FILTRAGEM AUTOMÁTICA DE DADOS DE PRODUTIVIDADE DE CANA-DE-AÇÚCAR

Eudocio Rafael Otavio da Silva [1]; Marcelo Chan Fu Wei [1]; Ricardo Canal Filho [1]; José Paulo Molin [2]

[1]Doutorando em Engenharia de Sistemas Agrícolas. Av. Pádua Dias, 235 - Agronomia, Piracicaba - SP, 13418-900. Escola Superior de Agricultura Luiz de Queiroz, Universidade de São Paulo; [2]Professor Titular. Av. Pádua Dias, 235 - Agronomia, Piracicaba - SP, 13418-900. Escola Superior de Agricultura Luiz de Queiroz, Universidade de São Paulo

RESUMO

O monitoramento da produtividade da cana-de-açúcar em alta resolução espacial gera grande volume de dados, que incorporam ruídos intrínsecos à operação de colheita, sendo necessário removê-los. Os métodos de filtragem existentes ainda necessitam de intervenções humanas, impedindo sua automatização para uso nos grandes bancos de dados gerados atualmente. O presente estudo propõe uma técnica de filtragem automática de dados de colheita em canaviais visando automatizar o processo e aumentar a eficiência computacional. Foi desenvolvido um filtro baseado em algoritmo de janela deslizante implementado a um método de filtragem estatístico. Dois estudos de caso com dados reais de lavoura foram apresentados para demonstrar a eficácia do filtro proposto em detectar *outliers* de diferentes magnitudes (MO), comparados à filtragem por outro método da literatura. *Outliers* da produtividade de cana-de-açúcar com MO ≥ ± 0,50 foram quase 100,00% detectados e cerca de 40,00% dos *outliers* artificias com MO de ± 0,01 foram identificados. O algoritmo proposto melhor preservou a variabilidade espacial da cultura em comparação ao método comparativo. Ainda, foram equivalentes na identificação de regiões com diferentes potenciais produtivos da cana-de-açúcar, refletindo a variabilidade existente. Portanto, o método proposto demonstra potencial para a filtragem de dados de colheitas em canaviais.

PALAVRAS-CHAVE: aprendizado de máquina; janela deslizante; *outlier*

INTRODUÇÃO

A cana-de-açúcar é cultivada em fileiras, colhida a cada uma ou duas fileiras e produz elevada biomassa com variações em curtas distâncias entre plantas e fileiras (MALDANER et al., 2022a). O monitoramento da biomassa na colheita mecanizada tem sido realizado por monitores de produtividade. Houve avanços substanciais nos últimos anos, como dados de produtividade em alta resolução espacial obtidos pela redistribuição da produção do canavial pelo dado de pressão de propulsão no picador da colhedora devidamente georreferenciado (MALDANER et al., 2022b). Este sistema gera volumosos conjuntos de dados que precisam ser processados para servirem como uma camada de informação na tomada de decisão agrícola. Entretanto, podem ocorrer erros nesses dados devido a diversos fatores, como tempo de posicionamento, problemas no controle de tráfego na fileira da cana, que pode provocar desalinhamento da colhedora, dificuldades no processo de alimentação da colhedora devido ao relevo, erros de leituras nos sensores embarcados na colhedora, paradas para manutenção da máquina em campo, problemas de *hardware* e *software* do computador de bordo, manobras de cabeceira, entre outros fatores (SPEKKEN et al., 2015; BRAUNBECK & OLIVEIRA, 2006). Logo, estes dados precisam ser filtrados para posterior utilização. A decisão de um analista de dados sobre quais dados devem ser removidos é complexa, e utiliza como suporte detectores de valores discrepantes (ou *outliers*, em inglês). Um *outlier* é uma observação que se desvia consideravelmente das demais no conjunto de dados. No entanto, nem todo *outlier* é um erro, pois determinados valores discrepantes podem apontar um fenômeno não caracterizado no conjunto de dados (MOKOENA et al., 2022). Os desafios da filtragem de dados para o contexto da agricultura são: (a) leis que regem a espacialidade precisam ser consideradas na análise da estrutura espacial de uma variável (e.g., Lei de Tobler) e critérios da matriz de vizinhança (e.g., contiguidade) (TOBLER, 1970); (b) os valores discrepantes podem afetar suposições de modelos estatísticos e geostatísticos; (c) ausência de estratégia unificada para detectar *outliers* (SMITI, 2020); (d) ausência de referência verdadeira sobre o *outlier* (LIU et al., 2021); (e) as

configurações de limite do filtro não são conhecidas; e (f) a maioria dos métodos de filtragem de dados exigem parâmetros de entrada ajustáveis pelo usuário, inserindo subjetividade no processo (SOUIDEN et al., 2022), o que demanda automação na análise de dados. Por exemplo, o procedimento de filtragem no programa MapFilter 2.0 (MALDANER et al., 2022a) requer que o usuário insira o parâmetro de variação de limite nas etapas de filtragem global e local, e requer também a inserção manual do valor da dependência espacial da variável a ser filtrada. Logo, observou-se a necessidade de otimizar o procedimento para as especificidades da cana, propondo a automatização dos processos.

OBJETIVOS

O presente estudo propõe uma técnica de filtragem automática de dados de colheita em canaviais visando automatizar o processo e aumentar a eficiência computacional.

