Seasonal Inhomogeneous Nonconsecutive Arrival Process Search and Evaluation

Kimberly Holmgren
MIT Lincoln Laboratory
Lexington, MA, USA
Kimberly.Holmgren@ll.mit.edu

Paul Gibby
MIT Lincoln Laboratory
Lexington, MA, USA
Paul.Gibby@ll.mit.edu

Joseph R. Zipkin
MIT Lincoln Laboratory
Lexington, MA, USA
Joseph.Zipkin@ll.mit.edu

Abstract—**Time series often exhibit seasonal patterns, and identification of these patterns is essential to understanding the data and predicting future behavior. Most methods train on large datasets and can fail to predict far past the training data. This limitation becomes more pronounced when data is sparse. This paper presents a method to fit a model to seasonal time series data that maintains predictive power when data is limited. This method, called *SINAPSE*, combines statistical model fitting with an information criteria to search for disjoint, and possibly nonconsecutive, regimes underlying the data, allowing for a sparse representation resistant to overfitting.**

I. Introduction

The danger of cyber threats to modern networks is grave and growing. In 2018, 31% of organizations reported attacks on operational technology infrastructure, and 53% of attacks resulted in damages of $500,000 or more [1]. Recently, a hacker exploited a configuration vulnerability in Capital One's firewall to gain access to credit card applications containing a wealth of sensitive information associated with more than 100 million customers [2]. A reactive posture has proven insufficient given the scale and persistence of cyber attacks, necessitating a proactive approach.

A proactive approach requires an understanding of the basic rhythms of cyber attacks. Humans drive both benign and malicious network traffic, so both exhibit seasonality inherited from human patterns of life [3]. For example, the timing of phishing attacks against an enterprise network should be coupled with the work day, when people use their work email. However, there are likely other timing factors not so readily apparent. The problem thus requires a means to model seasonality while imposing minimal structure *a priori*.

This paper presents the Seasonal Inhomogeneous Nonconsecutive Arrival Process Search and Evaluation (SINAPSE) algorithm to fit piecewise constant seasonal models with minimal prior knowledge of the underlying structure. Piecewise models offer several advantages over other models, including an explicit representation of the seasonal component, computationally inexpensive prediction, and parsimonious parameterization. Additionally, some data exhibit sharp discontinuities rather than slow change over time. Segmenting the period into intervals rather than treating observations individually enables the model to identify patterns within a season with fewer parameters and data. Critically, the intervals may be nonconsecutive. For example, the algorithm may find a single, common rate of cyber attack among all weeknights rather than fitting (and potentially overfitting) a rate for each night. The model trained by this algorithm can be used to model seasonality, predict events, perform anomaly detection, or serve as a baseline model to compare more sophisticated predictive systems against.

This work is sponsored by the Intelligence Advanced Research Projects Activity (IARPA) in the Office of the Director of National Intelligence (ODNI) under Air Force Contract FA8702-15-D-0001. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA or the United States Government.

II. Literature Review

A. Seasonal Modeling

Many real-world time series exhibit *seasonality*, or regular, recurring patterns, and as such it is necessary to account for them in modeling. Here, we briefly review existing methods for modeling seasonality.

Smoothing techniques apply a set of fixed filters, or moving averages, either to decompose the time series or to denoise it. Common methods include the Holt–Winters model, X11-style methods, and other variations on exponential smoothing. The Holt–Winters method is particularly common and has been around for almost 60 years, during which time it has been improved by supplementary research [4]–[7].

The ARIMA (AutoRegressive Integrated Moving Average) and related SARIMA (seasonal ARIMA) models are among the most frequently used time series techniques for modeling and forecasting. SARIMA models seasonality with the inclusion of extra parameters for the seasonal component of the time series. However, it can be difficult to determine the best value for each of the many parameters, especially when the unit of time is small.

Other models have been proposed to handle seasonality, such as spectral analysis, which involves decomposing a time series with cyclical components into underlying sinusoidal functions. Nonstationary data can be studied using Fourier analysis, but these models are very difficult to fit. Another approach is to introduce categorical variables for the different intervals (for example, days of the week) [8]. This decision of how to divide the data is chosen qualitatively and may miss variations within the interval.

Many of these models are best fit to relatively smooth data and involve experimentation and analysis to parameterize. The SINAPSE algorithm aims to fit a piecewise distribution to data with minimal analyst burden.

B. Nonhomogeneous Point Processes

A *point process* or *arrival process* is a process by which events (points) are distributed randomly through time. The most common of these is the homogeneous Poisson process where events occur probabilistically at a constant rate for any fixed interval. Though such a model is elegant in its simplicity, it often fails to capture external changes to the underlying process, necessitating a nonhomogeneous Poisson process. Also, though Poisson is the most common base for such processes, other distributions can be used, such as Negative Binomial.

