




Article

Enhancing Wildfire Detection via Trend Estimation Under Auto-Regression Errors

Xiyuan Liu ¹, Lingxiao Wang ², Jiahao Li ¹, Khan Raqib Mahmud ³ and Shuo Pang ^{4,*}

¹ Department of Mathematics and Statistics, Louisiana Tech University, Ruston, LA 71272, USA; liuxyuan@latech.edu (X.L.)

² Department of Electrical Engineering, Louisiana Tech University, Ruston, LA 71272, USA; lwang@latech.edu

³ Department of Computer Science, Louisiana Tech University, Ruston, LA 71272, USA; krm070@email.latech.edu

⁴ Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA

* Correspondence: shuo.pang@erau.edu

Abstract: In recent years, global weather changes have underscored the importance of wildfire detection, particularly through Uncrewed Aircraft System (UAS)-based smoke detection using Deep Learning (DL) approaches. Among these, object detection algorithms like You Only Look Once version 7 (YOLOv7) have gained significant popularity due to their efficiency in identifying objects within images. However, these algorithms face limitations when applied to video feeds, as they treat each frame as an independent image, failing to track objects across consecutive frames. To address this issue, we propose a parametric Markov Chain Monte Carlo (MCMC) trend estimation algorithm that incorporates an Auto-Regressive ($AR(p)$) error assumption. We demonstrate that this MCMC algorithm achieves stationarity for the AR(p) model under specific constraints. Additionally, as a parametric method, the proposed algorithm can be applied to any time-related data, enabling the detection of underlying causes of trend changes for further analysis. Finally, we show that the proposed method can “stabilize” YOLOv7 detections, serving as an additional step to enhance the original algorithm’s performance.

Keywords: objective detection; auto-regression; time series analysis; trend estimation; Markov Chain Monte Carlo; Hidden Markov Model

MSC: 60K15; 82-11; 82-10; 82C4



Academic Editor: Mario García-Valdez

Received: 14 February 2025

Revised: 16 March 2025

Accepted: 19 March 2025

Published: 24 March 2025

Citation: Liu, X.; Wang, L.; Li, J.; Mahmud, K.R.; Pang, S. Enhancing Wildfire Detection via Trend Estimation Under Auto-Regression Errors. *Mathematics* **2025**, *13*, 1046. <https://doi.org/10.3390/math13071046>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The increasing frequency and intensity of wildfires, exacerbated by climate change, highlight the urgent need for effective detection and management strategies to protect ecosystems, human lives, and economic assets [1]. As wildfires become more unpredictable and widespread, early detection is crucial for minimizing damage and improving response times. However, traditional fire detection methods, such as satellite imaging and ground-based sensors, often suffer from delayed detection, limited coverage, and high operational costs [2].

Recent advancements in image processing, particularly in object detection [3] and image segmentation [4], have significantly improved fire detection accuracy and efficiency. In this context, Uncrewed Aircraft Systems (UASs) have emerged as a promising tool for real-time smoke detection, providing critical support for wildfire prevention and management [5,6]. Unlike conventional methods, UASs can rapidly navigate hazardous environ-

ments, offering a flexible and cost-effective solution for monitoring vast and inaccessible areas. By leveraging aerial imagery and sophisticated computer vision algorithms, UAS can detect smoke at its early stages, enabling authorities to respond proactively before fires escalate.

Moreover, by integrating advanced vision-based techniques with artificial intelligence, UASs enable automated smoke detection and tracking in dynamic environments [7]. With their vision-based capabilities, UASs offer significant potential for wildfire surveillance, identification, and management [8]. This approach mitigates both the environmental and social impacts of wildfires, highlighting the critical role of UASs in modern fire management strategies.

A key challenge in vision-based fire detection using video data is the limitation of object detection algorithms. One widely used real-time object detection algorithm is You Only Look Once (YOLO) [9]. Leveraging a Convolutional Neural Network (CNN) architecture, YOLO quickly detects and classifies objects in an image by dividing it into a grid of pixels. YOLOv7 [10] improves upon the original YOLO algorithm by enhancing accuracy and speed through the introduction of Extended Efficient Layer Aggregation Networks (E-ELAN).

While YOLOv7 performs well on static images, its effectiveness declines in video applications. This is because the algorithm processes each video frame independently, disregarding temporal relationships between frames. For instance, due to the temporal dependency of adjacent frames, YOLOv7's bounding boxes may flicker, leading to inconsistent detections. This can cause errors such as misclassifying clouds as smoke or missing smoke detections in some frames, even when correctly identified in previous ones. To address these limitations, we propose a time-series analysis approach to enhance YOLOv7's performance by incorporating temporal information, thereby improving the robustness and reliability of UAS-based wildfire detection.

In simple terms, a time series refers to a sequence of variable values recorded at different points in time, such as daily temperature measurements at a weather station [11]. Time series data often exhibits long-term trends, and identifying and extracting these underlying trends can be crucial for better understanding and analysis. Trend extraction is widely used across various fields. For example, the power sector relies on it to forecast daily electricity production [12], while the Department of Transportation uses it to predict state traffic patterns, helping individuals plan their travel more effectively [13].

Trend extraction methods are generally classified into parametric and non-parametric approaches. Classical parametric trend estimation methods, such as Ordinary Least Squares (OLS), include median-of-pairwise slopes regression [14] and segmented regression [15]. The advantage of parametric methods lies in their strong interpretability; when the chosen model accurately represents the underlying data, these methods can achieve high accuracy. However, real-world data often contains unpredictable factors, making it difficult to ensure that a parametric model perfectly fits the data. In such cases, non-parametric trend estimators, such as Hodrick-Prescott (HP) filtering [16] and l_1 -trend filtering [17], are often preferred.

In this paper, the proposed method falls under the category of parametric trend estimation, as the prescribed fire environment during UAS deployment remains highly consistent. Compared to OLS, the proposed method incorporates additional constraints in the dataset, making it useful not only for stabilizing the YOLOv7 algorithm by predicting smoke instances missed by YOLOv7, but also for identifying key factors influencing smoke trends, such as wind speed, forest type, and UAS movement. The following are the key contributions of this paper:

1. **Development of a Parametric Trend Estimation Method for Non-Stationary Time Series:** We introduce a novel approach to accurately decompose time-series data into its long-term and short-term components, assuming the short-term trend follows an autoregressive ($AR(p)$) model. This method leverages the inherent autocorrelation in time-series data to identify patterns and similarities between past and present data points, as is commonly observed in such datasets [18].
2. **Theoretical Justification of the Model:** We provide a rigorous theoretical foundation for the proposed parametric estimation method. Under two key assumptions, we demonstrate that the method can be optimized using an MCMC algorithm and that the algorithm converges over successive iterations.
3. **Application to Wildfire Smoke Detection in Video:** The proposed trend estimation method is applied to real-world smoke video data under the Hidden Markov Chain (HMM) structure. Experimental results demonstrate that the method effectively enhances smoke detection by correcting both missed and erroneous detections.

This paper is organized as follows: Section 2 presents some popular trend estimation methods currently in use. Section 3 describes the overall model, assumptions, theoretical support for applying MCMC estimation, and the MCMC algorithm. Section 4 provides two simulation studies to visualize the algorithm's efficiency. Section 5 compares our model estimation with the YOLOv7 [10] using a wildfire video in the real world. Section 7 shows the future research direction based on the model, and Section 8 summarizes our contribution.

2. Related Works

As mentioned, due to the unpredictable factors when collecting data, popular examples of trending estimation in time series fields are the HP filtering and l_1 -trend filtering. Compared to HP filtering, l_1 -trend filtering estimates the trend using a simple linear regression. The benefit of this approach is its interpretability. However, the optimization method for l_1 -trend filtering is slightly more complicated due to the use of the l_1 penalty. A modern solution to this optimization problem is to apply the Markov Chain Monte Carlo (MCMC) process [19]. Without the optimization problem, HP filtering is popular in economics [20]. However, since this method is nonparametric, the model lacks interpretability [21].

