IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 April 2022, accepted 24 June 2022, date of publication 5 July 2022, date of current version 24 February 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3188651

Model-Adaptive Event Triggering for Monitoring Recurrent Mobility Patterns in Public Transport

PANAYIOTIS KOLIOS[1], (Member, IEEE), LOIZOS PAPACHRISTOFOROU[2], CHRISTOS PANAYIOTOU[1], (Senior Member, IEEE), AND GEORGIOS ELLINAS[1], (Senior Member, IEEE)

[1]KIOS Research and Innovation Center of Excellence, Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia, Cyprus
[2]Department of Electronic and Electrical Engineering, Imperial College London, London SW7 2BX, U.K.

Corresponding author: Panayiotis Kolios (pkolios@ucy.ac.cy)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Program under Grant 739551 (KIOS Research and Innovation Center of Excellence–TEAMING); and in part by the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

ABSTRACT Event-triggering (ET) is a highly promising technique for efficient operation of Internet of Things (IoT) devices, where instead of continuous or even periodic triggering of events, communication and control is only applied after some event interrupt. In this work, the proposed event-triggering technique is examined and applied on public transportation services, having as an objective to provide good tracking accuracy for a fleet of buses, while limiting communication between the system's components. Specifically, an ET technique is proposed, where a local and a remote host use behavior modeling to track the evolution of the system and synchronization messages are only sent when deviations are detected between the nominal model and the actual behavior of the system. This work focuses on a multi-model event triggering and proposes and develops a multi-model ET (MMET) technique, where multiple models are derived to accurately represent the system state. This is achieved by utilizing data-driven approaches that are used to analyze recurrent patterns and predict the system's behavior. In this way, both local and remote hosts can adapt to system changes by switching to the most accurate model that best represents the underlying system settings. The proposed MMET technique is subsequently compared to the single-model event triggering (SMET) approach, as well as traditional periodic triggering techniques, demonstrating that MMET can provide better performance in terms of tracking accuracy at a lower number of communication events, reducing in this way communication energy consumption as well.

INDEX TERMS Event triggering, behavior modeling, energy efficiency.

I. INTRODUCTION

The use of IoT devices increases rapidly across all aspects of everyday life. Their ability to perform specific functions without human intervention in order to improve the quality of life, increases their appeal to the masses. However, most IoT devices are resource constraint and this leads to a limited operating performance, which is greatly affected by the various computation, communication, and sensing/actuation tasks that need to be executed [1].

Various works exist in the literature on the development of algorithms that preserve maximum performance of these devices, as investigated in [2]–[4]. Importantly, by limiting the communication between devices, while achieving target performance levels, improves longevity, as well as avoids contention over the wireless channel and queueing delays in processing these messages [5]. Event-triggering (ET) algorithms facilitate these features by enabling communication only at particular event interrupts, while at every other time the communication circuitry is put to sleep or even completely turned off [6].

In this work, a model-based event triggering technique is investigated for the exchange of information between a local

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

and a remote host. The local host resides in the system to be monitored, while the remote host is off premise, and both the local and remote hosts have a model of the system. The remote host uses this model to estimate the system state when there is no communication between the local and remote host, and the local host compares this model with the true state of the system, as observed by its sensors, to decide when (re)synchronization events need to be sent out in order to inform the remote host of changes in the system.

As demonstrated in this work, to accurately represent the system state, and in order to minimize the number of synchronization events, multi-model event triggering techniques can be developed through simple data-driven approaches. These approaches consist of three phases, namely behavior modeling, event detection, and event handling [7] (as elaborated in Section III). To demonstrate the validity and the effectiveness of the multi-model event triggering technique, it is applied to a public transportation scenario, where a fleet of buses service a particular route that needs to be monitored and managed. In this case, the local host consists of an onboard mobility tracker and a communication interface, while the remote host is a fleet management center, where information is collected to monitor the complete system state. The multi-model ET technique is also compared to the single-model ET approach (also implemented for the specific transportation scenario), demonstrating the advantages of the multi-model approach in terms of tracking accuracy using significantly lower number of events resulting to reduced communication energy expenditure.

Specifically, the contributions of this work are as follows:

- It develops a novel multi-model event triggering (MMET) technique where multiple models are derived to accurately represent the system state. In particular, these multiple models are derived using data-driven approaches that analyze recurrent patterns and can predict the system's behavior.
- It demonstrates how this multi-model approach addresses shortcomings of the single-model technique (in terms of number of events generated and response times), allowing the remote host as well as the local (central) host to quickly adapt to any changes that may occur in the system, by switching to the most accurate model that best represents the underlying system settings whenever deemed necessary.
- It extensively analyzes this novel model-adaptive framework (both the single- and multi-model) for different target number of events, bounds, etc. This performance evaluation is performed both analytically and utilizing real data by considering a real-world practical scenario (a bus route in the city of Nicosia, Cyprus), demonstrating the validity and feasibility of the proposed MMET technique as compared to the SMET as well as traditional periodic triggering techniques.
- It investigates the network operating conditions/ parameters required for achieving a balance between tracking accuracy and number of events generated, based on the system operator's specific performance targets (i.e., deviation from true location, energy consumption, bandwidth utilization, etc.).

The rest of the paper is structured as follows: Section II includes the related work and the novelty of the proposed work, while the proposed ET architecture is presented in Section III. Details of the application scenario are included in Section IV. The single-model event triggering approach is discussed in Section V, while the multi-model case is subsequently discussed in Section VI. Performance results of the single-model approach (benchmark), together with the performance evaluation of the MMET technique, based on a real-world practical application scenario, are detailed in Section VII. Finally, Section VIII offers some concluding remarks and possible future directions.

II. RELATED WORK

Recent studies show that the use of IoT devices is significantly affected by the limitations in energy availability [8], [9]. Thus, several works in the literature have given particular emphasis on the deployment of solutions to minimize the energy consumption of those devices [2], [3], [10]–[12].

For instance, mobility trackers are extensively used for monitoring and logging fleets of vehicles to effectively manage their use, especially in the public transport context. Public transport services are characterized by recurrent trips, which can be utilized to extract information and predict behavior. Under this condition, various techniques and algorithms have been developed to minimize energy consumption, while preserving tracking accuracy (defined as the average error between the actual and the reported location) [13].

One popular approach is trace compression, as used in several works reported in [12], [14]–[20]. Trace compression algorithms aim to reduce the size of trajectory data, while retaining the quality of the information. Clearly, the less data exchanged, the less energy expended. However, despite of the fact that this technique reduces the size of each transmission, it is not capable of reducing the number of transmissions without sacrificing tracking accuracy [13].

Complementary to trace compression, ET algorithms, in contrast to traditional periodic triggering approaches, trigger communication only when an abnormal event (that can potentially change the state of the system) has taken place [21]–[23]. Event triggering is particularly useful in applications where recurrent mobility patterns exist and can be obtained utilizing and analyzing collected data via machine learning or other techniques (e.g., regression, Kalman filtering) [24]–[29]. Thus, the number of communication events is minimized, resulting in extended running life for the IoT devices. Various strategies to handle ET signals have been proposed [30], and several ET architectures and techniques have been presented in the literature. Specifically, centralized supervisory architectures have been proposed and applied due to their simplicity [31], while distributed architectures have also been extensively researched [32]–[37].

Further, collaborative architectures have also been introduced, which are capable of distributing computations across a network of devices, resulting in a decreased computational complexity [38].

This work extends our previous works in [7] and [39] by investigating the development of a holistic model-adaptive (single- and multi-model) framework for the ET technique. Specifically, this work presents, develops, analyzes, and experimentally evaluates a complete model-adaptive triggering framework, where a local and a remote host use behavior modeling to track the evolution of a dynamical system and trigger events (i.e., send communication messages) only when the actual and modeled behaviors of the system deviate. As detailed in Section I, the focus of this work is on the development of a novel multi-model event triggering (MMET) technique and its performance advantages as compared to the SMET and periodic triggering approaches, as by representing the system state with multiple models the system dynamics are more accurately captured, and at the same time the number of resynchronization events is decreased. Additional work on MMET was also presented in our previous work in [40], where, contrary to this work, we considered the case where vehicles exchange information in order to ascertain their next operating models.

As previously mentioned, the proposed multi-model technique is evaluated for a public transport use case, where a fleet of buses service a particular route. It is shown, that the proposed MMET technique, that extends the single-model ET framework, provides enhanced tracking accuracy as compared to SMET, while at the same time limiting event triggering between the local (i.e., onboard mobility tracker with a communication interface) and remote host (i.e., fleet management center). Specifically, as demonstrated in this work, MMET provides more that 3× improvement in model tracking accuracy for the same number of triggering events as compared to SMET. Alternatively, for the same tracking accuracy, the number of triggering events reduces by approximately 9× for the MMET as compared to SMET, which translates directly to less communication bandwidth utilized and less communication energy consumed by the local devices, an important consideration for the successful implementation of IoT in a number of application scenarios. Further, compared to a periodic triggering technique, for the use case examined, the MMET technique achieves a 5× improvement in terms of the number of vehicle state updates, thus again outperforming the periodic approach in terms of communication bandwidth and communication energy expended.