In [9], the author outlines a method to model the arrival rate as piecewise constant, and although the paper proves that the method converges to the true solution with a continuously varying rate parameter, the predicted rate is updated at every data point, which creates a model as large as the training set. [10] improves on this method by splitting the timeline into evenly sized intervals and calculating the parameters within each interval. As the interval size decreases and data volume increases, it can be shown that the model converges in the same way as [9], though at each step the model size can be kept much smaller. Although this reduces the number of parameters and hence chances of overfitting, convergence requires increasing the number of parameters to the same level as [9].

C. Changepoint Detection

As noted, human patterns of life produce discontinuities in the behavior of the networks we are modeling. Detecting the *changepoints*, the temporal boundaries between behavior regimes, allows us to fit an appropriate model to each regime. Here we review techniques for unsupervised detection of changepoints in point processes. We concentrate on offline detection, as our goal is modeling seasonal variation rather than short-term prediction.

Kawahire & Sugiyama [11] use likelihood ratios to accept or reject a changepoint in a tested time series. Other methods, such as discussed in [12], assume the time series is the output of an unknown stochastic system for which the observability matrix both before and after the proposed changepoint can be estimated. The distance between the subspaces spanned by the columns is used as a discriminator for accepting or rejecting the changepoint. Chib [13] takes a Bayesian approach, assigning a hidden state variable to each point in time, indicating between which changepoints that point lies, and so identifying changes in this hidden variable identifies the changepoint. However, the user must specify the number of changepoints.

The above methods all require large quantities of data to achieve statistically significant results. One possibility to allow training on smaller datasets is to allow for nonconsecutive intervals governed by the same set of parameters. Such work has appeared in the literature, but with significant limitations. Yao's [14] detection method presupposes a system with exactly two changepoints. Ismail & Isa [15] identifies multiple changepoints between parameter regimes but only allows for two such regimes. All these methods also require more background knowledge of the process underlying the time series than we can expect to achieve for our application [16].

III. Algorithm

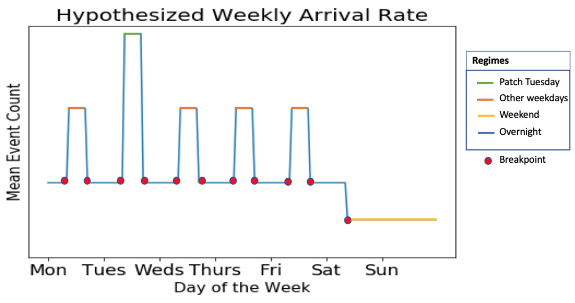

Fig. 1. Diagram of the piecewise model, as hypothesized for cyber attack data

SINAPSE takes in observations of a time period and outputs a piecewise arrival rate model specified by (i) an arrival process, (ii) a set of intervals partitioning the time period, and (iii) the parameters of the arrival process over those intervals. A *breakpoint* is a changepoint in the parameters between two intervals. A primary contribution of SINAPSE is the ability to find breakpoints without any prior knowledge of their location or number. An example of the hypothesized piecewise output model for cyber attack data is shown in Fig. 1. There are breakpoints at the start and end of each workday, as well as a breakpoint at the start of the weekend. A *regime* is a set of intervals sharing the same parameters. In Fig. 1, all the weekdays, colored orange, are part of one regime and all of the overnight periods, colored blue, are part of another regime. This model repeats periodically, in this example weekly, to model seasonality in data. The ability to link disjoint intervals exhibiting the same behavior is another key contribution of SINAPSE.

Fig. 2 summarizes SINAPSE. The algorithm is as follows:

```
1: Initialize intervals $int_0$ (III-B) based on knowledge of data or to homogeneous model with $int_0.breakpoints \leftarrow \{0, m\}$ and $int_0.map \leftarrow \{1\}$ where $m$ is the number of observations per period
2: $model_{best} \leftarrow$ None, $score_{best} \leftarrow \infty$
3: **for** $i = 0$ to $num\_iterations$ **do**
4:     Using all training data and proposed interval $int_i$, train the piecewise arrival model $model_i$ as specified in III-C
5:     Use the AICc as defined in III-D to compute $score_i$ reflecting how well $model_i$ fits the data
```

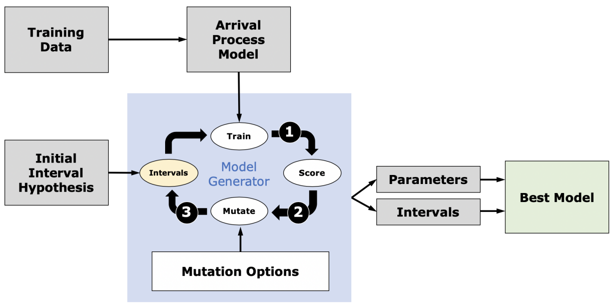

Fig. 2. Diagram of the SINAPSE algorithm.

```
6:      if $score_i < score_{best}$ then
            $score_{best} \leftarrow score_i$ and $model_{best} \leftarrow model_i$
7:      end if
8:      Mutate the intervals in accordance with the selected
        mutation algorithm (III-E) to produce $int_{i+1}$
9:  end for
10: Return $model_{best}$
```

A. Input Data

Occurrence times are transformed into an $n \times m$ counts matrix, where n is the number of observation periods and m is the number of observations per period. For an application to w weeks of data and hourly counts, this would be a $w \times 168$ matrix.