MATERIAL E MÉTODOS

Área de estudo e conjunto de dados

O estudo foi conduzido em dois canaviais comerciais no estado de São Paulo, Brasil, identificados como conjunto de dados 1 e 2. O conjunto de dados 1 abrange 10 talhões, totalizando 208,78 hectares, enquanto o conjunto de dados 2 é composto por 6 talhões com uma área total de 52,98 hectares (Tabela 1).

TABELA 1. Caracterização do conjunto de dados brutos para os conjuntos de dados 1 e 2.

Conjunto de dados	nº talhões	Área (ha)	nº pontos	Densidade (pontos ha^{-1})	Média prod. (Mg ha^{-1})
1	10	208,78	217521	1041,87	69,29
2	6	52,98	54281	1204,56	37,30

nº talhões: número de talhões; nº pontos: número de pontos; Média prod.: média da produtividade.

Neste estudo, a colheita foi realizada por colhedoras de cana, que cortam e processam uma única fileira por vez, que predomina no mercado. Os registros contíguos dos dados de produtividade de cana foram realizados durante o deslocamento da máquina ao longo da fileira de cana e estes dados foram armazenados em um computador de bordo integrado à colhedora. O mapeamento da produtividade foi realizado com um sistema comercial (Solinftec, Araçatuba, São Paulo, Brasil) baseado nas variações da propulsão hidráulica do sistema picador de colmos da colhedora (MALDANER et al., 2022b). A resolução espacial e a frequência de coleta dos dados de colheita foram determinadas pela variação de pressão do picador de colmos da colhedora. Neste estudo, os dados de colheita foram registrados com receptor GNSS com uma frequência de 0,20 Hz. Os dados possuem diferentes colunas e suas respectivas linhas com variáveis distintas. Os dados utilizados para o filtro proposto foram: coordenadas de latitude e longitude, produtividade e tempo de colheita. Assim, estes dados brutos foram transformados em uma matriz de dados. Ao executar o algoritmo, uma coluna da matriz foi selecionada e cada um dos registros de dados de colheita nas linhas correspondeu a um elemento dentro da janela deslizante (JD), com o tamanho da janela definido (janela deslizante univariada). Logo, se o tamanho da JD fosse igual a cinco, por exemplo, isso corresponderia ao registro de cinco pontos de colheita obtidos na lavoura e armazenados no arquivo, como mostra o esquema de obtenção da JD na colheita da cana (Figura 1).

