On the Advancement of Project Management through a Flexible Integration of Machine Learning and Operations Research Tools

Nikos Kanakaris, Nikos Karacapilidis and Alexis Lazanas
Industrial Management and Information Systems Lab, MEAD, University of Patras, 26504 Rio Patras, Greece

Keywords: Project Management, Machine Learning, Operations Research, Intelligent Optimization.

Abstract: Project Management is a complex practice that is associated with a series of challenges to organizations and experts worldwide. Aiming to advance this practice, this paper proposes a hybrid approach that builds on the synergy between contemporary Machine Learning and Operations Research tools. The proposed approach integrates the predictive orientation of Machine Learning techniques with the prescriptive nature of Operations Research algorithms. It can aid the planning, monitoring and execution of common PM tasks such as resource allocation, task assignment, and task duration estimation. The applicability of our approach is demonstrated through two realistic examples.

1 INTRODUCTION

Project Management (PM) is a complex practice that is highly fluid and hard to predict, thus imposing a series of challenges to organisations and experts. Such challenges may concern alignment between projects and their business objectives, handling of conflicts and dependencies in resource allocation, fine tuning of multiple projects to avoid fragmented planning, as well as informed and diffused decision making to handle potential opportunities or threats during the execution of a project (Svejvig and Andersen, 2015).

At the same time, PM is inherently collaborative and knowledge-intensive. Issues to be addressed are characterized by ever-increasing amounts of different types of data and knowledge, which may be obtained from various sources and vary in terms of subjectivity, ranging from individual opinions and estimations to broadly accepted practices and indisputable measurements and results (Karacapilidis, 2014). Their types can be of diverse level as far as human understanding and machine interpretation are concerned.

Up to now, the majority of methods and tools aiming to facilitate and augment the quality of PM are based on the application of advanced analytical approaches developed and elaborated in the realm of the Operations Research (OR) discipline. These approaches employ techniques such as mathematical optimization and statistical analysis to look for optimal or suboptimal solutions to diverse PM issues. In addition, the application of Artificial Intelligence (AI) techniques to automate project management has been proposed more than 30 years ago. At that time, the proposed AI-leveraged project management systems used knowledge processing and procedural techniques to provide new kinds of decision support for project objective-setting and control (Levitt and Kunz, 1987).

Nowadays though, the adoption of AI in the data-intensive and cognitively-complex PM settings enables a series of advancements. AI, and in particular Machine Learning techniques, can aid project managers easily delegate thousands of tasks, while sustaining a holistic view of their resources and projects. This contributes to the achievement of the required accuracy and precision when dealing with bottlenecks or constraints that may obstruct business processes. At the same time, these techniques can aid managers and experts to interpret big volumes of data and gain valuable insights towards improving their overall PM practice. Based on past data, they can predict undesired situations, provide timely warnings and recommend preventive actions regarding problematic resource loads or deviations from business priority lists.

Admittedly, each of the abovementioned disciplines (OR and AI) has significantly contributed to the improvement of the PM practice, by addressing the associated issues from a different

Kanakaris, N., Karacapilidis, N. and Lazanas, A.
On the Advancement of Project Management through a Flexible Integration of Machine Learning and Operations Research Tools.
DOI: 10.5220/0007387103620369
In *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems (ICORES 2019)*, pages 362-369
ISBN: 978-989-758-352-0

philosophy and research perspective. However, we argue that their joint consideration has not been thoroughly explored yet, and has much potential to further augment PM-related business intelligence. Such an approach will concentrate on both the planning and execution of individual projects as well as their association with past data and their impact on the wider business. Moreover, this approach can appropriately represent and process the associated data and knowledge, while at the same time remedy the underlying cognitive overload issues. Particular attention should be also given to the expression and maintenance of tacit knowledge (i.e. knowledge that employees do not know they possess or knowledge that they cannot express with the means provided), which predominantly exists and dynamically evolves in PM settings.