One can argue that even though energy savings are achievable for the use case under examination, energy consumption is not a critical consideration for this specific use case, as the remote host is on a vehicle where there is a continuous supply of power (even though it could be argued that for electric vehicles this could be a consideration as any unnecessary energy consumption will constrain the travel distance of the electric vehicle). Nevertheless, the reader should note that the multi-model event triggering approach proposed in this work is completely general and it applies to any type of IoT network utilizing any type of device, including battery-powered mobile devices and smartphones, where energy constrains are very much an issue (i.e., it is not confined to the public transport use case). Thus, in general it can provide both communication and energy savings, especially for networks utilizing devices with battery constrains.

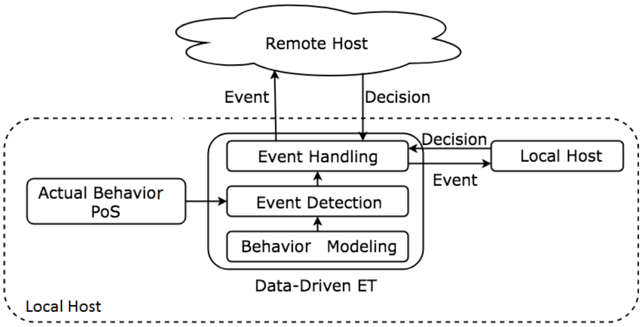


FIGURE 1. Coordinated data-driven event-triggering framework.

III. PROPOSED DATA-DRIVEN ET ARCHITECTURE

In the proposed ET architecture, a local and remote host are assumed, where the local host is regarded as the on-board entity, while the remote host is the off-board one, usually residing in a computing infrastructure. Both hosts assume that the system follows a mutually agreed behavior model under normal conditions. As long as the actual behavior of the system matches the model behavior, no communication between the two hosts is required. The local host uses the predefined behavior model in order to detect abnormalities of the system, based on some bounds that force the triggering of an event interrupt. The local host then decides whether to handle the event interrupt by itself or/and inform the remote host about the occurrence of the event interrupt. Similarly, the remote host can decide whether to handle the event interrupt by itself or/and inform other services about the occurrence of the event interrupt. The actions taken by the local/remote host can include updates or changes to the behavior model. Figure 1 shows a diagram of the proposed technique to aid understanding.

Specifically, as previously mentioned, ET has three phases, namely: behavior modeling, event detection, and event handling, which are described as follows:

1) **Behavior Modeling**: A model of the normal behavior of the system is developed from collected data. This model will be compared with the actual behavior of the system and if there is a deviation between the model and the actual behavior, an event is triggered.
2) **Event Detection**: Different event detection algorithms can be deployed to classify the deviation between the actual and model behavior. For this work, a simple threshold-checking approach is followed.
3) **Event Handling**: Event handling is a characteristic of the ET architecture. The event can be handled

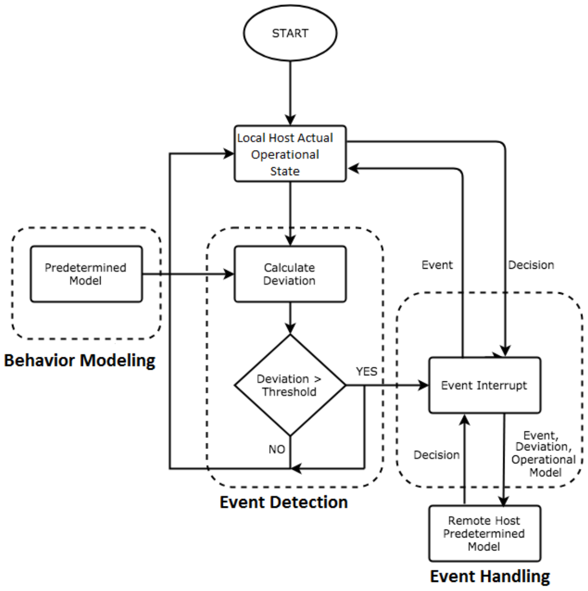


FIGURE 2. Logical diagram of proposed event triggering technique.

locally, remotely, or collaboratively. In this work events are handled collaboratively, as it will be explained in Section IV below.

In the proposed system, a predetermined model (for the SMET approach) or a set of models (for the MMET approach) of the application scenario considered is generated and used to monitor/track the system's elements. As long as the system's operation matches the operating model, no communication between the two hosts is required. However, when a deviation is observed between the model and the true system operation/state that exceeds a predefined threshold, an event interrupt is generated. The local host can then decide whether to simply send a resynchronization event or inform the remote host about the event and switch to a different prediction model as well (in the case of a multi-model system approach). Figure 2 shows a logical diagram of the proposed ET technique.

IV. PROBLEM DEFINITION AND OBJECTIVES

Without loss of generality, the proposed ET technique is utilized in order to monitor a fleet of public transport vehicles (buses) serving a particular route of the system as detailed in [7]. This is an important practical scenario, as monitoring public transport systems reduces unpredictability (that can potentially arise due to accidents, traffic congestion, etc.) and enhances the utilization of public transport [41], [42]. Simple solutions to the tracking problem of public transport vehicles includes the utilization of GPS sensors to collect location and timing data, that can be subsequently sent to controllers that are dedicated for traffic monitoring, providing the public transport users with arrival-time predictions, as well as the overall optimization of the transportation network [43]- [45]. As mentioned in Section II, data collection can be achieved either via periodic signaling (traditional approach) or event triggering in applications where recurrent mobility patterns exist and can be obtained utilizing and analyzing collected data. This is indeed the case for the scenario considered in this work, with buses serving a particular route. Thus, events can be generated only when unanticipated patterns are detected i.e., no communication is required when the movement of a bus matches its operating model.

In the scenario considered, and based on the system architecture description of Section III above, the local host can be an IoT device onboard a bus and the remote host a server collecting and processing the messages generated by the local host. Thus, in this work, a predetermined model (or set of models) of the bus mobility is generated and used to track the movement of the bus along its route. If the actual mobility (GPS readings) does not follow the model, then an event is triggered, the local host switches models and the information is relayed to the server (remote host) for resynchronization purposes. The (predicted) bus arrival time is also readjusted and relayed to the corresponding bus terminal for passenger notification.

The objective is to accurately track the movement of the buses, while effectively minimizing communication events and the related energy consumption at the local hosts, as well as processing at the local and remote hosts. Noticeably, the deviation threshold between the actual and predicted mobility can affect both the tracking accuracy and the volume of triggered events; clearly, the larger the threshold, the larger the deviation between actual and modeled behavior. Therefore, the deviation should be kept to a minimum by controlling the magnitude of the threshold.

A. BEHAVIOR MODELING

The following transportation scenario considered is described hereafter. There exist a bus route which consists of bus stops that belong to the set $b \in [1, 2, 3, \ldots, \mathcal{B}]$. The travel time between bus stops i and j, is defined as $d_{i,j}$, where $[i, j] \leq \mathcal{B}$ and $j > i$. Thus, a data matrix $\mathbf{D}_n \in \mathbb{R}^{\mathcal{B}\times\mathcal{B}}$ can be constructed having the following form.

$$\mathbf{D}_n = \begin{bmatrix} 0 & d_{1,2} & d_{1,3} & \ldots & d_{1,B} \\ 0 & 0 & d_{2,3} & \ldots & d_{2,B} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & d_{B-1,B} \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}$$

Each bus trip has a unique data matrix $\mathbf{D}_n$. As a result, for N bus trips there exists N different data matrices, $\mathbf{D}_n$, that belong to set $\mathcal{D} = \mathbf{D}_{n=1}^{N}$. As demonstrated experimentally in the sequel, the distribution of travel times $d_{i,j}$ has a Gaussian shape envelop and hence a Gaussian distribution is assumed for generating moments of the random variables $d_{i,j}$.

B. EVENT DETECTION

An event is defined as a diversion from the range $\tau_{i,j} \pm \alpha$, where $\tau_{i,j}$ is the mean of the distribution of the random variable $d_{i,j}$ (i.e., the 1^{st} moment), and α is the time bound.

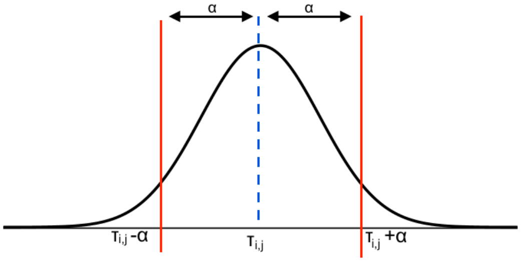


FIGURE 3. Random variable $d_{i,j}$ and violation bounds. Event is generated when actual movement is not within $\tau_{i,j} \pm \alpha$ (red lines).