Another issue is that both HP filtering and l_1 filtering assume that the error terms are independent of time, meaning the time series consists of a single trend function and an error term with constant variance over time (i.e., the error is stationary). However, this stationarity assumption may not be appropriate for analyzing wildfire detection videos, as the motion of wildfire smoke in these videos is influenced by two key time-dependent factors. The first factor is the long-term trend, which is primarily determined by the motion of the UAS capturing the video. The second factor is the short-term trend, which depends on dynamic environmental conditions such as current wind speed, wind direction, wildfire spread, and other contextual variables. This short-term trend introduces a non-stationary error component into the model, making the stationary error assumption inadequate.

In addition to classical HP filtering and l_1 filtering, a newer trend estimation method is Singular Spectrum Analysis (SSA) [22–25]. SSA transforms the time series into a trajectory matrix and applies Principal Component Analysis (PCA) to decompose the matrix into a linear combination of elementary matrices, effectively extracting the latent function. However, similar to HP filtering and l_1 filtering, SSA also assumes a stationary error term.

To address the issue of non-stationarity, a non-stationary Gaussian process regression model has been proposed [26], which is a non-parametric approach capable of handling time-varying errors. While non-parametric and semi-parametric models excel at making accurate predictions, they face significant limitations in terms of interpretability and facilitating further analysis. These challenges make them less suitable for applications

that require a deeper understanding of the underlying processes, such as the dynamics of wildfire smoke or the movement of UAS.

Therefore, we propose a parametric model and combine it with YOLOv7 based on the Hidden Markov Model (HMM) framework, which overcomes these limitations. The proposed method estimates transition probabilities within the HMM using time series data, where the long-term trend is estimated using Ordinary Least Squares (OLS), and the short-term trend is modeled by an $AR(p)$ process. The emission states correspond to smoke locations identified by the YOLOv7 object detection model (see Figure 1).

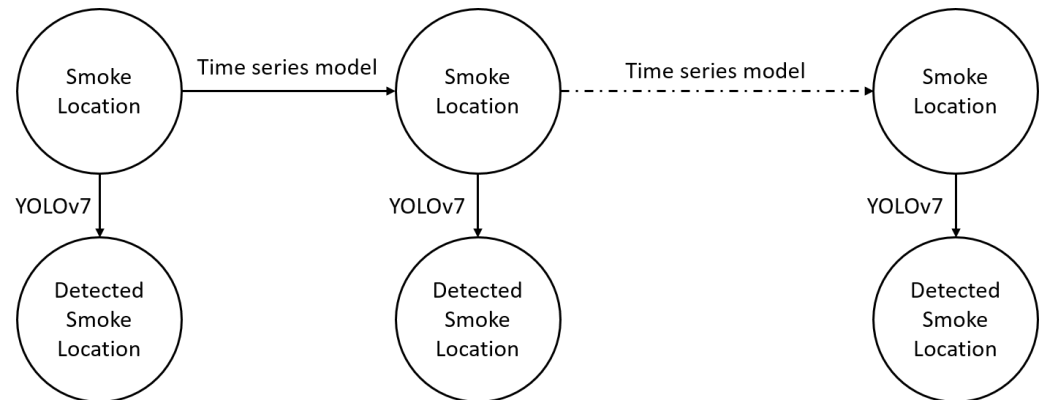


Figure 1. The Hidden Markov Model assumption by combining our proposed method and the objective detection method (YOLOv7).

This approach allows for the prediction of smoke in frames where YOLOv7 detected it in the previous frame but fails to identify it in the current one. It also corrects false detections and integrates these refined, newly identified frames into the training set, enhancing YOLOv7’s accuracy. Additionally, the trend estimation facilitates further analysis of the factors driving changes in the trend function, providing valuable insights into the dynamics of wildfire smoke.

3. Methodology

In this article, we assume that the location and motion of smoke in a video are governed by two main factors: the long-term trend and the short-term trend. The long-term trend is primarily influenced by the motion of the UAS, while the short-term trend consists of two key components. The first component is the previous smoke location in the video, which ensures temporal continuity and can be predicted using an autoregressive function [27]. The second component includes random factors affecting the smoke, such as wind direction, wind speed, and other environmental variables. Additionally, we assume that these factors are additive, leading to the formulation of the following time-series model.

$$Y(t) = f(t) + AR(p), \tag{1}$$

where $f(t)$, a deterministic function, represents the long-term trend, and the short-term trend is modeled by an autoregressive function $AR(p)$:

$$AR(p) = Y^*(t) = \alpha_1 Y^*(t - 1) + \alpha_2 Y^*(t - 2) + \dots + \alpha_p Y^*(t - p) + \epsilon_t. \tag{2}$$

Here, $Y^*(t) = Y(t) - f(t)$ represents the deviation of $Y(t)$ from the trend $f(t)$, and the error term ϵ_t is assumed to be independently and identically distributed (IID) following a distribution with finite mean 0, and variance σ^2 (i.e., $\epsilon_t \stackrel{iid}{\sim} Dist(0, \sigma^2)$). In addition, by the definition of autoregressive function, we have the following assumption.

Assumption 1. The coefficients α_i in Equation (2) satisfy

$$0 \leq \alpha_i \leq 1, i = 1, 2, \dots, p, \text{ and } \sum_{i=1}^p \alpha_i \leq 1.$$

In this setup, we treat $AR(p)$ as the random error, while the trending function $f(t)$ is conditionally independent of $AR(p)$ given

$$\mathbf{Y}^*(t) = [Y^*(t - 1), Y^*(t - 2), \dots, Y^*(t - p)]^T \in \mathbb{R}^p. \tag{3}$$

Furthermore, by Assumption 1, and leveraging the properties of simple linear regression, we constrain the covariance structure of $Y^*(t)$. That is,

$$\frac{Cov[Y^*(t), Y^*(t + i)]}{Var[Y^*(t)]} \approx \alpha_i \in [0, 1] \Leftrightarrow 0 \leq Cov[Y^*(t), Y^*(t + i)] \leq Var[Y^*(t)]. \tag{4}$$

This ensures that the covariance of is bounded, reflecting the stability of the short-term trend within the time series model.

3.1. Model Assumptions and the Stationary Property

Based on the model assumptions outlined above, we further introduce the following assumption.

Assumption 2. The initial value of the short-term trend $Y^*(t)$ given by

$$Y^*(0) \sim Dist(0, \sigma^2),$$

and for $t \geq 1$, $Y^*(t)$ follows the autoregressive process

$$Y^*(t) = \begin{cases} \sum_{i=1}^t \alpha_i Y^*(t - i) + \epsilon_t, & t = 1, 2, \dots, p, \\ AR(p), & t > p. \end{cases} \tag{5}$$

This implies that $Y^*(t)$ is a non-stationary time series with variance changing over time.

With both Assumptions 1 and 2, and given $f(t)$ and $\mathbf{Y}^*(t)$, the conditional expectation of $\mathbf{Y}^*(t)$ is

$$E[Y(t)|\mathbf{Y}^*(t)] = f(t) + \sum_{i=1}^p \alpha_i Y^*(t - i). \tag{6}$$

Thus, the long-term trend $f(t)$ can be expressed as

$$f(t) = E[Y(t)|\mathbf{Y}^*(t)] - \sum_{i=1}^p \alpha_i Y^*(t - i), \tag{7}$$

and the autoregressive component is given by

$$\sum_{i=1}^p \alpha_i Y^*(t - i) = E[Y(t)|\mathbf{Y}^*(t)] - f(t), \tag{8}$$

which is the classical detrending process.