In line with the above remarks, this paper attempts to shape a hybrid approach for better handling PM issues by meaningfully integrating tools originally developed in the context of OR and AI. The remainder of the paper is organized as follows: Section 2 discusses background work considered in the context of our approach, which is analytically described in Section 3; the applicability of the proposed approach is demonstrated through two realistic examples presented in Section 4; concluding remarks, limitations and future work directions are summarized in Section 5.

2 BACKGROUND ISSUES

Numerous software solutions to PM exist in the market nowadays. The list of the most widely adopted ones includes Wrike (www.wrike.com), Asana (www.asana.com), Trello (www.trello.com), and Jira (www.atlassian.com/software/jira). Such solutions offer a user-friendly environment that mainly enables issue tracking and supports various project management functions. In addition, by providing interactive graphics, issue boards and timelines, they simplify planning, collaboration, reporting and time management. It is broadly admitted that existing commercial PM solutions may increase an organization's productivity and prevent the teams from diverging from their actual goals. However, they unintentionally hide important PM-related information, due to the complex multidimensional data found in the hosted projects.

At the same time, by adopting an AI-perspective, a range of digital project management assistants has been already developed, including solutions such as Stratejos.ai (www.stratejos.ai), PMOtto.ai (www.pmotto.ai), and x.ai (www.x.ai). This category of solutions is based on seamless, easy-to-use interfaces that assist project managers in common tasks (e.g. a project's supervision). They rely on the expressiveness, immediacy, interactivity and descriptiveness that natural language provides to offer a 'zero-level' entrance environment. They are used to automate repetitive work such as creating project's tasks by analyzing textual conversations, to remind and organize important events such as meetings, to extract shallow insights (e.g. 'top contributors of the week'), and to answer simple queries (e.g. 'what is my team working on today?').

We argue that this second category of solutions offers narrow predictions and automations. In particular, their underlying reasoning mechanisms mainly build on rules to store and manipulate knowledge, and ignore advanced AI technologies that can uncover insights, perform more complex tasks, make explainable recommendations, and support informed decision making, sometimes in ways that outperforms what people are able to do today. Furthermore, each of these digital personal assistants is relevant to a specific project management need (e.g. reporting, scheduling meetings, organizing events); thus, they are unable to embrace a 'single-access-point' approach that mitigates the overall PM complexity.

From an OR perspective, a series of techniques and tools have been proposed and extensively used to solve various PM related issues. OR techniques provide solutions in problems such as prediction, resource allocation, forecasting, scheduling, task assignment, networking etc. These techniques are supported by very useful software libraries such as pyschedule (github.com/timnon/pyschedule), PuLP (github.com/coin-or/pulp), Google OR-tools (developers.google.com/optimization), JuMP.jl (Dunning et al., 2017), Hungarian.jl (github.com/Gnimuc/Hungarian.jl), and CVXPY (www.cvxpy.org).

The abovementioned software libraries support a variety of OR techniques including integer, linear, convex and dynamic programming. However, these techniques tend to add more complexity on the overall PM practice, mainly due to the complicated mathematical models needed to operate. Another drawback is that these techniques are unable to learn by the systems' experience, which often results to the proposition of optimal or near-optimal solutions that are not realistically feasible.

With the advent of big data and cloud computing era, Machine Learning (ML) techniques gain ground in a variety of scientific and commercial sectors.

These techniques (and corresponding algorithms) can categorize items, predict values, identify meaningful relationships, and detect data patterns or unexpected behavior (anomaly detection). ML approaches are usually grouped into four categories, namely *supervised learning*, *semi-supervised learning*, *unsupervised learning* and *reinforcement learning* (Goodfellow *et al.*, 2016).