B. Interval Definition

The data are partitioned by breakpoints over the time period of the season, $b_0, b_1, ..., b_n$, where b_0 is the minimum of the range and b_n is the maximum of the range. The $n-1$ intervals between them are $i_0, ..., i_{n-1}$, where $i_j = [b_j, b_{j+1})$. A minimum interval length c can be set. A regime, designated as R_k with index k, is a collection of disjoint intervals that are enforced to have the same parameters. The size of R_k, denoted $|R_k|$, is the sum of the lengths of its constituent intervals. Each interval must belong to exactly one regime. Regimes allow more flexible modeling of the data to identify similar behavior across the period as well as reducing the number of parameters, which makes the models more compact and less likely to overfit the training data.

C. Training the Arrival Process Model

As part of the training step, the arrival process is fit to find a set of parameters for each regime. Many arrival processes in the time space are not homogeneous. For example, a restaurant is likely to have many more patrons during the time frame 6:00 PM–8:00 PM than during the time frame 2:00 AM–4:00 AM. As such, we consider arrival processes with piecewise constant parameters that change between regimes.

We demonstrate SINAPSE with a Poisson arrival process, in which events occur independently at a fixed stochastic rate, λ. The Maximum Likelihood Estimate (MLE) of the rate parameter of a Poisson process is

$$\hat{\lambda}_k = \frac{\sum_{i \in R_k} x_i}{|R_k|} \tag{1}$$

where $\hat{\lambda}_k$ is the estimate of the rate parameter for the regime R_k and x_i is the number of occurrences at time i. This is equivalent to the average arrival rate of events in the regime.

Other arrival processes are compatible with SINAPSE such as zero-inflated Poisson, which is used to model Poisson processes in which some events are expected to be missed and not recorded in the data. It has a probability mass function of

$$P(k=h) = \begin{cases} \pi + (1-\pi)e^{-\lambda} & h = 0 \\ (1-\pi)e^{-\lambda}\frac{\lambda^h}{h!} & h \geq 1 \end{cases} \tag{2}$$

and the method of moments can be used to find values of π and λ.

D. Score

The Akaike Information Criterion (AIC) is a statistic used to evaluate models of a fixed set of data:

$$\text{AIC} = 2P - 2\log L(\theta|X) \tag{3}$$

where P is the number of parameters in the model and $L(\theta|X)$ is the likelihood of the model given the data. Minimizing the AIC encourages parsimony as well as quality of fit by penalizing the number of parameters used to fit the model. The AIC is biased when the data set is small, generally when $\frac{n}{P} < 40$, where n is the number of data points. The bias-corrected AIC, notated AICc, adds the term: $\frac{2P(P+1)}{n-P-1}$ to the AIC.

The number of parameters is defined as the number of variables required to uniquely define a model within its model class. For our purposes, P is the number of intervals, plus the product of the number of regimes and the number of parameters per regime. $P = count_{intervals} + count_{regimes} * m$ where $count_{intervals}$ is the number of intervals, $count_{regimes}$ is the number of regimes, and m is the number of parameters used in each regime, which varies between model classes. This "discounts" the addition of a new interval if it is to an existing regime. For the SINAPSE algorithm, the AICc may be modified slightly to include a penalty parameter, which is a weight the user may set on the parameter term of the AICc to encourage either fewer or more breakpoints in the model. This may be useful if the user wants a simpler model, but decreasing the penalty is not recommended as it may result in overfitting.

E. Mutation

Intervals can be mutated in one of four ways with examples shown in Fig. 3.

Split an interval by randomly adding a new breakpoint. For the interval $i_m = [b_m, b_{m+1})$, if a breakpoint b were added, then the interval i_m is split into two: $i'_m = [b_m, b)$ and $i'_{m+1} = [b, b_{m+1})$. The new i'_m belongs to the regime of the pre-split i_m; the new i'_{m+1} forms a new regime.

Merge two distinct intervals by randomly selecting them and adding the second to the regime of the first. If the intervals were consecutive, remove the breakpoint between them.

