PREVISÃO DO DESLOCAMENTO DE TEMPESTADES SEVERAS: ABORDAGENS POR APRENDIZADO DE MÁQUINA

Nicole Amanda Rozin

Paulo Henrique Siqueira

nicole.rozin.simepar@gmail.com

paulohs@gmail.com

Universidade Federal do Pananá

Cesar Augustus Assis Beneti

Jorge Vinícius Ruviaro Bonato

cesar.beneti@simepar.br

jorge.ruviaro.simepar@gmail.com

Sistema Meteorológico do Paraná

Rua Francisco H. dos Santos, 210 - Jardim das Americas, 81531-980, Curitiba, Paraná, Brasil

Resumo. *No Brasil, a principal atividade econômica é a agroindústria, um setor vulnerável a precipitação e eventos relacionados. Nesse contexto, a previsão de tempestades severas possibilita a tomada de decisões e medidas operacionais para mitigar danos, uma vez que esses eventos podem afetar a economia e apresentar riscos a vida humana. Esse projeto objetiva estudar o uso de técnicas de aprendizado de máquina para a previsão do deslocamento desses fenômenos a curtíssimo prazo. Esses métodos são capazes de entender e aprender com suas características e seus relacionamentos. Além disso, uma vez que o modelo é aprendido por ferramentas de Aprendizado de Máquina, o processamento das novas entradas ocorre rapidamente. Foram selecionadas nove técnicas de regressão, todas utilizando características das tempestades como entrada. O desempenho dessas técnicas fora avaliado de acordo com os dados reais observados e das previsões do Titan para o mesmo período de dados. O estudo verificou que ferramentas de Aprendizado de Máquina são abordagens promissoras ao problema proposto, visto que apresentaram resultados semelhantes e até mesmo melhores que o Titan para estimar as próximas posições de uma tempestade, utilizando um número menor de características de entrada.*

Keywords: *Aprendizado de Máquina, Previsão, Tempestades, Regressão*

1 INTRODUÇÃO

O sul do Brasil é uma região propícia a ocorrência de tempestades severas, segundo Beneti (2012), esse tipo de evento pode afetar a economia além de apresentar riscos a vida. Uma vez que sua principal atividade econômica é a agroindústria, setor vulnerável a precipitação e eventos relacionados.

A previsão de eventos dessa natureza pode auxiliar na tomada de decisões e medidas operacionais, bem como mitigar e até mesmo antecipar danos, permitindo que as ações possíveis possam ser tomadas.

Desse modo, de acordo com Bonato (2014), existe a necessidade de buscar técnicas confiáveis e rápidas para o monitoramento de tempestades, que consiste, basicamente, em três processos: a identificação de células ativas de tempestades, o rastreamento e a previsão.

Assim, o foco desse trabalho é o estudo de técnicas de Aprendizado de Máquina aplicados à terceira etapa, que seria a previsão desses fenômenos a curtíssimo prazo.

Para esse objetivo, as primeiras etapas foram obtidas através da ferramenta TITAN (Thunderstorm Identification, Tracking, Analysis and Nowcasting), na qual foram extraídos os dados de células de tempestades identificadas e rastreadas em diferentes estágios de vida. A proposta se estende para a região discutida e utiliza dados de radares meteorológicos na execução do software.

Devido à natureza dos fenômenos representados neste trabalho, os métodos de Aprendizado de Máquina foram escolhidos porque eles são capazes de entender e aprender melhor com suas características e seus relacionamentos. Além disso, uma vez que o modelo é aprendido pelas ferramentas de Aprendizado de Máquina, com a confiabilidade desejada, o processamento das novas entradas ocorre rapidamente.

A avaliação dos resultados é feita da comparação com a previsão fornecida pelo TITAN para cada célula, pois trata-se de uma ferramenta reconhecida e estabilizada na área.

Diversas técnicas para regressão foram testadas, todas utilizando características das tempestades, como a posição, orientação, histórico de deslocamento, área, volume, valor máximo de refletividade, entre outras.

Na Seção 1.1 são descritos os métodos que apresentaram melhor desempenho aplicados ao problema. Destes, verificou-se que o estudo é um caminho promissor aos objetivos propostos, conforme pode ser constatado no campo de resultados.

1.1 O TITAN

O TITAN (Thunderstorm Identification, Tracking, Analysis and Nowcasting) é um software que aborda a identificação, o rastreamento, análise e a previsão a curto prazo de tempestades em tempo real, a partir de dados de radares meteorológicos (NCAR, 2016).

Uma vez que, ”uma tempestade é definida como uma região contígua excedendo limites de refletividade e tamanho”(Dixon e Wiener, 1993), o software realiza o processo de identificação admitindo-se limitantes para os valores de volume e refletividade observados de tempestades, obtidos por meio de dados de radares meteorológicos. Para cada tempestade identificada são geradas algumas características, sendo algumas observadas pelos radares meteorológicos e outras calculadas a partir desses dados e com base em seus conceitos físicos.

Essas características podem ser dividas em três grupos: as que se tratam de sua posição, como latitude e longitude do centróide e sua orientação em relação ao Norte, ou de deslocamento, como direção e velocidade e, ainda, dados relacionados a tamanho e intensidade da tempestade, como volume, área média, área projetada, VIL, valor máximo de refletividade e altura respectiva, massa de granizo e, no caso em que se escolhe representar a tempestade através de elipses, tamanho do eixo maior e menor em quilômetros.

