APRENDIZADO DE MÁQUINA APLICADO À PREVISÃO DE RESPOSTAS EM PROBLEMAS DE SIMULAÇÃO A EVENTOS DISCRETOS

Wilson Trigueiro de Sousa Junior
Departamento de Engenharia Mecânica, Universidade Federal de São João del-Rei - UFSJ
Praça Frei Orlando, 170, Centro - Campus Santo Antônio, São João del-Rei, MG
wilson.trigueiro@ufsj.edu.br

Rafael de Carvalho Miranda
Instituto de Engenharia de Produção e Gestão, Universidade Federal de Itajubá - UNIFEI
Av. BPS, 1303 - Pinheirinho, Itajubá - MG
rafael.miranda@unifei.edu.br

José Arnaldo Barra Montevechi
Instituto de Engenharia de Produção e Gestão, Universidade Federal de Itajubá - UNIFEI
Av. BPS, 1303 - Pinheirinho, Itajubá - MG
montevechi@unifei.edu.br

RESUMO

Com a crescente utilização de ferramentas de aquisição e armazenagem de dados, na era da indústria 4.0, faz-se necessário a utilização de metodologias para a captura de padrões e análise de dados, de modo que esses dados possam ser transformados em informações úteis e auxiliarem os processos de tomada de decisão. Nesse sentido, o presente trabalho aplicou 12 métodos de aprendizagem de máquina para um problema de avaliação de possíveis soluções de investimentos em um sistema industrial genérico para uma variação do problema M/M/k, considerando os lucros gerados pelo mesmo. Como resultado, encontrou-se métodos com taxa de acerto de 99% e a possibilidade de integração dos mesmo com métodos de otimização tradicionais para estudos futuros.

PALAVRAS CHAVE. Aprendizado de máquina, Simulação a eventos discretos e Previsão.

Tópicos. Simulação.

ABSTRACT

With the growth on the utilization of tools to acquire and store data on the era of the 4.0 industry, methodologies are necessary to capture patterns and for data analysis, in a way that those data can be transformed on useful information and help in the process of decision making. The present paper applied 12 machine learning methods to a problem of the evaluation of possible investment solutions on a variation for the generic industrial system of the type M/M/k, considering the profits. As result, methods found a prediction rate of 99% and the possibility to integrate with traditional optimization techniques in future research.

KEY WORDS: Machine learning, Discrete event simulation and Prediction.

Topics. Simulation.

https://proceedings.science/p/85540?lang=pt-br

1. Introdução

O processo de análise e tomada de decisão envolve a determinação de quais variáveis são mais relevantes para a caracterização de um problema, de modo que a partir desse momento, informações sejam coletadas para a correta dimensão dessas variáveis [Lopes *et al.* 2017].

A tomada de decisão em cenários que envolvam a interação de variáveis estocásticas é beneficiada pela aplicação das técnicas de modelagem e simulação a eventos discretos (MSED) [Saadouli *et al.* 2015; Lopes *et al.* 2017]. Técnicas estas capazes de modelar computacionalmente sistemas reais complexos, e posteriormente, analisar os dados coletados, com a vantagem de: avaliação em um ambiente seguro e controlado das possíveis alterações no estado inicial de um problema, o baixo custo de implementação e avaliação de alternativas diferentes da original e melhor visualização e compreensão de processos complexos que envolvam diferentes taxas de processamentos e que possam gerar "gargalos" [Fonseca, Navaresse e Moynihan 2003; Altiparmak, Dengiz e Bulgak 2007; Mehrsai *et al.* 2013].

Como possíveis desvantagens do uso da simulação, podemos citar: a necessidade de conhecimento especializado para geração e análise dos modelos, o custo da coleta de dados, e a não garantia de que o sistema real final será igual ao simulado [Fonseca, Navaresse e Moynihan 2003; Altiparmak, Dengiz e Bulgak 2007; Mehrsai *et al.*, 2013]. Mesmo com as desvantagens apresentadas, vários estudos vêm apresentando resultados que corroboram com a ideia de eficiência da aplicação da MSED. Como exemplos, podemos citar estudos nas áreas de produção de bens e serviços tais como: hospitalar, militar, siderúrgica, logística, manufatureira, gás e petróleo [Alexopoulos *et al.* 2001; Martins *et al.* 2013; Nageshwaraniyer, Son e Dessureault 2013; Lucidi *et al.* 2016; Salam and Khan 2016].

Em tais situações, são avaliados um ou mais cenários, a partir da alteração das variáveis, para respostas a perguntas do tipo: "e se...?". De acordo com o tipo de problema, as variáveis envolvidas e características do espaço solução do problema, é possível a utilização de várias técnicas de otimização, tais como: análise envoltória de dados, heurísticas como *hill-climb*, [Kamrani, Ayani e Moradi 2012; He, Huang e Chang 2015] e meta-heurísticas como: Algoritmos Genéticos, GRASP, VNS, Colônia de Formigas, etc. [Vitanov, Vitanov e Harrison 2009; Akiya, Scott Bury e Wassick 2011; Gansterer, Almeder e Hartl 2014; Miranda, Montevechi e Pinho 2015; Muta *et al.* 2015].

Para as técnicas de otimização citadas, as variáveis devem ser prioritariamente do tipo ordinal ou contínua. Para variáveis do tipo categórica, técnicas especiais do tipo de Aprendizagem de Máquina (AM), comumente referenciadas na literatura como *Machine Learning*, podem ser utilizadas para um propósito diferente, pois em vez da otimização, pretende-se a classificação e previsão de um conjunto de dados (soluções) previamente definidas.

Dessa maneira o presente trabalho possui um objetivo primário e secundário. Primariamente, remete a aplicação de 12 técnicas de AM, para o treinamento (com parte do espaço solução) e posterior previsão das soluções desejadas. Comparativamente, a AM se assemelha aos métodos de otimização derivados da meta-modelagem [Kang e Lobo 2016], constituindo primeiramente em uma ferramenta de auxílio a tomada de decisão com avaliação rápida de soluções. Secundariamente, pela análise dos resultados da previsão é proposta a avaliação da possibilidade de integração dos métodos de AM com técnicas de otimização.

