A Systematic Review of Literature on Recommendation Systems and Machine Learning Applied to Multiagent Systems *

Karine Pestana Ramos[1] **and Marilton Sanchotene de Aguiar**[1]

[1] Programa de Pós-Graduação em Computação
Centro de Desenvolvimento Tecnológico
Universidade Federal de Pelotas
Pelotas, RS – Brazil

{kpramos,marilton}@inf.ufpel.edu.br

Abstract. *This study developed a systematic review of the literature (SRL), a formal study that is used to map a specific area of knowledge. The main question is defined that will guide the entire search during SRL including other formal steps. This SLR has been defined to synthesize and integrate the areas of multi-agent systems, machine learning, and recommendation systems. At the end of the SRL, six studies with different characteristics were found that were adequate to the main question and that satisfy the selection criteria.*

1. Introduction

A Systematic Review of Literature (SRL) is a methodology study that aims to synthesize formally through well-defined steps, the main studies available in a specific area [Okoli and Schabram 2010].

According to [Galvão and Pereira 2014] the systematic reviews must be comprehensive and non-biased in its preparation. So that the researcher can repeat the steps taken and achieve the same results.

To do a good SRL is necessary a well-defined question beyond criteria for identifying, selecting, evaluating, and synthesizing the study results. This work aims to, from an SRL according to the steps mentioned above, raise the possibilities of integrating the areas of Multi-Agent Systems (MAS), Machine Learning (ML), and Recommendation Systems (RS).

This SRL is developed as a part of a major study. It aims to be used as a base for an application in ML for a MAS. To provide a good basis for future works this research is very important to know some information such as, the state of art in the area, how authors have been creating this kind of application, and the system and tools that are used.

The paper is structured as following: in Section 2 are presented the main concepts in the theoretical framework of MAS (Sec. 2.1), ML (Sec. 2.2) and, RS (Sec. 2.3). Section 3 presents the methodology (Sec. 3.1) applied to SRL in this paper (such as

* O trabalho A Systematic Review of Literature on Recommendation Systems and Machine Learning Applied to Multiagent Systems de Karine Pestana Ramos e Marilton Sanchotene de Aguiar está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial 4.0 Internacional.http://creativecommons.org/licenses/by-nc/4.0/

databases and keywords), the evaluation steps (Sec. 3.2), the search results (Sec. 3.3), and the discussion of selected articles. Finally, in Section 4 the conclusions are presented.

2. Theoretical Framework

This Section presents briefly the main concepts about Multi-Agents Systems, Machine Learning, and Recommendation Systems, necessary for understanding the SRL.

2.1. Multi-Agent Systems

According to [Singh 1994], a Multi-Agent System (MAS) is an example of distributed computing, composed of the interactions of various computational entities. They are systems capable of modeling theories and human societies.

The term agent is widely used in computing and is usually abstracted from their knowledge, intentions, and desires [Singh 1994]. An agent is considered a basic unit of system intelligence in the scope of this study.

This area aims to solve problems collectively, in a way that a single agent would not be able to perform such a task alone [Hübner et al. 2004]. Thus, it has its focus on the collectivity of the environment, defining the integration between agents and their forms of organization among themselves.

Some characteristics are common to MAS [Weiss 1999]: agents have only incomplete information; there are restrictions on their actions; the control of the system decentralized or distributed; the data is also distributed; and, there is asynchronicity.

As part of an intelligent system, an agent can perceive and act on the environment and, at least partially, his actions are defined by his experience in the environment [Weiss 1999]. In MAS, an autonomous agent is an agent that is independent of other agents. It is important to emphasize that an agent will always be limited by his knowledge.

2.2. Machine Learning

Machine Learning (ML) can be seen as a branch of Artificial Intelligence (AI) as it is a statistical model that has had its progressive growth in the field of computer science.

It has become a common means for any task that requires significant pattern recognition in large databases [Shalev-Shwartz and Ben-David 2014]. Learning applications must be able to learn and adapt. The goal is to program software able to learn given a certain input. Effectively performing specific tasks without the use of explicit instructions, with the use of patterns and inferences [Harrington 2012].