O rastreamento realizado pela técnica se dá por meio do uso dessas características e pontos de tempestades já identificadas, esse processo consiste, genericamente, em ”determinar o movimento correspondente entre células de tempestades em imagens de radares sucessivas”(Han, Zhao et. al, 2009), com isso é possível obter um histórico de deslocamento para cada tempestade conhecida.

A construção de um histórico para esse tipo evento permite que seja realizada a previsão do seu deslocamento para até 60 minutos, sendo cada período de tempo 10 minutos, ou seja, permite a previsão de suas seis posições seguintes. Isso é realizado pela técnica a partir da identificação de uma tempestade em pelo menos três períodos subsequentes de tempo.

Esse processo é realizado pela extrapolação linear e com um coeficiente de ajuste que têm como base o histórico das características da tempestade, um exemplo desse ajuste pode ser visualizado na Figura 1.

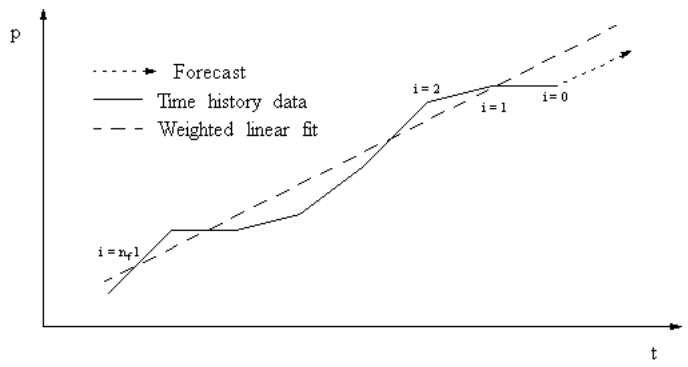

Figura 1: Previsão e ajuste linear no Titan (Fonte: NCAR, 2016)

A extrapolação da tempestade, com base em NCAR (2016), é feita utilizando as seguintes propriedades: Área projetada do centróide, centróide volumétrico (Z), refletividade do centróide, altura, máximo valor de dBZ, fluxo de precipitação, massa, velocidade, direção, área projetada e volume.

2 MATERIAIS E MÉTODOS

2.1 Descrição dos dados

A ferramenta TITAN além de realizar o acompanhamento de tempestades, ainda pode fornecer diversos dados referentes a esses fenômenos e suas características, como foi mencionado na Seção 1.1. O conhecimento desse tipo de informação, de acordo com Damian (2011), possibilita “estimar com mais precisão quais foram os eventos que deram origem ao fenômeno meteorológico e qual será seu comportamento no futuro”.

Nesse contexto, os dados utilizados foram obtidos pela execussão do software e contemplam as tempestades identificadas no período de agosto de 2016 à agosto de 2017.

A execução da ferramenta se deu em períodos de dez minutos e utilizou um mosaico de dados de radares meteorológicos que contempla a região do Brasil. O mosaico foi gerado com dados do tipo PPI (Plan Position Indicator) que, de acordo com Oliveira (2014), é "a forma mais básica de visualização das variáveis medidas pelo radar"e "fornece a projeção num plano horizontal, obtitido através de uma varredura"para um ângulo de elevação constante do radar.

Portanto, os dados da pesquisa se restringem as células de tempestades identificadas e rastreadas pela ferramenta, no período de tempo e região de interesse. Nesse sentido, a quantidade de dados fornecidos para a pesquisa, bem como as que correspondem a divisão em conjunto de treinamento e teste, estão apresentadas na tabela 1, para cada período de tempo.

Tabela 1: Número de células de tempestade

Tempo (minutos)	Total	Treinamento	Teste
10	54376	43500	10876
20	33033	26426	6607
30	19489	15591	3898
40	11714	9371	2343
50	7469	5975	1494
60	4783	3826	957

Como os dados utilizados na identificação das tempestades não foram volumétricos, não foi possível obter as características que se referem a altura e o volume da tempestades. Os dados extraídos para pesquisa estão descritos na Tabela 2 e representam a tempestade no formato de uma elipse.

As diferentes combinações possíveis das características descritas na Tabela 2 foram testadas, a fim de encontrar a melhor configuração para as técnicas aplicadas. Apesar da variedade de possibilidades, o melhor resultado foi encontrado ao selecionar apenas três dos recursos, que seriam o histórico de posição da célula (latitude e longitude) e a de sua velocidade.

2.2 Aprendizado de Máquina

Abordagens por aprendizado de máquina têm sido amplamente aplicadas na meteorologia, alguns exemplos disso são: Pessoa (2004) na previsão de clima, Guilhon (2007) para a previsão de vazões naturais afluentes, Anochi e Silva (2009) no estudo de padrões climáticos sazonais, Lohmann (2013) na previsão probabilística de alagamentos no Município de Curitiba, Santos (2014) na estimativa de chuva, Anochi e Velho (2016) na previsão climática de precipitação, Silva (2017) na Identificação de eventos de tempo severos, entre outros.

Embora todos eles sejam aplicações da meteorlogia, tratam-se de problemas bem variados, como clima, vazão, precipitação etc. O uso desse tipo de técnica na área pode ser atribuído a

Tabela 2: Descrição das características das tempestades

Características	**Descrição**
Centróide	Latitude e Longitude referente a posição do centro da elipse.
Orientação	Ângulo de rotação da elipse em relação ao Norte.
Velocidade	Velocidade calculada do deslocamento.
Direção	Direção do deslocamento.
Eixo maior	Tamanho, em quilômetros, do eixo maior da elipse.
Eixo menor	Tamanho, em quilômetros, do eixo menor da elipse.
Área	Área calculada da tempestade.
VIL	Vertically Integrated Liquid-Water representa as características de um tempestade tridimensional para o bidimensional, uma vez que converte os dados de refletividade em estimativa de água líquida.
dBZ	Representa o valor de refletividade em escala logarítmica.
Massa de granizo	Estimativa da massa de granizo na tempestade.

quantidade de dados e por muitas vezes as características dos fenômenos não serem exatas ou terem padrão desconhecido.

