Modelos Baseados em Árvores de Regressão Extrema Aplicados na Previsão de Carga Média

M. R. Vidal and R. M. Salgado

***Abstract*— Load forecasting is a very important task in the planning and operation of power systems. However it is not trivial work. The electric market is structurally different from other commodities and erroneous analyzes can cause serious financial risks. In this article, we propose the analysis of the machine learning techniques XGBoost and Random Forest, on two prediction strategies. To evaluate them, an algorithm that selects the component that best obtains the results within a set of random configurations has been proposed. The results were very relevant, of which it obtained a high performance.**

Keywords*— *Power System Operation, Load Forecast, XGBoost, Random Forest, Machine Learning.

I. INTRODUCAO

A gestão e manutenção da demanda de energia elétrica em empresas de distribuição de energia são tarefas muito importantes. Esta área tem gerado competição no mercado entre as empresas que comercializam a energia elétrica, tornando-as intoleráveis a altos erros de estimação da carga real.

Previsões de carga são essenciais para todo o planejamento e operações elétricas. Alto grau de precisão possibilita otimização de produção, redução de custos, melhoras na coordenação hidrotérmica e na programação de transferências de energia entre concessionarias ou companhias elétricas [1].

A realização delas é indispensável, pois a partir dessas previsões se estabelecem quando e quanto da capacidade de geração e transmissão deve se utilizar no atendimento da demanda prevista sem interrupção no fornecimento [2].

Existe uma grande preocupação em garantir o atendimento das reais deficiências da demanda, já que a energia é um patrimônio vital no desenvolvimento econômico de uma pais. Tornando o planejamento indispensável, pois ele maximiza a redução de perdas e riscos em sistemas elétricos, bem como a segurança da energia gerada[3][4].

De acordo com o horizonte de previsão, esse problema pode ser agrupado em 4 classes:

- ***Long-term*:** Previsões de longo prazo condensam na análise e planejamento da lucratividade de investimentos. Geralmente os prazos são mensurados meses, trimestres ou anos [5].
- ***Mid-term*:** Previsões de médio prazo são optadas no gerenciamento de riscos, previsão de preço da eletricidade e cálculos de balanço. Usualmente são estimadas previsões de alguns dias ou meses a frente [6].
- ***Short-term*:** Previsão de curto prazo são de principal importância nas operações diárias. Em geral envolve algumas horas ou dias a frente [7].
- ***Very Short-term*:** Previsões de curtíssimo prazo são importantes para o gerenciamento do sistema de energia ou no comércio, nas seguintes 48 às 72 horas. Normalmente consistem em minutos ou horas a frente.

Neste artigo, o foco principal foi a previsão de carga em curto prazo (*short-term*), visto que o foco é a previsão de um único mês a frente para cada análise. Modelos precisos desta classe de problema têm grande impacto no planejamento e operação de sistemas de potência. Contudo não é uma tarefa trivial. O mercado elétrico é estruturalmente diferente de outras *commodities,* o balanço dinâmico em tempo real das redes elétricas envolve muitos fatores externos. Análises errôneas podem causar sérios riscos financeiros e/ou perda de confiabilidade na empresa [8].

Existem diversas técnicas estudadas e desenvolvidas na literatura que visam melhorar e aprimorar a previsão de carga elétrica. Estas técnicas podem ser divididas em duas categorias, os modelos estatísticos e os baseados em aprendizado de máquina [9] [10].

Em síntese os modelos estatísticos são utilizados principalmente para descrever e compreender informações importantes sobre a séries temporais. Das quais, podem se destacar padrões sobre tendências, periodicidade, sazonalidades e/ou irregularidades. Em exemplo podem-se destacar: *Exponential Smoothing*[11], *Autoregressive Moving Average (ARMA)*[12], *Autoregressive Integrated Moving Average (ARIMA)* [13] entre outros.

Atualmente os modelos estatísticos em geral são utilizados em combinação de algoritmos de aprendizado de máquina. Pois estes modelos possuem a desvantagem de realizar análises somente lineares. Em razão disto não conseguem prever as séries de carga não lineares sozinhos.

Por esta razão, os algoritmos inteligentes são utilizados. Estes conseguem modelar cenários de não linearidade e linearidade, de modo a realizar modelos mais generalizados, e assim, conseguirem um maior desempenho em precisão. Na literatura pode-se destacar os seguintes algoritmos: *Artificial Neural Network (ANN)*[14], *Support Vector Machine (SVM)* [15], *Decision Trees (DT)*[16][17] entre outros.

O foco principal deste artigo é a análise sobre as técnicas de aprendizado de máquina, especificamente nos modelos baseados em ensemble de árvores. Na literatura mais recente, modelos de ensemble têm sido um dos principais métodos utilizados no aumento da qualidade dos resultados em previsão. Alguns trabalhos que utilizaram desta metodologia são: [18] por intermédio do *Extreme Learning Machine(ELM)* combinou os modelos *Partial Least-Squares Regression* e *Hybrid Neural Networks*, [19] combinou os modelos *Least-Square Support Vector Machine (LSSVM)* e o *Particle Swarm Optimisation*

M. R. Vidal, Universidade Federal de Alfenas (Unifal), Alfenas, Minas Gerais, Brasil, mauricio.vidal@bcc.unifal-mg.edu.br

R. M. Salgado, Universidade Federal de Alfenas (Unifal), Alfenas, Minas Gerais, Brasil, ricardo@bcc.unifal-mg.edu.br

(PSO), [10] que utilizou ensemble de redes neurais em conjunto com o uso de algoritmo genético, [20] e [21] apresentou a efetividade do uso do *Random Forest* para previsão de carga.

Fundamentado nisso, esse artigo objetivou analisar a eficiência dos modelos de ensemble *Random Forest* e *XGBoost* sobre duas estratégias de previsão de carga média. Para isso é proposto um algoritmo que recebe 100 parametrizações aleatórias de ambos modelos, das quais o algoritmo realiza a seleção da parametrização que obteve o melhor resultado aplicado em uma componente preditiva e ajusta de maneira inteligente informações de feriado e pós-feriado que os modelos *XGBoost* e *Random Forest* não conseguem prever. Os dados históricos utilizados para prever a média diária gerada por estas técnicas são informações reais de carga elétrica medidas no sudeste do Brasil por uma grande metrópole. Com esses dados foram separados 4 meses para previsão: Jun/2000, Ago/2000, Nov/2000 e Jan/200. Dos quais os resultados apresentados neste trabalho conseguiram superar de forma relevantes os resultados obtidos por [22]. Com a estratégia 1-step houve redução de em média 0.8% para o *XGBoost* e 0.67% para o *Random Forest*. E para a estratégia N-steps verificou-se redução de em média 5.14% para o *XGBoost* e 4.45% para o *Random Forest* utilizando-se a mesma base de dados e similar análise que [22].

Este trabalho encontra organizado da seguinte forma: na seção I foi apresentada a introdução do trabalho. Na seção II é realizado uma breve descrição teórica dos modelos de previsão utilizadas neste artigo. A elaboração do algoritmo proposto e descrita na seção III. Análises sobre os dados utilizados encontram se na seção IV. E as seções V e VI fecham o trabalho trazendo respectivamente os resultados e conclusões obtidas neste artigo.