According to [Shalev-Shwartz and Ben-David 2014] learning is the process of incorporating experiences so it becomes specialties or knowledge. The entries in the algorithm are expected to serve as experience to be possible to generate inferences, generating knowledge in its output.

Knowledge of the algorithm should not be limited to its initial training, so like humans, a good algorithm can during its execution be constantly learning as new entries are inserted and learn about their errors [El Naqa and Murphy 2015].

There are two main machine learning classifications: supervised learning and unsupervised learning. In supervised learning, it is necessary a previous step named data training, more specifically, a set of data that must be used as a learning base for the eventual classification of new entries.

In unsupervised learning, however, learning occurs without previous knowledge. Thus, the algorithm does not necessarily attempt to associate each entry with a label (as in supervised learning). The algorithm learns about the input data entered.

Finally, there is learning classification, semi-supervised learning that blends the two previous types of learning, where one part of the data has one classification and the other part doesn't. Thus the already classified part can be used to assist in the classification of the unclassified part.

2.3. Recommendation Systems

Recommendation Systems (RS) are algorithms to establish filtering for the user. It is intended to assist individuals in their choices among a usually large set of options that could cause difficulty for the user to decide. These are systems that seek to facilitate the laborious activity of searching for interesting content.

Word of mouth is a common form of indication where the individual relies on someone else's recommendation (usually of a similar taste or profile). The purpose of recommendation systems is precisely to make this connection even that no one directly indicates something to the individual.

For the recommendation to succeed the group that should receive the recommendation must include individuals with common interests (explicit) or common behavior (implicit) [Cazella et al. 2010]. However, it is a big challenge to do this mapping correctly, creating a correct match for the expectations of users with the products or services that may be recommended to them. To define and discover this relationship of interests can be difficult.

Recommendation systems are widely used in e-commerce sites, using various techniques to provide the most appropriate product recommendations to customers, thereby increasing their profitability.

3. Systematic Review

This Section contains the specification of each step of the methodology defined for the execution of the SRL and a presentation of the main studies found by this review.

3.1. Methodology

The first step to be defined is the question that will guide the entire literature search, which was established as "What are the main articles involved in the context of multi-agent systems using machine learning and recommendation systems?".

The Table 1 is intended to summarize how the main elements were defined for the review, including the main issue, among other information such as objective search criteria, inclusion, and exclusion criteria. The next step is to define the databases for literature research. The selected databases for this study are listed in Table 2 as suggested by [Kofod-Petersen 2012].

Table 1. Systematic literature review protocol

Main Question	"What are the main articles involved in the context of multi-agent systems using machine learning and recommendation systems?"
Goal	Search the literature for studies that are in the context of multi-agent systems using machine learning and recommendation systems only from the last 10 years (2010–2019).
Inclusion Criteria	Articles that contain the following key terms and synonyms: i) **Multi-agent Systems**; ii) **Machine Learning**; and, iii) **Recommendation Systems**
Exclusion Criteria	i) the study was not in English; ii) the study was not found in the databases searched; and, iii) the study does not have the searched keywords.

Table 2. Databases used in the study

Name	**URL**
ACM	`https://dl.acm.org/`
IEEE Xplore	`https://ieeexplore.ieee.org/Xplore/home.jsp`
Science Direct	`https://www.sciencedirect.com/`
Springer Link	`https://link.springer.com/`

Since the protocols of the research and the databases have already been established, the next step is necessary to define the key terms that will be used in the searches. According to [Kofod-Petersen 2012] the main point is to form a string that represents the key terms searched. For instance, if we wanted to search for the term multiagent, the string should not contain only this variation of the term but should also include other variations that appear in the literature, such as multi-agent, multiagent system, among others.

The same method was used for the other searched terms, so the terms are included with their synonyms. In the search result, we desire to find articles that contain any of the terms of that synonym group and equally for the two other term groups. Boolean logical operators like OR and AND are used to achieve this. Finally, it is possible to observe the compilation of these terms in a string generated as a result in Table 3.