A seleção dessas técnicas para o presente trabalho teve por motivação o crescente uso das metodologias de mineração de dados na era da indústria 4.0, que com sua automatização, possibilita a geração e armazenagem de uma quantidade de bancos de dados a serem exploradas , gerando informação útil para os agentes tomadores de decisão [Xu *et al.* 2016].

Nesse artigo foi utilizado o ambiente de programação Python 3.6.4 fornecido pela *Python Software Foundation*, juntamente com os pacotes NumPy, SimPy, SciLearn-kit, Pandas, SciPy e MatPlotLib, por se tratar de um ferramental utilizado na mineração de dados, de fácil acesso e gratuito, de forma que com o algum conhecimento da linguagem e pacotes utilizados é possível a reprodução do presente artigo e adaptação para situações similares.

A estrutura do trabalho segue como: seção 2, fundamentação teórica, seção 3, metodologia, seção 4, resultados e discussões, seção 5, conclusão, seção 6, agradecimentos e seção 7, referências.

2. Fundamentação teórica

2.1. Simulação de eventos discretos e otimização

O termo "simulação", utilizado no presente trabalho, refere à coleção de técnicas que imitam um comportamento específico de um sistema real ou ideal, que possui pelo menos uma variável estocástica e identicamente e independentemente distribuídas (IID), aplicando recursos (tempo e conhecimento) para responder questões feitas para a estrutura estudada, quando experimentos reais são muito caros ou impossíveis de serem realizados. Isso pode ser utilizado em uma variedade de áreas de estudo, industrias e aplicações, que principalmente consiste na coleta de dados e análise auxiliada pelo intermédio de computadores [Law e Kelton 1991; Banks *et al.* 2010; Kelton, Sadowski e Swets 2010]. Os questionamentos realizados são comumente relacionados à otimização de características específicas que representam os cenários do sistema proposto.

Nesse sentido, otimização pode ser definida como a minimização, maximização ou ambos relacionados a uma função mono ou multi-objetivo, utilizando da forma matemática, resume os questionamentos propostos para o sistema. Sendo assim, diferentes combinações de alternativas são consideradas viáveis se satisfazem todas as restrições do problema, ou inviável se pelo menos uma restrição não é satisfeita. A alternativa que possui o melhor valor para a função objetivo, sendo viável, é considerada ótima. Se a simulação possui dados suficientes para representar o sistema analisado, a melhor solução simulada pode ser inferida como boa, a ponto de ser implementada no sistema real, cumprindo a sua função de ferramenta para auxílio à tomada de decisão [Taha 2007].

Portanto, as técnicas de otimização se fazem necessárias quando se deseja encontrar uma combinação de variáveis que minimizam ou maximizam uma função objetivo. Fato este que nem sempre se faz necessário, quando por exemplo um agente tomador de decisão quer apenas avaliar qual o valor esperado de uma série de soluções específicas. Nesse caso, faz-se possível o uso de técnicas de classificação e previsão empregadas em AM.

2.2. Aprendizagem de máquina

De acordo com [Kubat 2017], as técnicas de AM surgiram como um anseio e tentativa dos pesquisadores da área da computação, desde a criação dessa ciência, para se conseguir que uma máquina seja capaz de realizar a tarefa de "aprender". Da maneira como conhecemos esses estudos na atualidade, seus fundamentos podem ser primeiramente creditadas as décadas de 80 e 90, a alguns trabalhos seminais como [Michalski 1969; Mitchell 1982; Carbonell, Michalski e Mitchell 1983].

Para [Buchberger 1983], os métodos de AM tem por fundamentação inicial a classificação de um conjunto de dados, para que inferências sejam geradas. Em [Kubat 2017] é proposto que independentemente do método de AM empregado, do conjunto inicial de dados das quais de dispõe, o mesmo deve ser dividido em duas partes, para que a primeira parte sirva para o treinamento do algoritmo e a segunda parte para o teste de previsão do algoritmo, para posteriormente calcular o erro entre o real e o previsto. O conjunto inicial de dados (amostra) é escolhido aleatoriamente a partir do espaço solução, e para um novo conjunto de dados, o processo é repetido para se determinar qual o melhor algoritmo de AM.

Para melhor identificar os métodos de AM que poderiam ser implementados, foram consultadas as bases de dados *Web of Science* e *Scopus*, utilizando-se as combinações dos termos: *machine learning*, *discrete event simulation*, *industry* e *manufacturing*, sendo os resultados apresentados de acordo com a relevância com os termos pesquisados. Como trabalhos correlatos a essa temática pode-se citar: predição do ciclo de produção [Can e Heavey 2016] e definição de ambiente flexível para transporte intra-logístico [Scholz *et al.* 2017] e determinação do melhor plano de produção [Bergmann, Feldkamp e Strassburger 2017]. Com trabalho de temática

intermediária, foi encontrado [García Nieto *et al.* 2018], sobre a previsão de estrutura do metal fundido.

No que tange aos problemas relacionados ao chão de fábrica que foram auxiliados pelas técnicas de AM, pode-se citar: confiabilidade de componentes industriais [Alsina *et al.* 2017], manutenção [Lin *et al.* 2017], desenvolvimento de software [Pospieszny, Czarnacka-Chrobot e Kobylinski 2018], desenvolvimento de produto [Khanzadeh *et al.* 2017], financeiro, varejo, manufatura e gestão industrial [Li *et al.* 2017].

A diminuta bibliografia encontrada relacionando problemas da engenharia de produção aplicando MSED e AM, corrobora com a hipótese de que a utilização dessas metodologias é uma proposta de trabalho que ainda pode ser explorada por trabalhos futuros.