Supervised learning refers to the process of learning aiming to predict values (e.g. house prices) or classify items into categories (e.g. categories of projects) by using labelled training data. Common algorithms and methods used in supervised learning include k-nearest neighbors, naive Bayes, decision trees, linear regression, and support vector machines. Semi-supervised learning combines both labeled and unlabeled input data for training, where in most cases there is a small amount of labeled data and a huge amount of unlabeled data available.

Unsupervised learning analyzes unlabeled data to identify patterns or cluster similar items into groups using alternative distance metrics (e.g. Euclidean distance, Manhattan distance). Common algorithms used in unsupervised learning include k-means, DBSCAN, OPTICS, Apriori (Agrawal and Srikant, 1994) and hierarchical clustering. Finally, reinforcement learning approaches iteratively interact with their environment to identify specific actions that maximize the reward or minimize the risk. Common algorithms and methods used in this category include Q-learning, temporal difference, and deep adversarial networks.

The above ML techniques and algorithms are fully supported today by various software libraries and environments, such as scikit-learn (Pedregosa *et al.*, 2011), H2O.ai (Candel *et al.*, 2016), Tensorflow (Abadi *et al.*, 2016), PyTorch (Paszke *et al.*, 2017) and WEKA (Holmes *et al.*, 1994).

As a last note, it is worth mentioning that most AI-based approaches to PM build on artificial neural networks. Related works discuss how neural networks are capable to assist project managers in problems such as resource allocation, prediction, clustering, classification (Burke and Ignizio, 1992) and forecasting (Zhang *et al.*, 1998). Neural network techniques have been also applied to predict construction cost and schedule success (Wang *et al.*, 2012). Other representative works concern development of a neural network to estimate project performance (Cheung *et al.*, 2006), or to classify the level of a project's riskiness by exploiting the knowledge extracted from data concerning past successful and unsuccessful projects (Costantino *et al.*, 2015). An interesting overview of the different types of neural network models applied in business can be found in (Smith and Gupta, 2000).

3 THE PROPOSED APPROACH

Considering the pros and cons of the techniques discussed in the previous section, we propose a hybrid approach to handle contemporary PM issues, which builds on a proper integration and orchestration of PM tools originally developed within the ML and OR disciplines. ML, which has become a buzzword nowadays, adopts a predictive analytics approach of the form *'if A happens, then B is likely to happen'*, which attempts to exploit available past data to create useful insights (i.e. make human-like decisions). On the other hand, OR adopts a prescriptive analytics approach to provide optimal solutions (courses of action) to problems of the form *'what does A need to be if we want B to happen'* (i.e. make perfect decisions).

We consider tools coming from the ML and OR fields as complementary, arguing that there is room for integration in a way that ML can create and refine A→B relationships that are often considered as optimal and remain unchanged upon the entry of new data in classical OR approaches. Despite the features that ML possesses in terms of data refinement and value prediction, it lacks algorithms aiming to provide optimal solutions, something that is inherent in OR techniques. Overall, our approach considers OR and ML as complementary to each other, and proposes an iterative interplay between them, where ML supplies OR algorithms with refined, accurate and up-to-date data (based on past records), while OR contributes to making optimal decisions with the continuously updated data input.

The proposed approach enables interpretation of big volumes of PM data to support preventive actions such as giving advice about resource assignments by identifying similar skills and expertise necessary to perform a task, make explainable recommendations about the capacity levels of certain resources based on historical performance data, and support informed decisions concerning a company's expansion to a new region or design of an efficient supply chain. The proposed approach augments the overall PM decision-making process, by enabling the drawing of reliable conclusions about conditions and future events, while also identifying potential risks and opportunities.