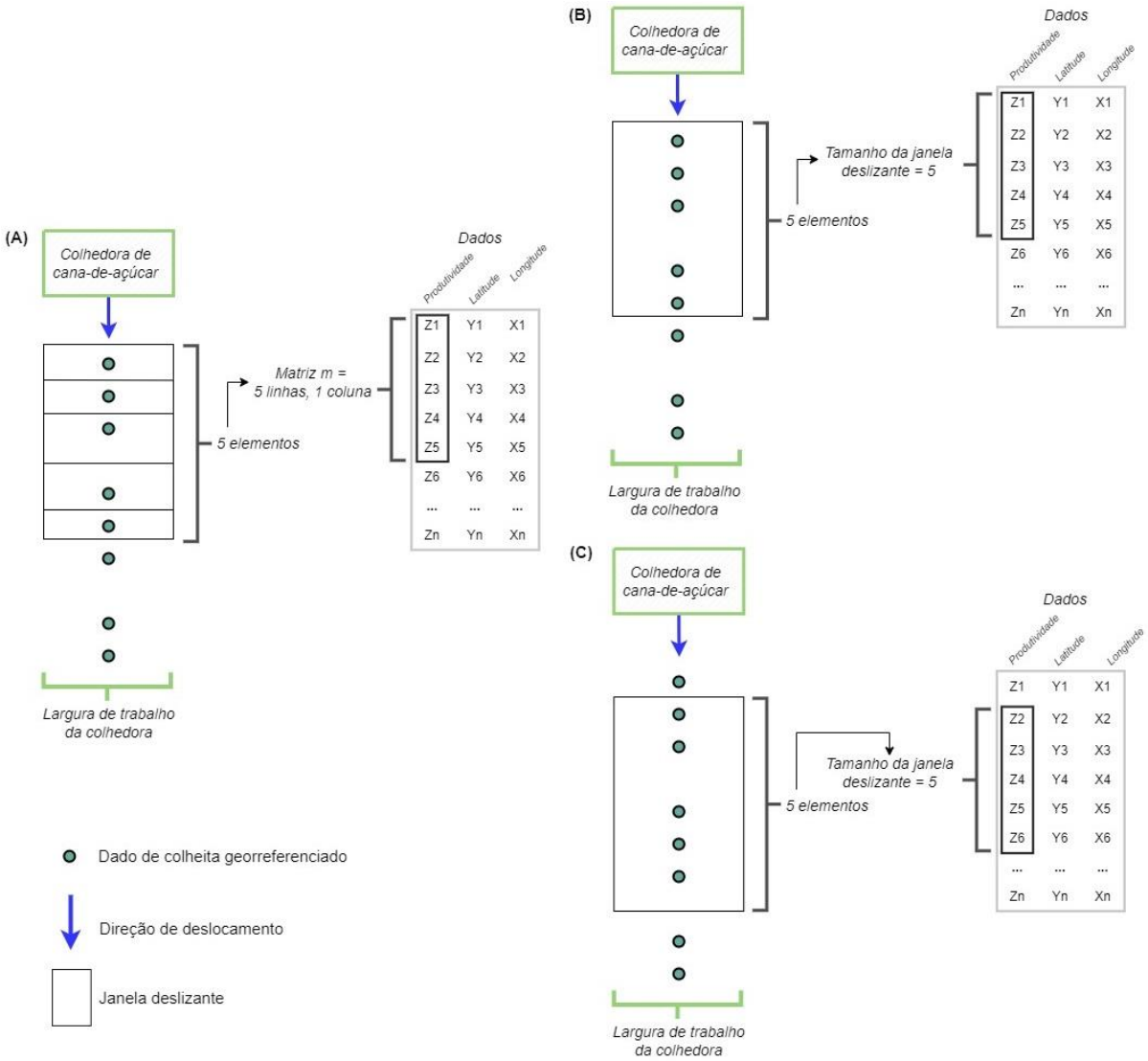

FIGURA 1. Esquema de obtenção da janela deslizante na colheita da cana-de-açúcar. Matriz de dados (A), construir a janela inicial (B), deslizar a janela de uma sublista para a próxima e deslizar repetidamente até o final da lista (C).

Desenvolvimento do algoritmo de janela deslizante

Uma janela é uma subsequência entre i-ésimo e o j-ésimo itens recebidos, denotado como W[*i*, *j*] = (x_i, x_{i+1},..., x_j), em que *i* < *j*, sendo *i* e *j* os itens na sequência de uma lista. No modelo de janela deslizante (JD), a janela W[*p* - *w* + 1, *p*], em que *p* é ponto de contagem atual dentro da janela e *w* é o tamanho da janela, sempre que uma nova instância chega, a JD é enfileirada em uma estrutura de dados de política FIFO (do inglês, *first-in*, *first-out*), na qual a mais antiga é descartada (SOUIDEN et al., 2022). O algoritmo de JD opera com base em determinados requisitos: uma matriz como entrada; elementos contíguos; uma janela que representa um intervalo de elementos; um estado para ser mantido nessa janela, neste caso, o filtro estabelecido para a filtragem de dados; e a iteração completa da matriz. No presente estudo, o algoritmo de JD foi desenvolvido e aplicado utilizando-se matrizes, no qual em sua execução os dados são atualizados constantemente dentro da janela, sendo gerados conjuntos de dados subjacentes a serem aplicados na filtragem. Zeng et al. (2023) destacam que o modelo de janela deslizante pode ser de dois tipos: baseado em contagem e baseado em tempo. No contexto deste estudo, a JD foi baseada em contagem, o que significa que sempre conterá um número fixo de dados que representam o tamanho da JD. O método de filtragem é baseado em estatística, utilizando a

mediana do conjunto de dados. Neste método, estabelece-se o limite superior (LS) e inferior (LI) correspondente ao intervalo de valores considerados representativos do conjunto de dados (do inglês, *inliers*). Valores acima e abaixo do LS e LI são considerados discrepantes e podem ser removidos do conjunto de dados (Equações 1 e 2).

$$Limite\ superior\ (LS) = Med_i + Med_i \times f \quad \text{Eq. (1)}$$

$$Limite\ inferior\ (LI) = Med_i - Med_i \times f \quad \text{Eq. (2)}$$

em que: Med_i - mediana dos valores localizados dentro da janela deslizante; e *f* - fator de variação máximo aceita para a mediana.

O algoritmo de filtragem automatizado proposto constitui-se de três etapas, representadas por *i*, *ii* e *iii* (Figura 2). As etapas *i* e *iii* tem como objetivo destrinçar os dados e a etapa *ii* visa detectar *outliers* com um valor fixo de limiar de dispersão.

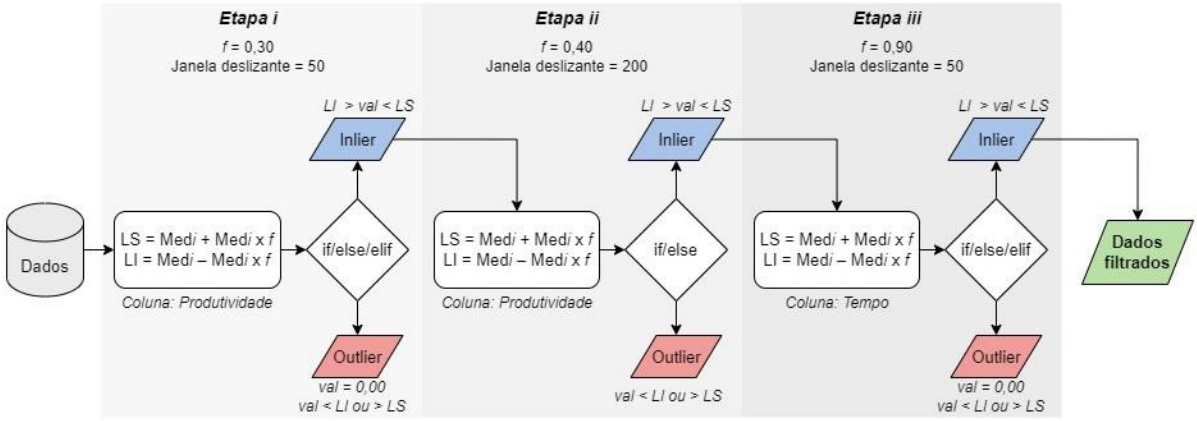

FIGURA 2. Estrutura do algoritmo de filtragem proposto. Med_i: mediana dos valores localizados dentro da janela deslizante; *f*: fator de variação máximo aceito para a mediana; LI: limite inferior; LS: limite superior; val: valor.