3. Metodologia

3.1. Determinação dos métodos utilizados

O presente trabalho pode ser considerado empírico pois não foram encontrados trabalhos com o mesmo propósito, para gerar por base uma metodologia para determinar quais os melhores métodos a serem empregados, escolhendo assim os melhores para serem comparados. Dessa forma, 12 métodos foram selecionados para se determinar qual o que melhor se ajusta ao problema proposto.

Consultando artigos que empregaram aprendizagem por máquina foram selecionados para avaliação os métodos presentes na Tabela 1, assim com qualquer parâmetro utilizado, quando explicitamente especificado e necessário para a utilização do método, pois nem todos necessitavam de algum ajuste prévio.

Tabela 1: Métodos, parâmetros utilizados e artigo consultado

Método utilizado com sucesso	**Parâmetros utilizados**	**Artigo onde o método foi empregado**
Polynomial Regression (PR)	('poly', PolynomialFeatures (degree=3)), ('linear', LinearRegression (fit_intercept=False))	[Cohen, Nahum-Shani e Doveh 2010]
Ridge Regression (RR)	alpha = 0.5	[Puggini e McLoone 2015]
Linear Discriminant Analysis (LDA)	solver="svd", store_covariance=True	[Ferreira e Warzecha 2017]
Kernel Ridge Regression (KRR)	alpha=1.0	[Morin *et al.* 2016]
Support Vector Machines (SVM)	----------	[Rostami, Dantan e Homri 2015]
Stochastic Gradient Descent (SGD)	loss="hinge", penalty="l1"	[Jia e Lam 2010]
Nearest Neighbors Classification (NNC)	n_neighbors=15, weights='uniform'	[Li *et al.* 2016]
Gaussian Naive Bayes (GNB)	----------	[Xie *et al.* 2018]
Random Forest Regressor (RFR)	max_depth=2, random_state=0	[Shevchik *et al.* 2017]
Ada Boost Classifier (ABC)	n_estimators=100	[P.Rajendiran, K.R.Sekar 2017]
One-Vs-One Classifier (OVOC)	LinearSVC(random_state=0)	[Abdul Raziff *et al.* 2017]
Multi-layer Perceptron Regressor (MLPR)	solver='lbfgs', alpha=1e-5,hidden_layer_sizes=(15,), random_state=1	[Naderi, Khalili e Tavakkoli-Moghaddam 2009; Can e Heavey 2012; Li *et al.* 2017]

Na Tabela 1 são apresentados os 12 métodos implementados, que geraram resultados de acordo com o esperado para problema simulado (índice Max, máximo erro gerado, menor que 200). Para todos os métodos que necessitam de parâmetros de ajuste, foram considerados os valores padrões dos mesmos, de acordo com os códigos fonte apresentados no site do pacote SciKit-learn. A não alteração destes parâmetros durante os testes se deve pelo fato de não ser objetivo do trabalho a otimização dos métodos, ou melhor ajuste destes a essa massa de dados. A escolha desses métodos se deu pela proposta de amostragem de pelo menos um método de cada classe de métodos nomeados a seguir pela ordem apresentada na Tabela 1: *Generalized Linear Models, Linear and Quadratic Discriminant Analysis, Kernel Ridge Regression, Support Vector Machines, Stochastic Gradient Descent, Nearest Neighbors, Naive Bayes, Ensemble Methods, Multiclass and Multilabel Algorithms* e *Neural Network Models*.

Além dos métodos apresentados na Tabela 1, foram testados os seguintes métodos que geraram resultado nenhum (erro) ou ruim (índice Max maior que 200) para o problema tratado: *Least Angle Regression* (LARS), *LARS Lasso, Cross Decomposition, Gradient Tree Boosting Classifier, Multioutput Regression* e *Multioutput Classifier*. Devido aos seus desempenhos insuficientes, esses resultados não serão mostrados.

3.2. Critérios de avaliação dos métodos usados

Os 12 métodos empregados, presentes na Tabela 1, foram avaliados por 10 parâmetros, considerando os seguintes fatores para suas análises:

1) IC (95%): Refere ao intervalo com 95% de confiança para a média das diferenças entre o resultado real da simulação e o resultado da previsão. Assim quanto mais perto o intervalo de confiança estiver do número "0", melhor será a previsão dos dados, pois geraram um menor erro na previsão, preferencialmente sem tendência para valores positivos ou negativos.

2) *K-fold* com 3 níveis: Nos casos em que os dados são IID, assumindo que foram gerados de um processo que não possui memória das amostras geradas no passado, fato este necessário para se trabalhar com a MSED, pode-se utilizar do método *K-fold* de validação cruzada. Este consiste na divisão da amostra em "k" grupos de tamanho igual, onde a aprendizagem é feita com os "k-1" grupos e a previsão com o grupo restante. Quanto mais perto de "1", melhor é a taxa de acerto do método, com valor previsto igual ao real.

3 e 4) Min e Max: Fazem referências aos valores mínimo e máximo do vetor de diferença (erro) entre o resultado real e o previsto. O ideal é que esses sejam em valores absolutos um próximo do outro para não haver tendência para mais ou para menos e com índices próximos de "0" para gerar o menor erro possível.

5) *Mean Absolute Error* (MAE): Como o MAE é a média do somatório dos valores absolutos da diferença entre o real e o previsto, é o valor médio esperado para o erro na resposta da previsão, portanto quanto mais perto do valor "0" melhor é o índice.

6) *Mean Squared Error* (MSE). Diferentemente do MAE, o MSE tira a média dos valores quadráticos, o que dá maior peso para erros com maior valor discrepante. Assim como o MAE, um valor de MSE próximo de "0" é desejável.

7) Coeficiente de Determinação (R^2): Mede a qualidade dos dados gerados pelo modelo de previsão, de maneira tal que significa a quantidade de dados que pode ser explicada pelo método utilizado. Quanto mais próximo do valor "1", significa um ajuste a 100% dos dados reais comparados com o previsto.

8) Tempo de treino: É a quantidade de tempo que o algoritmo necessitou para treinar com a massa de dados, no *software* e *hardware* especificados, em segundos.