Depending on the specific PM issue under consideration, our approach advocates a proper

streamlining of ML and OR algorithms. As far as ML algorithms are concerned, these can be distinguished in four categories concerning data classification, value prediction, structure discovery, and detection of anomalies or abnormal behavior. More specifically:

- Data classification aims to predict which category the input data belongs to. For example, in a software development project, a new task can be classified into distinct categories (e.g. story, bug, epic) based on its attributes using a decision tree classifier.
- Value prediction concerns regression algorithms to predict continuous numerical values. For example, in a common PM scenario, these algorithms can estimate the budget of a project by exploiting knowledge of similar, already accomplished projects using simple linear regression techniques, thus providing advice to the project manager during the planning phase on possible cost reduction decisions.
- Anomaly detection algorithms aim to identify unusual events or patterns that do not conform to usual or expected behavior. For example, in a certain maintenance setting, these algorithms can detect outages of some components before they occur and proactively act towards keeping the whole system functioning.
- Structure discovery aims to uncover data patterns, reveal hidden or not obvious relationships and divide data items into groups with similar traits (features). This is achieved using widely-adopted ML techniques (e.g. k-means and Apriori algorithms). For example, in a construction PM problem, the Apriori algorithm can mine frequent itemsets concerning constructors and project durations to build useful association rules (e.g. constructor x is always late when delivering dam construction projects).

4 EXAMPLES

In this section, we demonstrate the applicability of the proposed approach through two realistic examples concerning resource assignment. Emphasis is given to the complementarity of ML and OR algorithms to advance the associated PM practice.

Example 1

Based on real data concerning implementation of public construction projects in the Region of Attica, Greece, for the period 2003-2014, we consider the following problem: Let $P = \{P_1, P_2, \ldots, P_n\}$ be a set of future public construction projects. Each project (P_n) is described by a list of attributes, namely P_n = *[PID, location, category, est_cost, funding_source, duration]*, corresponding to a unique project identifier, the municipality to manage the project, the type of construction needed, the project's estimated cost, the source funding the project, and its estimated duration, respectively.

Similarly, let $C = \{C_1, C_2, \ldots, C_m\}$ be the set of registered construction companies, each of them being associated with the set of attributes *[CID, {Location$_i$}, {Category$_j$}, {Cost_Range$_k$}, {Duration_Range$_l$}, AvgDiscount, AvgDelay]*, corresponding to a unique constructor identifier, the municipality where the constructor is active, the type of projects the constructor deals with (e.g. flood control, health infrastructure), the projects' budget category the constructor is interested in (e.g. large scale (>1,5M€), medium scale (0,5M€-1,5M€)), the projects' duration range (e.g. short term (<6 months), mid term (6-18 months)), the average discount provided by the constructor, and the average delay caused by the constructor, respectively.

Let a project management scenario where there are $n = 3$ projects of various categories and $m = 3$ available constructors. Obviously, each P_n requires a different expertise, while each C_m possesses a distinct number of skills based on their profile, which is populated with attributes extracted from past data. To determine the constructors that best fit to the projects' requirements, we need to populate a (P_n, C_m) score matrix (each entry taking values in the range [0, 1]). This is through the calculation of (i) the Jaccard similarity index $J(P_n, C_m)$ (Jaccard, 1901), and (ii) an additional score value $Score_{C,M}$ for the attributes *avg_discount* and *avg_delay* of each C_m (these attributes do not participate in the calculation of the Jaccard similarity index).

We define:

$$Rating_{n,m} = [\, J(P_n, C_m) + Score(C_m) \,] / 2 \quad (1)$$

$$J(P_n, C_m) = | P_n \cap C_m | / | P_n \cup C_m | \quad (2)$$

$$Score(C_m) = [discount\_score(AvgDiscount\_C_m) + delay\_score(AvgDelay\_C_m)] / 2 \quad (3)$$

$$\text{discount\_score} = \begin{cases} 1, & AvgDiscount_{Cm} \geq 0.5 \\ 0.75, & 0.4 \leq AvgDiscount_{Cm} < 0.5 \\ 0.50, & 0.25 \leq AvgDiscount_{Cm} < 0.4 \\ 0.25, & 0.05 \leq AvgDiscount_{Cm} < 0.25 \\ 0, & 0 \leq AvgDiscount_{Cm} < 0.05 \end{cases} \quad (4)$$

$$\text{delay\_score} = \begin{cases} 0, & AvgDelay_{Cm} \geq 0.5 \\ 0.25, & 0.2 \leq AvgDelay_{Cm} < 0.5 \\ 0.5, & 0.1 \leq AvgDelay_{Cm} < 0.2 \\ 0.75, & 0.05 \leq AvgDelay_{Cm} < 0.1 \\ 1, & AvgDelay_{Cm} < 0.05 \end{cases} \quad (5)$$