After collecting articles found in the four databases cited in Table3, we use two software for reference management to help us organize the results: JabRef and Mendeley. JabRef is used to remove duplicate articles, considering that sometimes an article is found in more than one database [Team et al. 2011]. Mendeley is used for reading and annotating in the articles, working synchronously for several devices [Singh et al. 2010].

3.2. Evaluation Steps

The evaluation steps of the articles found were applied as indicated in the study of [Kofod-Petersen 2012]. The method consists of three steps: title and abstract evaluation, introduction evaluation, and full-text evaluation, applied in exactly that order.

In title and abstract evaluation, the authors read the title and abstract to evaluate if each study is in the context of SRL's main question, considering the inclusion and exclusion criteria defined in Table 1. So, studies outside the theme are eliminated from the research.

In the introduction evaluation, the authors read introductions from every article approved in the stage before. The evaluation occurs by analyzing if the introduction is

Table 3. Key terms used to define the search string

(“multiagent system”	**OR**	“multiagent systems”	**OR**
“multiagents system”	**OR**	“multiagents systems”	**OR**
“multi-agent system”	**OR**	“multi-agent systems”	**OR**
“multi-agents system”	**OR**	“multi-agents systems”	**OR**
“agent-based system”	**OR**	“agent-based systems”	**OR**
“agent system”	**OR**	“agent systems”	**OR**
“agents system”	**OR**	“agents systems”	**OR**
“agent based model”	**OR**	“agent based model”	**OR**
“agent-based model”	**OR**	“agent-based models”)	
	AND		
(“machine learning”	**OR**	“automatic learning”	**OR**
“machine learning techniques”)			
	AND		
(“decision support system”	**OR**	“decision support systems”	**OR**
“recommendation system”	**OR**	“recommendation systems”)	

related to SRL’s main question, besides considering the inclusion and exclusion criteria. The studies approved in this stage still need to pass by a final evaluation.

The last evaluation stage is the reading of the full text of the articles that remain under evaluation. In full-text evaluation, the methodology used is very specific of computing area, where for each article read we analyzed and answered ten questions, whose answers can be yes (1 point), partially (1/2 point) or no (0 points).

Composing a final grade for each article evaluated. The cutoff note is defined by the researchers. In this study, the cutoff note is 6 points. Below are listed the 10 specified questions defined by [Kofod-Petersen 2012]:

1. “Is there is a clear statement of the aim of the research?”
2. “Is the study put into context of other studies and research?”
3. “Are system or algorithmic design decisions justified?”
4. “Is the test data set reproducible?”
5. “Is the study algorithm reproducible?”
6. “Is the experimental procedure thoroughly explained and reproducible?”
7. “Is it clearly stated in the study which other algorithms the study’s algorithm(s) have been compared with?”
8. “Are the performance metrics used in the study explained and justified?”
9. “Are the test results thoroughly analyzed?”
10. “Does the test evidence support the findings presented?”

3.3. Search Results

When performing the searches we collected 206 articles referring to the review in question, which is distributed according to Table 4. Of the 206 articles extracted from the database after removing the duplicates, it remained 190 which were finally analyzed by the study.

In the first evaluation step, title and summary led to the removal of 163 of the articles, leaving only 27 to be analyzed in the next step. The evaluation of the introduction

Table 4. Database search results

Database	Quantity of Articles
ACM	21
IEEE Xplore	22
Science Direct	110
Springer Link	53

restricted the study to only 11 articles to be evaluated for their complete text. The analysis of the last step is summarized in Table 5, where the articles are evaluated as defined in Section 3.2. Numbers 1 through 10 correspond to the questions set out in the same Section (3.2), respectively.