9) Tempo para a previsão: É o tempo necessário para o algoritmo treinar com a massa de dados, no *software* e *hardware* especificados, em segundos.

10) Acertos: É o percentual de valores previsto comparado aos reais, de maneira tal que o método de AM acertou no sinal do valor da função objetivo. Esse índice é importante para auxílio na determinação da possibilidade de aplicação da aprendizagem de máquina em parceria com métodos de otimização. Esse item será melhor discutido na conclusão do trabalho.

3.3. Definição do estudo de caso e dados de entrada

O objeto analisado foi o problema M/M/k adaptado de Gandhi *et al.* (2010) que originalmente trata da utilização de *data centers*. O problema tratado consiste na simulação de um processo produtivo genérico onde "k" máquinas produzem em paralelo, com fornecimento de matéria prima puxada, e probabilidade de quebra, demorando um tempo para ser reparado pelo técnico de manutenção durante um determinado período de tempo, dados estes apresentados na Tabela 2.

Tabela 2: Distribuições de probabilidade associadas ao problema

Atividade	**Distribuição de probabilidade (min)**
Tempo de processamento por peça	Normal (média=10, desvio padrão=2)
Tempo entre as quebras	Exponencial (lambda =1/300)
Tempo de reparo da máquina	Constante (30)

A função objetivo desse problema visa a maximização dos lucros desse sistema, dados pela relação apresentada na Equação 1.

$$\begin{aligned} Lucro = {} & (Total\ de\ peças \times \$100) \\ & - (Técnico\ de\ manutanção \times \$20 \times \text{Tempo de operação}) \\ & - (\text{Tempo de operação} \times 5) \end{aligned} \quad (1)$$

Na Equação 1 estão relacionados o ganho de $100 com a produção e venda de cada item e os custos de contratação de $20 por hora do técnico de manutenção e $5 por hora de funcionamento da instalação industrial. As restrições do problema estão na Tabela 3.

Tabela 3: Valores aceitáveis para as variáveis independentes.

Variável do Problema	**Valores aceitáveis**
Técnico de manutenção (pessoa)	2, 4, 6, ..., 20
Máquinas (unidade)	2, 4, 6, ..., 40
Tempo de operação (semanas)	30, 32, 34, ..., 150

A partir dos dados da Tabela 3 é possível a geração de um espaço solução com 12.200 possíveis respostas para o problema. Para 2000 cenários avaliados foram gastos 6,94 horas em uma máquina i7-7700, 16GB de RAM e Windows 10 x64. Fato este que gera um alto custo computacional para a avaliação de novas avaliações via SED.

Para o presente problema, foram utilizados os seguintes passos para a geração, tratamentos e análise dos dados do problema proposto:

1) Determinação do tamanho da amostra dos dados. Das possíveis 12,2K soluções para o problema, foram selecionadas 2.000 soluções, 16,4% do total que gera uma representatividade do espaço solução do problema, para o treinamento, previsão e avaliação dos melhores métodos de AM. Para a escolha dos 2K dados de entrada, foram criados dois vetores com as 12,2K posições e embaralhado utilizando-se da geração de posições aleatórias até que todos os dados do vetor original estivessem no vetor embaralhado. Assim as primeiras 2K posições do vetor embaralhado foram escolhidas, o que garante uma amostra sem repetição, e com uma aleatoriedade tal que possibilitasse a representatividade da variabilidade do espaço solução do problema.

2) Determinação da marcação da amostra de dados. A partir da amostra gerada, a mesma foi passada como parâmetro para ser simulada em pacote de simulação do Python e seus dados armazenados em arquivos de dados CSV, fazendo parte assim das marcações (resultados) dos dados gerados. Dessa maneira foram considerados dados a quantidade de máquinas, a quantidade de técnicos de manutenção e o tempo da simulação, e a função objetivo como marcação (resultado).

3) Tratamento dos Dados. Para serem aplicados os métodos de AM, os dados de marcação devem estar compreendidos em classes, fato este que em geral não é resultado comum para problemas envolvendo simulação a eventos discretos. Foi averiguado que o maior e menor valores dos dados foram respectivamente $-463.883.600,00 e $399.473.800,00. Pelo fato da SED originalmente trabalhar com variáveis contínuas, ou discretas, gerando um grande número de classes, torna o cálculo da frequência relativa impraticável, foi adotado o método de discretização das variáveis proposto por [Kubat 2017, pg 30]. A partir da ideia inicial do referido autor, foi proposto e utilizado uma escala, de maneira tal que quão mais próximo do valor definido na escala, a variável receberia esse determinado valor. Com uma variação dessa magnitude, foi aplicada uma escala com 300 níveis, assim cada nível compreende valores com uma diferença de $2.877.858, dessa forma para um projeto ser considerado diferente do outro, deve haver uma diferença de cerca de $3KK.

4) Treino e testes. Com os 2K dados e marcações (*data frame* do problema), foi considerado uma taxa de 80% dos dados para treino (1600) e o restante (400) para as previsões e teste de aderência do ajuste. Foram considerados para comparação entre os testes: o intervalo de confiança para a diferença entre o real e o previsto, os valores de pontuação gerados pelo método *K-fold* com 3 níveis, os valores mínimo e máximo para as diferenças encontradas, *Mean Absolute Error*, *Mean Squared Error*, tempo necessário para o treino e para a previsão e percentagem de acertos. Para a ordenação dos melhores métodos foi considerado o Min, Max e IC (95%), nessa ordem, pois é desejado para esse problema que a previsão tenha uma probabilidade, caso o erro seja diferente de zero, com o menor valor possível, sendo preferencialmente zero ou próximo desse valor.

4. Resultados e discussão

São apresentados na Tabela 4 e Figuras 1 e 2 os resultados da aplicação dos 12 métodos apresentados na Tabela 1, respeitando os passos apresentados no item 3.3, e apresentados em ordem decrescente considerando a melhor solução para o problema proposto. Os dados da Tabela 1 estão ordenados considerando a pontuação referente aos índices Min, Max e IC (95%), assim como as condições explanadas nos itens 3.2 e 3.3.