To estimate the trend, we propose the following iterative procedure. Let $Y_{obs}(t)$ represent the observed time series. The iteration steps are as follows:

1. $\sum_{i=1}^p \alpha_i Y^*(t - i) = Y_{obs}(t) - f_i(t)$.
2. $f_{i+1}(t) = Y_{obs}(t) - \sum_{i=1}^p \alpha_i Y^*(t - i)$.

In this case, the parametric function $f_i(t)$ is a stationary function that does not change over time, while the error term $\sum_{i=1}^p \alpha_i Y^*(t-i)$ is non-stationary, with variance changing over time. Additionally, by taking the variance of both sides of Equation (8), we obtain

$$\text{Var} \left[\sum_{i=1}^p \alpha_i Y^*(t-i) \right] = \text{Var} \{ E[Y^*(t) | \mathbf{Y}^*(t)] \} \leq \text{Var}[Y^*(t)], \tag{9}$$

where $\text{Var}[Y^*(t)] = \text{Var}[AR(p)]$. Under these assumptions, we can now state the following theorem.

Theorem 1. *Let Assumptions 1 and 2 hold. The variance of $Y^*(t)$ is given by*

$$\text{Var}[Y^*(t)] = (1 + \lambda t)\sigma^2, \quad t = 1, 2, \dots \tag{10}$$

where $0 \leq \lambda \leq 1$.

With Theorem 1, the proposed method can be applied to non-stationary time series scenarios where variance changes over time. The proof of Theorem 1 is provided in Appendix A, and the empirical conclusions are discussed in the simulation study section. The following inequality follows from Theorem 1.

$$\text{Var} \left[\sum_{i=1}^p \alpha_i Y^*(t-i) \right] \leq \text{Var}[Y^*(t)] \leq (1 + t)\sigma^2. \tag{11}$$

This inequality implies that the variance of $\sum_{i=1}^p \alpha_i Y^*(t-i)$ is finite for a fixed t . Hence, as the iteration of $f(t)$ progresses, it becomes stationary over extended prediction times t , filtering the non-stationary error terms whose variance changes over time. Therefore, given the form of $f(t)$, the observed data $Y_{obs}(t)$, and the prediction horizon t , we can construct an MCMC algorithm to estimate the trending function $f(t)$.

3.2. The MCMC Algorithm

Algorithm 1 presents the pseudo-code. In our experiment, we set $f(t)$ as a linear trend since the data collected from the UAS is obtained under consistent motion. However, in theory, this method can also be applied to non-linear cases.

Algorithm 1 MCMC iteration algorithm

- 1: Initialize the trending function $\hat{f}_0(t)$.
 - 2: **for** $i = 0, 1, \dots, I - 1$ **do**
 - 3: Compute $Y^*(t) = Y_{obs}(t) - f_i(t)$.
 - 4: Estimate $\widehat{AR}_i(p)$ given $Y^*(t)$.
 - 5: Compute $RMSE_{AR}$, the rooted mean squared error of $\widehat{AR}_i(p)$, as the estimate of σ^2 .
 - 6: **for** $j = 1, 2, \dots, J$ **do**
 - 7: Simulate $\hat{Y}_{ij}^*(t) \sim N(\widehat{AR}_i(p), RMSE_{AR}^2)$
 - 8: Estimate $\hat{f}_{i+1,j}(t) = Y_{obs}(t) - \hat{Y}_{ij}^*(t)$
 - 9: Compute the 95% quantile for the $Y(t)$ estimation.
 - 10: **if** $Y_{obs}(t) \notin (\hat{Y}_{0.025}(t), \hat{Y}_{0.975}(t))$ **then**
 - 11: set $Y_{obs}(t)$ as the outlier and remove it from the training data.
 - 12: Determine $\hat{f}_{i+1}(t) = Y_{obs}(t) - \hat{Y}_i^*(t)$, where $\hat{Y}_i^*(t) = \text{Median}_j(\hat{Y}_{ij}^*(t))$.
 - 13: After I iterations, and when k th iteration becomes stationary, where $k < I$, compute the final prediction and the quantile intervals.
-

This algorithm performs two main tasks. First, it estimates the trending function $f(t)$. Second, it identifies outliers and non-stationary errors, removes them from the training data, and corrects the trending function $f(t)$.

In Line 2 of Algorithm 1, the total number of iterations, I , should be set large enough to ensure the estimated function $\hat{f}(t)$ has converged. In our simulation study, the estimates became stationary after 100 iterations. Therefore, empirically, we recommend setting I between 150 and 250 iterations. In Line 6 of Algorithm 1, we set the for-loop parameter $J = 50$ since, according to our empirical (i.e., simulation) study, there is no significant difference when $J > 50$. In Line 7, we assume ϵ_t follows a normal distribution with mean $\widehat{AR}_i(p)$ and standard deviation $RMSE_{AR}$, where $RMSE_{AR}$ is the root mean squared error (RMSE) of $\widehat{AR}(p)$ (see Line 5). In Line 12, we use the median to compute $\hat{Y}_i^*(t)$, but it can also be determined by averaging over J iterations of $\hat{f}_{i+1,j}(t)$. In this paper, the final model is

$$\hat{Y}(t) = \hat{f}(t) + \widehat{AR}(p) = \sum_{i=k}^I \frac{\hat{f}_i(t)}{I - k + 1} + \sum_{i=k}^I \frac{\widehat{AR}_{i-1}(p)}{I - k + 1}, \tag{12}$$

where the 95% Percentile Interval is given by the following formula:

$$\begin{aligned} \hat{Y}_{0.025}(t) &= \text{quantile}_i(\hat{f}_i(t) + \hat{Y}_{AR_{i-1}}(t), q = 0.025) \\ \hat{Y}_{0.975}(t) &= \text{quantile}_i(\hat{f}_i(t) + \hat{Y}_{AR_{i-1}}(t), q = 0.975) \end{aligned}$$

4. Simulation Study

We construct two simulation studies: one is to simulate the performance of Algorithm 1, and the other is to demonstrate Theorem 1.

4.1. Simulation of $Y(t) = f(t) + AR(p)$

In the simulation study, we let $f(t) = 5t$, a linear trend, and let

$$AR(3) = 0.4Y^*(t - 1) + 0.1Y^*(t - 2) + 0.5Y^*(t - 3) + \epsilon, \tag{13}$$

where $\epsilon \sim N(0, 5^2)$. Hence, the simulated data can be expressed as

$$Y(t) = 5t + 0.4Y^*(t - 1) + 0.1Y^*(t - 2) + 0.5Y^*(t - 3) + \epsilon. \tag{14}$$

The simulation contains 150 time series with time length $T = 100$ (see Figure 2a). The training data includes the first 25 time points, i.e., $Y(0), Y(1), \dots, Y(25)$. Notice that we simulate the camera shaking error at time 90. That means the algorithm should be able to split the data into two classes: one is the first 90 time points, and the other one is the last 10 time points.