Dado a natureza dos dados que se dispõe e do problema da pesquisa, optou-se por esses métodos, pois eles apresentam grande potencial em aprender características e seus relacionamentos. Além disso, a execução de novas entradas de dados ocorre de forma muito rápida, uma vez que um modelo é treinado baseando-se nesse tipo de abordagem e com uma taxa de confiabilidade desejada.

A ferramenta utilizada foi *scikit learn*, "um módulo Python que integra uma ampla gama de algoritmos de aprendizado de máquina de ponta, para problemas supervisionados e não supervisionados"(Pedregosa et. al, 2011). O módulo possui documentação bem completa dos métodos, além de oferecer fácil manipulação dos seus parâmetros.

Os métodos utilizados são do tipo supervisionado, que tratam-se daqueles em que o aprendizado ocorre a partir "do mapeamento da entrada perceptiva atual em relação à observação das ações decorrentes"(Maia, 2012), ou seja, são apresentados à técnica as saídas desejadas para seu treinamento, para que ela possa aprender e ajustar seus pesos o melhor possível para o conjunto de dados.

A avaliação desse tipo de metodologia é feita usando-se um conjunto de dados desconhecidos pelo modelo treinado. Neste trabalho o conjunto de treinamento é de 80% dos dados e o restante é utilizado para teste.

Como definido anteriormente, as variáveis de entrada, denominado recursos em aprendizado de máquina, são as características de tempestades e os rótulos, resultado ou saída, que pretendemos são seus deslocamentos. Como os rótulos são do tipo contínuo e não discreto, o

problema não é de classificação, mas sim de regressão.

Estabelecido o tipo de metodologia necessária para o problema de pesquisa, foi feito um levantamento das técnicas possíveis e com alguns testes foram selecionadas as nove que melhor se adaptaram ao tipo de recursos. Esses algoritmos foram separados em três áreas: Ensemble, Modelo Linear e Redes Neurais Artificiais, conforme estão descritas, brevemente, nas Seções 2.3, 2.4 e 2.5.

Como esses tipos de técnicas são bastante sensíveis aos dados de entrada, cada treinamento e teste foram realizados diversas vezes, mudando esses conjuntos, que são escolhidos aleatoriamente, a fim de garantir a capacidade de generalização dos modelos. Assim, os resultados apresentados são apenas aqueles em que os parâmetros e dados de entrada mostraram resultados estáveis, ou seja, não foram constatadas mudanças significativas no erro ao longo dessas execuções.

Na maioria das técnicas foram utilizados os parâmetros padrões definidos pelo módulo *scikt-learn*, as técnicas que mostraram melhores resultados para outros parâmetros, são apresentados juntamente com suas descrições.

2.3 Métodos de Ensemble

De acordo com Coenen, Preece e Macintosh (2011), os métodos que são denotados por Ensemble tratam-se da combinação da predição de vários algoritmos, com alguma variação nos parâmetros ou no seu conjunto aprendizado.

As técnicas que foram utilizadas desse eixo são baseadas no algoritmo de Árvore de decisões que, de acordo Koronacki, Ras e Wierzchon (2009), em comparação a outras técnicas de Aprendizado de Máquina, apresenta um rápido processamento na fase de treinamento do modelo. Dentre as técnicas utilizadas, quatro são classificados em Ensemble: Random Forest, Extra Trees, Gradient Boosting e Bagging.

Random Forest

Random Forest ou Floresta Aleatória trata-se de um conjunto de árvores de decisão, segundo Telocken (2016), criado à partir da seleção de subconjuntos de atributos, gerados aleatoriamente. Como essa metodologia gera diferentes subconjuntos de árvores de decisões, possibilita melhores resultados do que a árvore de decisão simples. Um parâmetro importante é quantidade de árvores, para a pesquisa foram estabelecidas 200 árvores.

Enquanto na construção de uma árvore de decisão simples a divisão de cada nó é feita levando em conta a melhor divisão do conjunto de recursos, na técnica essa escolha é feita apenas para o subconjunto aleatório.

Extra trees

Assim como o Random Forest, Extra trees trata-se de um conjunto de árvores de decisões, mas enquanto na anterior a divisão de um nó é feita selecionando a melhor divisão para o subconjunto aleatório, nessa a divisão é escolhida aleatoriamente no subconjunto.

Gradient Boosting

O princípio básico de funcionamento do Gradient Boosting, segundo Paula (2016), "é bastante simples: dada uma função de perda e dado um estimador fraco o algoritmo procura um modelo que minimiza essa função de perda", o estimador utilizado é a árvore de decisão, em que calculando-se o gradiente de perda é realizado o ajuste do modelo a fim de minimizar essa função. Foi estabelecido o número de estimadores igual à 2000.

Bagging

O algoritmo Bagging, basicamente, "cria amostras repetidamente (com substituição) a partir de um conjunto de dados", de acordo com Tan, Steinbach e Kumar (2009), com base na distribuição uniforme de probabilidade, treina um regressor para cada conjunto e ao fim seleciona o melhor estimador. Na pesquisa foi estabelecido o número de árvores igual à 40 para a execução dessa técnica.