Tabela 4: Resultados para o problema M/M/k, com 2K dados e 300 níveis

Método	IC (95%)	*K-fold*	Min	Max	MAE	MSE	R^2	Treino (seg.)	Prev. (seg.)	Acertos (%)
PR	(-0,29 ; 0,80)	0,99	-15,2	19,8	2,5	15,2	0,99	0,00	0,00	98,5
MLPR	(-0,47 ; 1,86)	0,97	-19,1	40,5	5,9	69,6	0,96	0,12	0,00	98,0
GNB	(-1,79 ; 2,23)	0,02	-40,0	42,0	11,0	206,9	0,90	0,01	0,01	82,5
LDA	(-2,20 ; 2,03)	0,05	-52,0	57,0	11,3	229,9	0,88	0,04	0,00	89,5
RR	(-2,58 ; 2,68)	0,84	-57,2	55,3	13,1	353,4	0,82	0,03	0,00	98,5
KRR	(-3,05 ; 2,23)	0,84	-58,5	56,5	13,1	358,1	0,82	0,17	0,00	98,5
OVOC	(-0,37 ; 6,76)	0,05	-73,0	105,0	17,7	661,4	0,67	22,01	1,36	96,0
RFR	(-3,19 ; 3,56)	0,70	-75,9	90,0	17,9	581,8	0,71	0,01	0,00	85,0
NNC	(4,79 ; 10,14)	0,03	-82,0	57,0	15,2	422,8	0,79	0,00	0,01	85,5
SVM	(-23,92 ; -11,40)	0,04	-158,0	114,0	35,9	2316,2	-0,17	0,57	0,15	50,0
ABC	(-3,82 ; 6,91)	0,02	-118,0	135,0	28,5	1475,8	0,26	0,98	0,05	53,0
SGD	(12,08 ; 23,39)	0,02	-102,0	124,0	29,4	1439,4	0,27	0,13	0,00	66,0

De acordo com a Tabela 4, podemos analisar que os cinco primeiros métodos que melhor se ajustaram foram: *Polinomial Regression* (PR), *Multi-layer Perceptron Regressor* (MLPR), *Gaussian Naive Bayes* (GNB), *Linear Discriminant Analysis* (LDA) e *Ridge Regression* (RR).

Analisando os resultados, podemos averiguar que o maior erro possível negativo (Min) para o primeiro método (PR) foi de 15,2 níveis, o que significa que esse método errou em algum momento uma previsão em cerca de $-43.743.441,60 comparado com o seu valor real e para mais (índice Max) em cerca de 19,8 níveis o que representa um erro para mais de $55.254.873,60. Esses valores representam outliers comparados com o IC (95%) de (-0,29 ; 0,80), significando que na

média o método errou 0,255 ±0,545, o que pode ser considerado um erro baixo pois na maioria das vezes a previsão da classe foi assertiva com o valor real. Portanto analisando o melhor método apresentado, existe uma probabilidade de 95% do erro da previsão ser menor do que um nível em comparação com o valor real. Este fato é confirmado pelo índice *K-fold* com 3 níveis, que apresenta o valor de 0,99 para o acerto das K-1 previsões, que mostra um excelente ajuste. Foi testado também a média *K-fold* com 10 níveis, porém seus resultados não foram apresentados pois a diferença em comparação com os valores para 3 níveis foram inferiores a 0,01 para a maioria dos métodos, corroborando com a ideia da presença de poucos *outliers* gerados pelos métodos.

Os índices MAE e MSE apresentam a média absoluta e quadrática, contribuindo para a análise do erro médio e da presença e influência de valores discrepantes que podem influenciar os resultados. Assim como o IC (95%) e *K-fold*, o MAE e MSE, com seus valores baixos, próximos de zero, da média dos erros ser baixa e apresentar poucos *outliers* dentre os resultados dos erros de previsão. O R^2 demonstra a quantidade de dados gerados que podem ser explicados pelo método específico, que no caso da PR, representa 99% dos dados. No caso dos métodos GNB e LDA os valores do *K-fold* foram baixos (0,02 e 0,05) porém com bons índices para o IC (95%) e MAE e MSE, fato este que sustenta a ideia de que fizeram previsões muito próximas do real, mas na maioria não foram assertivas em relação à classe.

Para o tempo de treino foi considerado o tempo necessário de aprendizado dos métodos, sem considerar os processos de carregamento do algoritmo ou da massa de dados, assim como o tempo computado para a previsão, que não considera o tempo de aprendizado. Dessa forma, tanto o tempo necessário para aprendizado quanto previsão foram menores que o tempo médio de 24.984 segundos necessários para gerar cada um dos 2K cenários simulados, fato este relevante para a implementação de métodos de otimização, que visam a diminuição do tempo computacional para se encontrar boas respostas para um problema. O último índice “Acertos” avaliou que no caso do método PR houveram cerca de 98,5% de acertos na comparação se tanto o teste quanto a previsão haviam gerado valores da função objetivo com o mesmo sinal, de maneira tal que avalia a possibilidade de o método ter boas chances de indicar se a resposta gera lucro ou prejuízo. A maioria dos valores discrepantes foram gerados com as previsões próximas do ponto de lucro zero.

Na Figura 1 são apresentados os *boxplots* das previsões comparadas com o vetor de dados reais.