II. FUNDAMENTAÇÃO TEÓRICA

Essa seção tem o intuito de apresentar os pressupostos teóricos que serviram de base para o desenvolvimento e execução da pesquisa.

A. Extreme Gradient Boost

O *Extreme Gradiente Boosting,* mais comumente chamado de *Xgboost,* é uma implementação muito eficiente e extensiva do *framework Gradient Boosting (GB)* implementado por Friedman em 1999. Esta técnica se baseia no método de *ensemble Boosting*, que geralmente é utilizado para melhorar a precisão de modelos de aprendizado "fracos", tornando-os mais robustos e generalizados [23].

Logo após a percepção de que inserir aleatoriedade dentro de funções de estimação poderiam aumentar o desempenho e a qualidade de um modelo de aprendizado. Friedman realizou uma pequena modificação no algoritmo *GB*, agregando-a como parte essencial do procedimento.

Dessa forma o algoritmo passou a selecionar em cada iteração uma amostra de dados de treinamento para ajuste do modelo e cálculo da atualização deste na próxima repetição, de maneira aleatória e sem reposição.

Assim como o *boosting*, esse algoritmo foi capaz de gerar modelos mais genéricos e precisos utilizando um conjunto de previsores "fracos" e "fortes", produzidos a cada iteração, de modo a atingir uma estimativa mais precisa e robusta [24].

Atualmente o *Xgboost* é muito utilizado em competições de inteligência artificial. Um dos sites mais famosos que concedem essas competições é o *Kaggle,* no qual se trabalha problemas em base de dados reais, de modo a testarem as melhores técnicas de machine *learning*. Dentre as soluções vencedoras, na grande maioria, o framework se encontra individualmente ou em combinação com outras técnicas, decorrentemente ganhando um grande destaque [25].

B. Random Forest

Random Forest é um combinador de árvores de predição que utiliza da técnica de *bootstrap aggregating,* ou como é usualmente conhecido por seu acrônimo *bagging,* para gerar suas soluções mais estáveis e fidedignas[26].

Este método consiste na criação de várias árvores aleatórias, que possuem entre si a mesma distribuição. De modo a generalizar o erro e convergir a um certo número de árvores na floresta.

Para atingir este objetivo é necessário reduzir a correlação entre as árvores de decisão gerada pelo algoritmo. Assim, o *Random Forest* combinou a ideia de seleção aleatória de característica, apresentado por Ho em 1998, em conjunto com a técnica de *ensemble Bagging*, implementado por Breiman em 1996 [27] [28].

Ocasionando uma substancial exatidão sobre os modelos individuais, comparando-a favoravelmente ao Adaboost, muito mais rápido que implementações de *Bagging* ou *Boosting* e relativamente robusto em *outliers* e ruídos [26].

III. METODOLOGIA

Neste trabalho, propõem-se uma metodologia que visa avaliar os modelos de aprendizado de máquina *XGBoos*t e *Random Forest* sobre duas estratégias de previsão de carga diferentes. Para tal fim, desenvolveu-se um algoritmo que seleciona as componentes que obtiveram o melhor resultado em ambos modelos em estratégias de previsão 1-step e N-steps à frente .

C. Elaboração da tabela de conhecimento

Inicialmente foi definido para cada dia a ser previsto um conjunto de características (*features*). Tais como: dia do mês, temperatura máxima, temperatura média, temperatura mínima, informações sobre feriado, se é um dia posterior a um feriado, informações sobre brilho solar e a carga média.

Essas *features* serão agrupadas em uma tabela de conhecimento, onde cada coluna desta tabela representa um padrão correspondente a um determinado dia. Assim como pode ser visto em uma exemplificação de uma parte da tabela de conhecimento pela Tabela I.

TABELA I: EXEMPLIFICAÇÃO DA TABELA DE CONHECIMENTO.

Data	**31/07/2000**	**01/08/2000**	**02/08/2000**
Mes	07	08	08

TMax	23.6	22.1	25.6
TMin	10.8	12.8	13.3
TMed	15.213	15.121	18.208
EF	0	0	0
PF	0	0	0
Brl	8.0	4.4	6.2
Carga	5294.73	5127.57	5161.63

Sendo:

- **Mes:** corresponde ao número do mês correspondente a uma determinada data.
- **TMax:** corresponde a temperatura máxima em °C obtida naquele dia.
- **TMin:** corresponde a temperatura mínima em °C obtida naquele dia.
- **TMed:** corresponde a temperatura média em °C obtida naquele dia.
- **EF:** informa de maneira binaria se no dia correspondente é um feriado ou não (Sim=1 e Não=0).
- **PF:** informa de maneira binaria se no dia anterior ao correspondente houve um feriado ou não (Sim=1 e Não=0).
- **Brl:** informa o brilho solar captado em Wh/m² naquele dia correspondente.

Os modelos utilizam das informações reais a respeito da temperatura e do brilho, de modo, que o foco principal seja exclusivamente a respeito do desempenho das técnicas propostas na previsão de carga. Visto que erros na previsão destas características temporais, podem implicar fortemente na resposta do modelo e tendenciar as análises.

D. Geração das configurações

Nesta etapa foram definidas 100 configurações aleatórias para o *XGBoost* e *Random Forest*. Em torno dos seguintes parâmetros definidos pelas Tabelas II e III. Dos quais foram escolhidos através de testes.

TABELA II : PARAMETRIZAÇÃO XGBOOST.

n_estimator	[10, 100, 200, 300]
base_score	[0.2, 0.5, 0.7, 1.0]
max_depth	[3, 6, 9, 12]
colsample_bytree	[0.2, 0.5, 0.7, 1.0]

Sendo:

- **n_estimator:** Se refere ao numero de árvores de reforço no treinamento. De maneira aleatória o algoritmo seleciona um valor dentro do conjunto pré-estabelecido [10, 100, 200, 300].
- **base_score:** Se refere ao score da previsão inicial para todas as instancias. É selecionado um valor entre [0.2,0.5,0.7,1.0].
- **max_depth:** Refere-se à profundidade máxima da árvore para cada previsor base. O algoritmo seleciona um valor dentro de [3, 6, 9, 12].
- **colsample_bytree:** Refere-se à fração das subamostras das colunas para construir cada árvore. A seleção do valor é feita no intervalo de [0.2, 0.5, 0.7, 1.0].

TABELA III: PARAMETRIZAÇÃO RANDOM FOREST.

n_estimators	[10, 100, 200, 300]
max_depth	[3, 6, 9, 12]
min_impurity_split	[1e-04, 1e-05, 1e-06, 1e-07]

Sendo:

- **n_estimators:** Faz referencia ao numero de arvores que contém na floresta de decisão. De maneira aleatória o algoritmo seleciona um valor dentro do conjunto pré-estabelecido [10, 100, 200, 300].
- **max_depth:** Referencia o valor máximo de profundidade da árvore. É selecionado um valor entre [3, 6, 9, 12].
- ***min_impurity_split:*** É definido um limiar para a parada precoce no crescimento da árvore. Um nó irá dividir se sua impureza estiver acima do limite, caso contrário, é um nó folha. A seleção do valor é feita no intervalo de [1e-04, 1e-05, 1e-06, 1e-07].