We apply the Algorithm 1 to the simulation data with iteration $I = 250$, and the simulated data $J = 50$, the final model is

$$\hat{Y}(t) = 5.17t + 0.39Y^*(t - 1) + 0.14Y^*(t - 2) + 0.35Y^*(t - 3), \tag{15}$$

whereas the trend is estimated by simple linear regression (the classical detrending method):

$$\hat{Y}'(t) = 7.34t + 0.85Y^*(t - 1) + 0.02Y^*(t - 2) + 0.05Y^*(t - 3). \tag{16}$$

In addition, the standard deviation we estimated is $RMSE_{AR} = \hat{\sigma} \approx 38.16$. Hence, the estimated maximum variance of $Y(t)$ (see Figure 2b) is

$$\widehat{Var}[Y(t)] = (1 + t)38.16^2. \tag{17}$$

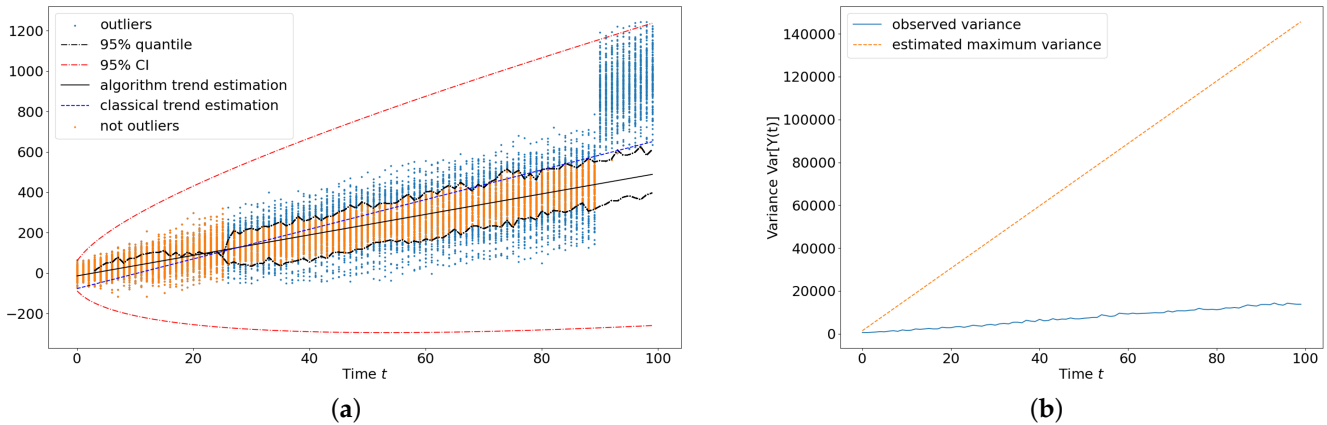


Figure 2. (a) The blue dashed line is the trending function estimated by the simple method, and the solid black line is the trending function estimated by Algorithm 1. (b) The observed variance (solid blue) and the maximum estimated variance (dashed orange) of $Y(t)$.

Table 1 shows the estimate and the confidence interval (CI) using linear regression. Because the camera shaking (error) is simulated, the linear regression overestimates the trending effect. On the other hand, the estimate and the percentile interval (PI) using Algorithm 1 show a better result. In addition, the RMSE shows that the simple linear regression (114.34) is overfitting, whereas the RMSE after using Algorithm 1 is not (145.57).

Table 1. Trending function estimated by classical linear regression vs. the proposed method.

Model	β_1		RMSE	
	Coef	95% CI or PI	Coef	95% CI or PI
Classical Method	7.34	(7.26, 7.42)	114.34	NA
Proposed Method	5.17	(4.99, 5.38)	145.57	(138.65, 151.60)

Table 2 shows the differences between the classical detrending method and Algorithm 1 for the AR model estimates. The PI of α_1 estimated by Algorithm 1 includes the true value (0.4), and both α_2 and α_3 are close to the true value. However, the classical detrending method using linear regression is not so good.

Table 2. $AR(3)$ estimated by the classical detrending method vs. the proposed method.

Model	α_1		α_2		α_3	
	Coef	95% CI or PI	Coef	95% CI or PI	Coef	95% CI or PI
Classical Method	0.85	(0.838, 0.870)	0.02	(−0.01, 0.04)	0.05	(0.04, 0.07)
Proposed Method	0.39	(0.38, 0.40)	0.14	(0.139, 0.141)	0.35	(0.34, 0.36)

Figure 2 left visualizes the final result of Algorithm 1 compared with the classical detrending method using linear regression. Notice that, since the camera error, the simple linear regression is not working correctly. Figure 2 right shows the maximum variance estimated and the observed variance. Notice that the maximum variance estimated is the upper bound and always larger than the observed variance.

Figure 3 shows the model estimating iterations by the proposed method. The estimates become stationary after 100 iterations. Hence, the final estimation takes the average of these estimators from 150 to 250 iterations (displayed by the green dashed line).

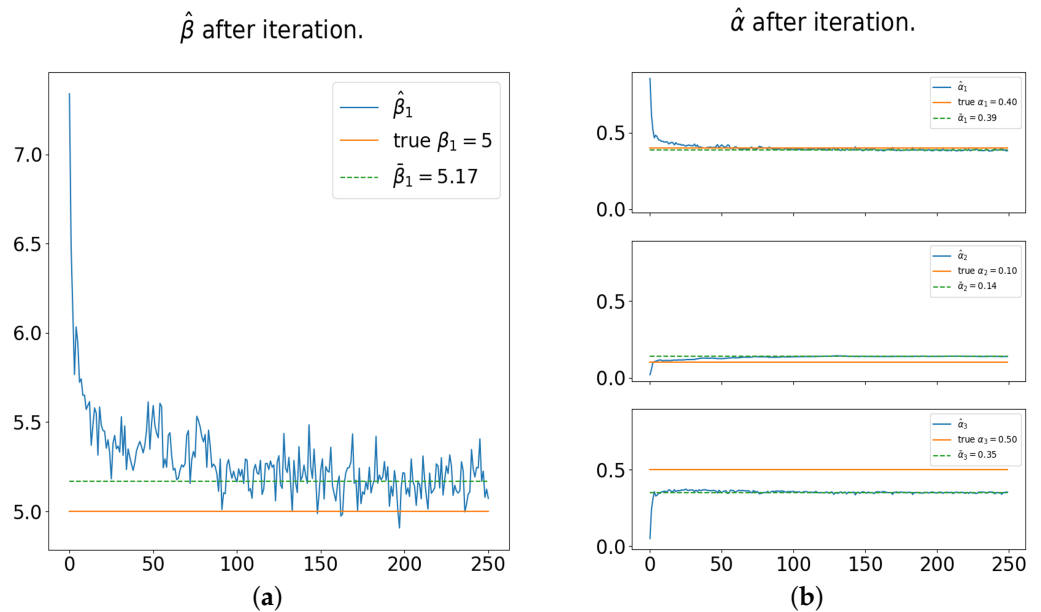


Figure 3. (a) The trending function estimation (i.e., $\hat{f}(t) = \hat{\beta}t$) estimated by the proposed method. (b) The AR(3) function estimation by Algorithm 1. The overall estimation becomes stationary after 100 iterations.

4.2. Simulation of $Var[AR(p)]$

In this simulation, we set

$$Y^*(t) = \alpha_1 Y^*(t - 1) + \alpha_2 Y^*(t - 2) + \epsilon, \tag{18}$$

where $\epsilon \stackrel{iid}{\sim} N(0, 5^2)$, and $Y(0), Y(1) \stackrel{iid}{\sim} N(0, 5^2)$. In addition, we set α_1 from 0 to 1 by 0.01, and $\alpha_2 = 1 - \alpha_1$. Moreover, we create 1500 replicates of $Y^*(0), Y^*(1), \dots, Y^*(100)$, and compute the variance of each time t (i.e., $Var[Y^*(t)]$, where $t = 0, 1, \dots, 100$).

After computing $Var[Y^*(t)]$, we apply simple linear regression to estimate the λ , that is,

$$Var[Y^*(t)] = \lambda' t + \sigma^2. \tag{19}$$

Then let $\lambda = \lambda' / \sigma^2$, which leads to the final model

$$Var[Y^*(t)] = (1 + \lambda t)\sigma^2. \tag{20}$$

Table 3 shows the minimum and the maximum estimates of λ , and when $\alpha_1 = 1$ is the random walk without drifting (Gaussian process).