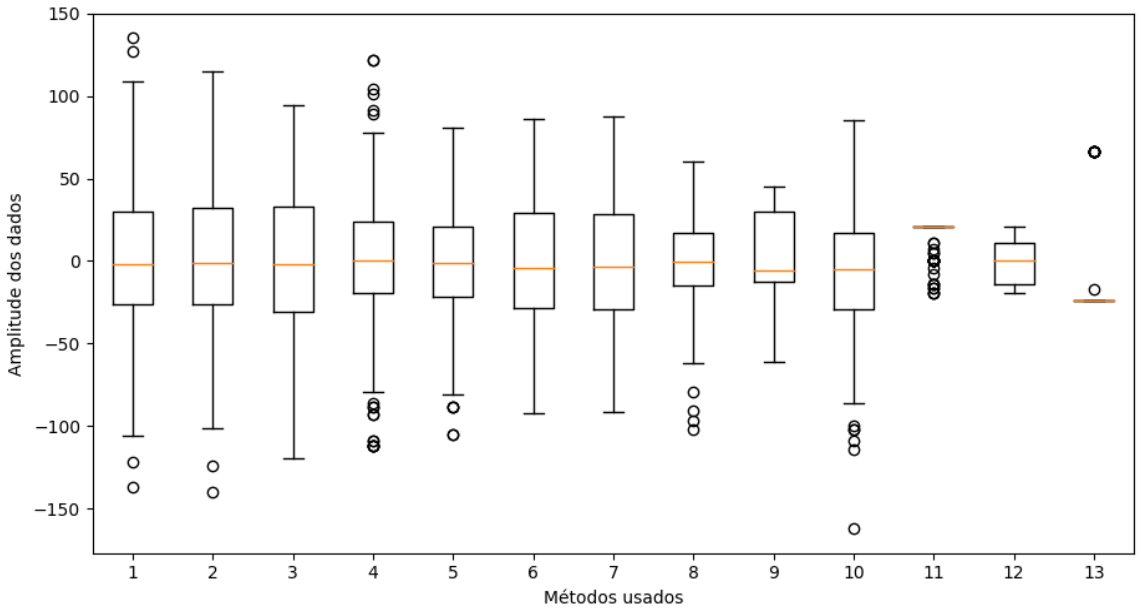


Figura 1: *Boxplot* dos dados reais vs. os 12 métodos

Na Figura 1, o *boxplot* número 1 remete aos dados de teste reais do problema, as quais deverão ser comparados com os 12 métodos de previsão. A ordem dos *boxplots* 2 a 13 segue a mesma apresentada na Tabela 4, dessa forma o "2" remete ou método PR (melhor método) e o "13" ao SGD (pior método).

Analisando a Figura 1 é possível perceber uma variabilidade dos dados muito próxima entre os dados reais em 1 e os dados da previsão do método PR em 2. A medida que os métodos vão piorando em qualidade de resposta, as variabilidades das previsões diminuem. Na Figura 2 são apresentados os *boxplots* dos erros das previsões.

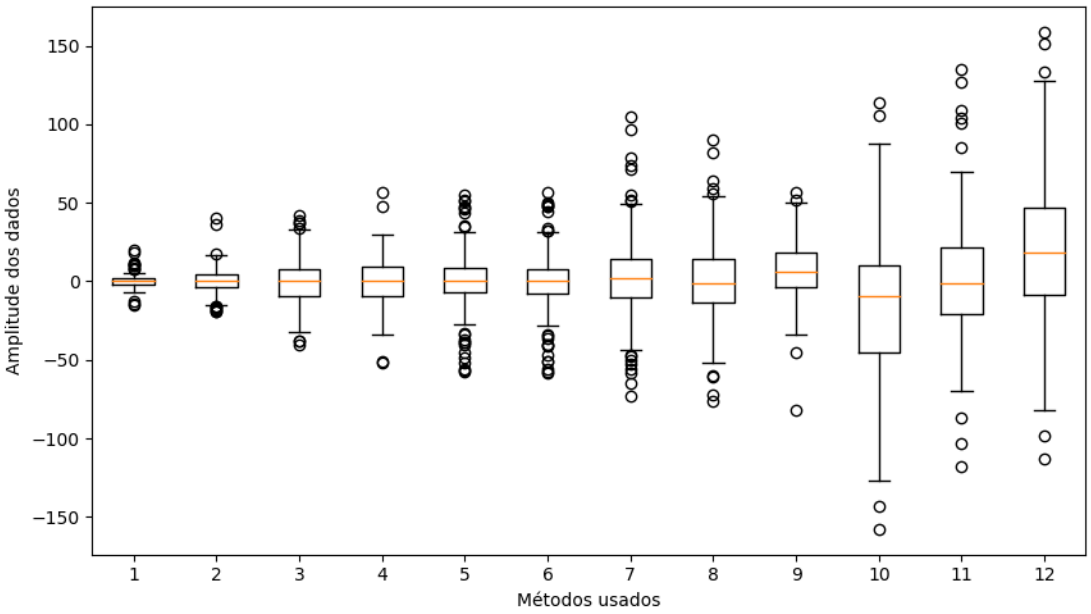


Figura 2: *Boxplots* das diferenças entre o real e o simulado para os 12 métodos

Na Figura 2, os dados são apresentados assim como determinado na Tabela 4. Analisando a Figura 2 é possível observar que quão melhor o método é para a previsão dos dados do problema, menor é a amplitude dos dados dos erros, gerando o menor erro acumulado comparado aos outros métodos, fato este que dá suporte aos dados de IC, MAE e MSE apresentados na Tabela 4.

5. Conclusão

Dentre os 12 métodos empregados, considerando os 4 primeiros (PR, MLPR, GNB e LDA), suas respostas de previsão foram consideradas boas por se aproximarem dos valores reais, mesmo muitas vezes não sendo iguais aos valores reais. Em um processo de tomada de decisão em que o tempo para se avaliar vários cenários deve ser considerado para a escolha do melhor método de auxilio no processo, a junção de técnicas de AM com SED se mostrou uma ferramenta com grande potencial a ser ainda explorado para problemas de origem industrial.

Nesse mesmo sentido, considerando os tempos de aprendizado e de treino para cada instância, que de acordo com a Tabela 4, foram em geral menores que 1 segundo, se configura como uma característica desejável de método a ser implementado como ferramenta de auxílio em otimização de MSED. Essa proposta advém das respostas geradas pelo último item de avaliação dos métodos e apresentado na Tabela 4, a percentagem de acertos, que remete ao fato da previsão ser do mesmo sinal que o real, ou seja, se tanto o previsto quanto o real apontam para uma mesma solução de lucro ou prejuízo. Este fato ajuda na pré-avaliação de uma possível solução antes que a mesma seja simulada, reduzindo assim a necessidade da simulação de uma solução que se caracteriza como muito longe da função objetivo desejada, ou de pouco proveito para a metodologia de otimização.