Etapa i: Automaticamente, com o fator *f* igual 0,30 e o tamanho da JD igual a 50 a coluna "produtividade" do conjunto de dados é selecionada para a realização da filtragem. Os valores de limites superior e inferior são obtidos e são variáveis ao longo da execução do algoritmo, sendo esta variação devido ao tamanho da JD, fator *f* e aos valores do conjunto de dados dentro da JD. Na execução do algoritmo, a janela desliza sob a matriz de dados e realiza a operação de filtragem, verificando os dados um a um. A cada momento que a janela desliza na matriz de dados, novos limites superior e inferior são obtidos e a filtragem do subconjunto de dados é realizada e cada valor é verificado se corresponde a um valor presente no intervalo do limite superior e inferior. Se o valor estiver dentro do intervalo de filtragem, corresponde a um *inlier* e é mantido no conjunto de dados (*if*); entretanto, se o valor não estiver dentro do limite inferior e superior, é considerado um dado discrepante e é removido do conjunto de dados (*else*). Se o dado de produtividade for igual a zero (0) também é considerado um dado discrepante e é removido (*elif*). Esta operação é finalizada ao iterar em toda a coluna de produtividade da matriz de dados. Ao final da etapa *i* é obtido um conjunto de dados filtrados que é automaticamente inserido na etapa *ii*.

Etapa ii: Os dados filtrados na etapa anterior são transformados novamente em uma matriz de dados e a coluna 'produtividade' é selecionada novamente para a filtragem dos dados. O tamanho da JD é igual a 200 e fator *f* igual a 0,40. Assim como a etapa *i*, a estrutura condicional *if* e *else* ocorre enquanto a janela desliza e realiza a filtragem do conjunto de dados. A etapa *ii* termina ao iterar toda a matriz de dados e o conjunto de dados filtrados são automaticamente inseridos na etapa *iii*.

Etapa iii: A coluna "tempo" é selecionada e filtrada. Esta coluna refere-se ao tempo de armazenamento da coleta de um dado em um ponto na lavoura. O tamanho da JD corresponde a 50 e o fator *f* nesta etapa é igual a 0,90. A estrutura condicional *if, else* e *elif* ocorre enquanto a janela desliza na matriz de

dados e realiza a filtragem do conjunto. Ao final desta etapa, são geradas como saída o conjunto de dados filtrados da colhedora, a estatística descritiva destes dados e o mapa de produtividade da cana dos campos analisados.

Validação

Para avaliar o desempenho do método proposto, foram inseridos artificialmente valores discrepantes nos conjuntos de dados. Desta maneira, a avaliação do desempenho na detecção de valores discrepantes baseou-se no número de *outliers* artificiais de diferentes magnitudes detectados, uma vez que o número de *outliers* introduzidos é conhecido (LIU et al., 2021). Para isso, foram selecionados aleatoriamente diferentes elementos nos conjuntos de dados, a partir dos quais *outliers* artificiais foram gerados. Nos pontos selecionados, os valores de produtividade foram submetidos a diferentes magnitudes de *outliers* (MO) (LIU et al., 2021), sendo MO = ± 0,01, ± 0,05, ± 0,10, ± 0,50, ± 0,80 e ± 1,00, que representam *outliers* de pequena, média e grande magnitude (Equação 3).

$$Z_0 = Z + MO \times Z \quad \text{Eq. (3)}$$

em que: Z - valor original de um elemento; Z_0 - valor do *outlier* artificial; e MO - magnitude do *outlier*, sendo ± 0,01, ± 0,05, ± 0,10, ± 0,50, ± 0,80 e ± 1,00.

Os conjuntos de dados 1 e 2 foram contaminados com 4944 e 1464 valores discrepantes artificiais, representando cerca de 2,30 e 2,70% do número total de dados, respectivamente. A partir disso, avaliou-se o desempenho na detecção de *outliers* artificiais para o algoritmo proposto comparando com o MapFilter 2.0 (MALDANER et al., 2022a), um método de filtragem de dados de alta densidade em lavouras que emprega filtragem global (a mediana do conjunto de dados para calcular os limites inferior e superior) e filtragem local (limites inferior e superior são determinados a partir da mediana dos dados localizados em uma faixa de raio ao redor de um ponto de forma aniso e isotrópica, ou seja, no sentido da fileira de cultivo e não considerando a disposição das fileiras, respectivamente). Os parâmetros inseridos no MapFilter 2.0 utilizados no presente estudo foram: variação de limite igual 30,00% (filtragem global), dependência espacial igual a 200,00 m e variação de limite correspondente a 30,00% (filtragem local). Foram analisadas as estatísticas descritivas dos dados submetidos ao método proposto e ao método desenvolvido por Maldaner et al. (2022a). Os dados filtrados foram espacializados utilizando o Sistema de Informação Geográfica (SIG) QGIS v. 3.22.10. Os *outliers* removidos foram identificados quanto às operações de deslocamento, parada e manobra da colhedora no campo e espacializados. O desenvolvimento do algoritmo e todas as análises desenvolvidas neste estudo foram realizados por meio do ambiente virtual *JupyterLab*, utilizando a linguagem de programação Python v. 3.10.5 (KLUYVER et al., 2016; PYTHON, 2024).