By using formulas 1-5, we calculate the (P_n, C_m) score matrix (Table 1).

Table 1: The (P_n, C_m) score matrix ($Rating_{n,m}$).

	P_1	P_2	P_3
C_1	0.8	0.8	0.3
C_2	0.6	0.7	0.5
C_3	0.7	0.4	0.8

Aiming to minimize the total construction cost of these projects, the problem is considered as a typical linear assignment problem (LAP), which can be easily solved through tools available in widely used software packages such as Google OR-Tools (https://developers.google.com/optimization/assignment/simple_assignment). Using the linear assignment solver of the above software package, we get the outcome presented in Table 2.

Table 2: (P_n, C_m) assignment matrix.

Project	P_1	P_2	P_3
Constructor	C_1	C_2	C_3

Aiming to further improve the accuracy of our estimations, we next consider the exploitation of Machine Learning algorithms, which are capable to provide knowledge-based patterns of construction projects' data. More specifically, we propose the use of the Apriori Algorithm to discover meaningful patterns (itemsets) relating P_n and C_m attributes.

Due to the fact that the Apriori algorithm requires a full search of the transactions' database in order to generate a *k*-large itemset, we limit our search to transactions containing only constructor C_1. We consider the transaction set $T = \{T_1, T_2, \ldots, T_{16}\}$ from a total of 685 transactions available in our database, concerning constructor C_1. The outcome of Apriori algorithm provides us with a "strong" supported 4-itemset that has been generated for constructor C_1 (see Table 3; it is noted that, due to space limitations, we present only the final step of the algorithm, omitting intermediate calculations of *k*-itemsets).

Table 3: L_4 itemsets for constructor C_1.

Large Itemset (L_4)	Support
BUILDINGS, LARGE_SCALE, MID_TERM, DELAY_LEVEL_0	3

In order to construct the association rules for C_1, we define a set of rules $R = \{\{R_1, Conf(R_1)\}, .. \{R_i, Conf(R_i)\}\}$, where:

$R_i = \{X\} \rightarrow \{Y\}$, where $\{X\}, \{Y\} \in \{L_4\}$ (6)

$$Conf(R_i) = \{X \cup Y\}/\{X\} \quad (7)$$

According to Equations (6) and (7), the set of rules produced is: $R = \{(R_1, 1), (R_2, 1), (R_3,1), (R_4, 1), (R_5,1), (R_6, 0.75), (R_7, 0.75), (R_8, 0.75), (R_9, 1), (R_{10},1), (R_{11},1), (R_{12}, 1), (R_{13}, 0.75), (R_{14}, 0.75)\}$

For each project P_n we apply the R_i, where $\{P_n \cap R_i\} \neq \emptyset$, and we calculate the corresponding confidence of the rule's application.

We define:

$$Conf(R_i \rightarrow P_n) = (P_n \cap R_i / R_i) \cdot Conf(R_i) \quad (8)$$

All R_i with $Conf(R_i) \geq 0.5$ are considered as legitimate association rules and can be applied to the initial available construction projects. In our case, the rule R_9 with $Conf(R_9) = 0.5$ has been applied to P_2 project (Table 4).

Table 4: Applying $R_i \rightarrow P_n$.