Table 3. Some important $\alpha = (\alpha_1, \alpha_2)^T$ simulation results.

Important α 's	λ	
	Coef	95% CI
(0, 1)	0.513	(0.510, 0.517)
(0.07, 0.93)	0.275	(0.273, 0.278)
(0.99, 1)	0.965	(0.961, 0.970)
(1, 0)	0.932	(0.928, 0.936)

Notice that the λ estimates' minimum and maximum are all in the range of $[0, 1]$ (see Figure 4b). In addition, Figure 4a visualizes the simulated variance and the linear regression of the variance when $\alpha_1 = 0.3, \alpha_2 = 0.7$.

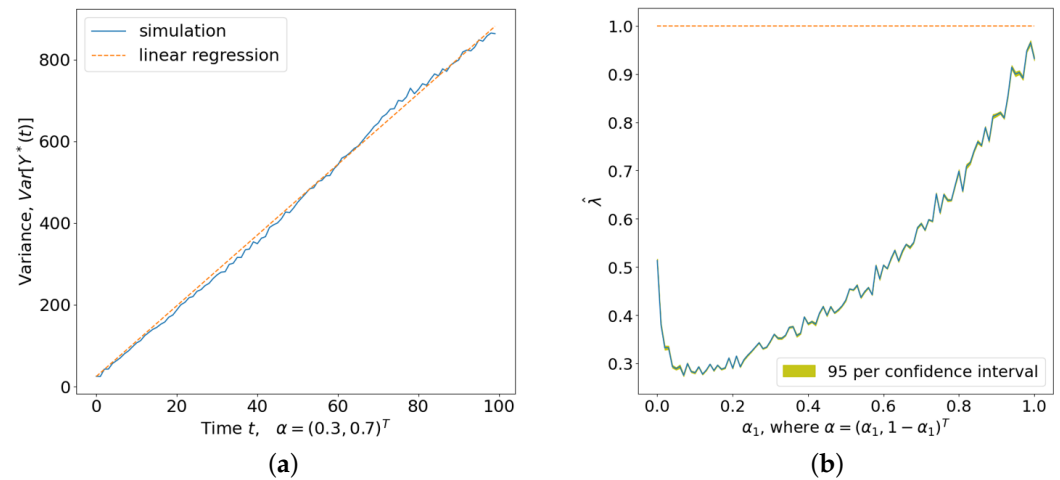


Figure 4. (a) $Var[Y^*(t)]$ vs. t when $\alpha = (0.3, 0.7)^T$. (b) The estimates of λ under different $\alpha = (\alpha_1, 1 - \alpha_1)^T$. The red dashed line indicates $\lambda < 1$.

5. Real-World Data Analysis

5.1. Data Collection

In May 2022, we collaborated with the Tall Timbers Research Station to conduct a controlled burn on their property in Tallahassee, Florida. The burn area, shown in Figure 5a, covers approximately 9 acres of forested land. These controlled burns are routinely performed during the spring to manage weeds and improve soil fertility.

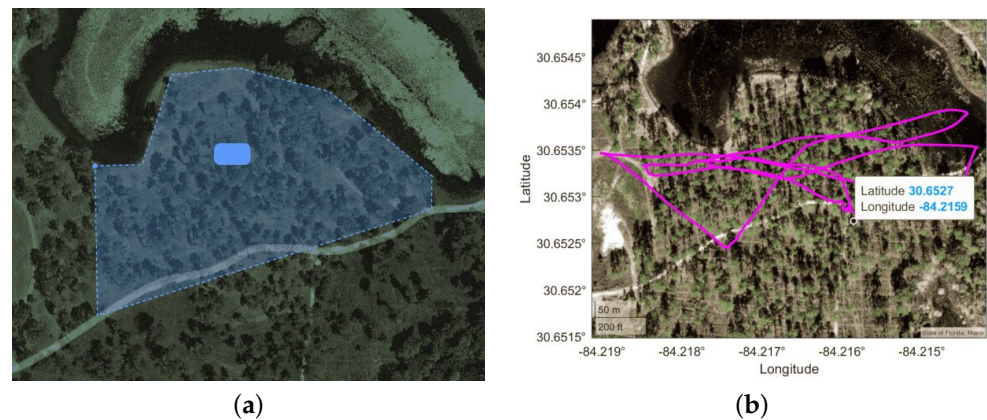


Figure 5. (a) The area designated for the controlled burn (outlined in blue). (b) The flight path of the UAS, which is under the control of a human operator. The marked spot on the diagram indicates the starting position.

Smoke movement is influenced by various factors, including temperature, external wind, and humidity. While it is mathematically impossible to precisely model all external conditions, our algorithm leverages past smoke movement to summarize these factors as an error term. In other words, our algorithm minimizes the impact of environmental conditions by estimating the trend of the smoke movement and using this trend to predict future movement.

During the burn, we utilized a multirotor UAS to collect data from the event. The UAS's flight plan initiated from the downwind side, progressing upwind towards the area of active burning. It was operated remotely by a human controller, with sensor

data being relayed to a ground station for immediate fire monitoring. Figure 5b shows the flight trajectories captured by the UAS’s onboard GPS, indicating that the drone repeatedly passed over the burn areas to gather environmental information. This operation occurred in the early stages of the controlled burn, just after it was ignited, and lasted for about 15 min before the UAS returned to the ground.

5.2. Data Process and Objective Detection

We utilized YOLOv7, a pre-trained real-time object detection model, to analyze wildfire smoke detection data captured at 30 frames per second by a UAS. YOLOv7, which is based on convolutional neural networks, is renowned for its accuracy in detecting objects in both images and videos. It outputs detailed information, including pixel coordinates of the bounding box center (horizontal x and vertical y), the width and height of the bounding box, confidence scores, and bounding box annotations.

In this dataset, we assume that all accurately detected smoke instances follow the same trend at any given timeframe. Therefore, we average the x -coordinates and y -coordinates of the center points of the bounding boxes provided by YOLOv7 within each frame, resulting in a unique x and y coordinate for each frame. This processed data is then treated as a time series, with the frame ID representing time, to train the proposed model using Algorithm 1. Finally, the trained model is applied to predict future frames, starting with a confirmed bounding box from YOLOv7 as the initial reference point.

Figure 6 shows the PACF of both the x -coordinate and the y -coordinate. According to Figure 6, we use $AR(3)$ to estimate the random effect. Hence, the model in this data is

$$Y(t) = f(t) + AR(3), \tag{21}$$

where $f(t)$ is a linear function determined by different timelines (frame id).

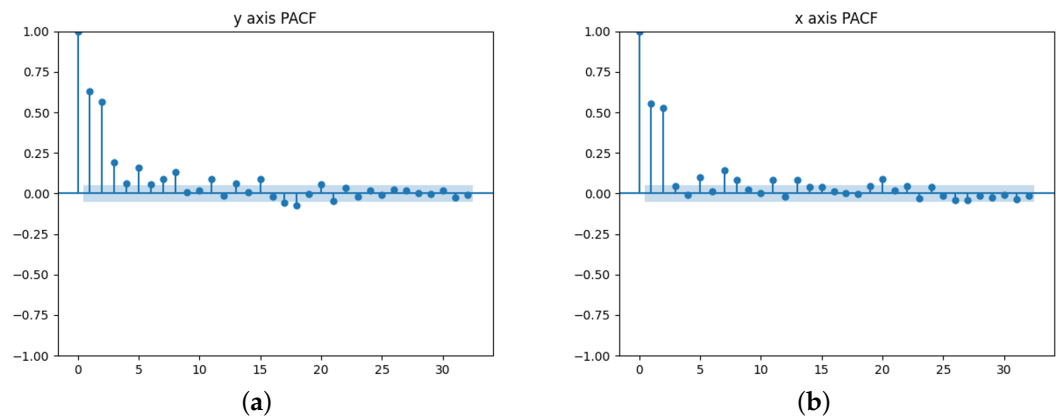


Figure 6. (a) The PACF of the y -coordinate. (b) The PACF of the x -coordinate.