Os resultados aqui apresentados foram testados para apenas uma base de dados gerada. Cabe a estudos futuros o teste como mais de uma instância de dados para averiguar se os melhores métodos aqui apresentados irão se comportar da mesma maneira para problemas correlatos. Dessa forma os resultados aqui apresentados só podem ser associados com os parâmetros aqui utilizados, por não se saber como os métodos de AM se comportaram durante o processo de otimização de MSED, nas quais dados são gerados na medida em que o método é chamado recursivamente, possibilitando ser que no início da otimização com poucos dados, possa ocorrer alguma inversão de ordem de melhor método a ser empregado.

Como não é objetivo desse trabalho a otimização dos parâmetros de nenhum método em específico, para melhor serem ajustados ao problema apresentado, tanto os métodos apresentados quanto os descartados para o presente trabalho podem ser melhor estudados para trabalhos futuros.

6. Agradecimentos

Os autores agradecem o apoio da FAPEMIG, CNPq e CAPES.

7. Referências

Abdul Raziff, A. R. *et al.* (2017) 'Gait identification using smartphone handheld placement with linear interpolation factor, single magnitude and one-vs-one classifier mapping', *International Journal of Intelligent Engineering and Systems*, 10(4), pp. 70–80.

Akiya, N., Scott Bury and Wassick, J. M. (2011) 'Proceedings of the 2011 Winter Simulation Conference S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.', *Proceedings of the 2011 Winter Simulation Conference, 11-14 Dec*, pp. 2194–2205.

Alexopoulos, C. *et al.* (2001) 'Proceedings of the 2001 Winter Simulation Conference B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer, eds.', *Simulation*, pp. 680–683.

Alsina, E. F. *et al.* (2017) 'On the use of machine learning methods to predict component reliability from data-driven industrial case studies', *International Journal of Advanced Manufacturing Technology*, pp. 1–15.

Altiparmak, F., Dengiz, B. and Bulgak, A. A. (2007) 'Buffer allocation and performance modeling in asynchronous assembly system operations: An artificial neural network metamodeling approach', *Applied Soft Computing Journal*, 7(3), pp. 946–956.

Banks, J. *et al.* (2010) *Discrete-Event System Simulation*. 5th edn, *PrenticeHall international series in industrial and systems engineering*. 5th edn. Pearson.

Bergmann, S., Feldkamp, N. and Strassburger, S. (2017) 'Emulation of control strategies through machine learning in manufacturing simulations', *Journal of Simulation*. Palgrave Macmillan UK, 11(1), pp. 38–50.

Buchberger, B. (1983) *Machine Learning, J. Symbolic Computation*. Edited by R. S. Michalski, J. G. Carbonell, and T. M. Mitchell. Berlin, Heidelberg: Springer Berlin Heidelberg.

Can, B. and Heavey, C. (2012) 'A comparison of genetic programming and artificial neural networks in metamodeling of discrete-event simulation models', *Computers and Operations Research*. Elsevier, 39(2), pp. 424–436.

Can, B. and Heavey, C. (2016) 'A demonstration of machine learning for explicit functions for cycle time prediction using MES data', in *2016 Winter Simulation Conference (WSC)*. IEEE, pp. 2500–2511.

Carbonell, J. G., Michalski, R. S. and Mitchell, T. M. (1983) 'Machine Learning: A Historical and Methodological Analysis', *AI Magazine*, 4(3), p. 69.

Cohen, A., Nahum-Shani, I. and Doveh, E. (2010) 'Further Insight and Additional Inference Methods for Polynomial Regression Applied to the Analysis of Congruence', *Multivariate Behavioral Research*, 45(5), pp. 828–852.

Ferreira, J. G. and Warzecha, A. (2017) 'An application of machine learning approach to fault detection of a synchronous machine', *2017 International Symposium on Electrical Machines (SME)*, pp. 1–6.

Fonseca, D. J., Navaresse, D. O. and Moynihan, G. P. (2003) 'Simulation metamodeling through