2.4 Métodos baseados em Modelo Linear

"Nestes Modelos supõe-se que a média de uma variável aleatória Y é dependente de uma ou mais variáveis (X_1, X_2, ..., X_r)"(Braga, 2005) e suas equações básicas do modelo pode ser representada por:

$$Y = \beta_0 + \sum_{i=1}^{r} \beta_1 X_i + \epsilon, \tag{1}$$

Em que objetiva-se encontrar valores para todo β a fim de estimar o valor médio de Y, minimizando o erro (ϵ).

Foram aplicados quatro modelos lineares: Theil Sen, Huber, Ransac e Bayesian Ridge, os quais são apresentados brevemente a seguir.

Theil Sen

No algoritmo formulado por Theil (1950) e Sen (1968), segundo Wilcox (2011), a estimativa é realizada calculando-se o coeficiente angular e linear para todas as combinações possíveis de pontos de uma subamostra e, por fim, dado pela mediana espacial desses valores.

Huber

O algoritmo Huber é um modelo Robusto, de acordo com Tanja e Kostina (2012), pois permite a redução da influência de pontos dados como outliers ao conjunto. O estimador proposto por Huber em 1962, minimiza a função custo de avaliação do termo de mínimos quadrados do Modelo Linear de Regressão, pois busca identificar os pontos que podem ser considerados como outliers, permitindo que o método seja menos sensível a esses dados.

RANSAC

Random Sample Consensus (Consenso de Amostra Aleatórias) é um modelo Robusto, realizado em duas etapas, a primeira se trata da geração de hipóteses, e a segunda a avaliação dessas hipóteses. Como definido por Choi, Kim e Yu (1997), a técnica particiona os dados em subconjuntos, gerando as hipóteses com os dados que não são outliers, depois de um número de iterações seleciona a hipótese que melhor representa o conjunto de dados, ou seja, a que minimiza erro.

Bayesian Ridge

O algoritmo Bayesian Ridge se funtamenta no teorema de Bays, de acordo com Silveira (1999), essa abordagem se baseia em que problema de decisão pode ser representado nos termos probabilísticos e que todas as probabilidades relevantes são conhecidas.

2.5 Método de Redes Neurais Artificiais

Segundo Haykin (2001), "uma rede neural é um máquina que é projetada para *modelar* a maneira como o cérebro realiza uma tarefa particular ou função de interesse", como reconhecimento de padrões, classificação, entre outros, interligando neurônios - células computacionais - para obter bom desempenho.

Multilayer Perceptron (MLP)

Multilayer Perceptron (Perceptron de Múltiplas Camadas) é uma Rede Neural Artificial de Aprendizagem em que possui uma camada de entrada, em que o número de neurônios é igual ao número de dados de entrada, pelo menos uma camada escondida de neurônios ocultos e uma camada de saída.

Após a parametrização em grade da técnica, ou seja, o teste com todas as combinações possíveis dos parâmetros do método, foram estabelecidos o uso da função de ativação identidade (*indentity*), dada por $f(x) = x$, o *solver* para otimização dos pesos como *lbfgs*, otimizador da família de métodos quase-Newton e cronograma de atualização da taxa de aprendizagem adaptativo (*adaptive*), ou seja, mantém-se a taxa de aprendizagem constante enquanto a rede não estagnar, caso contrário atualiza o seu valor dividindo o anterior por cinco.

Além desses parâmetros ainda foram utilizados o número máximo de iterações igual à 2000, a tolerância na sexta casa decimal e a configuração da rede com uma única camada escondida, com 100 neurônios.

3 RESULTADOS

Os resultados serão analisados para cada período de tempo de previsão, em uma janela de 10 minutos. Ou seja, cada t_j, com $j = 1, ..., 6$, de modo que t_1 representa a primeira posição de previsão, com 10 minutos de deslocamento, t_2 a segunda, com 20 minutos de deslocamento e assim para as demais posições.

Para a avaliação para t_1 são consideradas todas as tempestades com histórico maior ou igual à 40 minutos, pois os primeiros 30 minutos tratam-se dos recursos utilizados para a previsão, enquanto os outros 10 refrem-se ao deslocamento que deseja estimar. Seguindo esse raciocínio, para t_2 é utilizado um histórico de 50 minutos ou mais e, da mesma forma, para os demais valores de t.

Seja n o número de tempestades, x os recursos e y os rótulos, tem-se a posição observada da tempestade i, dada por:

$$y_i = (y_{i1}, y_{i2}), \forall i = 1, ..., n \tag{2}$$

Sendo, y_{i1} a coordenada x da posição do centróide e y_{i2} a coordenada y, no cartesiano.

Analogamente, y_{pi} representa a posição prevista do centróide da tempestade i no plano cartesiano, portanto tem-se o erro da previsão para cada tempestade dado por:

$$E_i = \sqrt{(y_{i1} - y_{pi1})^2 + (y_{i2} - y_{pi2})^2} \tag{3}$$

Em que E_i é a distância euclidiana entre o ponto observado e o previsto. Assim, para avaliar o erro das técnicas foram calculados o desvio padrão (σ), a média (μ) e a mediana (med) do erro para cada t_j.

3.1 Previsão por ferramentas de Aprendizado de Máquina

Os resultados apresentados para as ferramentas de Aprendizado de Máquina se referem aos erros calculados para o conjunto de teste, em relação a distância do ponto previsto por cada técnica, ao ponto observado.

Foram realizadas diversas execuções com diferentes configurações de parâmetros para as técnicas, assim como combinações das características de entrada, os resultados apresentados são para a configuração do modelo que apresentou melhor o desempenho.