Different values of α produce different number of events. Thus, the objective is to choose the appropriate value for α in order to produce the desirable number of events for a particular bus route. Figure 3 shows the distribution of time, $d_{i,j}$, for a segment of the route and the bounds beyond which a violation in the model is considered.

Clearly, the number of events, E, belongs to the set $E \in [0, 1, \dots \mathcal{B} - 1]$. For example, for $\alpha = 0$, $\mathcal{B} - 1$ events are generated, since a violation will occur in each segment of the bus route. On the other hand, for an adequately large value of α, such that the bounds include approximately the entire distribution, no events will occur. Consequently, the right value of α should be computed that produces the desired number of out-of-bound events (i.e., balancing the number of communication events with tracking accuracy).

C. EVENT HANDLING

In general, the local host can choose whether or not to inform a remote host of an event interrupt (since such an event might be an outlier). For the application scenario considered, the local host re-synchronizes with the remote host whenever an event interrupt is triggered. The local host informs the remote host of the deviation from the actual movement and indicates whether or not to switch to another operating model as well (for the MMET case). This procedure is illustrated in Fig. 2.

In this work, two different sets of information passed by the local host to the remote host are examined. Initially, in Section V the only information passed to the remote host is the deviation between the actual and modeled behavior (single-model case), while in Section VI the remote host is additionally informed whether and to which other model to switch (multi-model case).

V. SINGLE-MODEL ET

The performance of the ET technique using a single predetermined model is initially examined. The mobility model is first computed and then an algorithm is derived to compute the bounds α that lead to a particular number of events. The information provided to the remote host whenever there is an event interrupt is limited to the deviation (magnitude of α), since a single model is used.

Set $\mathcal{D}$ is sampled with independent and identically distributed (IID) normal random variables (RVs) with constant variance. Thereafter, $\mathcal{D}$ is split into a training subset, $\mathcal{D}_{train}$, and test subset, $\mathcal{D}_{test}$. $\mathcal{D}_{train}$ consist of 80% of the data of $\mathcal{D}$ while $\mathcal{D}_{test}$ consists of 20%. A behavior model is generated using set $\mathcal{D}_{train}$. Thereafter, and to further investigate the performance of the proposed ET algorithms under a broader set of parameters, a simulated set of traces is generated with these samples defined in set $\mathcal{S}$ and split into $\mathcal{S}_{train}$ and $\mathcal{S}_{test}$, where $\mathcal{S}_{train}$ is used for the training of the algorithms, while set $\mathcal{S}_{test}$ is used for their evaluation, similar to set $\mathcal{D}$.

For the analysis that follows, a bus route of 8 bus stops ($\mathcal{B} = 8$) is considered. The probability of a particular number of events $P_T(E, \alpha)$, as described in [7], depends on the total number of event interrupts, E, and the violation bound, α. For a particular target number of events, E, all possible combinations of E events can be calculated using the binomial coefficients. Hence, for a route with $\mathcal{B}$ bus stops, the number of combinations of E events is described as $c = \binom{\mathcal{B}-1}{E}$. These combinations are stored in a data matrix $\mathbf{C} \in \mathbb{R}^{c \times E}$ which has the following form:

$$\mathbf{C} = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ c \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & E \end{array} \\ \left[\begin{array}{c} \textit{Combination } 1 \\ \textit{Combination } 2 \\ \vdots \\ \textit{Combination } c \end{array} \right] \end{array}$$

Let $\bar{P}(i, j)$ represent the probability of no event occurring within segment $i \rightarrow j$ (from bus stop i to bus stop j):

$$h\bar{P}(i, j) = P(\tau_{i,j} - \alpha \leq d_{i,j} \leq \tau_{i,j} + \alpha) \tag{1}$$

Consequently, the probability of no events occurring from i to $j - 1$ until an event occurs at j can be defined as:

$$P(i, j) = \bar{P}(i, i + 1) \times \dots \bar{P}(i, j - 1) \times (1 - \bar{P}(i, j)) \tag{2}$$

The first and last events should also be taken into account. The first event represents the probability of no event to be generated until the i^{th} segment, and it is derived in Eq. (3). The last case represents the probability of no event to be generated between the j^{th} segment (where the last event occurred) and the end of the route, as detailed in Eq. (4).

$$\begin{aligned} P(1, i) &= \bar{P}(1, 2) \times \bar{P}(1, 3) \times \dots \bar{P}(1, i - 1) \\ &\quad \times (1 - \bar{P}(1, i)) \qquad (3) \\ P(j, |\mathcal{B}|) &= \bar{P}(j, j + 1) \times \bar{P}(j, j + 2) \times \dots \bar{P}(j, |\mathcal{B}|) \qquad (4) \end{aligned}$$

Summarizing, the probability of a particular number of events to occur is then:

$$\begin{aligned} P_T(E, \alpha) = \sum_{i=1}^{c} \prod_{e=1}^{E-1} \big[& P(\mathbf{C}_{ie}, \mathbf{C}_{ie+1}) \times P(1, \mathbf{C}_{i1}) \\ & \times P(\mathbf{C}_{ie}, |\mathcal{B}|) \big] \end{aligned} \tag{5}$$

where $P(\mathbf{C}_{ie}, \mathbf{C}_{ie+1})$, $P(1, \mathbf{C}_{i1})$, and $P(\mathbf{C}_{ie}, |\mathcal{B}|)$ can be obtained by Eqs. (2), (3), and (4), respectively.

To aid understanding, Fig. 4 evaluates Eq. (5) by varying α in the range between [0, 250] and demonstrating the shape of $P_T(E, \alpha)$ for all values of $E \in [0, 1, \dots \mathcal{B} - 1]$.

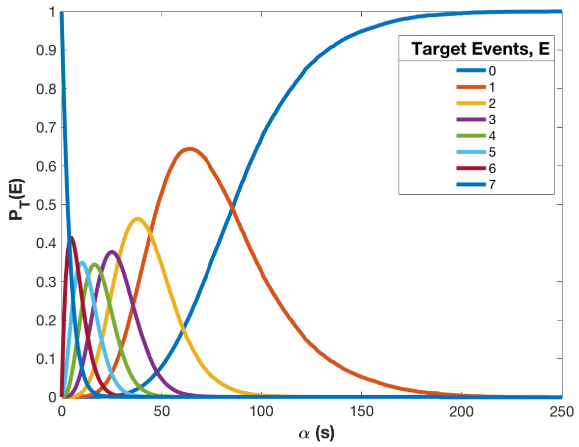


FIGURE 4. $P_T(E)$ versus α, for $\mathcal{B}$=8 (for different number of events E).

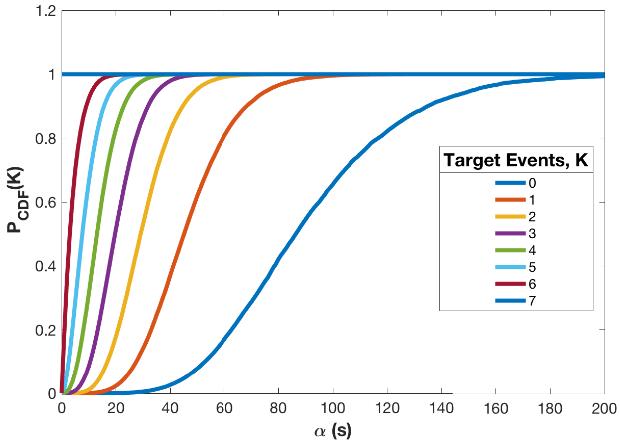


FIGURE 5. $P_{CDF}(K)$ versus α, for $\mathcal{B}$=8 (for different number of target events K).

As expected, the probability of no events occurring for very large bounds is equal to 1, $P_T(0, \alpha > 200) = 1$. As the bound gets smaller, the probability of a greater number of events to occur increases and for a specific target number of events decreases. Additionally, for $\alpha = 0$ the probability of $\mathcal{B}-1$ events to occur is 1, $P_T(\mathcal{B}-1, 0) = 1$. Noticeably, for any bound $\alpha = \alpha_0$, the sum of $P_T(E, \alpha_0)$ is equal to 1, i.e.,

$$\sum_{E=0}^{\mathcal{B}-1} P_T(E, \alpha) = 1, \quad \alpha \in \mathbb{R}, \alpha \geq 0 \tag{6}$$

For example, for the case of $\alpha = 50$ the following probabilities are observed in Fig. 4: $P_T(0, 50) = 0.087$, $P_T(1, 50) = 0.5391$, $P_T(2, 50) = 0.3288$, $P_T(3, 50) = 0.0433$, $P_T(4, 50) = 0.0018$, $P_T(5, 50) = 0$, $P_T(6, 50) = 0$, $P_T(7, 50) = 0$.