P_n	R_i	$\{P_n\} \cap \{R_i\}$	Confidence (R_i)	Prediction ($R_i \rightarrow P_n$)
P_2	R_9	{BUILDINGS, LARGE_SCALE}	0.50	{MID_TERM, DELAY_LEVEL_0}

We notice that in P_2 project (originally assigned to constructor C_2 – Table 2), the application of R_9 rule suggests (with high confidence) that an assignment augmentation should take place. In other words, $Prediction(R_9 \rightarrow P_2)$ denotes that: *"If C_1 constructor is selected for LARGE_SCALE BUILDINGS, there is a 50% possibility to complete P_2 project in MID_TERM duration and DELAY_LEVEL_0 delay time".*

Taking into consideration the above prediction, we update the original assignment matrix (Table 2). The new assignments are shown in Table 5. We note that for construction project P_2 ={*1002, ATHENS,BUILDINGS, 3,67M€, EU, 1050 DAYS*}, the above augmentation has a positive estimated

impact as it: (i) reduces its overall cost by 34.2% based on C_1 profile ($Avg\_Discount_{C1}$), (ii) provides minimum construction delay (*DELAY_LEVEL_0*) in the range [0, 0.05] and (iii) reduces the construction duration to *MID_TERM* (duration < 700 days).

Table 5: Augmented (P_n ,C_m) assignments.

Project	P_1	P_2	P_3
Constructor	C_2	C_1	C_3

Our approach is sketched in a pseudo-code form bellow:

```
for each (Cm)in transactions_DB do
create_profiles(Cm);
for each (Pn) in projects_DB do
{
   find top-n(Cm);
   calculate_Score(Pn,Cm) matrix;
}
assign(Pn,Cm);
for each C1 in transactions_DB do // Apriori
                        // Algorithm
{
   generate large k-itemsets (Lk) with
   minimum support (s);
   construct rules (Ri) with minimum
   confidence(Ri);
}
for each Pn in projects_DB do
{
      for each Ri do
      calculate_prediction(Ri,Pn);
}
assign(Pn,Cm);
```

To summarize the basic concepts of the above example, we addressed a PM issue as a typical OR assignment problem (a group of constructing companies need to accomplish a set of construction projects) using a score matrix with estimations for each (*Pn*, *Cm*) element. LAP solver algorithm provided a solution for the problem prescribing the optimal assignment matrix. Next, we exploited ML – Apriori algorithm to discover association rules between transactions' data to spot trends, relationships and structure similarity between data sets. In this way, we demonstrated that ML models and algorithms can be used to re-feed initial OR solutions, integrating OR's prescriptive analytics with ML's predictive analytics orientation.

Example 2

Consider another project management scenario concerning a software house, where there is a set $E=\{e_1, e_2, ..., e_n\}$ of available employees (i.e. software engineers). Each employee is described by an *ID* and an array of *skills*. In addition, there are two sets of new and past (completed) short-term tasks (e.g. fixing of software bugs), that are denoted by $N=\{n_1, n_2, ..., n_n\}$ and $P=\{p_1, p_2, ..., p_n\}$, respectively, which are described by an array of attributes, namely *[description, assignee, skills, duration]*.

By exploiting existing knowledge arising from past similar tasks, the company desires to minimize the total amount of time required to complete the *N* new tasks. This process can be accomplished through the following steps:

- Finding *top-K* employees for each task;
- Discovering clusters of similar tasks;
- Estimating tasks' durations, considering candidate employees;
- Assigning employees to tasks, by adopting the linear assignment problem (LAP) algorithm.

