. D. (2005). A regression approack for estimating procurement cost. *Journal of Public Procurement, 5*(2).

Naive Bayes—Scikit-learn 0.22.2 documentation. (n.d.). Retrieved April 9, 2020, from https://scikit-learn.org/stable/modules/naive_bayes.html

Regression analysis software | Regression tools | NCSS software. (n.d.). Retrieved April 9, 2020, from https://www.ncss.com/software/ncss/regression-analysis-in-ncss/#Ridge

ur Rehman, M. H., Chang, V., Batool, A., & Wah, T. Y. (2016). Big data reduction framework for value creation in sustainable enterprises. *International Journal of Information Management, 36*(6, Part A), 917–928. Retrieved from https://doi.org/10.1016/j.ijinfomgt.2016.05.013

Summary points for logistic regression | STAT 504*.* (n.d.). Retrieved April 9, 2020, from https://online.stat.psu.edu/stat504/node/164/

Supervised learning vs unsupervised learning—Best 7 useful comparison. (2018, April 29). EDUCBA. Retrieved from https://www.educba.com/supervised-learning-vs-unsupervised-learning/

Waller, M. A., & Fawcett, S. E. (2013, June 1). Data science, predictive analytics, and big data: A revolution that will transform supply chain design and management. *Journal of Business Logistics*. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/jbl.12010

Wang, R., & Yang, W. (2016). Estimation of demand and supply in U.S. airline industry (Unpublished working paper).

Wei, K., Wu, J., Ma, W., & Li, H. (2019). State of charge prediction for UAVs based on support vector machine. *Journal of Engineering, 2019*(23), 9133–9136.

Weinberger, K. (2018). *Lecture 2: K-nearest neighbors/Curse of dimensionality*. Retrieved from https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote02_kNN.html

Xiao, H., Cai, C., & Chen, Y. (2006). Military vehicle classification via acoustic and seismic signals using statistical learning methods. *International Journal of Modern Physics C, 17*(02), 197–212.

Zainol, Z., Azahari, A. M., Wani, S., Marzukhi, S., Nohuddin, P. N., & Zakaria, O. (2018). Visualizing military explicit knowledge using document clustering techniques. *International Journal of Academic Research in Business and Social Sciences, 8*(6), 1127–1143.

Zhao, P., & Yu, B. (2006). On model selection consistency of Lasso. *Journal of Machine Learning Research, 7*(Nov), 2541–2563.

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net