artificial neural networks', *Engineering Applications of Artificial Intelligence*, 16(3), pp. 177–183.
Gandhi, A., Doroudi, S., Harchol-Balter, M. *et al*. (2014) 'Exact analysis of the M/M/k/setup class of Markov chains via recursive renewal reward', Queueing Systems, 77(2), pp.177-209. https://doi.org/10.1007/s11134-014-9409-7.
Gansterer, M., Almeder, C. and Hartl, R. F. (2014) 'Simulation-based optimization methods for setting production planning parameters', *International Journal of Production Economics*. Elsevier, 151, pp. 206–213.
García Nieto, P. J. *et al.* (2018) 'A comparison of several machine learning techniques for the centerline segregation prediction in continuous cast steel slabs and evaluation of its performance', *Journal of Computational and Applied Mathematics*. Elsevier B.V., 330, pp. 877–895.
He, J., Huang, Y. and Chang, D. (2015) 'Simulation-based heuristic method for container supply chain network optimization', *Advanced Engineering Informatics*. Elsevier Ltd, 29(3), pp. 339–354.
Jia, N. and Lam, E. Y. (2010) 'Machine learning for inverse lithography: Using stochastic gradient descent for robust photomask synthesis', *Journal of Optics A: Pure and Applied Optics*, 12(4).
Kamrani, F., Ayani, R. and Moradi, F. (2012) 'A framework for simulation-based optimization of business process models', *Simulation*, 88(7), pp. 852–869.
Kang, H. and Lobo, J. M. (2016) 'Identifying the optimal configuration of an express care area in an Emergency Department: A des and metamodeling approach', in *2016 Winter Simulation Conference (WSC)*. IEEE, pp. 1961–1969.
Kelton, W. D., Sadowski, R. P. and Swets, N. B. (2010) *Simulation with Arena*. 5th edn. McGraw-Hill.
Khanzadeh, M. *et al.* (2017) 'Quantifying Geometric Accuracy With Unsupervised Machine Learning: Using Self-Organizing Map on Fused Filament Fabrication Additive Manufacturing Parts', *Journal of Manufacturing Science and Engineering*, 140(3), p. 31011.
Kubat, M. (2017) *An Introduction to Machine Learning*. Second. Cham: Springer International Publishing.
Law, A. M. and Kelton, W. D. (1991) *Simulation Modeling and Analysis*. 2nd edn. New York: McGraw-Hill.
Li, N. *et al.* (2016) 'Human performance modeling for manufacturing based on an improved KNN algorithm', *International Journal of Advanced Manufacturing Technology*, 84(1–4), pp. 473–483.
Li, Y. *et al.* (2017) 'On neural networks and learning systems for business computing', *Neurocomputing*. Elsevier B.V.
Lin, C.-C. *et al.* (2017) 'A MapReduce-Based Ensemble Learning Method with Multiple Classifier Types and Diversity for Condition-Based Maintenance with Concept Drifts', *IEEE Cloud Computing*, 4(6), pp. 38–48.
Lopes, H. dos S. *et al.* (2017) 'Scenario analysis of Brazilian soybean exports via discrete event simulation applied to soybean transportation: The case of Mato Grosso State', *Research in Transportation Business and Management*. Elsevier, 25(October), pp. 66–75.
Lucidi, S. *et al.* (2016) 'A Simulation-Based Multiobjective Optimization Approach for Health Care Service Management', *IEEE Transactions on Automation Science and Engineering*, 13(4), pp. 1480–1491.
Martins, M. S. R. *et al.* (2013) 'PSO with path relinking for resource allocation using simulation optimization', *Computers & Industrial Engineering*. Elsevier Ltd, 65(2), pp. 322–330.
Mehrsai, A. *et al.* (2013) 'Using metaheuristic and fuzzy system for the optimization of material pull in a push-pull flow logistics network', *Mathematical Problems in Engineering*, 2013.
Michalski, R. S. (1969) 'On the Quasi-Minimal Solution of the General Covering Problem', *Proceedings of the V International Symposium on Information Processing (FCIP 69)*, pp. 125–128.
Miranda, R. D. C., Montevechi, J. A. B. and Pinho, A. F. (2015) 'Development of an adaptive

https://proceedings.science/p/85540?lang=pt-br

genetic algorithm for simulation optimization', *Acta Scientiarum. Technology*, 37(3), p. 321.
Mitchell, T. M. (1982) 'Generalization as search', *Artificial Intelligence*, 18(2), pp. 203–226.
Morin, M. *et al.* (2016) 'Machine learning-based metamodels for sawing simulation', in *Proceedings - Winter Simulation Conference*, pp. 2160–2171.
Muta, H. *et al.* (2015) 'A multi-objective genetic algorithm using intermediate features of simulations', *Proceedings - Winter Simulation Conference*, 2015–Janua, pp. 793–804.
Naderi, B., Khalili, M. and Tavakkoli-Moghaddam, R. (2009) 'A hybrid artificial immune algorithm for a realistic variant of job shops to minimize the total completion time', *Computers and Industrial Engineering*. Elsevier Ltd, 56(4), pp. 1494–1501.
Nageshwaraniyer, S. S., Son, Y.-J. and Dessureault, S. (2013) 'Simulation-based optimal planning for material handling networks in mining', *SIMULATION*, 89(3), pp. 330–345.
P.Rajendiran, K.R.Sekar, B. S. & R. M. (2017) 'Customer Relationship Management in the Manufacturing Industry using Data Mining Techniques', *International Journal of Educational Science and Research (IJESR)*, 7(6), pp. 63–70.
Pospieszny, P., Czarnacka-Chrobot, B. and Kobylinski, A. (2018) 'An effective approach for software project effort and duration estimation with machine learning algorithms', *Journal of Systems and Software*, 137, pp. 184–196.
Puggini, L. and McLoone, S. (2015) 'Extreme learning machines for virtual metrology and etch rate prediction', *Irish Signals and Systems Conference (ISSC)*, 2(2), pp. 1–6.
Rostami, H., Dantan, J.-Y. and Homri, L. (2015) 'Review of data mining applications for quality assessment in manufacturing industry: support vector machines', *International Journal of Metrology and Quality Engineering*, 6(4), p. 401.
Saadouli, H. *et al.* (2015) 'A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department', *Computers & Industrial Engineering*. Elsevier Ltd, 80, pp. 72–79.
Salam, M. A. and Khan, S. A. (2016) 'Simulation based decision support system for optimization A case of Thai logistics service provider', *Industrial Management & Data Systems*, 116(2), pp. 236–254.
Scholz, M. *et al.* (2017) 'Software-in-the-loop testbed for multi-agent-systems in a discrete event simulation: Integration of the Java Agent Development Framework into Plant Simulation', *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*.
Shevchik, S. A. *et al.* (2017) 'Prediction of failure in lubricated surfaces using acoustic time-frequency features and random forest algorithm', *IEEE Transactions on Industrial Informatics*, 13(4), pp. 1541–1553.
Taha, H. A. (2007) *Operations Research: An Introduction*. 8th edn. Pearson.
Vitanov, I. V., Vitanov, V. I. and Harrison, D. K. (2009) 'Buffer capacity allocation using ant colony optimisation algorithm', *Proceedings of the 2009 Winter Simulation Conference*, (2003), pp. 3158–3168.
Xie, Y. *et al.* (2018) 'Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances', *Journal of Petroleum Science and Engineering*. Elsevier B.V., 160, pp. 182–193.
Xu, J. *et al.* (2016) 'Simulation optimization in the era of Industrial 4.0 and the Industrial Internet', *Journal of Simulation*. Palgrave Macmillan UK, 10(4), pp. 310–320.

https://proceedings.science/p/85540?lang=pt-br