More specifically, the company, for each new task, discovers the *top-K* suitable employees by comparing tasks' requirements with each employee's skills. Obviously, for a specific task, the most capable employee is the one who meets all the required skills. Given a set *X* of a task's required skills and a set *Y* of an employee's identified skills, we define a score function $SF \in [0,1]$ as:

$$\mathrm{SF}(\mathrm{X},\mathrm{Y}) = \frac{|\mathrm{X} \cap \mathrm{Y}|}{|\mathrm{X}|} \quad (9)$$

Next, our approach uses the *k-means* clustering method to compose groups of similar tasks. The features (attributes) used in our case include *[description, assignee_skills, task_skills]*. It is known that data pre-processing and preparation are two fundamental steps in order for the *k-means* algorithm to work properly. Hence, certain data preparation techniques are applied to the features of each task; these include (i) calculation of *tf-idf* weights, removal of stop-words and stemming regarding textual data (e.g. the *description* feature), and (ii) conversion of an array of skills to binary values (e.g. the *skills* features). It is noted that our approach adopts the *Elbow method* (Trupti and Prashant, 2013) to determine the number of *k* groups.

For the abovementioned clustering requirements, we use the *scikit-learn* package (which is characterized by a wide adoption, simplicity, usability, well-written documentation and code stability). From the outcome of this step, we can estimate the time t_{ij} that each employee *i* needs to complete a task *j*, which is equal to the time of the task's group centroid. The output of this step is shown

in Table 6.

Table 6: Estimation of task durations per employee (minutes).

	N_1	N_2	N_3
E_1	118	'N/A'	'N/A'
E_2	63	546	116
E_3	287	179	184
E_4	'N/A'	245	587

Applying the LAP algorithm (https://developers.google.com/optimization/assignment/simple_assignment) to the elements of Table 6, the final step assigns employees to tasks. The outcome of this step is shown in Table 7).

Table 7: The task assignment matrix.

Task	N_1	N_2	N_3
Employee	E_1	E_3	E_2

It has been broadly admitted that employees have the tendency to underestimate or overestimate their skills, as well as the complexity of a certain task in order to estimate the time or cost required to accomplish it (Hill *et al*, 2000). As a consequence, time estimations deviate significantly from the reality, which in turn leads to miscalculations of projects' costs and durations, missing of deadlines, etc.

By estimating tasks' duration per employee (Table 6) through the exploitation of past data, our approach avoids the use of ad-hoc estimations and feeds the LAP algorithm with more accurate input. It is important to mention that the real duration of each task is recorded (and compared to the estimated one) for future use.

5 DISCUSSION

Key enablers that are driving the development of the proposed approach are the availability of huge computing power, the existence of big volumes of PM data and knowledge, as well as the accessibility of a range of well-tried and powerful OR and ML software libraries. Undoubtfully, there is more computing power available today than ever before, something that contributes significantly in making OR and ML algorithms extremely powerful, in ways that were not possible even a few years ago. In fact, this computing power enables us today to process massive amounts of PM data and extract valuable knowledge needed to make our models more intelligent. At the same time, as discussed in Section 2, software needed to process the diversity of PM data is open and freely available; it is also noted here that PM-related AI algorithms become available and get commoditized via dedicated APIs (Application Programming Interfaces) and cloud platforms.

Despite the above advancements, much work still must be done on the proper manipulation of PM data and knowledge, as far as its labeling, interrelation, modeling and assessment are concerned; and this has mainly to be done by humans. Especially in the context of project management, one should always take into account that valuable data and knowledge emerge continuously during an organization's lifecycle, and concern both the organization per se (e.g. a project's duration, overall budget, KPIs etc.) and its employees (e.g. one's competences and performance, knowledge shared during a decision-making process etc.).

Building on a meaningful and flexible integration of OR and ML techniques and associated tools, our approach enables organizations to reap the benefits of the AI revolution. It allows for new working practices that may convert information overload and cognitive complexity to a benefit of knowledge discovery. This is achieved through properly structured data that can be used as the basis for more informed decisions. Simply put, our approach improves the quality of PM practice, while enabling users to be more productive and focus on creative activities. However, diverse problems and limitations still exist; these concern the value and veracity of existing data, as well as the availability of the massive amounts of data required to drive contemporary AI approaches.

Future work will concentrate on the consideration of more complex PM issues aiming to identify additional useful combinations of ML and OR algorithms. Another work direction concerns the embedment of explainability features in the recommendations provided by the proposed approach (Karacapilidis et al., 2017).