The objective is to find the value of α such that the number of events to be generated is no larger than the desired number of events, K. For this purpose, the cumulative distribution function (cdf) of $P_T(E, a)$ is calculated, as expressed below:

$$P_{CDF}(K, \alpha) = \sum_{E=0}^{K} P_T(E, \alpha), \quad \alpha \in \mathbb{R}, \alpha \geq 0 \tag{7}$$

The distributions for $P_{CDF}(K, a)$ versus α are plotted in Fig. 5 for different target events K. The calculation of $P_{CDF}(K, a)$ enables the calculation of α for any target $K \in [0, 1, \ldots \mathcal{B}-1]$ and also for any percentage of the target K, $p_{target} \in [0, 1]$. For example, a test set of 100 samples tuned with parameters $p_{target} = 0.8$ and $K = 4$, should produce 80 bus trips of less or equal to 4 events and 20 bus trips greater than 4 events. Consequently, the desired α_d is defined as the minimum value of α that drives $P_{CDF}(K, \alpha)$ to become p_{target}, i.e., $P_{CDF}\left(K, a_d\right) = p_{target}$.

In the aformentioned expression, the only unknown parameter is the bound α_d. Numerical methods such as the bisection method [46] can be implemented to calculate the value of α_d (as shown in Alg. 1 below). Using the bisection method, initially α_d is assumed to lie within the interval (α_l, α_u). $P_{CDF}(K, \alpha)$ of $\alpha = \frac{\alpha_l+\alpha_u}{2}$ is compared with p_{target} and the interval is updated accordingly. The process repeats until a predetermined precision value ϵ is reached ($\alpha_u - \alpha_l > \epsilon$).

Algorithm 1 Calculating Bounds (α_d) for Desired K, p_{target}

```
Input: $\alpha_l$, $\alpha_u$, $\epsilon$, $K$, $p_{target}$
while $\alpha_u - \alpha_l > \epsilon$ do
    Calculate $\alpha = \frac{\alpha_l+\alpha_u}{2}$ & $P_{CDF}(K, \alpha)$ using Eq. (7)
    if ($P_{CDF}(K, \alpha) > p_{target}$) then
        $\alpha_u = \alpha$
    else
        $\alpha_l = \alpha$
    end if
end while
return $\alpha_d = \alpha$
```

VI. MULTI-MODEL ET

In this section, the single-model ET technique presented in Section V is extended to consider a set of behavior models that can be used to estimate the evolution of the system. At each time instance, both the local and remote host operate on a single model, ξ, but switch between different behavior models when deemed necessary. This potential is examined hereafter to address situations where trace samples are correlated and that knowledge can be used to enhanced tracking accuracy.

A. MODEL BEHAVIOR

Unlike the single-model case described in Section V, hereafter we consider a set Ξ that comprises of a number of operational models, ξ_i, for each segment of the bus route, $i \in [2, \mathcal{B}]$. To select a specific operating model, the distributions of the travel times in each road segment are split into areas of distribution (AoD) of equal size. Then, both local and remote hosts operate on a specific predetermined model, $\xi_i^b \in \Xi$, at any one time. To aid understanding, initially a set of 3 models are considered and an arbitrary number of models

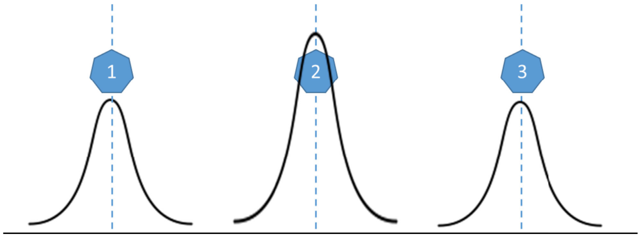


FIGURE 6. Three-model schematic.

is assumed thereafter. Figure 6 illustrates the case of a set that comprises of 3 pre-determined models.

B. EVENT DETECTION AND EVENT HANDLING

The event detection for the single-model case sets the bounds of violation, α, around the mean, $\tau_{i,j}$, of the distribution $d_{i,j}$. In the 3-model case, this corresponds to setting the bounds of violation around the mean of the model with index number 2 (mean of normal distribution, i.e., area 2 of Fig. 6).

Therefore, it is reasonable for the case of set $\mathcal{S}$ (i.e., set of simulated samples) to set bounds of violation around the mean of the operating model, ξ. The following procedure is followed for choosing the new operating model, ξ': if an event occurs, change operating model, ξ, to the one which is nearest to the violation and set bounds around its mean, $\tau_{i,j}(\xi')$. The local host will additionally inform the remote host about the event and the new operating model, ξ' (see Fig. 2).

To compute bounds for the case of correlated data, an updated method is followed to calculate $P_{CDF}(K, \alpha)$. This method is described in Algs. 2 and 3 below.

Algorithm 2 Calculation of $P_{CDF}(K, \alpha)$

```
Input: $models, S_{train}, \alpha, K$
Output: $P_{CDF}(K, \alpha)$
Compute $\tau_m$
for $i = 1$ to $|S_{train}|$ do
    Initialization: $\xi = 1$
    for $j = 2$ to $\mathcal{B}$ do
        if $\tau_m(\xi) - \alpha \leq S_{train}(1, j, i) \leq \tau_m(\xi) + \alpha$ then
            $\rho(i) = \rho(i) + 1$
            Compute nearest model
            Update $\xi$ to nearest model $\xi'$
        end if
    end for
end for
Compute $P_{CDF}(K, \alpha)$ using matrix $\rho$
return $P_{CDF}(K, \alpha)$
```

Initially, the mean of each model is computed and stored in τ_m. The operating model is defined by ξ and it is initialized to one (1). For each trace i of S_{train} (i.e., for each trace in set of simulated samples used for training), the number of events that are generated for the particular bound, α, are calculated. Each time an event is generated, ξ is changed to the model nearest to the violation, ξ'. The number of events generated for each trace is stored in matrix ρ. The cumulative sum of matrix ρ can then be computed and subsequently $P_{CDF}(K, \alpha)$ is calculated for the desired α and K. Thereafter, the bisection method is used to find a_d that produces the desired K and p_{target}. To compute these values, Alg. 3 is used.

Algorithm 3 Calculation of Bounds for Desired K, p_{target}

```
Input: $\alpha_l, \alpha_u, \epsilon, K, p_{target}$
Output: $\alpha_d$
    LOOP Process
while $\alpha_u - \alpha_l > \epsilon$ do
    Compute $\alpha = \frac{\alpha_l + \alpha_u}{2}$
    Compute $P_{CDF}(K, \alpha)$ using Algorithm 2
    if $(P_{CDF}(K, \alpha) > p_{target})$ then
        $\alpha_u = \alpha$
    else
        $\alpha_l = \alpha$
    end if
end while
return $\alpha_d$
```

VII. PERFORMANCE RESULTS

A. PERFORMANCE METRICS

To evaluate the performance of the prediction model compared to the actual system state, a loss function is considered. For this purpose, a suitable error measure is chosen based on the absolute difference between p_{target} and the achieved $P_{CDF}(K, \alpha_d)$. Then, the loss function is defined as the mean value of the error measured over all $K \in [0, 1, 2, \ldots, \mathcal{B} - 1]$, mathematically defined as follows:

$$\text{Loss Function} = mean\left\{\left|P_{CDF}(K, \alpha_d) - p_{target}\right|\right\}_{K=0}^{\mathcal{B}-1} \tag{8}$$

For example, for the evaluation of the proposed SMET algorithm for the analytical scenario considered, $N = 5000$ samples are utilized with $p_{target} = 0.98$, in order to compute bound α_d. As shown in Fig. 7, α_d drops non-linearly with increasing values of triggering events, K.

The performance of the algorithm is evaluated by applying the calculated bounds, α_d, on $\mathcal{D}_{test}$. Figure 8 shows the cumulative percentage of events generated using the calculated α_d for $K = 4$. The proposed algorithm achieves a $P_{CDF}(4, \alpha_d) = 0.981$ which corresponds to an error of 0.001. Repeating this procedure and averaging for all values of N, allows the calculation of the mean test error.

TABLE 1. Test error for different values of N.

N	100	1000	5000	10000
Mean Test Error	0.4	0.25	0.005	0.001

Table 1 shows the mean test error for different values for the number of samples used (i.e., N). Undoubtedly, the greater the number of samples used for training, the better the estimation of α_d and the smaller the mean test error. It is observed that for the case of $N = 10000$ the model produces

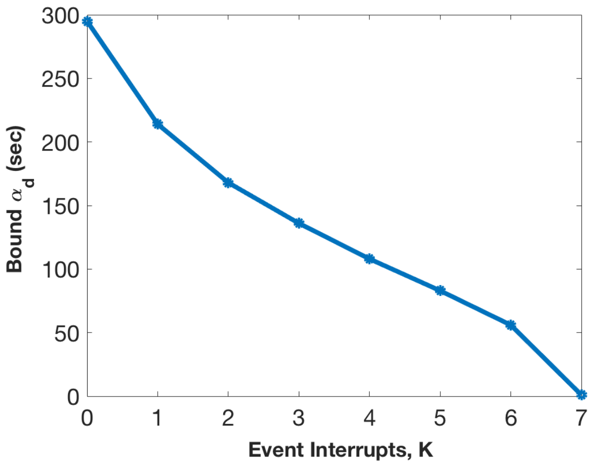


FIGURE 7. Values of the bound, α_d, for different K values and $P_{target} = 0.98$.