Table 5. Results of last stage

Article Title	1	2	3	4	5	6	7	8	9	10	Points
A Multiagent Approach to Ambulance Allocation Based on Social Welfare and Local Search [Shaft and Cohen 2013]	1	1	0.5	0	0.5	0.5	0.5	0.5	1	1	6.5
A network aware approach for the scheduling of virtual machine migration during peak loads [Duggan et al. 2017]	1	1	0.5	0	0.5	0.5	1	1	1	1	7.5
A novel meta learning system and its application to optimization of computing agents' results [Kazík et al. 2012]	1	1	1	0	0	0.5	1	1	1	1	7.5
Agent-based Decision Support System for Technology Recommendation [Legien et al. 2017]	1	1	1	0.5	0.5	1	1	0.5	0.5	0.5	7.5
ALBidS: A Decision Support System for Strategic Bidding in Electricity Markets [Pinto and Vale 2019]	1	0.5	0.5	0	0	0.5	0	0	0	0	2.5
Decision Support System for Opponents Selection in Electricity Markets Bilateral Negotiations [Silva et al. 2019]	1	0	1	0	0	0	0	0	0	0	2
Decision support system for predicting stock prices based on sentiments in social media [Chornous and Iarmolenko 2017]	1	1	1	0.5	0.5	0.5	1	0	0	0	5.5
Intelligent Digital Learning : Agent-Based Recommender System [Brigui-Chtioui et al. 2017]	1	1	1	0.5	0.5	0.5	1	0.5	0	0	6
Key Technologies of Confrontational Intelligent Decision Support for Multi-Agent Systems [Zhang 2018]	0.5	0.5	1	0	0.5	0.5	0.5	0	0	0	3.5
Real-time Machine Learning Prediction of an Agent-Based Model for Urban Decision-making [Zhang et al. 2018]	1	0	0	0	0.5	0.5	0	0.5	0	0.5	3
Strategic advice provision in repeated human-agent interactions [Azaria et al. 2016]	1	1	0.5	0.5	0.5	0.5	1	1	1	1	6

For the selection of the final articles, they were ranked according to their final score from Table 5. In case of the same score, the rank order presented in alphabetical order. As a cutoff note, it was decided to use at least 6 points, so the study needs to address at least 60% of punctuated criteria.

3.4. Selected Papers

In [Duggan et al. 2017] the study aims at mastering available resource utilization when there are high cloud network saturation peaks. This analysis and prediction of volume traffic used by the real-time network have been studied in ways to efficiently utilize the resources available at times of high access to cloud data centers. Enabling the migration of virtual machines (VMs) without congesting network traffic.

The difficulty of transmission is to balance network resources with network traffic without loss of resources. A single agent is implemented that through pro-reinforcement learning can detect the best moment for live migrations. Thus, the approach is adaptive and able to learn the best moments for the migration of VMs to occur, reducing the network resource saturation at peak times, and the migration time itself.

This study produces a testing environment to simulate a band and its traffic demands. Simulating how the agent can handle the migration situation VMs allowing the agent to decide to schedule the migration or just wait. The reinforcement learning system acts as decision support in the live migration process for testing the developed environment for simulation. The environment allows the agent to observe and maximize rewards based on actions taken. The agent is allowed to migrate or wait.

The study shows through simulated scenarios as results that the agent can learn to migrate when network traffic utilization is low, reducing congestion at peak times and improving the use of network resources during off-peak hours and decreasing the costs. Thus is concluded that the reinforcement learning approach is satisfactory as it provides a live migration framework that has no impact on a cloud computing system, increasing quality of service for cloud clients and providers.

In [Kazík et al. 2012], authors develop a multi-agent system that incorporates different data mining methods. The formulated environment is capable of testing different mining methods and finds the best settings. The AGR model is employed (agent, group, and role), where roles allow the flexible composition of multi-agent systems into group structures. The agent may join a group only by performing a function defined in the structure of the group in question. Two agents interact through communication protocol defined between two roles they assumed. For interaction between two agents, they must be in the same group.

The system learns from the results of previous tasks, gathering experience, becoming able to find the best method possible even for the new datasets. This learning works through basic training.

The developed system tests are divided into two sets. The first set used a simple training error as a comparison criterion, while the second group performed cross-validation five times on the data set. Three search algorithms were compared: search by genetic algorithm (with and without elitism), simulated annealing, and random searching. Study experiments have shown that learning can lead to better models. Considering simpler sets local searches are sufficient and for more sophisticated sets genetic algorithms give better results.