RESULTADOS E DISCUSSÃO

Foram identificados valores discrepantes de 59,67% e 59,11% no conjunto de dados 1 e 42,45% e 41,17% no conjunto de dados 2 pelo algoritmo proposto neste trabalho e pelo método comparativo, respectivamente. O MapFilter 2.0 removeu uma quantidade relativamente menor de pontos (Tabela 2). A redução de dados após o processo de filtragem é uma observação comum em limpezas desse tipo de dados. Em Maldaner & Molin (2020), utilizando o MapFilter 2.0 para filtrar conjuntos de dados de monitores de produtividade de colhedoras de cana, foram removidos valores que alcançaram até 40,00% do conjunto de dados, caracterizados como discrepantes. Mesmo após a filtragem, a densidade de pontos pelo método proposto é considerável, com valores iguais a 87723 e 31238 pontos, representando cerca de 420 e 590 pontos ha^{-1} nos conjuntos de dados 1 e 2, respectivamente. A execução das três etapas do algoritmo demandou um tempo de processamento de 91,49 s e 20,68 s para os conjuntos de dados 1 e 2. Esses valores estão diretamente relacionados à premissa do algoritmo de janela deslizante, que visa minimizar a complexidade de tempo da operação (ZENG et al., 2023).

TABELA 2. Estatísticas da produtividade para os dados brutos e pelos métodos de filtragem nos conjuntos de dados 1 e 2.

Conjunto de dados	Método	n	% dados removidos	média	min	max	DP	CV (%)
				Mg ha^{-1}				
1	Dados brutos	217521	-	69,29	0,00	752,84	42,64	61,54
	Algoritmo proposto	87723	59,67	83,88	29,04	178,05	18,66	49,63
	MapFilter 2.0	88940	59,11	79,75	53,77	99,83	11,26	14,12
2	Dados brutos	54281	-	37,30	0,00	647,30	22,88	61,34
	Algoritmo proposto	31238	42,45	47,29	1,39	126,18	7,87	30,99
	MapFilter 2.0	31934	41,17	47,42	31,73	58,31	5,62	11,86

n: número de dados; min: mínimo; max: máximo; DP: Desvio padrão; CV: Coeficiente de variação.

O algoritmo de JD preservou maior amplitude de valores de produtividade, indicados pelos valores mínimos e máximos, e demonstrou maior variabilidade dos dados nos canaviais em comparação com o MapFilter 2.0, evidenciado pelos valores de CV e DP. Isso é relevante, pois a cana-de-açúcar apresenta alta variabilidade de biomassa em curtas distâncias (MALDANER et al., 2022a), entre fileiras e entre plantas, sugerindo que o algoritmo de JD conseguiu lidar com a variabilidade da produtividade nos canaviais. Em relação aos dados médios, o filtro proposto apresentou valores superiores, porém próximos aos valores do filtro comparativo. Esses resultados estão associados às maiores amplitudes de dados obtidas. O algoritmo demonstrou habilidade na identificação de valores discrepantes, alinhando-se de maneira consistente com uma avaliação manual, especialmente no que diz respeito aos dados de produtividade que extrapolam o padrão de produção para a cana-de-açúcar (por exemplo, valores superiores a 300,00 Mg ha^{-1}, assim como valores de 752,84 e 647,30 Mg ha^{-1} verificados nos conjuntos de dados 1 e 2, respectivamente) e aos registros de tempo de armazenamento da informação de colheita no computador de bordo da colhedora (com valores acima de 20,00 s, por exemplo). Isso é relevante considerando-se que o tempo médio de coleta de dados é de 5 segundos (frequência de 0,20 Hz). Verifica-se que os valores discrepantes em determinados intervalos, como os valores exemplificados anteriormente, coincidem com aquilo que o olho humano identificaria como *outliers*, o que aumenta a confiança do filtro proposto para automação de tarefas morosas e repetitivas. Uma vantagem do método proposto em relação ao MapFilter 2.0 é que não requer a inserção de valores pelo usuário como parâmetros de entrada para a filtragem de dados, automatizando assim o processo e solucionando a problemática comum de diversos métodos de filtragem, a subjetividade (SOUIDEN et al., 2022). Além disso, o filtro proposto apresenta tempo de processamento otimizado, pois além do tempo de execução do algoritmo ser relativamente curto, a não necessidade de conhecer previamente o banco de dados da colheita da cana-de-açúcar para o estabelecimento de parâmetros a serem inseridos manualmente para realizar a filtragem poupa o tempo do usuário que analisa esses dados. Os resultados dos testes para detectar os valores discrepantes artificiais nos dados de colheita da cana são apresentados na Tabela 3.

TABELA 3. Desempenho de detecção de *outliers* artificiais (%) pelo método de filtragem proposto e pelo MapFilter 2.0.

Conjunto de dados	Método	Magnitude do *outlier*					
		± 0,01	± 0,05	± 0,10	± 0,50	± 0,80	± 1,00
1	Algoritmo proposto	48,79	46,72	45,75	89,44	96,60	97,33
	MapFilter 2.0	100,00	100,00	99,27	98,18	94,54	99,88