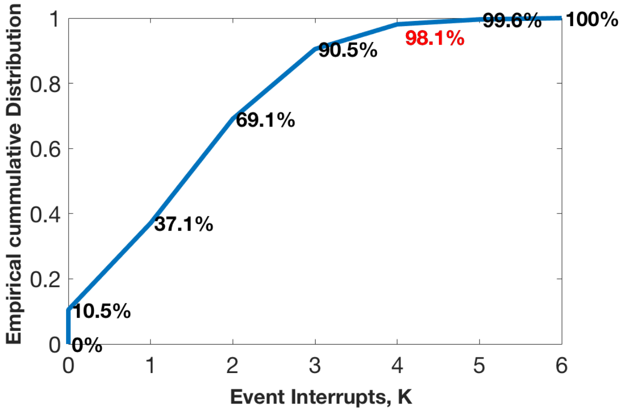


FIGURE 8. Cumulative distribution of events for α_d calculated for various values of K (for the case of $K = 4$, test error=0.1% and bound=108 sec).

almost perfect results as the difference between prediction and the real values (i.e., the error) is approximately zero.

B. EXPERIMENTAL EVALUATION

1) DATA PROCESSING

To evaluate the proposed technique, the algorithm was applied on a real-world public transport scenario, with data provided by the Transportation Organization of Nicosia (OSEL) in Cyprus. The data consists of locations (longitude, latitude) and timestamps of 4 buses operating on a particular service (Route 259), with $\mathcal{B} = 58$ bus stops.

Initially a pre-processing step is introduced in order to provide a useful input set of data. Thereafter, the application of the SMET and MMET techniques is assessed. Specifically, the pre-processing step involves removing invalid traces (i.e., movements when out-of-service) in order to generate set $\mathcal{D}_{real}$.

This is clearly illustrated in Fig. 9 that shows the data collected for Route 259 and Fig. 10 that shows a plot of the locations of the bus trips generated after the pre-processing procedure is applied. Evidently, this pre-procedure successfully filters out almost all unwanted traces, clearly distinguishing Route 259. In particular, from this pre-processing phase, $\mathcal{D}_{real}$ results in $N = 311$ bus trips, with the envelope of the empirical distribution of $d_{i,j}$ resembling the Poisson distribution as demonstrated in Fig. 11.

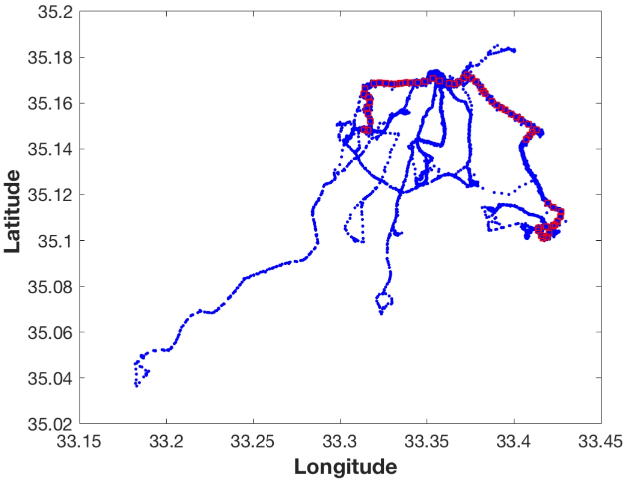


FIGURE 9. Bus trace data for Route 259 before pre-processing. Blue dots correspond to bus traces; Red dots correspond to bus stops.

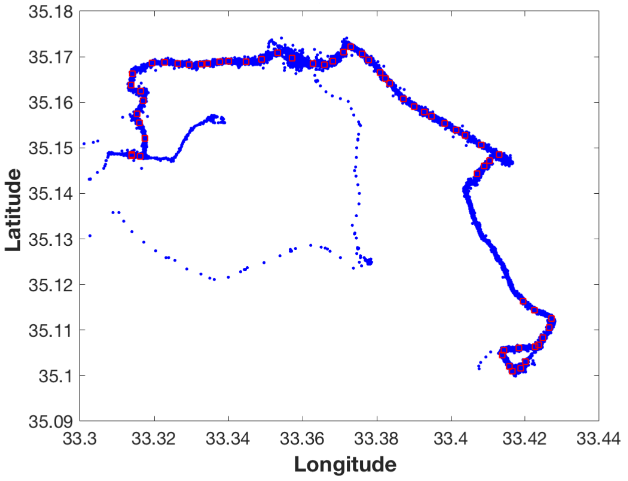


FIGURE 10. Bus trace data for Route 259 after pre-processing. Blue dots correspond to bus traces; Red dots correspond to bus stops.

Further, as discussed in Section VII-A, the greater the number of bus trips N, the greater the accuracy of the algorithm. Hence, and in order to minimize the test error, artificial samples are generated that follow the observed bus behavior.

To examine the behavior of the bus trips, distributions $d_{1,j}$, for $j : 2 \rightarrow \mathcal{B}$, are divided into equal-size areas, previously denoted as Areas of Distribution (AoD). Then, the Markov chain transition matrix (MCTM), $\mathbf{P}$, is calculated with a number of states equal to the number of AoDs. MCTM contains the probabilities of transition from area k of $d_{1,j}$ to area t of $d_{1,j+1}$, where t, k are the indices of the specific AoDs. Figures 12 and 13 shows the transition matrix, $\mathbf{P}$, and the state diagram for the case of 5 states. The element $P_{i,j}$ of data matrix $\mathbf{P}$ in Fig. 12 represents the probability of transition from state t to state k, which corresponds to the probabilities

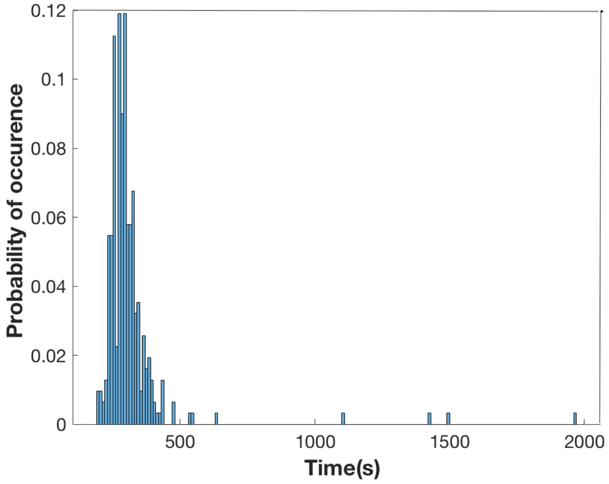


FIGURE 11. Histogram of traveled time between bus stops 1 and 8 ($d_{1,8}$).

$$\mathbf{P}_5 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} \;\;1\;\; & \;\;\;2\;\;\; & \;\;\;3\;\;\; & \;\;\;4\;\;\; & \;\;\;5\;\;\; \end{array} \\ \begin{bmatrix} 0.9055 & 0.0920 & 0.0018 & 0.0003 & 0.0004 \\ 0.0966 & 0.8779 & 0.0232 & 0.0021 & 0.0002 \\ 0 & 0.0950 & 0.8587 & 0.0463 & 0 \\ 0 & 0 & 0.2481 & 0.7296 & 0.0222 \\ 0 & 0 & 0 & 0.1149 & 0.8851 \end{bmatrix} \end{array}$$

FIGURE 12. Transition matrix for the case of 5 states.

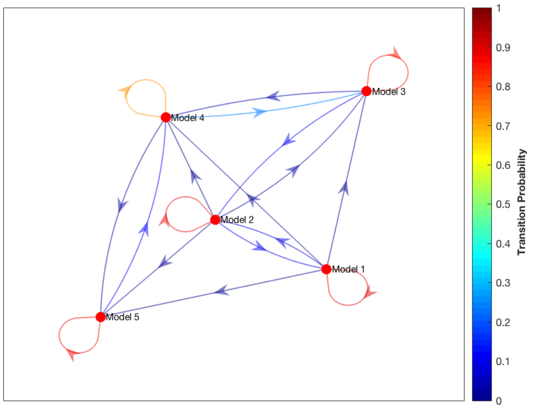


FIGURE 13. State diagram for P_5.

of transition from area k of $d_{1,j}$ to area t of $d_{1,j+1}$ as explained above.

From Figs. 12 and 13, it is evident that there exists a strong dependency between the previous and the next state. Evidently, when in a particular state, it is very probable to continue operating in the same state, as the probabilities of the diagonal of matrix $\mathbf{P}$ are significantly higher than the rest of the values in the matrix.

Another key insight is that a large number of states is required to produce artificial bus trips that resemble the real-world scenario. Hence, the performance of this technique is evaluated by its accuracy, defined as the mean absolute difference between the states produced by the real data and the one produced by model-based simulations, over all states and all segments of the bus route.