The work of [Legien et al. 2017] is a multi-agent implementation of a labeled deduction system in which agents are used to decomposing the system in knowledge distribution and processing. Agents share a common API that enables their communication. Agents are required to register in the system, so the list of registrations is always updated as agents register or unregister.

Machine learning algorithms can be performed during an inference process. In the developed implementation the concept of breadth-first search (BFS) is used. This math is performed by a specific agent and for distribution math, the reasoning process can be performed by a group of agents working in parallel.

The structural part of the project is divided into two parts, one being the API of the agents and another part are the components of the environment. The first involves the interface and the agent request classes for cooperation with other agents. The second, registration in the environment, sending requests, and sending responses.

The environment has an agent that acts as an administrator, able to register agents working in the environment, and statistics. The study experiments presented in [Legien et al. 2017] simulate a situation in which an environment based on agents is a back end of the metallurgical field recommendation system. Looking for better foundry recommendations. In the beginning, all agents need to register with the system. They submit a request for reasoning, make inferences, calculate a cost of proof, and display the results of the process, cooperating in a distributed environment. The developed model has its advantages. It is flexible and adding modules is easy.

The scenario involved in the study of [Shaft and Cohen 2013] involves a local search in a health care environment with multiple victims and hospitals throughout an accident. The goal is that given an accident with many victims with different severity of injuries they need to be allocated to an ambulance and eventually to a hospital. Generically the problem can be seen as a situation of multi-agent resource allocation.

For the correct allocation of ambulances, it is important to know the state of patients' health to be included in the decision process. The simulation assumes that everyone involved in the accident needs medical attention and needs to go to a hospital, so is created a one-to-one relationship with the number of injuries and ambulances available. Besides, the resources available at the hospital do not necessarily meet the needs of patients.

The problem addressed by the study [Shaft and Cohen 2013] is an optimization problem and therefore the goal is to find a solution that maximizes the total cost function. Three points influence the cost function. The first point is related to hospital capacity. If there are more patients assigned to a hospital than their capacity, the cost function should return to an unfavorable result. The second point involves the resources, i.e. if the patient is assigned for a hospital with fewer resources than necessary for its care it should also return unfavorable values. This situation implies that the patient will have to wait at the hospital for resources to be available that they need. The last point is a balance involving the experience of the ambulance driver, patient severity, and distance to the hospital. Experiments were developed to test the above approach and the results were satisfactory. Scenarios were tested with distributions of different accidents. The study also indicates the application of the work in real scenarios, despite citing several possible enhancements and expansions.

The study presented in [Brigui-Chtioui et al. 2017] is an approach about digital learning, so an agent-based recommendation system is developed which aims to help students supply knowledge in areas that they are deficient by suggesting relevant learning. The main objective is that with the right support you can make learning more efficient.

The system is cooperative and agent-based, agents are autonomous and able to update recommendations, enabling improvements in the recommendation's outcome based on previous learning platform experiences. The recommendation process occurs from a recommender agent who has unrestricted access to platform data and can be viewed as estimate scores for items not discovered by the user. The recommending agent can judge the relevance of each event. In general, each agent has a communication module that allows message exchange according to the communication protocol.

In the system in question the recommendation deals with learning resources and the users are students. The item rating is represented by a score. The system is based on item decomposition into singular values for analysis of score vs. item. A collaborative filtering process occurs, where similarity is calculated between the references of different users. This assumption works because it assumes that those who liked a particular item on the market tend to continue to like that particular item (or very similar items).

On the developed platform new users have initial data through assessments, but on other systems, one may have no knowledge about the user and then the most popular activities (set as default) are recommended. Initial assessments allow the recommendation system to learn student preferences. Despite highlighting points that could be further explored, the study achieves its objective in developing the proposed system, creating a system with a set of autonomous cooperative agents able to recommend useful resources to the students, to fill their gaps or deficiencies detected.

The study by [Azaria et al. 2016] aims to develop strategies that allow the system to advise on scenarios involving computers agents and people. Scenarios have been modeled in which humans or computing agents can share certain goals, but are essentially interested in themselves.