2	Algoritmo proposto	31,15	30,74	28,28	92,21	96,72	97,54
	MapFilter 2.0	100,00	100,00	98,36	97,95	100,00	99,59

Verificou-se, a partir da Tabela 3, que para MO ≥ ± 0,50, o desempenho de detecção de *outliers* artificiais do método proposto e do MapFilter 2.0 apresentaram valores próximos a 100,00%. Para MO = ± 0,01, ± 0,05 e ± 0,10, o desempenho foi reduzido nesta ordem. O método proposto exibiu degradação de desempenho em comparação com o método comparativo, MapFilter 2.0. Isso não é desejado, entretanto faz sentido ao considerar que o método proposto apresenta variação nos limites da JD quanto ao que é considerado *outlier* e *inlier*, à medida que a janela desliza no arquivo de dados. Portanto, para os conjuntos de dados 1 e 2, os valores de menor magnitude tornaram-se mais difíceis de serem detectados, uma vez que o limiar de *outlier* depende do subconjunto de dados gerado na JD. O MapFilter 2.0 detectou 100,00% de *outliers* artificiais de menor magnitude, evidenciando a sua robustez. No entanto, observou-se que o mesmo apresentou intervalos de valores da produtividade de menor amplitude e menor variação de limites no conjunto de dados comparado ao método de JD proposto. Isto pode não ser tão favorável para a filtragem de dados, uma vez que há o risco de eliminar dados legítimos que, embora correspondam a dados reais do campo, não se encontram nos limites estabelecidos pelo filtro. Neste aspecto, o método proposto se destacou positivamente, pois preservou os intervalos mais amplos de dados, permitindo assim a retenção de informações que refletem a variabilidade existente no campo. É relevante destacar a complexidade inerente à definição de um limiar que identifique de maneira acurada os dados discrepantes. O subconjunto gerado pela janela deslizante torna esse limiar mais específico, pois compreende os dados de maneira localizada. Quanto maiores forem as tolerâncias, menos falsos positivos ou falsos *outliers* serão identificados. Isso ocorre porque o algoritmo tolera valores maiores nos limites da JD. No entanto, grandes tolerâncias também resultam em muitos falsos negativos, ou seja, valores discrepantes perdidos. No presente estudo, o limiar depende do tamanho da janela deslizante e do fator *f*, o que leva o algoritmo a possuir configurações distintas em cada etapa (*i*, *ii*, *iii*) e, portanto, tolerâncias distintas. A primeira etapa possui uma menor tolerância em relação às outras, principalmente porque os dados são inseridos pela primeira vez no filtro. Assim, o limiar é variável e de menor tolerância, quando comparado aos das etapas seguintes. Na etapa *ii*, esse limiar variável é mais amplo porque o fator *f* é maior e o conjunto de dados iterou *n* vezes em subconjuntos maiores que o da etapa *i*. Desta maneira, são identificados e removidos os dados que talvez não pudessem ser detectados com determinados valores de limiar de dispersão. A configuração da etapa *iii* permite um limiar variável que possibilita uma maior amplitude dos dados, considerando o fator *f*, e uma quantidade de elementos na janela que geraram subconjuntos adequados para detectar os valores discrepantes de tempo de armazenamento dos dados de produtividade no computador de bordo da colhedora. Apesar das diferenças em determinados parâmetros estatísticos e das respostas quanto ao desempenho de detecção de valores discrepantes, os métodos de filtragem foram equivalentes na identificação de regiões com diferentes potenciais produtivos da cana-de-açúcar no campo. Logo, o método proposto permitiu a retenção de dados que refletem a variabilidade da produtividade existente nos canaviais (Figura 3).