Figure 7 displays a segment of the data (Frames 687–825). In this section, the video captures a single smoke area. However, within a 1 s window (Frames 756–787), the bounding box generated by YOLOv7 fluctuates multiple times due to detection errors, as indicated by the dots in Figure 7. Figure 7a highlights the significant difference between the trend prediction obtained using the classical detrending method (green solid line) and the prediction generated by Algorithm 1 (black dashed line).

The real images on the left in Figure 8 show the YOLOv7 predictions. In Frame 757, as illustrated in Figure 8, YOLOv7 fails to detect the smoke. However, Algorithm 1 successfully identifies the trend and makes an accurate prediction. Therefore, visually, the proposed method appears more “stable” than the original YOLOv7.

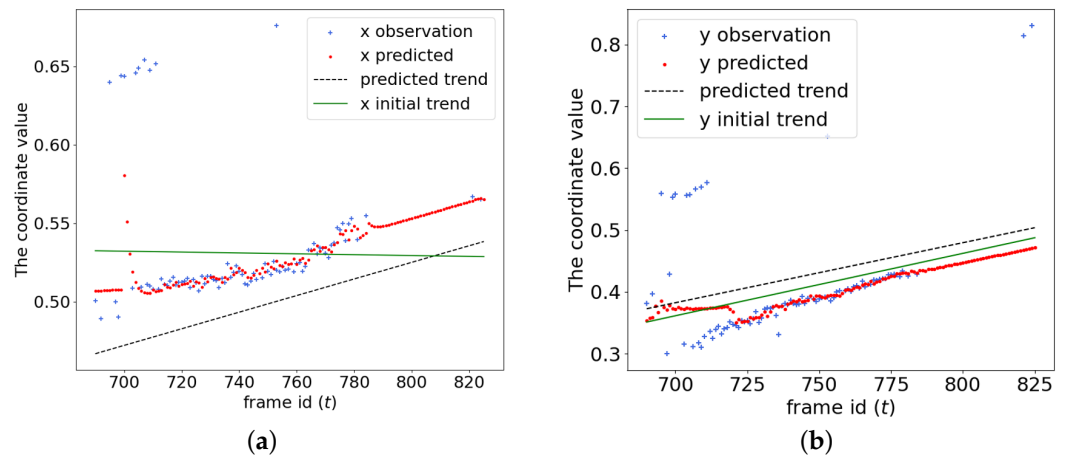


Figure 7. (a) The x -coordinate prediction. (b) The y -coordinate prediction.



Figure 8. The figures on the left are the YOLOv7 bounding box detection on Frames 756, 757, and 758. The figures on the right are the Algorithm 1 bounding box prediction on Frames 756, 757, and 758.

6. Results

Based on the flight path of the UAS (see Figure 5b), we divide the video frames into three sections. The first section, from Frames 29–500, corresponds to the UAS moving from the forest to the lake, during which YOLOv7 captures most of the smoke and generates the most consistent data. The second section, from Frames 687–825, focuses on the lake, resulting in fewer instances of smoke and less data generated by YOLOv7. In the third section, the UAS returns from the lake to the forest. This section includes both the sky and smoke, causing YOLOv7 to occasionally misidentify clouds as smoke.

Table 4 presents the numerical results of the trend estimation. For the x -coordinate, there are significant differences between the classical detrending method and Algorithm 1, with our proposed method slightly outperforming the classical approach.

Table 4. The trending function of Frame 687–825, estimated by classical linear regression vs. the proposed method.

Model	β_1		RMSE
	Coef	95% CI	
Classical Method (<i>x</i> -coordinate)	0	(−0.0003, 0.0003)	0.0324
Proposed Method (<i>x</i>-coordinate)	0.0005	(0.0005, 0.0006)	0.0321
Classical Method (<i>y</i> -coordinate)	0.0011	(0.0005, 0.0017)	0.0661
Proposed Method (<i>y</i> -coordinate)	0.0015	(0.0014, 0.0016)	0.0709

For the *y*-coordinate, no statistical differences were observed, as the 95% CIs of both the classical method and the proposed method overlap. The outcome is also illustrated in Figure 7b, where the classical method is represented by the green solid line (initial *y*-trend), and the proposed method is represented by the black dashed line (predicted trend). Additionally, since the classical method treats the points from Frame 687 to 710 as outliers, it adheres more closely to the observed data and, consequently, predicts a different intercept ($\hat{\beta}_0$), resulting in a slightly smaller root mean squared error (RMSE).

However, since both methods use only the time-related coefficient ($\hat{\beta}_1$) to predict the trend, and the proposed method is ultimately combined with the AR(3) model, discarding the predicted intercepts, the difference between the intercept estimates from both methods is not significant.

Table 5 implies that the AR(3) estimated by both methods has no significant difference except α_1 in the *y*-coordinate.

Table 5. The AR(3) of Frames 690–825, estimated by the classical detrending method vs. the proposed method.

Model	α_1		α_2		α_3	
	Coef	95% CI	Coef	95% CI	Coef	95% CI
Classical Method (<i>x</i> -coordinate)	0.0464	(−0.1431, 0.2361)	0.0609	(−0.1281, 0.2501)	0.4613	(0.2721, 0.6511)
Proposed Method (<i>x</i> -coordinate)	−0.0129	(−0.1991, 0.1741)	0.0332	(−0.1531, 0.2201)	0.5648	(0.3781, 0.7511)
Classical Method (<i>y</i> -coordinate)	0.2993	(0.0811, 0.5181)	0.0815	(−0.1671, 0.3301)	0.3700	(0.1201, 0.6201)
Proposed Method (<i>y</i> -coordinate)	0.0906	(−0.0991, 0.2801)	0.0428	(−0.1461, 0.2321)	0.5252	(0.3421, 0.7091)

Finally, we apply Algorithm 1 and identify three distinct trending patterns in three frame sections (29–500, 687–825, 3150–3600). The summary result is shown in Table 6. The inconsistent RMSE is due to changes in the motion of the observed data. The RMSE between Frames 687 and 825 is the smallest because, during this section, the UAS captured a constant wildfire while maintaining a constant speed. However, between Frames 3150 and 3600, the UAS captured both the wildfire and mistakenly identified some clouds as wildfire, causing the observations to become unstable and resulting in a higher RMSE.

Table 6. The trending function estimates using Algorithm 1.

Frames Section	β_1		RMSE
	Coef	95% CI	
29–500 (<i>x</i> -coordinate)	1.25×10^{-4}	(1.20×10^{-4} , 1.30×10^{-4})	0.0808
687–825 (<i>x</i> -coordinate)	5.29×10^{-4}	(5.00×10^{-4} , 6.00×10^{-4})	0.0321
3150–3600 (<i>x</i> -coordinate)	-2.53×10^{-4}	(-2.68×10^{-4} , -2.38×10^{-4})	0.0746
29–500 (<i>y</i> -coordinate)	1.96×10^{-4}	(1.91×10^{-4} , 2.00×10^{-4})	0.0985
687–825 (<i>y</i> -coordinate)	1.46×10^{-3}	(1.37×10^{-3} , 1.56×10^{-3})	0.0709
3150–3600 (<i>y</i> -coordinate)	2.05×10^{-3}	(2.00×10^{-3} , 2.10×10^{-3})	0.1213

7. Discussion and Future Works

There are two directions to further refine the algorithm in this paper. One is related to segmented regression in the time series. The main problem of segmented regression is determining the number of segments of time (non-identified case) [28]. With Algorithm 1, we can determine if we should make a new segment at time $t + h$ or not by the maximum confidence interval (i.e., $Y(t + h) \pm 1.96\sqrt{(1 + h)\sigma}$).