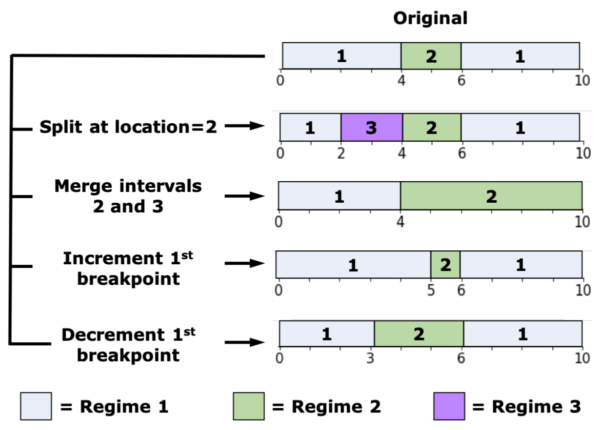

Fig. 3. Examples of each type of mutation

Increment. To increment by inc, select a breakpoint b_m (besides the final one). Increment by changing the value of b_m to $b_m + inc$.

Decrement. To decrement by dec, select a breakpoint b_m (besides the first one). Decrement by changing the value of b_m to $b_m - dec$.

F. Mutation Choice

We use a genetic algorithm [17] to choose mutations at each step. The algorithm begins by randomly generating a number, n, of interval partitions that collectively form a population. The models resulting from each interval partition are compared using a measure of fit. The least fit partitions are discarded, and the best partitions are chosen by tournament selection to be crossed and mutated with a probability, p, to form the next generation. Crossover is achieved by randomly switching two partitions' breakpoints. Mutation is as described in Section III-E.

Simulated annealing was also considered as a mutation algorithm, but was found to fit models with a higher root mean square error (RMSE) and higher variance of error.

The algorithm stops after a specified number of generations, returning the model with the best fitness value. For all datasets discussed in this paper, 100 generations was sufficient, as represented in Fig. 4.

IV. Results

A. Simulated Data

To validate the SINAPSE algorithm, data were simulated from a piecewise Poisson distribution. Dataset 1 contains three regimes with low values of λ. Dataset 2 contains the same regimes but with higher λ. Dataset 3 contains only one breakpoint. Dataset 4 contains many breakpoints. Summaries of the models and parameters data were synthesized from can be found in Table III at the end of the document. These intervals are shown in Fig. 5.

Each model was trained using an initial interval hypothesis of a homogeneous Poisson process (one regime). The genetic algorithm was used with 100 generations of population size

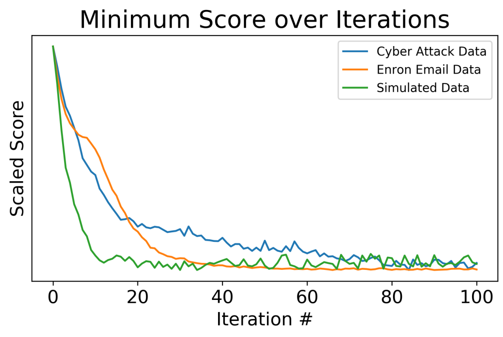

Fig. 4. A variety of datasets (simulated, Enron, and cyber data from the results section) with score levelling off around 100 generations. Errors are normalized.

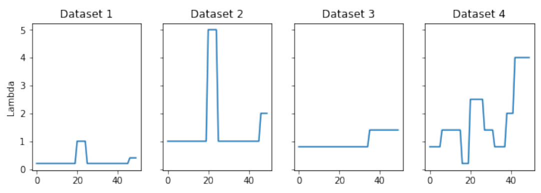

Fig. 5. True underlying intervals and model for each simulated dataset

100. The mutation and crossover probabilities are 0.5. The minimum interval length is 4. There is no additional penalty factor. These standard parameters were found to work well in a variety of conditions and are set as defaults.

1) Evaluation on Datasets 1–4: The error metric used to evaluate the performance is

$$e = \frac{\sqrt{\sum_{i=0}^{t}(\lambda(i) - \hat{\lambda}(i))^2}}{\sum_{i=0}^{t}\lambda(i)} \tag{4}$$

where the denominator serves as a normalization factor. This is proportional to the normalized RMSE between the trained model and the known true model. To evaluate the performance on the algorithm in a variety of conditions, a model was fit using SINAPSE 100 times for each dataset and the error for each was computed using (4).

Table I presents statistics on the errors on simulated data. Overall, the training error is low for all models with a similarly small range of error over 100 iterations. Comparing Dataset 1 and Dataset 2 demonstrates a greater average error but lesser variance when λ is greater. This is because an incorrect breakpoint placement causes proportionally higher error, and having more underlying observations in the data

TABLE I
Statistics for Error of Simulated Data

Dataset	Mean Error	Error Std.	Error Range
1	.005	.005	.015
2	.012	.001	.003
3	.002	.002	.007
4	.029	.004	.025

results in stronger evidence for a smaller set of models. Comparing Dataset 3 and Dataset 4 shows lower error and lower standard deviation in error when there are fewer breakpoints. This is also anticipated as there is more variety in the data in Dataset 4. There are more opportunities to mis-estimate breakpoints, resulting in a higher error.