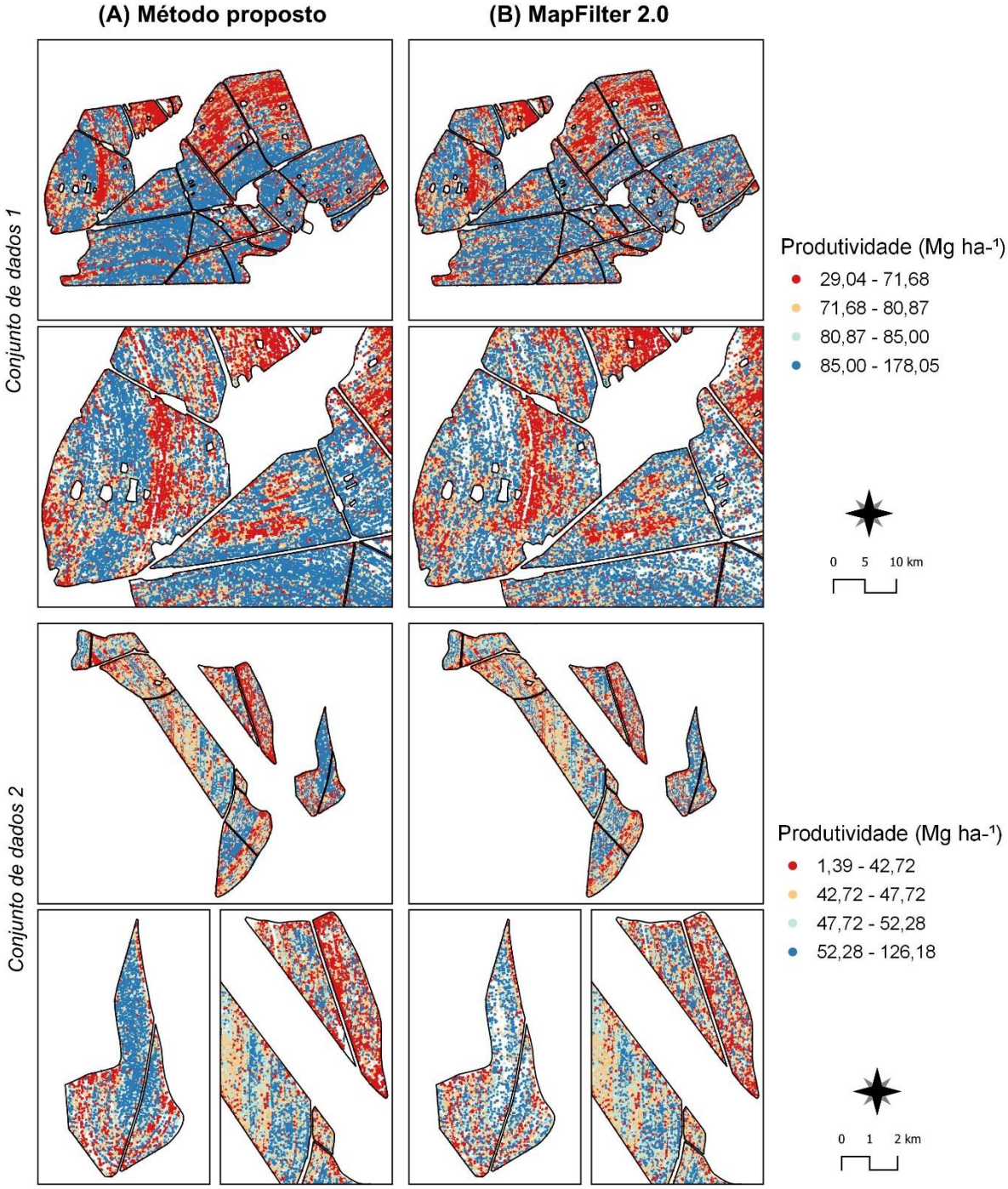

FIGURA 3. Caracterização da produtividade da cana indicados por pontos gerados a partir dos conjuntos de dados 1 e 2 filtrados pelo método proposto neste estudo (A) e pelo MapFilter 2.0 (B).

Adicionalmente, constatou-se que o método proposto identificou e removeu do conjunto de dados registros relacionados a operações de deslocamento da colhedora no campo (Fig. 4A). Esses dados incluem dados relacionados ao movimento da máquina da gleba anterior, deslocamento entre talhões e nos carreadores. Além disso, foram detectados e removidos dados correspondentes à parada da máquina na lavoura para a troca da lâmina de corte, manutenção corretiva, falta de caminhão para reabastecimento/descarregamento da cana colhida, lavagem e limpeza de equipamentos (Fig. 4B). Estes dados, com tempos de operação superiores aos de colheita e com valores de produtividade

apontados como iguais a zero, visto que é uma operação em que não estava sendo efetivamente realizada a colheita, foram facilmente identificados pelo método proposto.

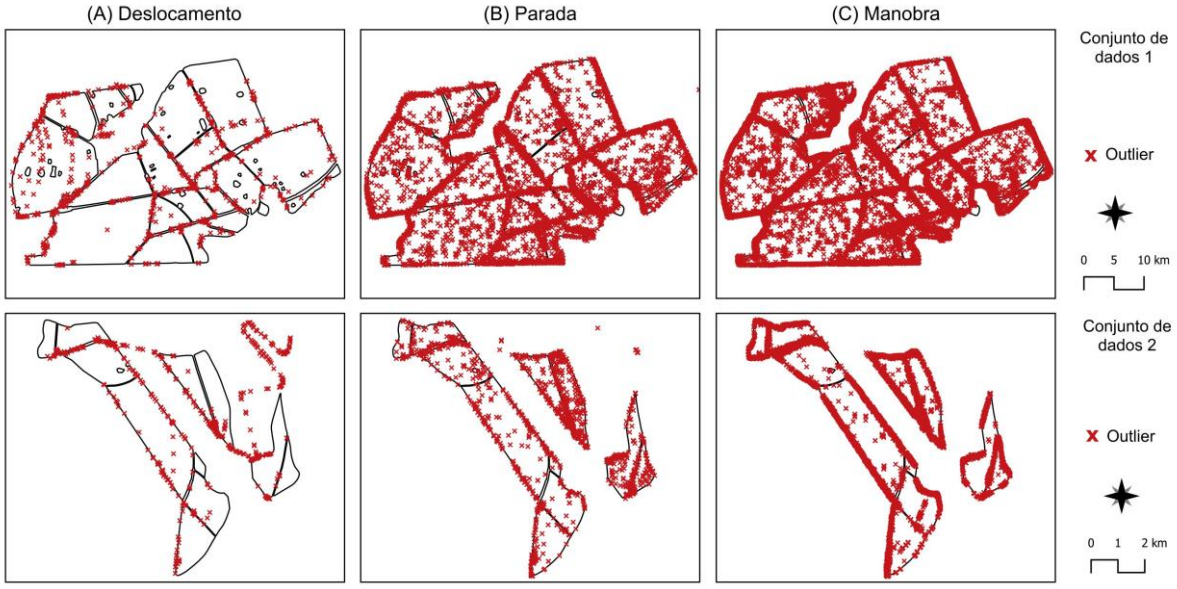

FIGURA 4. Detecção de valores discrepantes pelo método correspondentes às operações de deslocamento (A), parada (B) e manobra (C) da colhedora no canavial nos conjuntos de dados 1 e 2.