The authors proposed a social agent to provide counseling. This agent is responsible for modeling the selection of human choice. For the agent to achieve success, their advice must have a high probability of acceptance by people without disregarding the fulfillment of the agent's individual goals. The agent is tested in two different environments, using hundreds of human subjects, scenarios varying in complexity, human behavior modeling, and information that is available to people to make them decide whether or not to accept the agent's advice.

The study demonstrates two possible agent designs, one using a Markov Decision Process (MDP) and one using a social preference model. The proposed agent is a social agent for providing counseling (SAP), which generates counseling according to the following social model. The approach balances the costs for the two participants in the selection process based on social weight. Experiments for the evaluation of SAP and MDP agents were tested in a route selection environment. Four possible commute routes for the driver to commute to work. The system advises a route but the driver does not necessarily have to choose it. The driver's goal is to make the journey in less time. The system considers other factors such as minimum fuel consumption and route safety. Only the agent knows the cost of each route. The tests were applied to men and women who had only as information that their objective was to optimize the route time.

It is concluded that SAP achieves goals and outperforms other implementations, and is simple to implement because its advisory strategy does not depend on the history of interaction with your current user, making it possible to use it in many common situations where there is no knowledge of the number of times that the users have used the system.

The studies discussed so far present different approaches and each paper focus on a very singular problem. The Table 6 summarizes and highlights the use of multi-agent systems and recommendation systems in the final articles.

Table 6. Comparative summary of the selected papers

Papers	Features (Multi-Agent Systems and Recommender Systems)
[Duggan et al. 2017]	**MAS**: A single agent that learns through reinforcement learning. **RS**: It allows the system to schedule optimal times for live migration.
[Kazík et al. 2012]	**MAS**: Multiple agents are implemented as computational intelligence methods. **RS**: Capable of recommending the appropriate method according to the data.
[Legien et al. 2017]	**MAS**: Multiple agents are responsible for distributing and processing knowledge. **RS**: Depends on the application, in experiments of the article the environment based on agents is a back end of the metallurgical field recommendation system.
[Shaft and Cohen 2013]	**MAS**: Agents are treated as resource allocation problems. **RS**: The system is capable of deciding how to allocate and schedule ambulances during a mass casualty incident through a relative to a chosen social welfare function.
[Brigui-Chtioui et al. 2017]	**MAS**: Use of autonomous agents able to recommend and update recommendations for the learning in the platform. **RS**: Agents act as recommender agents.
[Azaria et al. 2016]	**MAS**: Propose a social agent provision (SAP) able to generate advice. **RS**: Uses the proposed agent.

4. Conclusions

SRL aims to synthesize the best studies of specific subarea in the literature. This study has detailed stage by stage of the procedures for elaboration of an SRL for the integration of multi-agent systems, machine learning, and recommendation systems areas, obtaining satisfactory results in this research and demonstrating it is possible to find this intersection point. Methodology and steps performed in this study were well described here making this SRL, as it should be, reproducible.

Although the six finalist studies are about multi-agent systems, recommendation systems, and machine learning with their topics and purposes differ in many ways. Studies diverge in the broadest aspects some are more detailed and others do not describe so deeply their implementation or testing or even the experiments of the work in question.

The studies have different characteristics, as the article by [Duggan et al. 2017] is only implemented with a single agent, and in the study by [Azaria et al. 2016] the implemented agents are for comparison purposes. Unlike other studies that also implement more than one agent, however, they work cooperatively or just perform different functions in the system.

In the study described by [Brigui-Chtioui et al. 2017] there is no information about experiments and there is no discussion about some system tests developed, other studies all included some system tests. Some use searches to support the decision process, which may be local searches, as in the study by [Shaft and Cohen 2013] or as in the study by [Kazík et al. 2012] which makes use of genetic algorithm searching, simulated tempering and random searching. The study covered by [Duggan et al. 2017] is the only that explicitly cites the use of reinforcement learning.