2) Aggregate Performance: To simulate a variety of conditions, 20 weeks of testing data were generated with between 5 and 14 randomly selected breakpoints separated by at least 4 hours each. Each interval was randomly assigned a regime, and each regime used a rate parameter drawn from a uniform distribution on $[0, 2]$. This was repeated for 100 parameter sets. As can be seen from Fig. 6, SINAPSE consistently finds a model that is close to the true underlying distribution. A homogeneous Poisson model was also trained as a baseline with mean scores, determined by (4), also plotted for comparison.

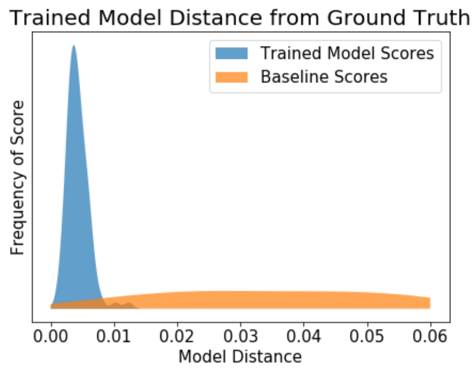

Fig. 6. SINAPSE trained algorithms are closer to ground truth than homogeneous Poisson models.

B. Enron Email Data

We also tested the SINAPSE algorithm on the Enron email dataset, available at https://www.cs.cmu.edu/∼enron/. It contains more than 600,000 emails over about two years. Timestamps following the phrase "Date:" were collected from every email between 1 January 2000 and 1 September 2000 and converted to PST. We trimmed the data to start and end on a Monday at midnight. There are 83,471 events (emails) during this nine-month period. This data is converted into an array of shape 34 (weeks in the dataset) $\times$ 168 (hours per week) containing the count of emails for each hour of each week.

Notably, the data are not stationary. Despite attempts to logarithmically transform data, smooth by n of less than 100 hours, difference, or subtract the linear trend, they fail the Augmented Dickey–Fuller test with a p-value less than 10^{-15}. There is an upward trend over the course of the dataset as well as seasonality. The upward trend is statistically significant with a coefficient of .0026 and a p-value less than 10^{-15}.

The model is trained using a minimum interval size of four hours, a Poisson model, and the genetic algorithm. The resulting weekly pattern is shown in Fig. 7. The model demonstrates a roughly constant rate for 12 hours each weekday during the working day and a lower rate over nights, with even lower rates over weekends. The significant drop during lunch hours on workdays is not modeled as the minimum interval constraint was four hours. If more or less specificity is desired, the sensitivity can be tuned by changing the penalty parameter. Increasing the penalty parameter results in a smoother model with three regimes, one for weekdays, one for overnight, and one for weekends. This model is easier to store and corresponds to states within the week.

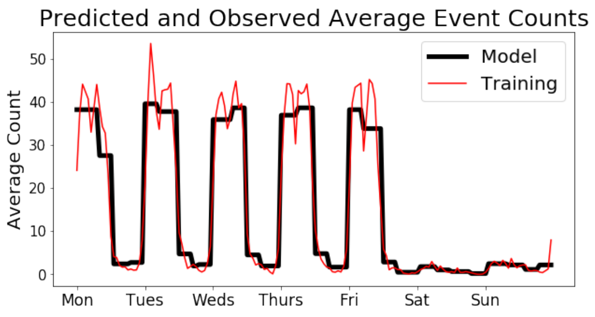

Fig. 7. Weekly rates for Enron email dataset aggregated over all weeks, compared to SINAPSE model.

1) Comparison to SARIMA Model: A SARIMA model is trained to compare SINAPSE to a traditional seasonal time series model. We used a SARIMA model with $p = 1$, $q = 1$, $P = 1$, and $S = 168$ to allow similar seasonal modeling over the prediction time period. The training RMSE of the SARIMA model was 12.14.

Because there is an increase in trend over the Enron data and SINAPSE only models seasonality, the SINAPSE model is combined with a linear trend. In the Enron dataset, the increase in model parameters is not proportionally linear: though there is an increase over time, the difference between the highs and lows, or days and nights, does not remain the same. Therefore, the SINAPSE-fit model is multiplied by a log increase per week. A linear model is fit on the log count of weekly emails, resulting in the trend 7.23+.032x, where x is the number of weeks from the beginning of the dataset. This adjustment accounts for upward trend and seasonality in the data. The training RMSE of the adjusted SINAPSE model was 12.01, which is very similar but slightly lower than that of the SARIMA model.