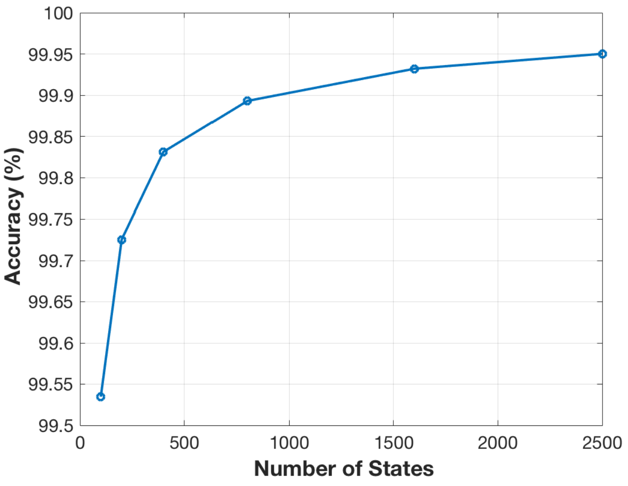


FIGURE 14. Simulation accuracy vs number of states.

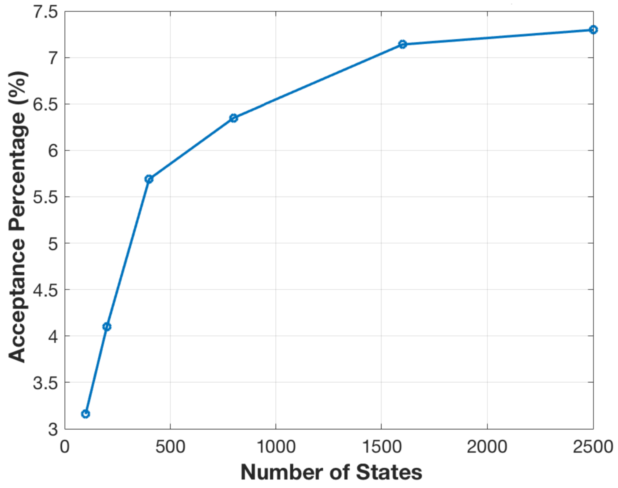


FIGURE 15. Simulation acceptance percentage vs number of states.

Furthermore, due to the location uncertainty there are several mobility traces where some vehicle positions appear at an arbitrary time instance before locations of vehicles in previous time steps. Therefore an acceptance percentage of valid simulated bus trips over all trips generated is defined (e.g., an acceptance percentage of 10% on a 1000-sample simulation, corresponds to 100 valid samples) and those traces with non-monotonically progressing trips are filtered out of the evaluation. To illustrate this, Figs. 14 and 15 show the accuracy and the acceptance percentage of simulated data over an increasing number of states. It is observed that a simulation accuracy of at least 99.5% is achieved. However, the acceptance percentage is low. This will influence the simulation time as a large number of the generated samples would be rejected (it should be noted though that this will not have a significant impact, as the process is performed offline).

To summarize, artificial samples produced using the MCTM method provide a reliable framework to simulate

$$\mathbf{P}_{D3} = \begin{array}{c|ccc|} & 1 & 2 & 3 \\ 1 & 0.0543 & 0.8564 & 0.0893 \\ 2 & 0.0700 & 0.8502 & 0.0798 \\ 3 & 0.0849 & 0.8417 & 0.0733 \end{array}$$

FIGURE 16. Transition matrix for the 3-state model, set $\mathcal{D}$.

$$\mathbf{P}_{S3} = \begin{array}{c|ccc|} & 1 & 2 & 3 \\ 1 & 0.9437 & 0.0555 & 0.0008 \\ 2 & 0.0921 & 0.8919 & 0.0160 \\ 3 & 0 & 0.0480 & 0.9520 \end{array}$$

FIGURE 17. Transition matrix for the 3-state model, set $\mathcal{S}$.

bus trips and can further be used to evaluate this stochastic processes accurately (as also discussed in [47]). For the performance evaluation that follows, simulations are carried out with 200000 samples and 1000 states. This leads to 11622 valid bus trip samples.

a: SSMET - EXPERIMENTAL RESULTS

Initially, the algorithm described in Section V is used with $\mathcal{S}_{train}$ to compute the operating bounds for the single-model case. In Fig. 18 below, the bound α_d computed for several values of K and $p_{target} = 0.98$ is shown (in blue for the SMET case). As it can be observed, the bounds calculated are relatively large, resulting to a low tracking accuracy. For example, the bound for $K = 45$ is calculated to be 533 seconds (around 9 minutes). The main reason behind this is the inconsistent variance of the distributions of $d_{i,j}$. In addition, in this case the samples in the simulation are correlated (samples are not IID). This results in an increasing number of events, since a violation that occurs in a particular segment $d_{i,j}$ will likely lead to violation of bounds at the next segment, $d_{i,j+1}$, and so on. As previously discussed, this shortcoming can be addressed by the MMET technique, where multiple models are considered (MMET) and switching between these models takes place whenever deemed necessary. This is shown below, where the experimental results of the MMET approach are presented, clearly demonstrating the advantages of MMET over the single-model approach.

b: MMET - EXPERIMENTAL RESULTS

For multi-model event triggering, the same real-world scenario concerning bus Route 259 as the one described in the previous section is considered. Again, the behavior of the bus trip should be examined in order to adapt the new algorithm and generate the set of predetermined models accordingly (i.e., define the AoDs and calculate the MCTMs). In this case, we first investigate a simple scenario of two 3-state MCTMs produced using the IID set $\mathcal{D}$ and the real-world values set $\mathcal{S}$. The MCTMs are shown in Figs. 16 and 17, respectively.

Clearly, there is a difference between the two MCTMs of the two sets, as samples generated in set $\mathcal{D}$ are IID. This results in the rows of the P_{D3} having the same behavior and the same distribution (i.e., a Gaussian shape envelope with the mean positioned in area 2 of Fig. 3). In other words, the probabilities of transition do not depend on the AoD where the current observed sample is obtained from.

TABLE 2. Mean test error and accuracy of sets.

Set	$\mathcal{S}_{test}$	$\mathcal{D}_{real}$
Mean Test Error	0.0022	0.0120
Accuracy(%)	99,7809	98,8039

However, this is not the case for P_{S3}. The probabilities of transition now do depend on the current AoD. It is observed that each row of P_{S3} forms a Gaussian shape envelope with the mean positioned in the current AoD. Therefore, each area of P_{S3} can be considered as a different pre-determined model.

To evaluate the proposed MMET technique, Alg. 3 is applied on set $\mathcal{S}_{train}$ to calculate α_d. Figure 18 shows the bounds generated using this algorithm for several values of K and $p_{target} = 0.98$. It is observed that the violation bounds using the new algorithm are much smaller compared to the single-model approach. Particularly, for the case of $K = 45$, bounds are calculated at $a_d = 154$, which is more than 3× smaller than the bounds calculated by the single-model case. The performance of the algorithm is tested using simulation-based synthetic set S_{test} and real world-based data, D_{real}. The mean of test errors is shown in Table 2.

Unsurprisingly, the error of $\mathcal{D}_{real}$ is higher than the one of $\mathcal{S}_{test}$, since the algorithm is trained on $\mathcal{S}_{train}$. However, both results provide high accuracy (greater than 98.8%), illustrating that the new algorithm provides a better trade-off between the magnitude of the bounds (which directly translates to the number of events generated) and the tracking accuracy, while keeping the test error negligible. Note that the approach just described can effortlessly be extended to larger number of behavior models. For instance, Fig. 18 shows the bounds generated for several values of K and $p_{target} = 0.98$ for an increasing number of models. As shown in the figure, the greater the number of models, the smaller the violation bounds. Importantly, by observing the difference of the single-model bounds with those of the 18-model case, for $K = 45$, in the first case $a_d = 533$ seconds and in the second case $a_d = 27$ seconds, resulting in 20× reduction in the magnitude of the bound.

Smaller bounds are in general preferable, as a higher precision of the violation leads to better tracking accuracy. Expectedly though, an increasing number of models provides diminishing returns. For instance, the bounds for the cases of 12, 15, and 18 models are very similar. This is more clearly demonstrated in Fig. 19, which shows the bounds produced for $K = 40$ and $p_{target} = 0.98$ for an increasing number of models. Tables 3 and 4 present the mean test error and the accuracy for the test- and real-world data sets for different numbers of models. Clearly, the mean test error for both sets is kept constant despite the reduction in the bounds' magnitude, demonstrating the efficient performance of the proposed multi-model ET technique in achieving lower tracking accuracy while retaining a small mean test error.