Foi treinado um modelo para prever cada t_j, todos eles com o histórico de 30 minutos das primeiras posições (lat/lon) identificadas e da velocidade do deslocamento, o uso de outras características não mostraram diferenças significativas no resultados.

Na Tabela 3, estão listados os erros para a previsão de 10 e 20 minutos apresentados por cada técnica.

Tabela 3: Erros para 10 e 20 minutos de previsão

Método	μ_1	med_1	σ_1	μ_2	med_2	σ_2
Random Forest	5,906	4,556	13,821	8,773	7,355	8,789
MLP	6,022	4,619	14,143	9,185	7,753	8,508
Gradient Boosting	5,968	4,626	13,83	9,035	7,616	8,413
Bagging	5,99	4,656	13,837	8,881	7,474	8,649
Extra Trees	6,321	5,003	14,032	9,324	7,867	8,904
Ransac	5,589	4,091	14,051	8,154	6,589	8,593
Theil Sen	10,403	6,353	17,151	10,213	8,135	9,738
Huber	5,548	4,037	14,138	8,134	6,514	8,663
Bayesian Ridge	5,614	4,123	13,916	8,252	6,688	8,6

Com base na Tabela 3 é possível verificar que para esses períodos de previsão apresentados as técnicas Huber, Ransac e Bayesian Ridge, respectivamente, apresentam melhores resultados

em relação a média e a mediana do erro. Enquanto isso, os algoritmos de Random Forest, Gradient Boosting e Bagging apresentam um desvio padrão menor para os primeiros 10 minutos, respectivamente. Já para 20 minutos, MLP melhora em todas as métricas avaliadas.

Tabela 4: Erros para 30 e 40 minutos de previsão

Método	μ_3	med_3	σ_3	μ_4	med_4	σ_4
Random Forest	11,926	10,134	10,851	15,505	13,751	9,971
MLP	10,965	9,104	8,3	13,749	11,789	9,496
Gradient Boosting	12,47	10,826	8,944	16,108	14,318	10,292
Bagging	12,105	10,281	11,012	15,72	14,07	10,086
Extra Trees	12,476	10,792	8,61	16,228	14,26	10,173
RanSAC	10,82	8,918	8,754	13,612	11,72	9,382
Theil Sen	24,673	13,623	27,563	19,924	14,804	17,308
Huber	10,791	8,888	8,949	13,5517	11,612	9,518
Bayesian Ridge	10,826	8,954	8,63	13,628	11,714	9,355

Em relação aos períodos anteriores é perceptível uma perda do algoritmo Theil Sen nos 30 minutos, com base na Tabela 4. Já para a previsão de 40 minutos ele apresenta uma grande melhoria, entretanto ele se mantém com resultados inferiores aos demais.

Destacam-se os algoritmos Huber, Bayesian Ridge, Ransac e MLP com os melhores desempenhos.

Tabela 5: Erros para 50 e 60 minutos de previsão

Método	μ_5	med_5	σ_5	μ_6	med_6	σ_6
Random Forest	19,714	17,819	11,912	23,084	20,766	13,374
MLP	16,682	14,41	11,003	18,722	16,278	11,568
Gradient Boosting	20,24	18,378	12,193	23,572	21,612	13,913
Bagging	19,965	17,883	12,022	23,591	21,122	13,666
Extra Trees	20,199	18,208	12,251	23,835	21,456	13,970
RanSAC	16,442	14,411	10,76	18,481	16,148	11,435
Theil Sen	42,104	22,538	47,103	25,611	20,618	19,303
Huber	16,379	14,176	11,1	18,386	15,675	11,627
Bayesian Ridge	16,49	14,461	10,706	18,573	16,176	11,37

Em análise aos 50 e 60 minutos de previsão, apresentados na Tabela 5, os mesmos quatro

métodos se mantêm bem próximos apresentando os melhores resultados, enquanto Theil Sen apresenta os piores, além de muitas oscilações nos períodos de tempos subsequentes.

Em relação aos primeiros 10 minutos de previsão, todas as técnicas apresentaram uma melhoria significativa no desvio padrão para os demais períodos de tempo, tanto que na sexta posição os valores se mantêm menores do que na primeira, para a maioria dos algoritmos.

Comparando de maneira geral os resultados das técnicas de Aprendizado de Máquina, podemos concluir que as técnicas Huber, Bayesian Ridge, MLP e Ransac apresentaram os erros bem semelhantes entre si, além de mostrarem os melhores desempenhos. Enquanto Random Forest, Gradient Boosting, Bagging e Extra Trees apresentaram um desempenho um pouco inferior, embora também tenham apresentado erros bem próximos entre si. Por outro lado, Theil Sen apresentou o pior resultado, bem distante das demais técnicas.

3.2 Previsão pelo software TITAN

A avaliação do resultados das técnicas de Aprendizado de Máquina foi realizada comparando com a previsão proveniente do TITAN, não com o objetivo de superá-la, mas sim mensurar a qualidade das previsões provenientes das técnicas de Aprendizado de Máquina, já que o software é reconhecido e utilizado para o problema proposto.

Assim, a Tabela 6 apresenta os erros do Titan que foram calculados para o mesmo domínio e período de dados.

Tabela 6: Erro da previsão do TITAN pata cada t_j

t_j	μ_j	med_j	σ_j
t_1	5,92	4,432	7,954
t_2	9,007	7,053	10,426
t_3	11,692	9,451	11,441
t_4	14,858	12,164	14,909
t_5	17,479	14,515	12,875
t_6	20,284	16,846	14,639

3.3 Análise dos resultados

Para uma melhor visualização dos erros das técnicas comparados com o TITAN são apresentadas na figuras 2, 3 e 4, respectivamente, a média, mediana e desvio padrão dos erros para cada período de tempo.