6 CONCLUSIONS

Considering contemporary PM challenges as well as the strengths of techniques originally developed in the context of the ML and OR fields, this paper presents a hybrid approach that aims to advance the

overall PM practice. The proposed approach can assist employees in common PM tasks such as resource assignment, estimation of task duration, and prediction about whether deadlines will be met. The proposed advancement of the PM practice lies in the proper orchestration of OR and ML algorithms by paying simultaneous attention to both optimization and big data manipulation issues.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J. and Kudlur, M. (2016). Tensorflow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 16(1), pp. 265-283.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules, In: *Proceedings of 20th int. conference on Very Large Data Bases (VLDB 1215)*, pp. 487-499.

Burke, L. I. and Ignizio, J. P. (1992). Neural networks and operations research: An overview. *Computer & Operations Research*, 19(3), pp.179-189.

Candel, A., Parmar, V., LeDell, E. and Arora, A. (2016). Deep learning with H2O. 6th ed. [pdf] H2O.ai Inc., Available at http://h2o.ai/resources/ [Accessed 19 Oct. 2018].

Cheung, S. O., Wong, P. S. P., Fung, A. S. and Coffey, W. (2006). Predicting project performance through neural networks. *International Journal of Project Management*, 24(3), pp. 207-215.

Costantino, F., Gravio, G. D. and Nonino, F. (2015). Project selection in project portfolio management: An artificial neural network model based on critical success factors. *International Journal of Project Management*, 33(8), pp. 1744-1754.

Dunning, I., Huchette, J. and Lubin, M. (2017). Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2), pp. 295-320.

Goodfellow I., Bengio Y. and Courville A. (2016). *Deep Learning*, The MIT Press. Cambridge, MA, USA.

Hill J., Thomas L.C. and Allen D.B. (2000). Experts' estimates of task durations in software development projects. *International Journal of Project Management*, 18(1), pp. 13-21.

Holmes, G., Donkin, A. and Witten, I. H. (1994). Weka: A machine learning workbench. In *Proceedings of the 1994 Second Australian and New Zealand Conference*, pp. 357-361.

Jaccard, P. (1901). Étude comparative de la distribuition florale dans une portion des Alpes et des Jura. *Bull Soc Vandoise Sci Nat*, 37, pp. 547-579.

Karacapilidis, N., ed. (2014). *Mastering Data-Intensive Collaboration and Decision Making: Cutting-edge research and practical applications in the Dicode project.* Studies in Big Data Series, Vol. 5, Springer, Cham.

Karacapilidis, N., Malefaki, S. and Charissiadis, A. (2017). A novel framework for augmenting the quality of explanations in recommender systems. *Intelligent Decision Technologies Journal*, 11(2), pp. 187-197.

Levitt, R. E. and Kunz, J. C. (1987). Using artificial intelligence techniques to support project management. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1(1), pp. 3-24.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z. and Lerer, A. (2017). Automatic differentiation in pytorch. In *31st Conference on Neural Information Processing Systems*, pp. 1-4.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(1), pp. 2825-2830.

Smith, K. A. and Gupta, J. N. (2000). Neural networks in business: techniques and applications for the operations researcher. *Computers & Operations Research*, 27(11), pp. 1023-1044.

Svejvig, P. and Andersen, P. (2015). Rethinking project management: A structured literature review with a critical look at the brave new world. *International Journal of Project Management*, 33(2), pp. 278-290.

Trupti M. K. and Prashant R. M. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1(6), pp. 90-95.

Wang, Y. R., Yu, C. Y. and Chan, H. H. (2012). Predicting construction cost and schedule success using artificial neural networks ensemble and support vector machines classification models. *International Journal of Project Management*, 30(4), pp. 470-478.

Zhang, G., Patuwo, B. E. and Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1), pp. 35-62.