Estas configurações servirão de entrada para o algoritmo, juntamente da Tabela I. Que incrementalmente, seleciona a parametrização que melhor ajusta o modelo *(XGBoost* ou *Random Forest*) acoplado a estratégia de previsão em um determinado intervalo de tempo. Assim como pode ser observado na Fig. I.

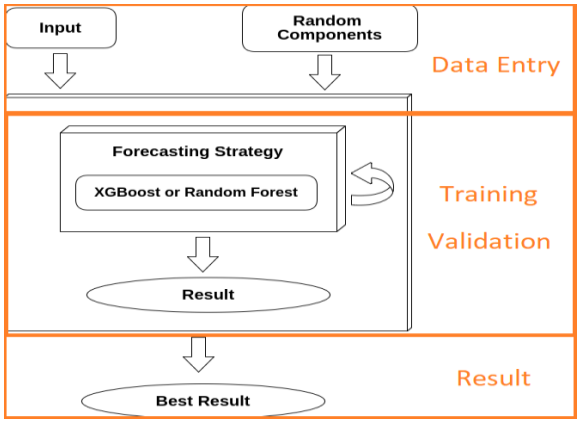


Figura I. Algoritmo proposto.

O algoritmo se resume:

1. **Entrada de dados:** Primeiramente o algoritmo proposto recebe como entrada a tabela de conhecimento, as cem configurações geradas aleatoriamente e o intervalo de tempo que será realizado a previsão, assim como o valor real para validação. Além de que é nessa fase que é informado ao algoritmo a estratégia de previsão e qual algoritmo de máquina que deve ser usado (*XGBoost* ou *Random Forest*).
2. **Treinamento:** Após a entrada dos dados, o algoritmo reestrutura a tabela de conhecimento, de modo que ela contenha informações apenas do dia da semana correspondente ao dia que será avaliado. Por exemplo: Se o dia ser previsto é 01/08/2000, a tabela de conhecimento se reorganizará de modo que apenas os dias que sejam domingos estejam contidos nela, pois dia 01/08/2000 é necessariamente um domingo. E esse procedimento continua até que o intervalo de tempo selecionado seja todo previsto.
3. **Validação:** Após a previsão de um dia. O algoritmo realiza o reajuste dos feriados e pós feriados que o algoritmo acabou de produzir. Cada dia previsto é recalculado pelos coeficientes definidos na Tabela IV, caso sejam dias de feriados ou pós feriados, posteriores a uma semana. Em seguida é utilizado uma métrica de avaliação, a qual será melhor definida na seção E, da previsão do intervalo de tempo definido e armazenado a componente que possuiu o melhor desempenho.
4. **Resultado:** Após todo esse processamento o algoritmo retorna a componente que obteve o melhor desempenho dentro das configurações definidas aleatoriamente.

TABELA IV: COEFICIENTES DE AJUSTE.

coefHoliday	[0.7, 0.8, 0.9, 1.0, 1.07, 1.1, 1.2]
coefAfterHoliday	[0.7, 0.8, 0.9, 1.0, 1.07, 1.1, 1.2]

Sendo,

- **coefHoliday:** É um valor escolhido sequencialmente atribuído em conjunto com as cem configurações definidas na seção D, de modo a recalcular o valor previsto para o feriado. O algoritmo seleciona um valor no intervalo de [0.7, 0.8, 0.9, 1.0, 1.07, 1.10, 1.2]. Dos quais foram pré-estabelecidos a base de testes.
- **coefAfterHoliday:** Similarmente ao coefHoliday, que pode ser exemplificado pela Fig. II. De maneira sequencial o algoritmo seleciona um valor dentro do conjunto pré-estabelecido a base de testes [0.7, 0.8, 0.9, 1.0, 1.07, 1.10, 1.2].

```
if Dp == holiday then
    Dp = Dp * coefHoliday
if Dp == after holiday then
  Dp = Dp * coefAfterHoliday
```

Figura II. Exemplificação do ajuste realizado.

Onde na figura II a variável Dp representa o valor do dia previsto pelo algoritmo.

E. Avaliação e Seleção das componentes

Após a etapa de geração de componentes aleatórias, o algoritmo proposto realiza a avaliação dos modelos previstos a fim de selecionar o mais bem ajustado. Para isso, utilizou -se a métrica de avaliação de desempenho *Mean Absolute Percentage Error* (MAPE), dado pela equação (1):

$$\boldsymbol{MAPE} = \frac{\mathbf{100}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \frac{\boldsymbol{x_i - \hat{x}_i}}{\boldsymbol{x_i}} \quad (1)$$

Sendo, **n** o número de previsões, $\boldsymbol{x_i}$ o valor real e $\boldsymbol{\hat{x}_i}$ o valor previsto do modelo. A partir da utilização desta métrica, será possível obter estimativas mais precisas e plausiveis dos modelos *XGBoost* e *Random Forest* sobre a estratégia proposta. E assim, o algoritmo será capaz de expor o mais efetivo automaticamente.

F. Estratégias de previsão

Para melhor compreender a efetividade dos algoritmos de aprendizado de máquina *XGBoost* e *Random Forest*. Neste artigo, foram propostas duas estratégias de previsão de carga diferente. A principal diferença entre elas está na forma em como é realizado a realimentação do algoritmo para a próxima previsão. Como pode ser exemplificado pela Fig. III.

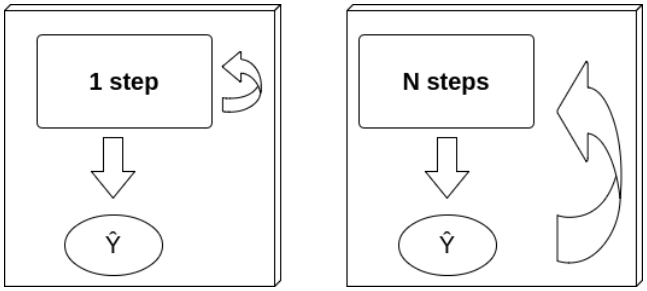


Figura III. Estratégias de previsão

- ***Estratégia 1 step:*** O algoritmo realiza a previsão sempre com o valor real, ou seja, não utiliza informações sobre a previsão que acabou de realizar para prever o próximo dia. Por exemplo, para prever o dia 01/08/2000 ele utiliza dos dados reais que antecedem esse dia. Após essa previsão, o próximo dia (02/08/2000) utiliza também dos dados reais que antecedem essa data.

- ***Estratégia N steps*:** O algoritmo realiza previsão utilizando informação que ele já preveu, ou seja, na segunda previsão ele utiliza a informação da primeira como entrada para prever a segunda. Por exemplo, para prever o dia 01/08/2000 ele utiliza dos dados reais que antecedem esse dia. Para a próxima previsão ele utiliza do resultado que a última previsão gerou e assim consequentemente.