Analisando a Figura 2, podemos perceber que o algoritmo Ransac, Huber e Bayesian Ridge se mantêm com um erro médio inferior ao TITAN, enquanto MLP inicia com um erro um pouco superior para as duas primeiras posições, mas apresenta uma melhora em sua performance nas

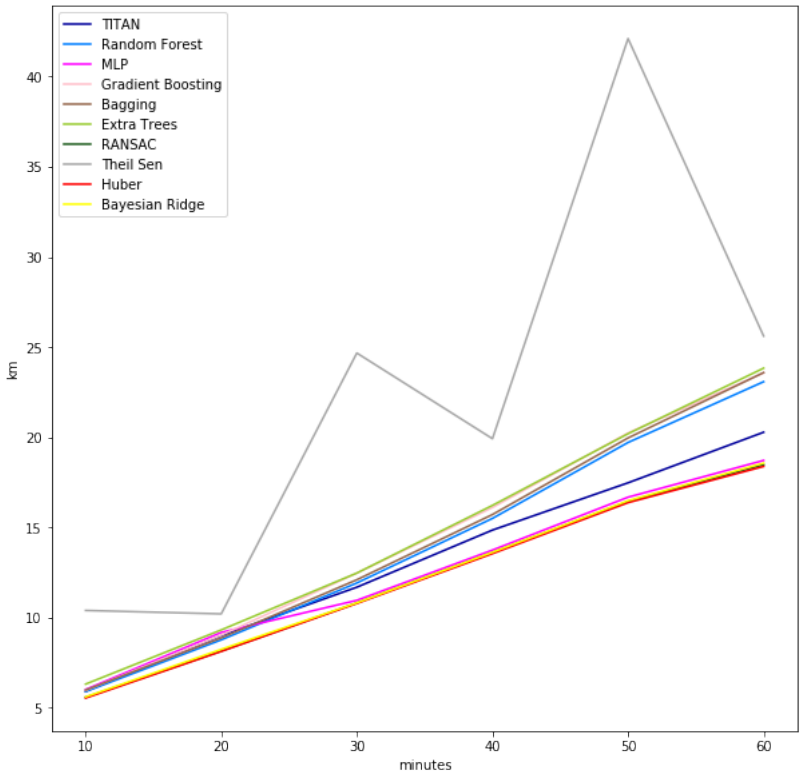

Figura 2: Erro médio de cada método em relação ao tempo

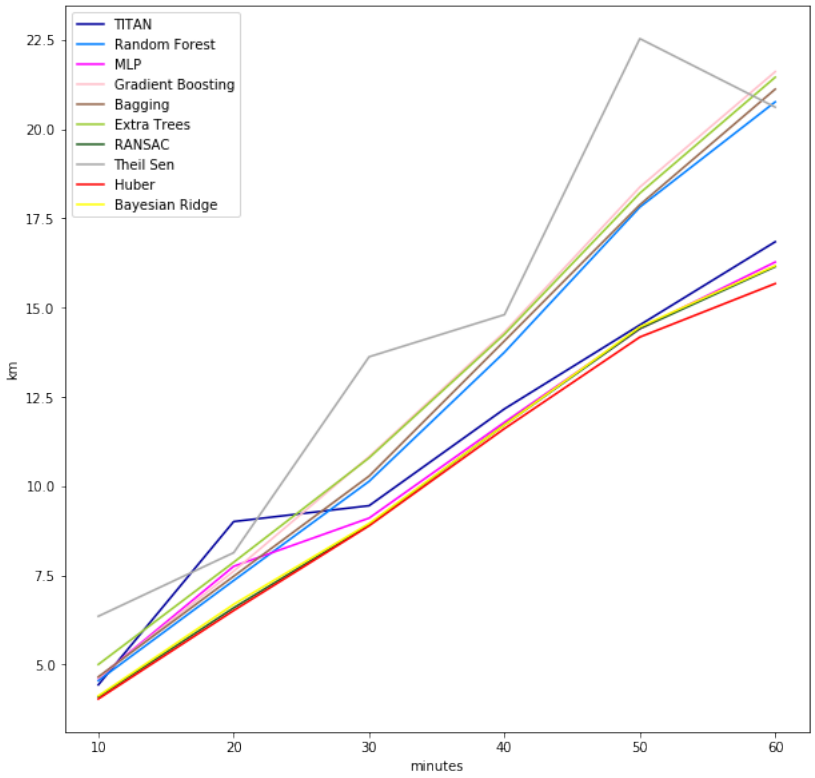

Figura 3: Mediana do erro de cada método em relação ao tempo

demais. Por outro lado, Theil Sen apresenta os piores resultados para todas as posições além de mostrar bastante variação de um período a outro.

As análises feitas em relação ao erro médio se mantém para a mediana (Figura 3) e permite perceber que os algoritmos Random Forest, Gradient Boosting, Extra Tree e Bagging mostram erros maiores aos do TITAN, tanto quando analisada a mediana, como também a média.