Para os dados armazenados como manobras (Fig. 4C), houveram duas fontes para a identificação de dados discrepantes pelo filtro proposto: a produtividade igual a zero e o tempo para manobra que superava o tempo de armazenamento de um dado efetivo de colheita (valores de até 60 s). Como pode ser observado, houve pontos localizados dentro dos talhões em regiões que não possuem carreadores, que foram registrados como um dado referente à manobra, possivelmente sendo um erro operacional, entretanto foram removidos porque os dados de produtividade nestes pontos eram nulos. Estes tipos de dado gerados no campo ocorrem em canaviais devido i) a talhões com fileiras mais longas e sem segmentação através do cruzamento de carreadores, atingindo a capacidade de colheita em locais distantes das estradas onde podem ser descarregados/reabastecidos (SPEKKEN et al., 2015), sendo gerados dados como os da Figura 4B. ii) Comprimentos de fileiras muito curtos, ocasionando a maior ocorrência de dados de manobra. Além de não serem rentáveis econômico e energeticamente, como verificado por Spekken et al. (2015); iii) necessidade de mais estudos de otimização de rotas na colheita mecanizada da cana-de-açúcar (SANTORO et al., 2017; SPEKKEN et al., 2015). Todos esses fatores e características da colheita da cana-de-açúcar resultam em dados com peculiaridades que precisam ser processados por meio de filtros para a geração de informação do canavial. Os dois conjuntos de dados relatados neste estudo foram originados com dados reais obtidos do cotidiano em operações agrícolas de colheita realizadas em canaviais. O método de detecção e remoção de *outliers* proposto é simples, não demanda de hardware especializado e é computacionalmente otimizado para grande volume de dados. Todos estes fatores garantem que a filtragem proposta pode ser utilizada em outros cenários de colheitas em canaviais. Como limitações, a filtragem proposta cumpre sua tarefa em fenômenos anisotrópicos (na direção da colheita na fileira) e são necessários estudos que verifiquem a estrutura espacial da produtividade da cana considerando o fenômeno isotrópico. Adicionalmente, verificou-se que foram removidos uma quantidade significativa de dados brutos e isso reflete um debate atual acerca do tratamento de dados obtidos por sensores em alta resolução espacial: um analista de dados agrícolas deve estar atento não apenas à quantidade de dados obtidos na lavoura, mas também à qualidade desses dados.

CONCLUSÃO

A filtragem proposta é automática e não necessita da inserção manual de parâmetros de entrada, o que evita tomadas de decisão subjetivas. O filtro proposto foi capaz de detectar quase 100,00% dos

outliers de maiores magnitudes, como esperado, e cerca de 40,00% dos *outliers* de menor magnitude. Observou-se que a filtragem proposta preservou os intervalos mais amplos de dados e apresentou resultados equivalentes na identificação de regiões com diferentes potenciais produtivos da cana-de-açúcar no campo em comparação ao método de filtragem comparativo (MapFilter 2.0). Portanto, o método proposto permitiu a retenção de dados que refletem a variabilidade existente lavoura e pode ser utilizado para a filtragem de dados de colheitas em canaviais.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. À Solinftec pelo suporte na obtenção dos dados.

REFERÊNCIAS

BRAUNBECK, O. A.; OLIVEIRA, J. T. A. Colheita de cana-de-açúcar com auxílio mecânico. **Engenharia Agrícola**, v. 26, n. 1, p. 300-308, 2006.

KLUYVER, T.; RAGAN-KELLEY, B.; PÉREZ, F.; GRANGER, B.; BUSSONNIER, M.; FREDERIC, J. et al. Jupyter Notebooks - a publishing format for reproducible computational workflows. *In*: LOIZIDES, F.; SCHMIDT, B. (Eds.). **Positioning and Power in Academic Publishing**: Players, Agents and Agendas. Amsterdã: IOS Press, 2016. p. 87-90.

LIU, J.; LIU, L.; DU, J.; SANG, J. TLE outlier detection based on expectation maximization algorithm. **Advances in Space Research**, v. 68, n. 7, p. 2695-2712, 2021.

MALDANER, L. F.; MOLIN, J. P.; SPEKKEN, M. Methodology to filter out outliers in high spatial density data to improve maps reliability. **Scientia Agricola**, v. 79, n. 1, e20200178, 2022a.

MALDANER, L. F.; CANATA, T. F.; MOLIN, J. P. An Approach to Sugarcane Yield Estimation Using Sensors in the Harvester and ZigBee Technology. **Sugar Tech**, v. 24, p. 813-821, 2022b.

MALDANER, L. F.; MOLIN, J. P. Data processing within rows for sugarcane yield mapping. **Scientia Agricola**, v. 77, n. 5, e20180391, 2020.

MOKOENA, T.; CELIK, T.; MARIVATE, V. Why is this an anomaly? Explaining anomalies using sequential explanations. **Pattern Recognition**, v. 121, 108227, 2022.

PYTHON. **A Biblioteca Padrão do Python**, 2024. Disponível em: <https://docs.python.org/pt-br/3/library/index.html>. Acesso em Jan 2024.

SANTORO, E.; SOLER, E. M.; CHERRI, A. C. Route optimization in mechanized sugarcane harvesting. **Computers and Electronics in Agriculture**, v. 141, p. 140-146, 2017.

SMITI, A. A critical overview of outlier detection methods. **Computer Science Review**, v. 38, 100306, 2020.

SPEKKEN, M.; MOLIN, J. P.; ROMANELLI, T. L. Cost of boundary manoeuvres in sugarcane production. **Biosystems Engineering**, v. 129, p. 112-126, 2015.

SOUIDEN, I.; OMRI, M. N.; BRAHMI, Z. A survey of outlier detection in high dimensional data streams. **Computer Science Review**, v. 44, 100463, 2022.

TOBLER, W. R. A computer movie simulating urban growth in the Detroit region. **Economic Geography**, v. 46, p. 234-240, 1970.

ZENG, Z.; CUI, L.; QIAN, M.; ZHANG, Z.; KAIMIN WEI, K. A survey on sliding window sketch for network measurement. **Computer Networks**, v. 226, 109696, 2023.