IV. ANÁLISE DOS DADOS

O conjunto de dados utilizados nesse artigo são informações reais de carga elétrica medidas no sudeste do Brasil. O histórico fornecido contém informações horárias do consumo elétrico em kWh que compreende entre 01/04/2000 à 30/04/2001. Contudo a análise será realizada sobre a série de carga média diária deste período de tempo, assim como pode ser visto pela Fig. IV.

O foco principal é verificar a eficiência e desempenho dos modelos definidos na seção anterior. Dando importância em realizar uma análise sobre os dados que serão previstos de antemão. As analises serão realizadas sobre 4 meses diferentes: Junho/2000, Agosto/2000, Novembro/2000 e Janeiro/2001. Na Tabela V é apresentado a estatística descritiva sobre a média diária dos meses que objetiva prever.

TABELA V: ANALISE DESCRITIVA SOBRE A MÉDIA DIÁRIA DOS MESES A SEREM PREVISTOS.

Mês	Min	Max	Media	M_d	CV (%)
Jun/2000	3620.07	5170.89	4730.28	5036.42	11.61
Ago/2000	3691.75	5218.49	4861.84	5128.06	10.90
Nov/2000	3726.47	5427.69	4857.32	5175.31	12.34
Jan/2001	3043.43	5401.11	4719.25	4990.28	13.28

A partir da análise descritiva realizada, percebe que os meses selecionados para a previsão são semelhantes entre si. Ou seja, são bem parecidos em questões de pontos de mínimos, máximo, média e os demais apresentados na tabela. Na Fig. V é apresentado o boxplot de cada mês selecionado para que seja realizado uma análise visual dos dados.

É notório que a partir dos boxplots apresentados pode-se observar que os meses são bem diferentes entre si. O mês de Agosto/2000 possui alguns *outliers*. Todos apresentam assimetria negativa e Janeiro possui uma cauda negativa bem expressiva.

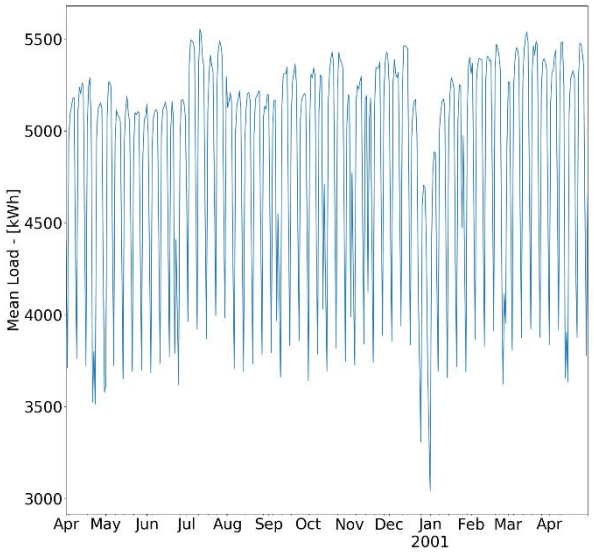


Figura IV. Histórico da série média dos dias compreendidos entre 01/04/2000 à 30/04/2001.

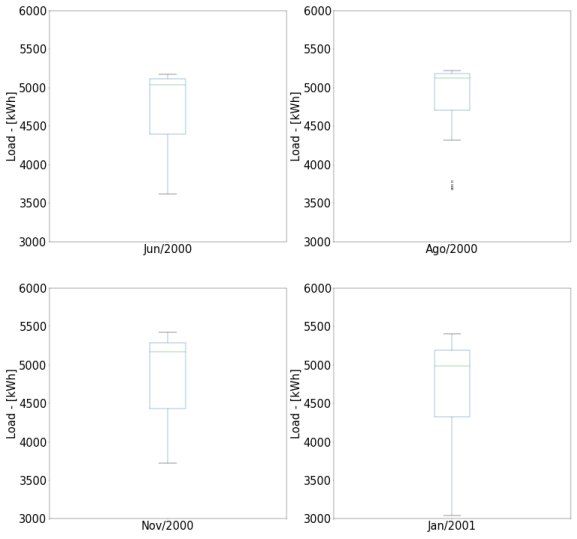


Figura V. Boxplot dos meses selecionados para previsão e análise.

Assim como as informações de carga, dados sobre a temperatura e a incidência solar são informações reais armazenados para cada dia. Na Fig. VI pode ser visto os informes sobre a temperatura máxima, mínima e média real do histórico do sudeste do brasil.

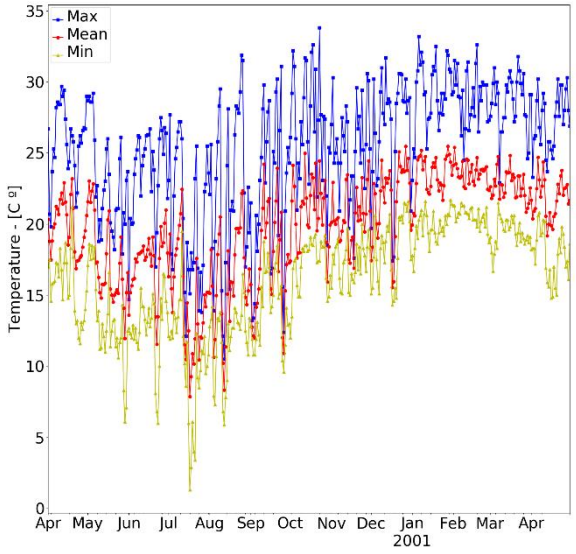


Figura VI. Histórico de temperatura média, mínima e máxima compreendidos entre 01/04/2000 à 30/04/2001.

Estes dados serão usados como características para cada dia, de modo que elas possam ajudar o algoritmo na tarefa de previsão. A partir da Fig. VI pode-se descobrir informações sobre tendências e sazonalidades que a temperatura pode possuir ao longo do tempo.

Outra informação que será utilizada por esse artigo é referente a incidência solar. Na Fig. VII pode ser observado o esse histórico para cada dia.

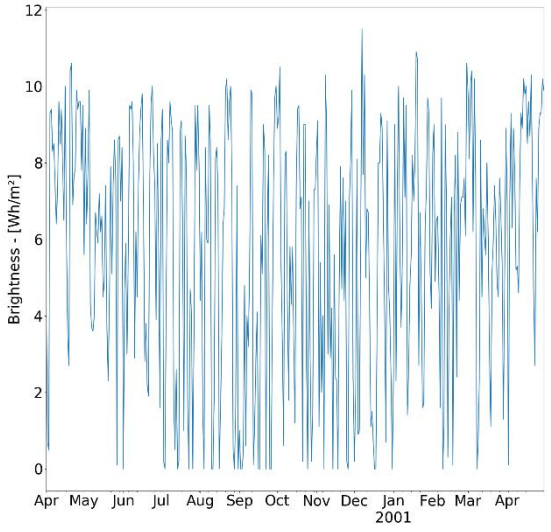


Figura VII. Histórico de incidência solar em Wh/m² diário compreendidos entre 01/04/2000 à 30/04/2001.