Another direction is to expand the applicability of the algorithm. As mentioned, the current constraint requires the parameters of the autoregressive function to be between 0 and 1, with their sum less than 1. Extending this constraint to the L1 norm, however, may not guarantee a stationary result. Moreover, since the data collected from the UAS is obtained under consistent motion, the algorithm is applied with a linear trend. While the trend function could, in theory, take any form, further studies are needed to explore this possibility. Additionally, in this paper, we employ the HMM-based framework solely to integrate the proposed trend estimation method with YOLOv7, without altering the HMM method itself. Therefore, the proposed approach can be further enhanced by incorporating Bayesian updates to fully leverage the capabilities of the HMM.

8. Conclusions

This paper introduces an MCMC-based trending estimation algorithm that effectively separates long-term and short-term trends for further analysis. In a real-world scenario, we apply this algorithm to wildfire smoke detection data, specifically a 30-frame-per-second video captured by a UAS. Compared to classical detrending methods, our algorithm provides more robust results with this data. Furthermore, the algorithm is used to predict missing data points and correct erroneous detections, ensuring that the bounding box in the video reliably tracks the smoke once detection is confirmed. Additionally, the paper demonstrates that, under certain constraints, the autoregressive function exhibits finite variance that is linearly related to time. As a result, the algorithm becomes stationary when the prediction period is limited.

Author Contributions: Formal analysis, X.L., K.R.M., L.W., J.L.; Writing—original draft, L.W., X.L.; Writing—review & editing, L.W., X.L., J.L., S.P.; Supervision, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The data will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1. Proof of Theorem 1, Where $t = 0, 1, \dots, p$

In this section, we will prove that, if

$$Y^*(t) = \sum_{i=1}^t \alpha_i Y^*(t-i) + \epsilon_t, t = 1, 2, \dots, p \quad (\text{A1})$$

and α_i following all the assumptions, then $\text{Var}[Y^*(t)] = (1 + \lambda t)\sigma^2$, where $0 \leq \lambda \leq 1$.

Proof. $Var[Y^*(0)] = \sigma^2 = (1 + 0\lambda_0)\sigma^2$. When $t = 1, 2, \dots, p$,

$$Var[Y^*(t)] = \sum_{i=1}^t \alpha_i^2 Var[Y^*(t-i)] + \sigma^2 \tag{A2}$$

$$+ \sum_{i=1}^{t-1} \alpha_i \left(\sum_{j=i+1}^t \alpha_j Cov[Y^*(t-i), Y^*(t-j)] \right) \tag{A3}$$

$$\leq \sum_{i=1}^t \alpha_i^2 Var[Y^*(t-i)] + \sigma^2 + \sum_{i=1}^{t-1} \alpha_i \left(\sum_{j=i+1}^t \alpha_j Var[Y^*(t-j)] \right) \tag{A4}$$

$$= \sum_{i=1}^t \left[\alpha_i \left(\sum_{j=1}^i \alpha_j \right) Var[Y^*(t-i)] \right] + \sigma^2. \tag{A5}$$

Let $t = 1$, $Var[Y^*(1)] = (1 + \alpha_1^2)\sigma^2 = (1 + \lambda_1)\sigma^2$, where $\lambda_1 = \alpha_1^2 \in [0, 1]$.

Similalry, let $t = 2$,

$$Var[Y^*(2)] \leq \sum_{i=1}^2 \left[\alpha_i \left(\sum_{j=1}^i \alpha_j \right) Var[Y^*(2-i)] \right] + \sigma^2 \tag{A6}$$

$$= \sum_{i=1}^2 \left[\alpha_i \left(\sum_{j=1}^i \alpha_j \right) [1 + \lambda_1(2-i)] \right] \sigma^2 + \sigma^2. \tag{A7}$$

Let $\lambda^* = \left(\sum_{j=1}^i \alpha_j \right)$, then $0 \leq \lambda^* \leq 1$. Consequently,

$$Var[Y^*(2)] \leq \lambda^* \sum_{i=1}^2 \alpha_i [1 + \lambda_1(2-i)] \sigma^2 + \sigma^2 \tag{A8}$$

$$\leq \lambda^* \left(\sum_{i=1}^2 \alpha_i \right) [1 + \lambda_1] \sigma^2 + \sigma^2 \tag{A9}$$

$$= (1 + 2\lambda_2)\sigma^2, \tag{A10}$$

where $\lambda_2 = \frac{\lambda^* (\sum_{i=1}^2 \alpha_i) [1 + \lambda_1]}{2} \in [0, 1]$, as $1 + \lambda_1 \leq 2$, and $0 \leq \lambda^* \left(\sum_{i=1}^2 \alpha_i \right) \leq 1$.

Let $t < p$ and

$$Var[Y^*(t)] = (1 + t\lambda_t)\sigma^2,$$

where $\lambda_t = \max_{i:i < t} \lambda_i \in [0, 1]$. Then

$$Var[Y^*(t+1)] \leq \sum_{i=1}^{t+1} \left[\alpha_i \left(\sum_{j=1}^i \alpha_j \right) [1 + \lambda_t(t+1-i)] \right] \sigma^2 + \sigma^2. \tag{A11}$$

Again, let $\lambda^* = \left(\sum_{j=1}^i \alpha_j \right) \in [0, 1]$, then

$$Var[Y^*(t+1)] \leq \sum_{i=1}^{t+1} \alpha_i \lambda^* [1 + \lambda_t(t+1-i)] \sigma^2 + \sigma^2 \tag{A12}$$

$$\leq \left(\sum_{i=1}^{t+1} \alpha_i \right) \lambda^* [1 + t\lambda_t] \sigma^2 + \sigma^2 \tag{A13}$$

$$= [1 + (t+1)\lambda_{t+1}] \sigma^2, \tag{A14}$$

where

$$\lambda_{t+1} = \frac{\left(\sum_{i=1}^{t+1} \alpha_i \right) \lambda^* [1 + t\lambda_t]}{t+1}.$$

Since $1 + t\lambda_t \leq 1 + t$ and $0 \leq \left(\sum_{i=1}^{t+1} \alpha_i\right) \leq 1, 0 \leq \lambda_{t+1} \leq 1. \quad \square$

Appendix A.2. Proof of Theorem 1, Where $t = p + 1, p + 2, \dots$

In this section, we will prove that, if

$$\sum_{i=1}^p \alpha_i Y^*(t - i) + \epsilon_t, t = p + 1, p + 2, \dots \tag{A15}$$

and α_i following all the assumptions, then $Var[Y^*(t)] = (1 + \lambda t)\sigma^2$, where $0 \leq \lambda \leq 1$.

Proof. Similar to the previous section, given the preliminary and that $t > p$, we have

$$Var[Y^*(t)] \leq \sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) Var[Y^*(t - i)] \right] + \sigma^2. \tag{A16}$$

Let $t = p + 1$, and

$$Var[Y^*(p + 1)] \leq \sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) Var[Y^*(p + 1 - i)] \right] + \sigma^2, \tag{A17}$$

where $Var[Y^*(0)] \leq Var[Y^*(1)] \leq \dots \leq Var[Y^*(p)] = (1 + \lambda p)\sigma^2, \lambda \in [0, 1]$. Consequently,

$$Var[Y^*(p + 1)] \leq \left[\sum_{i=1}^p \alpha_i \left(\sum_{j=1}^i \alpha_j \right) \right] (1 + \lambda p)\sigma^2 + \sigma^2 \tag{A18}$$

$$= [1 + (1 + p)\lambda_{p+1}]\sigma^2, \tag{A19}$$

where

$$\lambda_{p+1} = \max_{i:i \leq p} \left\{ \frac{\left[\sum_{i=1}^p \alpha_i \left(\sum_{j=1}^i \alpha_j \right) \right] (1 + \lambda p)}{1 + p}, \lambda_i \right\}. \tag{A20}$$

Since $(1 + \lambda p) \leq 1 + p$, and $\left[\sum_{i=1}^p \alpha_i \left(\sum_{j=1}^i \alpha_j \right) \right] \in [0, 1], \lambda_{p+1} \in [0, 1]$.