To compare the predictive power of the SARIMA model and the SINAPSE model, we also look one week past the end of the training data and compare the RMSE of both with the actual data. Both models were trained on data between 3 January, 2000 and 28 August, 2000. Data for the week of 28 August, 2000 to 4 September, 2000 are processed in the same way as the original data were. The predictions of the SINAPSE model adjusted for trend had an RMSE of 19.86, as compared to the predictions of the SARIMA model, which had an RMSE of 26.86, as seen in Table II. A visualization of both models' predictions as compared to the true values is shown in Fig. 8.

TABLE II
COMPARISON OF RMSE FOR SARIMA AS COMPARED TO SINAPSE MODELS ON ENRON DATASET

Model	Training RMSE	Prediction RMSE
SINAPSE	12.01	19.86
SARIMA	12.14	26.86

The SINAPSE model, adjusted for trend, fits the data as well as the SARIMA model does and outperforms it for prediction.

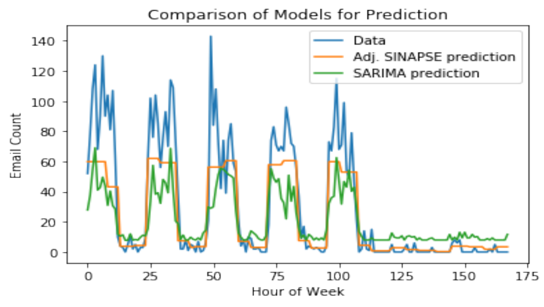

Fig. 8. Comparison of SARIMA and SINAPSE model predictions one week ahead on Enron dataset.

2) Anomaly Detection: As with many time series models, the model found by the SINAPSE algorithm can be used for anomaly detection, a utility with a broad range of applications. For cybersecurity in particular, there is demand for anomaly detection applied to network intrusion that could identify zero-day attacks. [18], [19], [20]. A variety of network traffic patterns could be modeled to detect unusual behavior on a network based on typical behavior during a week. An anomaly in the count of unique connections from a single host could indicate a port scan, a spike in packet size from a specific host could indicate data exfiltration, or activity at an unusual time could indicate an insider threat. An example of seasonal behavior that could be modeled to identify anomalies is the count of emails over the week. Monitoring emails for anomalies is important as it is the most common vector for cybersecurity incidents, with 96% of all incidents involving phishing [21]. Fig. 9 demonstrates the first six weeks of the Enron data overlaid with the weekly model, adjusted for trend. A test or metric needs to be decided upon to determine whether behavior is considered anomalous. An example of bounds for normal behavior is $\bar{x}_i \pm c * \sigma_i$, where $\bar{x}_i$ is the model's hourly rate on the interval i, c is some constant multiplier (set to 5 here), and σ_i is the standard deviation on interval i. If a data point occurs outside this range, it is determined to be anomalous. This can be improved by taking the moving average over three data points to smooth behavior near change points. Note in the context of email counts the value cannot be negative, but for other data this may be valid. The normal bounds should be determined in a way that is applicable to the data. In Fig. 9, the gray bounds define normal behavior and red dots highlight anomalies that occur outside that range. During that six week period, three anomalies were identified. By returning to the Enron emails it was validated that each spike was due to one individual behaving in an anomalous way. The first anomaly was due to congratulatory promotion emails and the next two were due to unusual weekend work. These anomalies did correspond to unusual behavior, which suggests promise for application as an anomaly detection algorithm.

C. Cyber Attack Data

The SINAPSE algorithm was also applied to cyber datasets. The first dataset contains timestamps of cyber attacks that occurred at an anonymous organization. The dataset spans from 1 July, 2017 to 31 December, 2018 and contains 506 cyber attacks. Occurrence times are converted into an array with dimensions 66 (weeks in the dataset) by 168 (hours per week), containing the count of events per hour. Given the sparsity of this dataset, it can be used to demonstrate how SINAPSE handles sparse seasonal data. As it is not always clear what constitutes a cyber event, and it is unlikely that any process will record every event, we used a zero-inflated Poisson model for this dataset. Fig. 10 shows the results of training. As can be seen, the model predicts high arrival rates during working hours that drops off every night. As the week progresses, the rate falls to an overall low over the weekend. This identified pattern mirrors an expected pattern of life for most 9–5 employees; there should be more activity during working hours, less at night, and the least on weekends.

V. DISCUSSION

A. Summary

The SINAPSE algorithm presents a novel method to train a nonhomogenous point process model for seasonal time series data. Although the literature outlines many methods to train similar models, they tend to either require domain knowledge and a significant time investment to set hyperparameters of the training process or overfit the training data. The genetic algorithm allows for efficient search of the parameter space, and because of its use of the AIC, SINAPSE can find a balance between the number of parameters and fidelity of the model to the training data. The ability to recognize nonconsecutive intervals governed by the same set of parameters allows fitting models on sparse data or models with underlying "states". This could be used to direct further investigation, such as if two intervals are identified as belonging to a common regime, there may be an underlying factor causing the similarity. The primary strength of this method is that it achieves high accuracy with minimal data and background knowledge. The only hyperparameters to be set are a penalty term, which can be increased to make the model more sparse, and the minimum interval, which can be determined based on modeling goals. These factors taken together make it highly effective on a broad range of datasets from simulated data, to high-volume email data, to sparse cyber incident data, all of which display seasonality and sharp changepoints.