At the end of the SRL, six studies were found that were adequate to the main question and that satisfy the selection criteria. Although the studies have several different points as discussed in Section 3.4, essentially these studies represent a set of the intersection of MAS, machine learning, and recommendation systems as the main point of this research.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoameto de Pessoal de Nível Superior (CAPES/Brasil) and Agência Nacional de Águas (ANA/Brasil) – Edital N° 16/2017.

References

Azaria, A., Gal, Y., Kraus, S., and Goldman, C. V. (2016). Strategic advice provision in repeated human-agent interactions. *Autonomous Agents and Multi-Agent Systems*, 30(1):4–29.

Brigui-Chtioui, I., Caillou, P., and Negre, E. (2017). Intelligent digital learning: agent-based recommender system. In *Proceedings of the 9th International Conference on Machine Learning and Computing*, pages 71–76.

Cazella, S. C., Nunes, M., and Reategui, E. (2010). A ciência da opinião: Estado da arte em sistemas de recomendação. *André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski..(Org.). Jornada de Atualização de Informática-JAI*, pages 161–216.

Chornous, G. and Iarmolenko, I. (2017). Decision support system for predicting stock prices based on sentiments in social media. In *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing*, pages 1–4.

Duggan, M., Duggan, J., Howley, E., and Barrett, E. (2017). A network aware approach for the scheduling of virtual machine migration during peak loads. *Cluster Computing*, 20(3):2083–2094.

El Naqa, I. and Murphy, M. J. (2015). What is machine learning? In *machine learning in radiation oncology*, pages 3–11. Springer.

Galvão, T. F. and Pereira, M. G. (2014). Revisões sistemáticas da literatura: passos para sua elaboração. *Epidemiologia e Serviços de Saúde*, 23:183–184.

Harrington, P. (2012). *Machine learning in action*. Manning Publications Co.

Hübner, J. F., Bordini, R. H., and Vieira, R. (2004). Introdução ao desenvolvimento de sistemas multiagentes com jason. *XII Escola de Informática da SBC*, 2:51–89.

Kazík, O., Peková, K., Pilát, M., and Neruda, R. (2012). A novel meta learning system and its application to optimization of computing agents' results. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 170–174. IEEE.

Kofod-Petersen, A. (2012). How to do a structured literature review in computer science. *Ver. 0.1. October*, 1.

Legien, G., Sniezynski, B., Wilk-Kołodziejczyk, D., Kluska-Nawarecka, S., Nawarecki, E., and Jaśkowiec, K. (2017). Agent-based decision support system for technology recommendation. *Procedia Computer Science*, 108:897–906.

Okoli, C. and Schabram, K. (2010). A guide to conducting a systematic literature review of information systems research. *SSRN Electronic Journal*.

Pinto, T. and Vale, Z. (2019). Albids: A decision support system for strategic bidding in electricity markets. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2375–2377. International Foundation for Autonomous Agents and Multiagent Systems.

Shaft, D. and Cohen, R. (2013). A multiagent approach to ambulance allocation based on social welfare and local search. In *2013 12th International Conference on Machine Learning and Applications*, volume 2, pages 384–389. IEEE.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Silva, F., Pinto, T., and Vale, Z. (2019). Decision support system for opponents selection in electricity markets bilateral negotiations. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2384–2386. International Foundation for Autonomous Agents and Multiagent Systems.

Singh, J. et al. (2010). Mendeley: A free research management tool for desktop and web. *Journal of Pharmacology and Pharmacotherapeutics*, 1(1):62.

Singh, M. P. (1994). *Multiagent systems*. Springer.

Team, J. D. et al. (2011). Jabref [software].

Weiss, G. (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press.

Zhang, Y. (2018). Key technologies of confrontational intelligent decision support for multi-agent systems. *Automatic Control and Computer Sciences*, 52(4):283–290.

Zhang, Y., Grignard, A., Lyons, K., Aubuchon, A., and Larson, K. (2018). Real-time machine learning prediction of an agent-based model for urban decision-making. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2171–2173. International Foundation for Autonomous Agents and Multiagent Systems.