Da mesma forma que a temperatura, esse atributo será utilizado no modelo preditivo, de modo a identificar os padrões que podem auxiliar na previsão de carga.

As informações sobre feriado nacionais são importantíssimas nesta análise também. Pois como pode-se observar na Fig. VIII os dados horários de um mesmo dia da semana no mesmo mês e ano possuem praticamente o mesmo padrão de consumo.

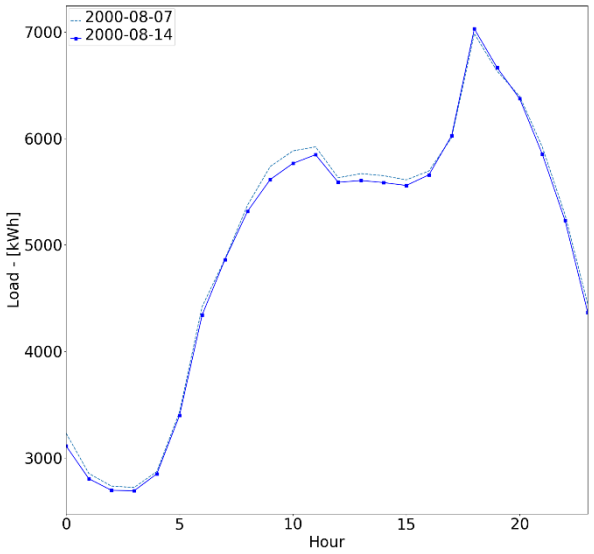


Figura VIII. Consumo horário para as duas primeiras segunda-feiras do mês de agosto de 2000.

Porém, isso não acontece quando naquele determinado dia ocorre um feriado. A partir da Fig. IX, pode-se comprovar essa análise.

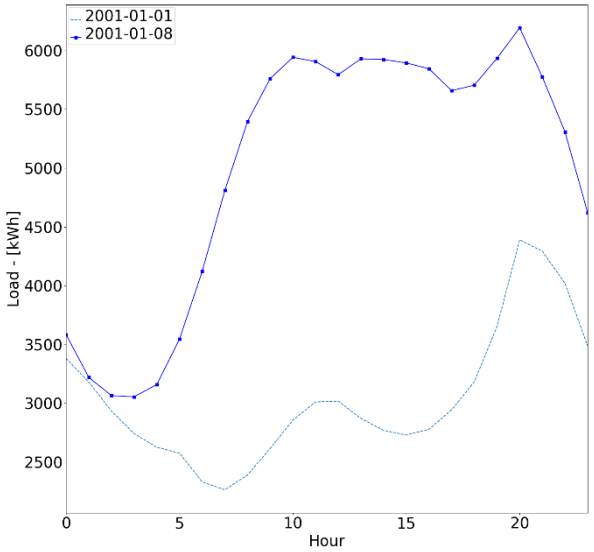


Figura IX. Consumo horário para as duas primeiras segunda-feiras do mês de janeiro de 2001.

Percebe que diferentemente da Fig. VIII as série horarias da Fig. IX são diferentes. Isso acontece porque no dia 01/01/2001 ocorre o feriado da Confraternização Universal, enquanto que no dia 08/01/2001 é um dia normal.

V. RESULTADOS

Como descrito anteriormente o principal objetivo foi prever a carga média diária gerada pelas técnicas *Random Forest* e *XGBoost* sobre duas estratégias de previsão de carga média. Os dados históricos utilizados para ajustar e testar os modelos são formados por informações reais de carga elétrica medida no sudeste do Brasil em uma grande metrópole, no período de 01/04/2000 à 30/04/2001. Assim como as informações de temperatura e brilho solar.

Para avaliação das técnicas utilizadas foi selecionado quatro meses: Junho/2000, Agosto/2000, Novembro/2000 e Janeiro/2001. Para cada um destes períodos, o algoritmo treinado com o histórico das datas que antecedem o dia ser previsto e assim sucessivamente. Completado o mês previsto, é seleciono a configuração que melhor ajusta o modelo dentro das 100 componentes criadas aleatoriamente e produz o resultado a partir desta. Para melhor entender esse processo, as seções G e H mostram a eficiência dos modelos *XGBoost* e *Random Forest* respectivamente, sobre a análise da estratégia 1 step e N steps à frente.

G. *XGBoost*

Após a execução do algoritmo proposto na metodologia. Selecionou-se para cada intervalo de tempo as configurações que melhor se ajustaram e obtiveram um melhor resultado. Como pode ser visto pela Tabela VI e VII.

TABELA VI: ESTRATÉGIA 1-STEP XGBOOST.

Mês	**06/2000**	**08/2000**	**11/2000**	**01/2001**
n_estimator	200	100	200	100
base_score	0.2	0.7	0.2	0.2
max_depth	3	3	3	6
colsample_bytree	0.7	0.2	1.0	0.7

TABELA VII: ESTRATÉGIA N-STEPS XGBOOST.

Mês	**06/2000**	**08/2000**	**11/2000**	**01/2001**
n_estimator	100	100	300	100
base_score	0.2	0.2	0.2	0.2
max_depth	3	6	3	6
colsample_bytree	1.0	0.2	1.0	0.7

Percebe-se nas Tabelas VI e VII, que para cada estratégia de previsão foi selecionado uma configuração diferente para um determinado intervalo de tempo. Na Tabela VIII, pode se destacar o MAPE de cada um destes modelos para cada mês.

Verifica-se que a Tabela VIII, em ambas as estratégias os modelos obtiveram MAPE aproximado. É possível notar também, que para determinados meses, elas obtiveram um melhor resultado.

TABELA VIII: COMPARATIVO 1-STEP E N-STEPS XGBOOST.

Mês	06/2000	08/2000	11/2000	01/2001
1-step	**1.16**	**1.37**	1.62	4.65
n-steps	1.25	1.48	**1.53**	**3.98**

Nas Fig. X, XI, XII e XIII pode-se constatar a eficiência de ambas as estratégias de previsão comparadas ao valor real da série de uma maneira visual. Além de que é possível verificar em qual dia as estratégias tiveram mais dificuldade em realizar a previsão da carga média.

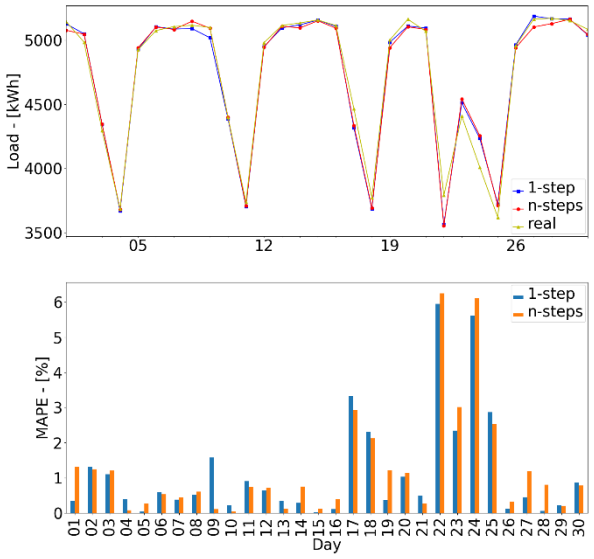


Figura X. XGBoost 1-step e N-steps para Junho de 2000.