Another measure of the efficiency of the proposed approach is its predictability, i.e., the ability of the approach

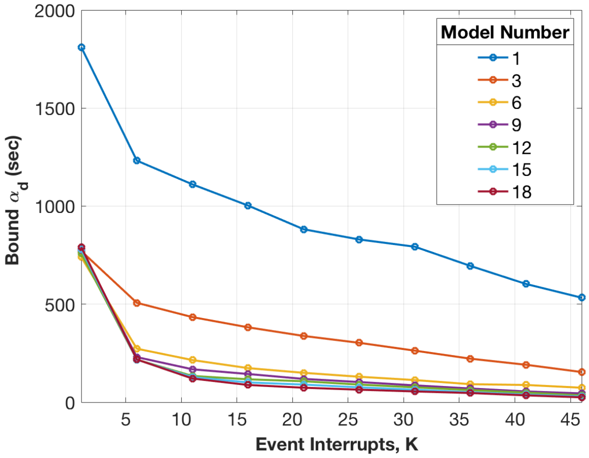


FIGURE 18. Bounds (α_d) vs event interrupts for different numbers of models.

TABLE 3. Mean test error and accuracy of set $\mathcal{S}_{test}$.

Model Number	Mean Test Error	Accuracy(%)
1	0.001	99.95
3	0.0022	99.7809
6	0.0071	99.2941
9	0.0038	99.6212
12	0.0066	99.3413
15	0.0043	99.5738
18	0.0039	99.6124

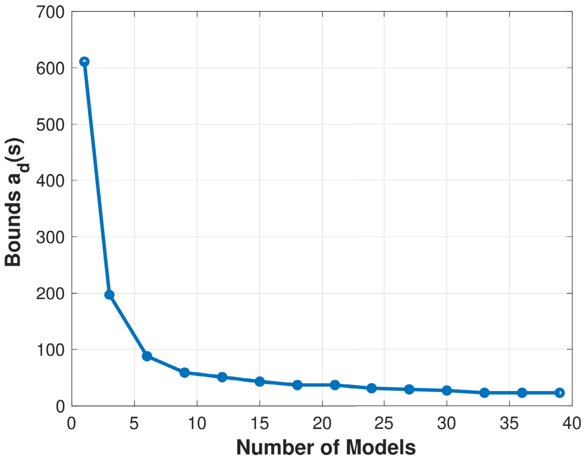


FIGURE 19. Bounds for $K = 40$ and $p_{target} = 0.98$ for an increasing number of models.

TABLE 4. Mean test error and accuracy of set $\mathcal{D}_{real}$.

Model Number	Mean Test Error	Accuracy(%)
1	0.01	99.981
3	0.0120	98.8039
6	0.0103	98.9749
9	0.0123	98.7717
12	0.0152	98.4823
15	0.0107	98.9325
18	0.0110	98.9003

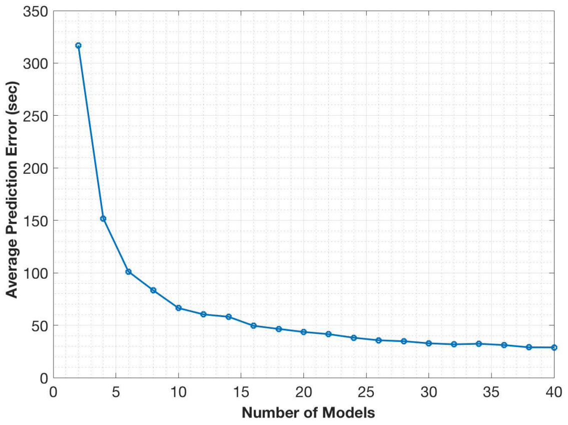


FIGURE 20. Average prediction error for different number of models.

to predict future times of bus schedules. For this concept, the operating model from current segment K of the route is used to calculate the predicted time of the next segment of the route, $K + 1$. The prediction error is defined as the average deviation between the predicted and actual bus travel times. Figure 20 illustrates the average prediction error (in seconds) for different number of models. From the figure, a highly non-linear decreasing behavior is observed between the average prediction error and the number of models. For instance, for a small number of models the average prediction error is high. This is reasonable, as the number of models is too small to describe the behavior of the transportation scenario considered. However, as the number of models increases, the average prediction error exponentially reduces, finally converging to a value of approximately 30 seconds.

In summary, the proposed multi-model technique manages to provide a very low tracking error for a significantly lower number of event interrupts k as compared to the single-model baseline; for a mere 25 events the prediction error reduces to approximately 30sec. At any other time there is no need for information exchange between the local and the remote host and there is no need for any additional computation to be performed for the monitoring process. In this way, the proposed MMET approach achieves $\sim 9\times$ reduction in triggering events compared to SMET (as shown in Fig. 18); in turn, this reduction directly translated to a nine-fold reduction in communication circuitry energy consumption in addition to the additional savings in energy from the reduced tracking and updating computations both at the local and remote hosts. Further, the proposed multi-model technique achieves a $5\times$ performance gain as compared to a periodic scheme that updates the vehicle state every 30 seconds over an 1-hour journey (since 120 interrupts will be required by the periodic scheme, as compared to only 25 for the proposed approach), again resulting in a five-fold reduction in energy consumption.

Finally, the reader should note that, as expected, for all three schemes there is clearly a trade-off between energy consumption (that directly relates to the number of events triggered) and tracking accuracy. Nevertheless, considering the performance comparison of MMET, SMET, and periodic (baseline) approaches, it is evident from Fig. 18 that MMET can provide better performance in terms of tracking accuracy at a lower triggering of communication events and thus reduced communication energy consumption.

VIII. CONCLUSION

In this work, single- and multi-model ET techniques are utilized for the prediction and tracking of recurrent patterns in public transportation systems. The single-model technique (benchmark), is able to provide good tracking accuracy for the application scenario under consideration, while limiting communication between the system's components by triggering events only when the actual and modeled behaviors of the system deviate. Nevertheless, it does present shortcomings in terms of the number of events generated and system response times. This has motivated the development of a multi-model approach, that through analytical, simulation, and experimental results was shown that it is capable of computing the smallest bounds that lead to a particular number of events with high precision as well as increasing the predictability of the ET technique. This is mainly achieved by deriving multiple models to accurately represent the system state and adapting to system changes by switching to the most accurate model that best represents the underlying system settings. Performance results, utilizing a real dataset for a fleet of buses operating in the city of Nicosia, Cyprus, have shown that for MMET the improvement in both tracking accuracy and energy efficiency is significantly higher as compared to the SMET and periodic triggering approaches, demonstrating the validity and feasibility of the proposed technique.

Future research directions initially include integrating the event triggering architecture into safety applications for CAVs in order to minimize messages exchanged with the road side units (RSUs) and enable faster processing of location information. Moreover, the aim is to introduce event triggering within the general age of information paradigm, in terms of queueing/processing delays incurred at the RSUs, in order to better optimize the CAVs' tracking accuracies.

REFERENCES

[1] C. Gray, R. Ayre, K. Hinton, and R. S. Tucker, "Power consumption of IoT access network technologies," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 2818–2823.

[2] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy aware localization for mobile devices," in *Proc. 8th Int. Conf. Mobile Syst., Appl.*, 2010, pp. 285–298.

[3] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *Proc. 8th Int. Conf. Mobile Syst., Appl.*, 2010, pp. 315–330.

[4] A. Fu and J. A. McCann, "Dynamic decentralized periodic event-triggered control for wireless cyber–physical systems," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 4, pp. 1783–1790, Jul. 2021, doi: 10.1109/TCST.2020.3016131.

[5] A. Kosta, N. Pappas, and V. Angelakis, "Age of information: A new concept, metric, and tool," *Found. Trends Netw.*, vol. 12, no. 3, pp. 162–259, 2017.

[6] N. Ahmed and M. I. Hussain, "Periodic traffic scheduling for IEEE 802.11ah networks," *IEEE Commun. Lett.*, vol. 24, no. 7, pp. 1510–1513, Jul. 2020.

[7] P. Kolios, C. Panayiotou, G. Ellinas, and M. Polycarpou, "Data-driven event triggering for IoT applications," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1146–1158, Dec. 2016.

[8] *Internet of Things (IoT) Connected Devices Installed Base Worldwide From 2015 to 2025 (in Billions)*. Accessed: Jan. 10, 2022. [Online]. Available: https://www.statista.com

[9] *Internet of Things Forecast*. Accessed: Feb. 15, 2022. [Online]. Available: https://www.ericsson.com

[10] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based positioning for smartphones," in *Proc. 8th Int. Conf. Mobile Syst., Appl.*, 2010, pp. 299–314.

[11] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "VTrack: Accurate energy-aware road traffic delay estimation using mobile phones," in *Proc. 7th 8th ACM Conf. Embedded Netw. Sensor Syst.*, 2009, pp. 85–98.

[12] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava, "SensLoc: Sensing everyday places and paths using less energy," in *Proc. 8th ACM Conf. Embedded Netw. Sensor Syst.*, 2010, pp. 43–56.

[13] J. Muckell, J.-H. Hwang, V. Patil, C. T. Lawson, F. Ping, and S. Ravi, "SQUISH: An online approach for GPS trajectory compression," in *Proc. 2nd Int. Conf. Comput. Geospatial Res. Appl.*, 2011, pp. 1–8.

[14] D. Ashbrook and T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," *Pers. Ubiquitous Comput.*, vol. 7, no. 5, pp. 275–286, 2003.