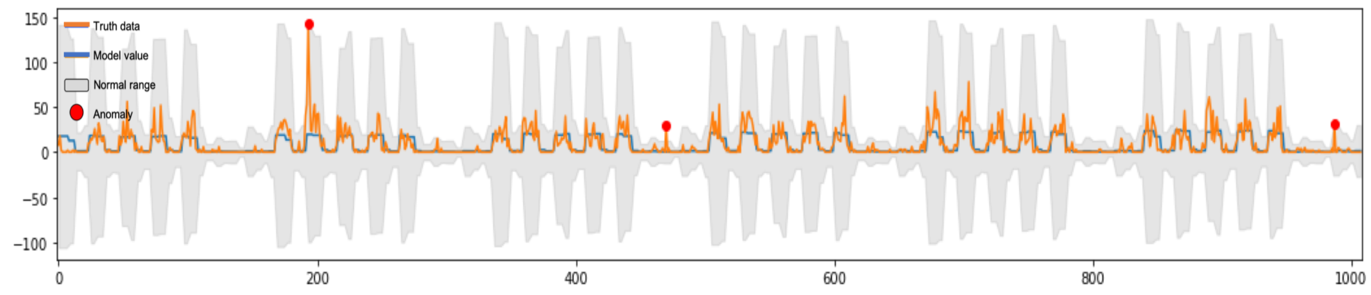

Fig. 9. Example of usage of SINAPSE for anomaly detection.

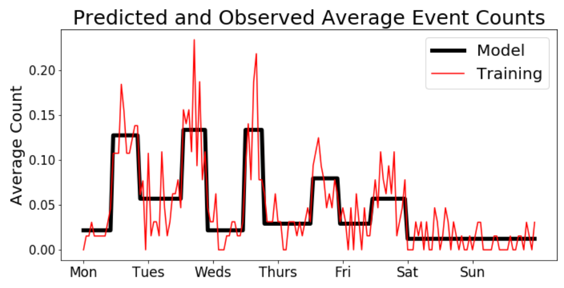

Fig. 10. Cyber event data aggregated weekly and overlaid with SINAPSE model.

B. *Use Cases/Specifications*

The SINAPSE algorithm fits a piecewise model over a period with suspected seasonality. In order to fit this model, that period must be fixed and known. This can be weekly, monthly, yearly, etc., but must be a period with unchanging length over time. This type of model is useful when there are sharp change points, such as the start and end of a workday, rather than gradual, continuous changes in the data, such as with monthly weather temperatures. It is useful when there are a large number of observation cycles in the length of the period. Monthly data within a year are too infrequent, as there are too few possible divisions of the twelve months in a year.

The algorithm currently is not designed to be used as an online algorithm. The search process for breakpoints is comprehensive and would be expensive to perform on every ingest of new data points. New data points could be used to recompute the parameters of the regime with a refitting after each season instance, but this requires storage of past data.

Additionally, there are some limitations with the choice of the genetic algorithm, as it is nondeterministic. The genetic algorithm is also susceptible to getting stuck in local minima, though this can be mitigated by increasing the mutation rate.

The model also only accounts for seasonality over the specified period. If there is an additional trend or known outliers like holidays, then it is necessary to combine the effects of each component.

VI. Future Work

Genetic algorithms, though often effective, are rarely the best way to approach an optimization problem. For example, a possible improvement may include by treating each mutation listed above as a "move" in a game scored by the AIC or applying a low-pass filter to the data before training and searching over the regime space.

A further direction to take this analysis would be to consider modeling multidimensional time series data. This would greatly broaden the application domain and allow for more powerful models that consider interactions between multiple measurements at each time.

As mentioned above, we currently only account for strict seasonality and cannot model exceptions or trend. As such, an important direction for future work would be to combine SINAPSE with other forms of seasonality or trending over time. [22] proposes modeling different types of variation and combining them multiplicatively using a Bayesian Markov Chain Monte Carlo estimation algorithm, which is a promising direction for future work.

References

[1] *Cisco 2018 Annual Security Report*, 2018.

[2] K. Matthews, "Incident of the week: Historic capital one hack reaches 100 million customers affected by breach," Aug 2019. [Online]. Available: https://www.cshub.com/attacks/articles/incident-of-the-week-historic-capital-one-hack-reaches-100-million-customers-affected-by-breach

[3] M. Papadopouli, H. Shen, E. Raftopoulos, M. Ploumidis, and F. Hernandez-Campos, "Short-term traffic forecasting in a campus-wide wireless network," *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*.