Pode-se verificar no gráfico da Fig. X, o dia que obteve o maior MAPE foi o dia 22/06/2000, o qual pode ser justificado pelo feriado de Corpus Christi. No qual o algoritmo não conseguiu tratar de forma eficiente. De modo geral as duas técnicas obtiveram resultados similares.

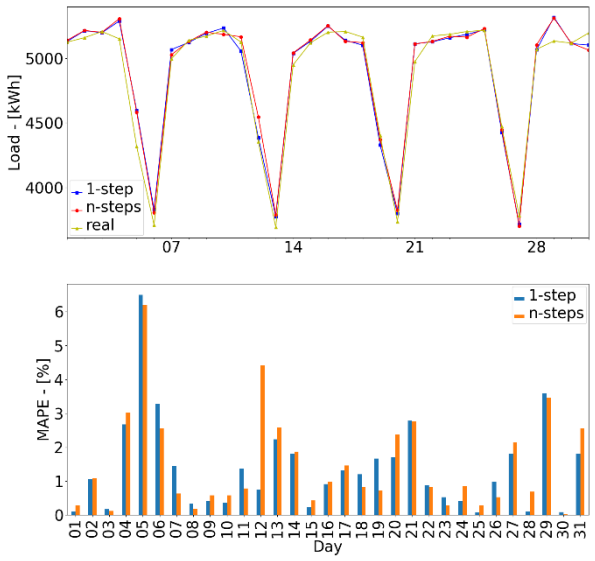
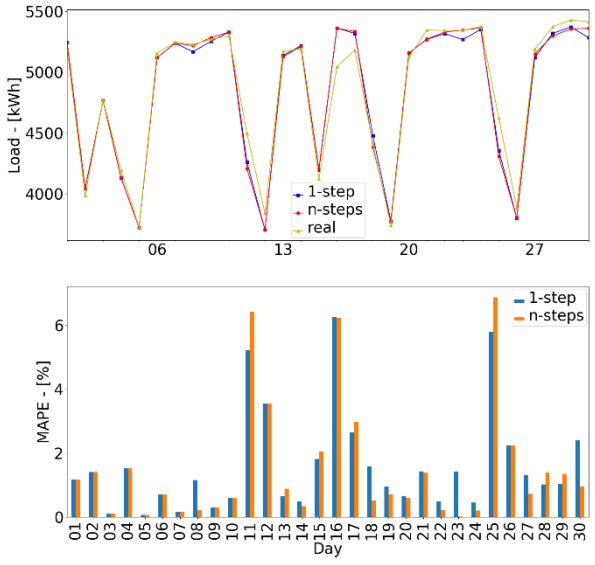


Figura XI. XGBoost 1-step e N-steps para Agosto de 2000.

No mês de agosto de 2000, representado pela Fig. XI, não ocorreu nenhum feriado. Porem os MAPEs com valor alto que ocorrem nesta previsão, podem acontecer devido aos outliers que o mês de Agosto apresentou, no qual pode ser visto na seção IV. É possível observar também que no dia 12/08/2000 a técnica 1-step possuiu maior desempenho comparada a N-steps.



FiguraXII. XGBoost 1-step e N-steps para Novembro de 2000.

No mês de novembro de 2000 os MAPEs mais altos se deram nos dias 16/11/2000 e 25/11/2000 para ambas as estratégias, o algoritmo não funcionou de maneira eficiente nesses dias. Porém, obteve um ótimo resultado nos dias de feriado representados pelo dia 02/11/2000 e 15/11/2000.

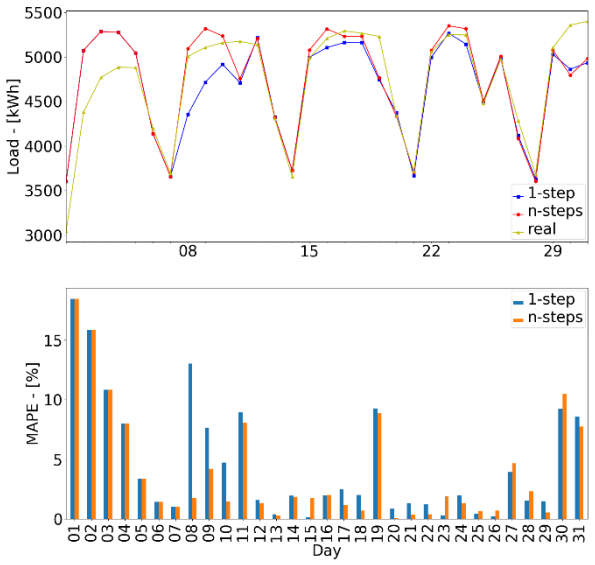


Figura XIII. XGBoost 1-step e N-steps para Janeiro de 2001.

E por último o mês de Janeiro de 2001, que obteve dentre os meses selecionados o MAPE de 3.98 para **N-steps** 4.65 para **1-step.** Isso acontece devido as características deste mês. Que se observar na seção IV, pode se notar que esse mês é considerado mais disperso do que os demais meses selecionados.

H. Random Forest

Similarmente a seção F, tratada sobre o XGBoost, após a execução do algoritmo proposto na metodologia. Expõem-se para cada intervalo de tempo as configurações que melhor se ajustaram e obtiveram um melhor resultado. Como pode ser visto pela Tabela IX e X.

TABELA IX: ESTRATÉGIA 1-STEP RANDOM FOREST.

Mês	06/2000	08/2000	11/2000	01/2001
n_estimator	200	100	100	300
max_depth	3	3	9	3
min_imputity_split	1e-05	1e-05	1e-06	1e-05

TABELA X: ESTRATÉGIA N-STEPS RANDOM FOREST.

Mês	06/2000	08/2000	11/2000	01/2001
n_estimator	200	100	100	300
max_depth	3	3	9	9
min_imputity_split	1e-04	1e-06	1e-06	1e-05

É notório que as parametrizações para cada intervalo de tempo em ambas as estratégias são diferentes. Contudo pode se observar que o parâmetro ***n_estimator*** se manteve como as mesmos valores em ambas as abordagem. Assim como, ***max_depth,*** porém este apenas no mês de janeiro se alterou. Na Tabela IX, pode se destacar o MAPE de cada um destes modelos para cada mês.

A partir da Tabela XI, observa-se que na grande maioria dos meses a estratégia **n-steps** obteve um ajuste levemente superior comparado a **1-step**. Contudo, em média a estrageia 1-step obteve melhor desempenho. É importante ressaltar que na maioria dos intervalos de tempo, ambas as abordagens produziram respostas bem próximas e parcimoniosas.

TABELA XI: COMPARATIVO 1-STEP E N-STEPS RANDOM FOREST

Mês	06/2000	08/2000	11/2000	01/2001
1-step	1.42	**2.08**	1.67	4.16
n-step	**1.41**	3.99	**1.57**	**4.04**

Em sequência nas Fig. XIV, XV, XVI e XVII é possível visualizar o cenário de desempenho de cada estratégia de previsão sobre as componentes selecionadas pelo algoritmo.