Let $t = p + 2$, and

$$Var[Y^*(p + 2)] \leq \sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) Var[Y^*(p + 2 - i)] \right] + \sigma^2. \tag{A21}$$

Similarly,

$$Var[Y^*(p + 1)] \leq \sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) Var[Y^*(p + 1 - i)] \right] + \sigma^2 \tag{A22}$$

$$= \sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) \right] Var[Y^*(p + 1)] + \sigma^2 \tag{A23}$$

$$= \sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) \right] [1 + (p + 1)\lambda_{p+1}]\sigma^2 + \sigma^2 \tag{A24}$$

$$= [1 + (2 + p)\lambda_{p+2}]\sigma^2, \tag{A25}$$

where

$$\lambda_{p+2} = \max_{i:i \leq p+1} \left\{ \frac{\sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) \right] [1 + (p + 1)\lambda_{p+1}]}{2 + p}, \lambda_i \right\} \in [0, 1]. \tag{A26}$$

Let $t > p$ be any integer and $\text{Var}[Y^*(t)] = [1 + t\lambda_t]\sigma^2$, where $\lambda_t \in [0, 1]$. Then

$$\text{Var}[Y^*(t+1)] \leq \sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) \right] \text{Var}[Y^*(t)] + \sigma^2 \quad (\text{A27})$$

$$= \sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) \right] [1 + t\lambda_t]\sigma^2 + \sigma^2 \quad (\text{A28})$$

$$= [1 + \lambda_{t+1}(t+1)]\sigma^2, \quad (\text{A29})$$

where

$$\lambda_{t+1} = \max_{i: i \leq t+1} \left\{ \frac{\sum_{i=1}^p \alpha_i \left[\left(\sum_{j=1}^i \alpha_j \right) \right] [1 + t\lambda_t]}{1 + t}, \lambda_i \right\} \in [0, 1], \quad (\text{A30})$$

which is the desired result. \square

References

- Hillayová, M.K.; Holécý, J.; Korísteková, K.; Bakšová, M.; Ostrihoň, M.; Škvarenina, J. Ongoing climatic change increases the risk of wildfires. Case study: Carpathian spruce forests. *J. Environ. Manag.* **2023**, *337*, 117620. [[CrossRef](#)] [[PubMed](#)]
- Hassan, A.; Audu, A. Traditional sensor-based and computer vision-based fire detection systems: A review. *Arid Zone J. Eng. Technol. Environ.* **2022**, *18*, 469–492.
- Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514. [[CrossRef](#)]
- Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3523–3542. [[CrossRef](#)]
- Chen, X.; Hopkins, B.; Wang, H.; O'Neill, L.; Afghah, F.; Razi, A.; Fulé, P.; Coen, J.; Rowell, E.; Watts, A. Wildland Fire Detection and Monitoring Using a Drone-Collected RGB/IR Image Dataset. *IEEE Access* **2022**, *10*, 121301–121317. [[CrossRef](#)]
- Mukhiddinov, M.; Abdusalomov, A.B.; Cho, J. A Wildfire Smoke Detection System Using Unmanned Aerial Vehicle Images Based on the Optimized YOLOv5. *Sensors* **2022**, *22*, 9384. [[CrossRef](#)]
- Hossain, F.A.; Zhang, Y.; Yuan, C. A Survey on Forest Fire Monitoring Using Unmanned Aerial Vehicles. In Proceedings of the 2019 3rd International Symposium on Autonomous Systems (ISAS), Shanghai, China, 29–31 May 2019; pp. 484–489.
- Yuan, C.; Zhang, Y.; Liu, Z. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Can. J. For. Res.* **2015**, *45*, 783–792. [[CrossRef](#)]
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
- Houghton, J.T.; Ding, Y.; Griggs, D.J.; Noguer, M.; van der Linden, P.J.; Dai, X.; Maskell, K.; Johnson, C.A. *Climate Change 2001: The Scientific Basis*; Cambridge University Press: Cambridge, UK, 2001; Volume 881.
- Babu, C.N.; Reddy, B.E. A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data. *Appl. Soft Comput.* **2014**, *23*, 27–38. [[CrossRef](#)]
- Wang, D.; Luo, H.; Grunder, O.; Lin, Y.; Guo, H. Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm. *Appl. Energy* **2017**, *190*, 390–407. [[CrossRef](#)]
- Brinkmann, W. Within-type variability of 700 hPa winter circulation patterns over the Lake Superior basin. *Int. J. Climatol.* **1999**, *19*, 41–58. [[CrossRef](#)]
- Hamilton, M.; Zhang, Z.; Hariharan, B.; Snavely, N.; Freeman, W.T. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv* **2022**, arXiv:2203.08414.
- Leser, C.E.V. A simple method of trend construction. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1961**, *23*, 91–107.
- Kim, S.J.; Koh, K.; Boyd, S.; Gorinevsky, D. ℓ_1 trend filtering. *SIAM Rev.* **2009**, *51*, 339–360.
- Xu, W.; Hu, H.; Yang, W. Energy time series forecasting based on empirical mode decomposition and FRBF-AR model. *IEEE Access* **2019**, *7*, 36540–36548.
- Heng, Q.; Zhou, H.; Chi, E.C. Bayesian trend filtering via proximal markov chain monte carlo. *J. Comput. Graph. Stat.* **2023**, *32*, 938–949.

20. Trimbur, T.M. Detrending economic time series: A Bayesian generalization of the Hodrick–Prescott filter. *J. Forecast.* **2006**, *25*, 247–273.
21. Chen, R.; Xiao, H.; Yang, D. Autoregressive models for matrix-valued time series. *J. Econom.* **2021**, *222*, 539–560.
22. Vautard, R.; Ghil, M. Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series. *Phys. D Nonlinear Phenom.* **1989**, *35*, 395–424. [[CrossRef](#)]
23. Broomhead, D.; Jones, R.; King, G.P. Topological dimension and local coordinates from time series data. *J. Phys. A Math. Gen.* **1987**, *20*, L563.
24. Broomhead, D.S.; King, G.P. Extracting qualitative dynamics from experimental data. *Phys. D Nonlinear Phenom.* **1986**, *20*, 217–236.
25. Poskitt, D.S. On singular spectrum analysis and stepwise time series reconstruction. *J. Time Ser. Anal.* **2020**, *41*, 67–94.
26. Paun, I.; Husmeier, D.; Torney, C.J. Stochastic variational inference for scalable non-stationary Gaussian process regression. *Stat. Comput.* **2023**, *33*, 44.
27. Preisler, H.K.; Schweizer, D.; Cisneros, R.; Procter, T.; Ruminski, M.; Tarnay, L. A statistical model for determining impact of wildland fires on Particulate Matter (PM_{2.5}) in Central California aided by satellite imagery of smoke. *Environ. Pollut.* **2015**, *205*, 340–349. [[CrossRef](#)]
28. Lerman, P. Fitting segmented regression models by grid search. *J. R. Stat. Soc. Ser. C Appl. Stat.* **1980**, *29*, 77–84.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.