[15] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica, Int. J. Geograph. Inf. Geovis.*, vol. 10, no. 2, pp. 112–122, Dec. 1973.

[16] J. Hershberger and J. Snoeyink, "An $O(n\log n)$ implementation of the Douglas-Peucker algorithm for line simplification," in *Proc. 10th Annu. Symp. Comput. Geometry*, 1994, pp. 383–384.

[17] G. Kellaris, N. Pelekis, and Y. Theodoridis, "Trajectory compression under network constraints," in *Proc. 11th Int. Symp. Spatial Temporal Databases*, 2009, pp. 392–398.

[18] N. Meratnia and A. Rolf, "Spatiotemporal compression techniques for moving point objects," in *Proc. 9th Int. Conf. Extending Database Technol.*, 2004, pp. 765–782.

[19] T. H. N. Vu, K. H. Ryu, and N. Park, "A method for predicting future location of mobile user for location-based services system," *Comput. Ind. Eng.*, vol. 57, no. 1, pp. 91–105, Aug. 2009.

[20] Y. Ohsawa, K. Fujino, H. Htoo, A. T. Hlaing, and N. Sonehara, "Real-time monitoring of moving objects using frequently used routes," in *Proc. 16th Int. Conf. Database Syst. Adv. Appl.*, 2011, pp. 119–133.

[21] A. B. M. Musa, J. Biagioni, and J. Eriksson, "Trading off accuracy, timeliness, and uplink usage in online GPS tracking," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 2124–2136, Aug. 2016.

[22] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 3270–3285.

[23] O. Vermesan and P. Freiss, *Internet of Things—Converging Technologies for Smart Environments and Integrated Ecosystems*. Denmark, Europe: River, 2014.

[24] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[25] W.-H. Lin and J. Zeng, "Experimental study of real-time bus arrival time prediction with GPS data," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1666, no. 1, pp. 101–109, Jan. 1999.

[26] S. I.-J. Chien, Y. Ding, and C. Wei, "Dynamic bus arrival time prediction with artificial neural networks," *J. Transp. Eng.*, vol. 128, no. 5, pp. 429–438, 2002.

[27] J. Patnaik, S. Chien, and A. Bladikas, "Estimation of bus arrival times using APC data," *J. Public Transp.*, vol. 7, no. 1, pp. 1–20, Mar. 2004.

[28] D. Sun, H. Luo, L. Fu, W. Liu, X. Liao, and M. Zhao, "Predicting bus arrival time on the basis of global positioning system data," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2034, no. 1, pp. 62–72, Jan. 2007.

[29] M. Chen, X. Liu, J. Xia, and S. I. Chien, "A dynamic bus-arrival time prediction model based on APC data," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 19, no. 5, pp. 364–376, Sep. 2004.

[30] S. Kartakis, A. Fu, M. Mazo, and J. A. McCann, "Communication schemes for centralized and decentralized event-triggered control systems," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 6, pp. 2035–2048, Nov. 2018.

[31] P. Huang, C. Wang, and L. Xiao, "RC-MAC: A receiver-centric MAC protocol for event-driven wireless sensor networks," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 1149–1161, Feb. 2015.

[32] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. Cham, Switzerland: Springer, 2008.

[33] D. Castanon and D. Teneketzis, "Distributed estimation algorithms for nonlinear systems," *IEEE Trans. Autom. Control*, vol. AC-30, no. 5, pp. 418–425, May 1985.

[34] A. Nayyar, A. Mahajan, and D. Teneketzis, "Decentralized stochastic control with partial history sharing: A common information approach," *IEEE Trans. Autom. Control*, vol. 58, no. 7, pp. 1644–1658, Jul. 2013.

[35] M. Zhong and C. G. Cassandras, "Asynchronous distributed optimization with event-driven communication," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2735–2750, Dec. 2010.

[36] Y. Khazaeni and C. Cassandras, "A new event-driven cooperative receding horizon controller for multi-agent systems in uncertain environments," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 409–422, Dec. 2018.

[37] P. Panagi and M. M. Polycarpou, "A coordinated communication scheme for distributed fault tolerant control," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 386–393, Feb. 2013.

[38] T. Schlage and J. Lunze, "Modeling of networked systems for remote diagnosis," in *Proc. Conf. Control Fault Tolerant Syst.*, 2010, pp. 795–800.

[39] P. Kolios and G. Ellinas, "Model-adaptive event-triggering for efficient public transportation tracking," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2019, pp. 682–686.

[40] L. Papachristoforou, P. Kolios, C. Panayiotou, and G. Ellinas, "Adaptive multi-model monitoring of recurrent mobility patterns," in *Proc. IEEE 5th World Forum Internet Things (WF-IoT)*, Apr. 2019, pp. 876–881.

[41] L. Tang and P. Thakuriah, "An analysis of behavioral responses to real-time transit information systems," in *Proc. Assoc. Collegiate Schools Planning Annu. Meeting*, 2006, pp. 1–42.

[42] F. Zhang, Q. Shen, and K. Clifton, "An examination of traveler responses to real-time bus arrival information using panel data," in *Proc. 87th Annu. Meeting Transp. Res. Board*, 2007, pp. 107–115.

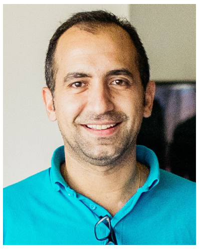
[43] O. Cats and G. Loutos, "Real-time bus arrival information system—An empirical evaluation," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1310–1315.

[44] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[45] P. Papadimitratos, A. De La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 84–95, Nov. 2009.

[46] G. M. Philips and P. J. Taylor, *Theory and Applications of Numerical Analysis*, 2nd ed. New York, NY, USA: Academic, 1996, pp. 196–220.

[47] M. Mazo, Jr., and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2456–2461, Oct. 2011.



PANAYIOTIS KOLIOS (Member, IEEE) received the B.Eng. and Ph.D. degrees in telecommunications engineering from King's College London, in 2008 and 2011, respectively. He is currently a Research Assistant Professor with the KIOS Research and Innovation Center of Excellence, University of Cyprus. His research interest includes basic and applied research on networked intelligent systems. Examples of such systems include intelligent transportation systems, autonomous unmanned aerial systems, and the plethora of cyber-physical systems that arise within the Internet of Things. Particular emphasis is given to emergency response aspects in which faults and attacks could cause disruptions that need to be effectively handled. He is an Active Member of the IEEE, contributing to a number of technical and professional activities within the association.

LOIZOS PAPACHRISTOFOROU received the M.Eng. degree in electrical and electronic engineering from Imperial College London in 2019, where his masters' work focused on the IoT and specifically on event triggering techniques and their utilization in the various IoT application scenarios. He worked as a Researcher Assistant at the KIOS Research and Innovation Center of Excellence (KIOS CoE), University of Cyprus, where his main focus of research was on the implementation of algorithms to improve energy efficiency of the IoT devices with particular emphasis on public transportation systems.

CHRISTOS PANAYIOTOU (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in ECE from the University of Massachusetts at Amherst, in 1994 and 1999 respectively, and the M.B.A. degree from the Isenberg School of Management, University of Massachusetts at Amherst, in 1999. He is currently a Professor with the Electrical and Computer Engineering (ECE) Department, University of Cyprus (UCY), and the Deputy Director of the KIOS Research and Innovation Center of Excellence of which he is also a Founding Member. Before joining the University of Cyprus, in 2002, he was a Research Associate at the Center for Information and System Engineering (CISE) and the Manufacturing Engineering Department, Boston University (1999–2002). He has published more than 250 papers in international refereed journals and conferences. His research interests include control, performance evaluation and optimization of intelligent transportation systems, modeling and control of discrete event and hybrid systems, cyber-physical systems, computer networks, the Internet of Things (IoT), smart buildings, fault diagnosis, event detection and localization, mobile terminal localization, resource allocation, and computer simulation. He was a recipient of the 2014 Best Paper Award for the journal *Building and Environment* (Elsevier). He is an associate editor for several IEEE journals and he held several positions in international conference program committees.

GEORGIOS ELLINAS (Senior Member, IEEE) received the B.Sc., M.Sc., M.Phil., and Ph.D. degrees in electrical engineering from Columbia University. He is currently a Professor at the Department of Electrical and Computer Engineering and a Founding Member of the KIOS Research and Innovation Center of Excellence, University of Cyprus. Prior to joining the University of Cyprus, he worked as an Associate Professor of electrical engineering at City College of the City University of New York, as a Senior Network Architect at Tellium Inc., and as a Research Scientist/Senior Research Scientist with Telcordia Technologies' (formerly Bell Communications Research, Bellcore) Optical Networking Research Group. He has coauthored/co-edited four books on optical networks, more than 270 archived articles/conference papers/book chapters, and he is the holder of 30 patents on optical networking. His research interests include optical networks, intelligent transportation systems, the IoT, emergency-response systems, and security of cyber-physical systems. He is a fellow of the IET (2019) and a Senior Member of OSA and ACM.