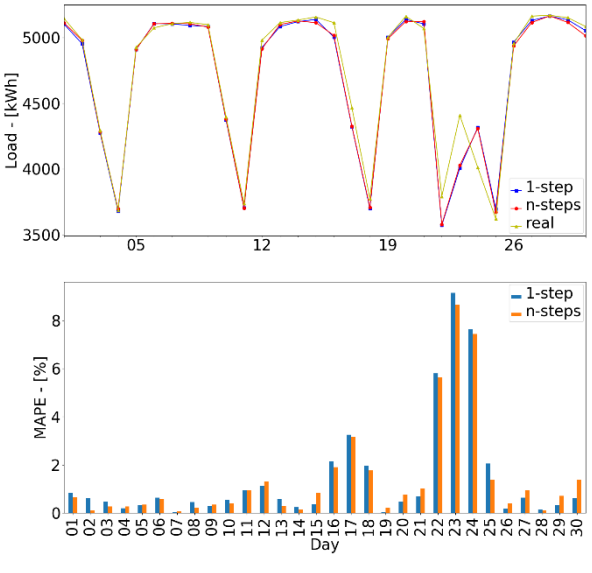


Figura XIV. ***Random Forest*** **1-step e N-steps para Junho de 2000.**

É possível notar que as técnicas 1-step e N-step produziram respostam bem similares utilizando o modelo *Random Forest*. Percebe-se também que o dia em que ocorreu o maior MAPE foi o dia posterior ao feriado do dia 22/06/2000. Na Fig. X percebe que o *XGBoost* teve maior desempenho.

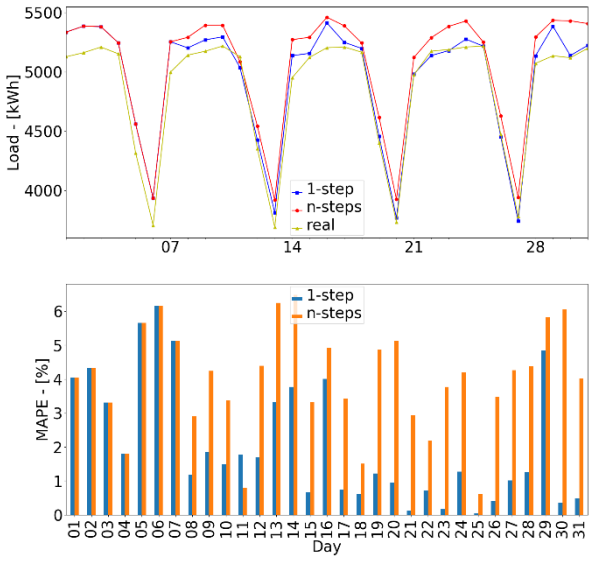


Figura XV. Random Forest 1-step e N-steps para Agosto de 2000.

No mês de Agosto de 2000 é notório que a técnica **1-step** obteve um melhor desempenho na previsão de carga do que a **N-steps** de modo geral. Inicialmente pode se observar também que ambas possuem similaridade nas previsões.

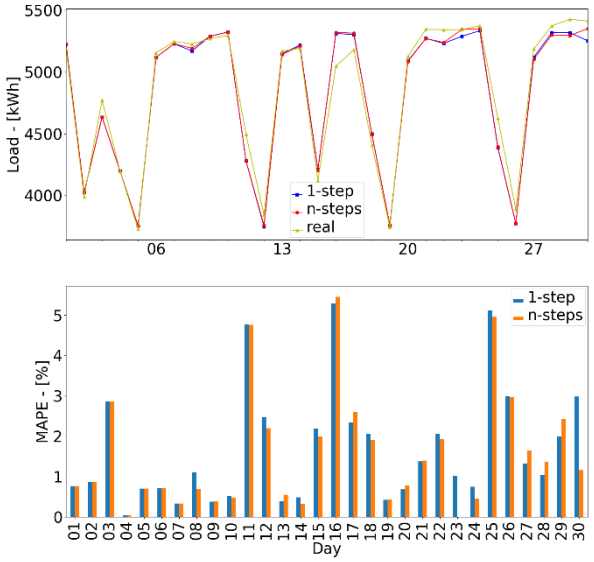


Figura XVI. ***Random Forest*** **1-step e N-steps para Novembro de 2000.**

Semelhantemente ao XGBoost, os MAPEs mais altos se deram nos dias 16/11/2000 e 25/11/2000 para ambas as estratégias de previsão. Obtivendo um menor erro nos feriados do dia 02/11/2000 e 15/11/2000

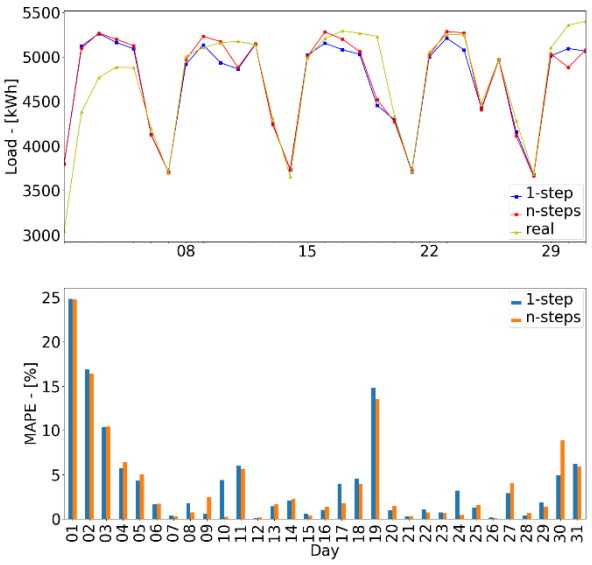


Figura XVII. ***Random Forest*** **1-step e N-steps para Janeiro de 2001.**

No mês de janeiro de 2001 representado pela Fig. XVII, as estratégias de previsão de carga obtiveram similares erros. Igualmente ao XGBoost os algoritmos obtiveram um maior erro no feriado que ocorre no dia 01/01/2001, porém no feriado correspondente ao dia 25/01/2001 eles obtêm um ótimo resultado.

I. Comparativo com a literatura.

De modo a comprovar a eficiências dos modelos expostos pelas seções G e H. O artigo de [22] realizou uma pesquisa similar ao proposto neste trabalho. No qual foi selecionado quatro meses para previsão da série de carga média diária: Jun/2000, Ago/2000, Nov/2000 e Jan/2001. Estes meses foram previstos utilizando as técnicas de previsão 1-step e N-steps juntamente com um modelo de aprendizado de máquina baseado em árvores de regressão, conhecido como Decision Forest. Na Tabela XII pode-se observar os resultados obtidos.

TABELA XII: RESULTADOS OBTIDOS PELO DECISION TREE

Mês	06/2000	08/2000	11/2000	01/2001
1-step	**1.67**	**2.46**	**3.20**	**4.67**
n-step	1.92	6.19	3.76	11.65

Analisando os resultados da Tabela XII. É possível comprovar a eficiência dos modelos *XGBoost* e *Random Forest*, que resultaram em média foram melhores comparados com o de [22], dos quais podem ser vistos pelas Tabelas VIII e XII respectivamente.