[4] S. Gelper, R. Fried, and C. Croux, "Robust forecasting with exponential and holt-winters smoothing," *SSRN Electronic Journal*, 2007.

[5] P. Goodwin, "The holt-winters approach to exponential smoothing: 50 years old and going strong," *Foresight: The International Journal of Applied Forecasting*, p. 3033, 2019.

[6] P. Gould and F. Vahid-Araghi, "Time series with multiple seasonal patterns," *Forecasting with Exponential Smoothing Springer Series in Statistics*, p. 229254, 2008.

[7] R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, "Prediction intervals for exponential smoothing using two new classes of state space models," *Journal of Forecasting*, vol. 24, no. 1, p. 1737, 2005.

[8] A. Pardo, V. Meneu, and E. Valor, "Temperature and seasonality influences on spanish electricity load," *Energy Economics*, vol. 24, no. 1, p. 5570, 2002.

[9] L. M. Leemis, "Technical note: Nonparametric estimation and variate generation for a nonhomogeneous poisson process from event count data," *IIE Transactions*, vol. 36, no. 12, p. 11551160, 2004. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/07408170490507693

[10] S. G. Henderson, "Estimation for nonhomogeneous poisson processes from aggregated data," *Operations Research Letters*, vol. 31, no. 5, p. 375382, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167637703000270

[11] Y. Kawahara and M. Sugiyama, "Sequential change-point detection based on direct density-ratio estimation," *Statistical Analysis and Data Mining*, vol. 5, no. 2, p. 114127, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.10124

[12] Y. Kawahara, T. Yairi, and K. Machida, "Change-point detection in time-series data based on subspace identification," *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 2007. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4470290

[13] S. Chib, "Estimation and comparison of multiple change-point models," *Journal of Econometrics*, vol. 86, no. 2, p. 221241, 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304407697001152

[14] Q. Yao, "Tests for change-points with epidemic alternatives," *Biometrika*, vol. 80, no. 1, p. 179, 1993. [Online]. Available: https://academic.oup.com/biomet/article-abstract/80/1/179/228230

[15] M. T. Ismail and Z. Isa, "Identifying regime shifts in malaysian stock market returns," *International Research Journal of Finance and Economics*, no. 15, p. 4457, 2008. [Online]. Available: https://www.researchgate.net/profile/Mohd_-Tahir_Ismail/publication/231216376_Identifying_-Regime_Shifts_in_Malaysian_Stock_Market_-Returns/links/0fcfd5066075e61a85000000/Identifying-Regime-Shifts-in-Malaysian-Stock-Market-Returns.pdf

[16] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and Information Systems*, vol. 51, no. 2, p. 339367, 2016.

[17] D. Whitley, *A genetic algorithm tutorial*. Univ., 1993.

[18] V. Kumar, "Parallel and distributed computing for cybersecurity," *IEEE Distributed Systems Online*, vol. 6, no. 10, p. 11, 2005.

[19] A. Aleroud and G. Karabatis, "A contextual anomaly detection approach to discover zero-day attacks," *2012 International Conference on Cyber Security*, 2012.

[20] M. Zolotukhin, T. Hamalainen, T. Kokkonen, and J. Siltanen, "Analysis of http requests for anomaly detection of web attacks," *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, 2014.

[21] "2018 verizon data breach investigations report," *2018 Verizon Data Breach Investigations Report*, 2018. [Online]. Available: https://enterprise.verizon.com/resources/reports/DBIR_2018_Report.pdf

[22] J. Weinberg, L. D. Brown, and J. R. Stroud, "Bayesian forecasting of an inhomogeneous poisson process with applications to call center data," *Journal of the American Statistical Association*, vol. 102, no. 480, p. 11851198, 2007.

APPENDIX

TABLE III
MODELS WE USED TO GENERATE SYNTHETIC DATA

Dataset	# Samples	Breakpoints	Interval Map	Parameters
1	50	[0, 20, 25, 46, 50]	[λ_1, λ_2, λ_1, λ_3]	λ_1=.2, λ_2=1, λ_3=.4
2	50	[0, 20, 25, 46, 50]	[λ_1, λ_2, λ_1, λ_3]	λ_1=2, λ_2=10, λ_3=4
3	50	[0, 35, 50]	[λ_1, λ_2]	λ_1=.8, λ_2=1.4
4	50	[0, 6, 16, 20, 27, 32, 38, 42, 50]	[λ_1, λ_2, λ_3, λ_4, λ_2, λ_1, λ_5, λ_6]	λ_1=.8, λ_2=1.4, λ_3=.2, λ_4=2.5, λ_5=2, λ_6=4