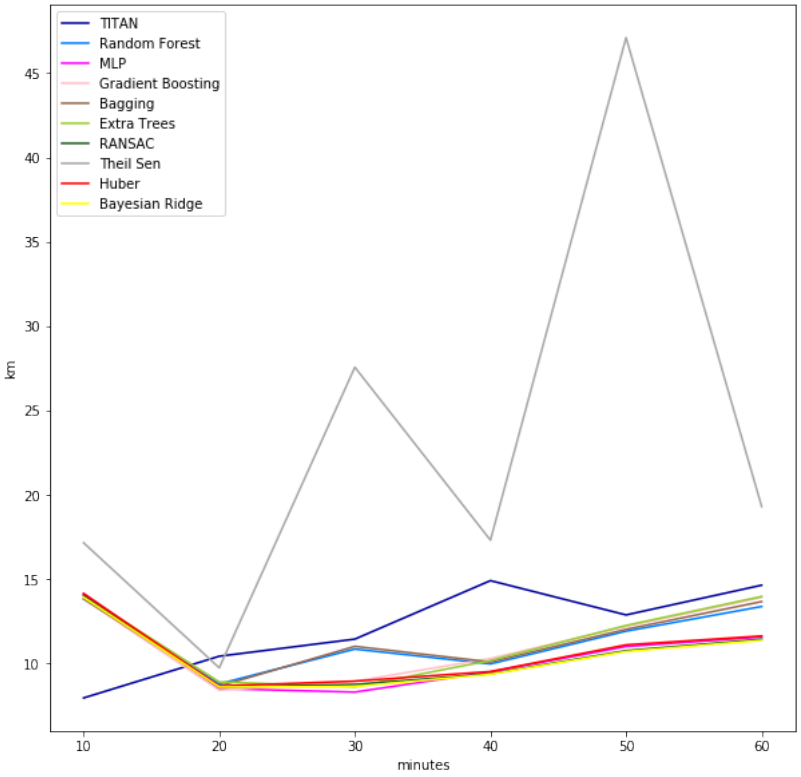

Figura 4: Desvio padrão do erro de cada método em relação ao tempo

Conforme a Figura 4, todas as técnicas de Aprendizado de Máquina apresentam menor desempenho do que o TITAN na primeira posição, em relação ao desvio padrão do erro, mas o superam para as demais posições, com exceção do algoritmo Theil Sen.

Portanto, é fácil notar que as técnicas de melhores resultados foram o Bayesian Ridge, Ransac, Huber e MLP. Dentre essas quatro técnicas, MLP mostrou uma pequena perda para segunda posição prevista, enquanto as demais se mantiveram sem maiores oscilações.

Podemos observar que 75% das técnicas que se mostraram mais viáveis ao problema foram do tipo baseadas em Modelo Linear. Desse conjunto de técnicas testadas apenas Theil Sen não mostrou se ajustar ao problema.

De acordo com os dados apresentados, podemos concluir que as técnicas de Aprendizado de Máquina apresentam um comportamento bem parecido com o TITAN em termos de resultados, podendo apresentar algumas vantagens ou desvantagens dependendo da técnica, mas com diferenças pouco discrepantes em quanto ao erro.

4 CONSIDERAÇÕES FINAIS

Portanto, com base nos dados apresentados é possível concluir que os resultados da pesquisa foram satisfatórios, uma vez que apontam erros semelhantes a uma técnica amplamente utilizada na área e, possivelmente, melhores.

Nesse sentido, os métodos de Aprendizado de Máquina baseados em Modelo Linear, apresentaram bom desempenho para a previsão do deslocamento de tempestades, utilizando dados do histórico da posição e velocidade desses eventos. Na sêquencia, tem-se MLP que também teve um dos melhores desempenhos.

Embora as técnicas baseadas em Ensemble não estejam no grupo que alcançou os melhores resultados para o problema, os erros apresentados foram aceitáveis e não muito maiores que os demais algoritmos. Entretanto, mostraram um perda maior na acurácia, em relação a primeira e a última posição de previsão.

Apesar dos métodos estudados não terem encontrado resultados tão superiores que o software para os testes realizados, permitindo a previsão mais posições com a mesma confiabilidade, por exemplo, eles apresentam outros aspectos positivos.

Um exemplo disso, é a quantidade de características utilizadas para estimar o deslocamento, apenas três, enquanto o TITAN utiliza de onze. Além disso, as técnicas são de rápido processamento uma vez que o modelo já foi treinado.

Um aspecto interessante é que o sistema de previsão do TITAN existe a mais de três década e durante este período de tempo tem sido aprimorado por diversas vezes. Devido a ser um método tão utilizado, optou-se por usá-lo como referência de quais resultados são suficiêntes na área.

Nesse sentido, é possível refletir sobre a possibilidade de melhorar uma técnica dentre as apresentadas, visto que já mostroram resultados tão satisfatórios, desde a primeira implementação ao problema.

Para trabalhos futuros é interessante pensar em outras técnicas ou formulação de Aprendizado de Máquina, como hibridização ou tratar do problema como função multiobjetivo, por exemplo. Isso seria interessante para verificar a possibilidade de gerar a previsão para mais posições ou com uma acurácia maior.

Um ponto relevante é a quantidade de dados de entrada e a qualidade desses dados, ou seja, com a melhoria das técnicas de identificação e acompanhamento de tempestades, permitindo se obter históricos mais completos para a células de tempestades ao longo do tempo, pode implicar em grandes melhorias na previsão, uma vez que o desempenho dessas técnicas está relacionado com a qualidade dos recursos aprendidos.

Por fim, se espera que as técnicas de Aprendizado de Máquina possam ser consideradas para o problema proposto, uma vez que se mostraram, no mínimo, suficientes.

REFERÊNCIAS

Anochi, J. A. & Silva, J. D., , 2009. Uso de teoria de conjuntos aproximativos e redes neurais artificiais no estudo de padrões climáticos sazonais. *Learning and Nonlinear Models*, v.7, p. 83-91.