VI. CONCLUSÃO

Este artigo apresentou o desempenho e as limitações das estratégias de previsão propostas sobre os algoritmos de aprendizado de máquina baseados em ensemble. Na qual para análise é proposto um algoritmo que seleciona a melhor componente gerada a partir das configurações aleatórias geradas para cada abordagem.

Os resultados mostram que os modelos gerados foram hábeis em relação a previsão de carga, obtendo analises consistentes às disponíveis na literatura, como pode ser observado na seção I.

É possível inferir também que os modelos que utilizaram o *XGBoost* em geral obtiveram um maior desempenho comparado aos que utilizaram o *Random Forest*.

De certo modo verificou -se que os resultados obtidos pelas estratégias de previsão 1-step e N-steps, mesmo com falta do histórico recente, foi capaz de estimar a carga com precisão e segurança. Além de que é importante salientar que os resultados obtidos nesse artigo foram melhores do que os propostos por [22] em ambos os algoritmos *XGBoost* e *Random Forest*. Para ambas, na estratégia 1-step houve redução de em média de 0.73% de erro e para N-steps houve redução de 4.79% de erro comparado com os resultados de [22].

AGRADECIMENTOS

Os autores deste trabalho gostariam de agradecer ao Laboratório de Inteligência Computacional (LInC) para disponibilização dos recursos computacionais necessários para a execução dos experimentos e elaboração desse artigo e a bolsa da Probic que auxiliou no desenvolvimento deste.

REFERÊNCIAS

[1] R. M. Salgado, J. J. Pereira, T. Ohishi, R. Ballini, C. Lima, and F. J. Von Zuben, “A hybrid ensemble model applied to the short-term load forecasting problem,” in The 2006 IEEE International Joint Conference on Neural Network Proceedings. IEEE, 2006, pp. 2627–2634.

[2] S. Santoso, H. W. Beaty, R. C. Dugan, and M. M. McGranaghan, Electrical Power System Quality. McGraw-Hill Professional, 2002, vol. 2.

[3] R. Zhang, Z. Y. Dong, Y. Xu, K. Meng, and K. P. Wong, “Short-term load forecasting of australian national electricity market by an ensemble model of extreme learning machine,” IET Generation, Transmission & Distribution, vol. 7, no. 4, pp. 391–397, 2013.

[4] D. Bunn and E. D. Farmer, “Comparative models for electrical load forecasting,” 1985.

[5] J. Xie, T. Hong, and J. Stroud, “Long-term retail energy forecasting with consideration of residential customer attrition,” IEEE Transactions on Smart Grid, vol. 6, no. 5, pp. 2245–2252, 2015.

[6] E. Pelikán, “Middle-term electric load forecasting by time series decomposition,” Electricity Load Forecast using Intelligent Technologies, pp. 167–176, 2002.

[7] R. Zhang, Z. Y. Dong, Y. Xu, K. Meng, and K. P. Wong, “Short-term load forecasting of australian national electricity market by an ensemble model of extreme learning machine,” IET Generation, Transmission & Distribution, vol. 7, no. 4, pp. 391–397, 2013.

[8] Q. Duan, J. Liu, and D. Zhao, “Short term electric load forecasting using an automated system of model choice,” International Journal of Electrical Power & Energy Systems, vol. 91, pp. 92–100, 2017.

[9] Z. Shen, X. Wu, J. M. Guerrero, and Y. Song, “Model-independent approach for short-term electric load forecasting with guaranteed error convergence,” IET Control Theory & Applications, 2016
[10] L.-G. Chen, H.-D. Chiang, N. Dong, and R.-P. Liu, “Group-based chaos genetic algorithm and non-linear ensemble of neural networks for shortterm load forecasting,” IET Generation, Transmission & Distribution, vol. 10, no. 6, pp. 1440–1447, 2016
[11] P. M. Macaira, R. C. Sousa, and F. L. C. Oliveira, “Forecasting brazil’s electricity consumption with pegels exponential smoothing techniques,” IEEE Latin America Transactions, vol. 14, no. 3, pp. 1252–1258, 2016.
[12] S.-J. Huang and K.-R. Shih, “Short-term load forecasting via arma model identification including non-gaussian process considerations,” IEEE Transactions on Power Systems, vol. 18, no. 2, pp. 673–679, 2003.
[13] M. T. Hagan and S. M. Behr, “The time series approach to short term load forecasting,” IEEE Transactions on Power Systems, vol. 2, no. 3, pp. 785–791, 1987
[14] S. Haykin, Neural Networks and Learning Machines. Pearson Education, 2011.
[15] C. Cortes and V. Vapnik, “Support vector networks,” Machine Learning, pp. 273–297, 1995.
[16] C. González, J. Mira-McWilliams, and I. Juárez, “Important variable assessment and electricity price forecasting based on regression tree models: classification and regression trees, bagging and random forests,” IET Generation, Transmission & Distribution, vol. 9, no. 11, pp. 1120– 1128, 2015.
[17] M. Korolev and K. Ruegg, “Gradient boosted trees to predict store sales.”
[18] S. Li, P. Wang, and L. Goel, “A novel wavelet-based ensemble method for short-term load forecasting with hybrid neural networks and feature selection,” IEEE Transactions on Power Systems, vol. 31, no. 3, pp. 1788–1798, 2016.
[19] W.-M. Lin, C.-S. Tu, R.-F. Yang, and M.-T. Tsai, “Particle swarm optimisation aided least-square support vector machine for load forecast with spikes,” IET Generation, Transmission & Distribution, vol. 10, no. 5, pp. 1145–1153, 2016.
[20] C. González, J. Mira-McWilliams, and I. Juárez, “Important variable assessment and electricity price forecasting based on regression tree models: classification and regression trees, bagging and random forests,” IET Generation, Transmission & Distribution, vol. 9, no. 11, pp. 1120– 1128, 2015.
[21] J. Ma and J. C. Cheng, “Identifying the influential features on the regional energy use intensity of residential buildings based on random forests,” Applied Energy, vol. 183, pp. 193–201, 2016.
[22] R. M. Salgado and R. R. Lemes, “A hybrid approach to the load forecasting based on decision trees,” Journal of Control, Automation and Electrical Systems, vol. 24, no. 6, pp. 854–862, 2013.
[23] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” Annals of statistics, pp. 1189–1232, 2001.
[24] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” Annals of statistics, pp. 1189–1232, 2001.
[25] M. Korolev and K. Ruegg, “Gradient boosted trees to predict store sales.”
[26] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324
[27] L. Breiman, “Bagging predictors,” Machine Learning, vol. 24, no. 2, pp. 123–140, 1996. [Online]. Available: http://dx.doi.org/10.1023/A: 1018054314350
[28] D. Tao, X. Tang, X. Li, and X. Wu, “Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval,” IEEE transactions on pattern analysis and machine intelligence, vol. 28, no. 7, pp. 1088–1099, 2006.