Anochi, J. A., & Velho, H. F. C., 2016. Previsão climática de precipitação para a região Sul por rede neural autoconfigurada. *Ciência e Natura*, v. 38.

Beneti, C. A. A., 2012. *Caracterização Hidrodinâmica e Elétrica de Sistemas Convectivos de Mesoescala.* PhD Thesis — Instituto de Astronomia, Geofísica e Ciências Atmosféricas, Universidade de São Paulo.

Bonato, J. V. R., 2014. *Clusterização de Dados Meteorológicos para Comparação de Técnicas de Nowcasting.* PhD diss., Programa de Pós-Graduação em Métodos Numéricos para Engenharia, Universidade Federal do Paraná.

Braga, L. P. V. B., 2005. *Introdução à Mineração de Dados-2a edição: Edição ampliada e revisada*. Editora E-papers.

Choi, S., Kim, T. & Yu, W., 1997. *Performance evaluation of RANSAC family.* Journal of Computer Vision 24.3: 271-300.

Coenen, F, Preece, A., & Macintosh, A., 2011. Research and Development in Intelligent Systems XX: Proceedings of AI2003, the Twenty-third SGAI *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer Science Business Media.

Damian, E. A., 2011. *Duas metodologias aplicadas à classificação de precipitação convectiva e estratiforme com radar meteorológico: SVM e K-means*. PhD diss., Programa de Pós-Graduação em Métodos Numéricos para Engenharia, Universidade Federal do Paraná.

Dixon, M., Wiener, G. (1993). TITAN: Thunderstorm identification, tracking, analysis, and nowcasting—A radar-based methodology. *Journal of atmospheric and oceanic technology*, 10(6), 785-797.

Guilhon, L. G. F., Rocha, V. F. & Moreira, J. C., 2007. Comparação de métodos de previsão de vazões naturais afluentes a aproveitamentos hidroelétricos. *Revista Brasileira de Recursos Hídricos*, v. 12, n. 3, p. 13-20

Han, L., Fu, S., Zhao, L., Zheng, Y., Wang, H., Lin, Y. (2009). 3D convective storm identification, tracking, and forecasting—An enhanced TITAN algorithm. *Journal of Atmospheric and Oceanic Technology*, 26(4), 719-732.

Haykin, S. , 2001. *Redes neurais: princípios e prática.* Bookman Editora.

Kleina, M., 2015. *Identificação, monitoramento e previsão de tempestades elétricas.* PhD Thesis, Programa de Pós-Graduação em Métodos Numéricos para Engenharia, Universidade Federal do Paraná.

Koronacki, J., Ras Z. W. & Wierzchon, S. T., eds., 2009. *Advances in Machine Learning II: Dedicated to the Memory of Professor Ryszard S. Michalski.* v. 263. Springer.

Lohmann, M., 2013. *Regressão logística e redes neurais aplicadas à previsão probabilística de alagamentos no Município de Curitiba, Pr.*

Maia, W. D. A. (2012). *Percepção Inteligência Artificial- Conceitos, Considerações e Arquitetura.* biblioteca24horas.

NCAR - National Center for Atmospheric Research (2016). *TITAN - Thunderstorm Identification Tracking Analysis and Nowcasting.* Disponível em: $http://www.ral.ucar.edu/projects/titan/$. Acesso em: 20 Agosto de 2017.

Oliveira, C., 2014. *Identificação e Correção da Banda Brilhante em Dados de Radar Meteorológico*. PhD diss., Programa de Pós-Graduação em Métodos Numéricos para Engenharia, Universidade Federal do Paraná.

Paula, E. L. D. (2016). *Mineração de dados como suporte à detecção de lavagem de dinheiro.* PhD diss., Mestrado Profissional em Computação Aplicada, Universidade de Brasília.

Pedregosa, F., Varoquaux, G., Gramfort, A., et. al, 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v.12, p. 2825–2830.

Pessoa, A. S. A., 2004. *Mineração de dados meteorológicos pela teoria dos conjuntos aproximativos na previsão de clima por redes neurais artificiais.* PhD Thesis, Pós-Graduação em Computação Aplicada, INPE.

Raschka, S., 2015. *Python machine learning.* Packt Publishing Ltd.

Santos, T. N, 2014. *Redes neurais artificiais e relação ZR aplicadas à estimativa de chuva*.

Silveira, R. B., 1999. *Recognition of clutter in weather radars using polarization diversity information and artificial neural networks.* PhD diss., University of Essex.

Silva, T. G. G. J., 2017. *Identificação de Evento de Tempo Severo utilizando técnicas de Aprendizagem de Máquina em dados de radar polarimétrico.* PhD diss., Programa de Pós-Graduação em Métodos Numéricos para Engenharia, Universidade Federal do Paraná.

Tan, P. N., Steinbach, M., & Kumar, V. (2009). *Introdução ao datamining: mineração de dados*. Ciência Moderna.

Tanja, B. & Kostina E., 2012. *Robust Parameter Estimation Based on Huber Estimator in Systems of Differential Equations.* Modeling, Simulation and Optimization of Complex Processes. Springer, Berlin, Heidelberg. 13-23.

Teloken, A. (2016). *Estudo Comparativo entre os algoritmos de Mineração de Dados Random Forest e J48 na tomada de Decisão.* Simpósio de Pesquisa e Desenvolvimento em Computação, 2(1).

Wilcox, R. (2011). *Modern statistics for the social and behavioral sciences: A practical introduction*.CRC press.