Resilient Learning-Based Control for Synchronization of Passive Multi-Agent Systems under Attack

Arash Rahnama and Panos J. Antsaklis, Fellow, IEEE

Abstract

In this paper, we show synchronization for a group of output passive agents that communicate with each other according to an underlying communication graph to achieve a common goal. We propose a distributed event-triggered control framework that will guarantee synchronization and considerably decrease the required communication load on the band-limited network. We define a general Byzantine attack on the event-triggered multi-agent network system and characterize its negative effects on synchronization. The Byzantine agents are capable of intelligently falsifying their data and manipulating the underlying communication graph by altering their respective control feedback weights. We introduce a decentralized detection framework and analyze its steady-state and transient performances. We propose a way of identifying individual Byzantine neighbors and a learning-based method of estimating the attack parameters. Lastly, we propose learning-based control approaches to mitigate the negative effects of the adversarial attack.

I. INTRODUCTION

Distributed coordination of multi-agent systems has been discussed extensively in control, communication and computer science literature. The wide range of applications in this area includes multiple robot coordination [1], cooperative control of vehicle formations [2], flocking [3] and spacecraft formation flying [4]. A strong body of literature exists on the state synchronization of homogeneous multi-agent systems with identical dynamics. In many practical

Arash Rahnama and Panos J. Antsaklis are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA (e-mail: {arahnama, antsaklis.1} @nd.edu). The support of the National Science Foundation under Grant No. CNS-1035655 and CNS-1446288 is gratefully acknowledged.
applications of multi-agent systems, however, individual systems may have different dynamics with different state-space dimensions. This has instigated the need for the design of output-based control frameworks which do not require the full knowledge of dynamic states and the focus on synchronization of multi-agent systems with different dynamics based on their output information. The problem of synchronization naturally arises when a group of networked agents are seeking output-based agreement according to a certain quantity of interest that depends on the overall goal of the multi-agent system. More specifically, synchronization for a multi-agent system is defined as the agents following a desired output behavior that is achieved thorough local cooperation of neighboring agents. This cooperation is based on a feedback mechanism consisting of a weighted sum of the differences of the outputs of the neighboring agents. Some examples of systems under cooperative control resulting in sophisticated dynamic patterns which cannot be achieved by individual members are migration (or flocking), swarming, and torus.

There exists a large body of valuable works in the area of synchronization and control. The problem of synchronization for multi-agent systems with dynamic communication edges has been explored in [5]. Adaptive synchronization of diffusively coupled systems is discussed in [6]. Synchronization of multi-agent systems that are physically coupled is discussed in [7]. Another interesting sub-field of synchronization in multi-agent system consists of leader-follower synchronization problems, such works include [5], [8]–[10]. The relationship amongst dissipativity, passivity and output synchronization has been explored in the literature as well [11]–[13]. Synchronization under switching topologies is discussed in [14]. Cluster-based synchronization in which only the synchronizations of separate clusters are achieved is discussed in [15]. Some of the other recent notable works in the area of synchronization in multi-agent systems are given in [16]–[24].

In none of the works above, the problem of security and the negative effects of malicious attacks on synchronization have been discussed. In this work, we consider the effects of a Byzantine attack on the multi-agent network. Byzantine attacks were first proposed by [25] and may cover different types of malicious behaviors [26]. In our work, Byzantine agents intelligently falsify their data —Similar to the adversaries defined in [27]–[29]. The Byzantine agents are assumed to be powerful in the sense that they have the complete knowledge of the whole system and can update their information in an arbitrary way and send different data to distinct neighbors at the same time. Additionally, the Byzantine nodes are capable of disturbing the structure of the underlying communication graph by manipulating their feedback weights —The
communication graph is usually required to meet certain conditions for synchronization to happen [11]–[13]. Lastly, we propose a distributed method of detection and mitigation as opposed to the more common centralized methods where a fusion center takes upon itself the responsibility of detecting and mitigating the attacks. There is obviously always a limitation to this approach as the central fusion unit may be compromised as well. Our proposed distributed detection and mitigation framework will eliminate this possibility. In the consensus literature, the decentralized method of detection has been proposed in works such as [30]–[33]. In [33] for example, it is assumed that through collaboration, the Byzantine agents are aware of the true hypothesis, which is similar to the assumption we make in the present work. As another example, in [32], the authors rely on a sequential decentralize probability ratio test that is modified via a reputation-based mechanism in order to filter out the false data and only accept reliable messages. Lastly, most detection and mitigation frameworks in the literature rely on exclusion of Byzantine agents from the synchronization algorithm [34], [35]. For example, in [36], the authors propose an adaptive outlier detection framework, based on which, the outside of the bound received information are excluded from the consensus process. In our work, we propose a mitigation scheme that takes advantage of the falsified information received from the Byzantine agents and mitigates the effects of the attack without excluding the Byzantine neighbors. This is due to the fact that excluding the Byzantine agents usually is not the best practice as most synchronization algorithms [11]–[13], rely on balancedness and connectedness of the underlying communication graph and exclusion of Byzantine agents may contradict these conditions.

Our framework is based on each individual agent locally deciding, based on its local test statistics that contain the information received by the agent from its neighbors, whether the entire multi-agent system has reached synchronization. We also show synchronization for an event-triggered control framework. This is motivated by the fact that event-triggered control frameworks can considerably reduce communication and computation load on the band-limited communication network [37]. Additionally, it has been shown that event-based control methods can maintain the same performance index as their continuous and periodic based control counterparts [38], [39]. First, we show that, under no attack, the entire event-triggered multi-agent network system is capable of reaching synchronization and that each agent may decide correctly on synchronization based on their local summary statistics, if our proposed triggering-based control framework and the underlying communication graph meet certain conditions. Next, we propose a method of identifying Byzantine agents based on the statistical distribution of
Byzantine agents’ outputs. We characterize and analyze the performance of the detection unit. Lastly, we propose a method of mitigation for the attacks in order to maintain the synchronization of the entire event-triggered multi-agent network system. In this vein, the contributions of our work are listed below,

- We show synchronization for an event-triggered multi-agent network system with output passive agents. We introduce a local decision making process based on which each individual agent decides whether the entire system has reached synchronization or not.
- We propose a simple design-oriented event-triggering control framework based on simple output-based triggering conditions which guarantees synchronization and positive lower-bounds for the inner-event time-instances (lack of Zeno behavior).
- We define a rather general Byzantine attack framework, and characterize the effects of the attack on passive qualities of the multi-agent system in particular and synchronization of the entire system in general.
- We introduce a decentralized detection framework for detecting the Byzantine attack.
- We analyze the performance of the proposed detection framework. We characterize both the steady-state and transient performance of the detection framework.
- We propose a specific method of identifying individual Byzantine neighbors and learning their attack parameters.
- Lastly, we introduce two different learning-based mitigation processes; one based on the passive properties of the agents, and one based on the statistical distribution of the data received from the neighboring agents. Based on which, we propose a learning-based control framework that can considerably mitigate the negative effects of the attack.

II. MATHEMATICAL AND STATISTICAL PRELIMINARIES

Consider the dynamical system $G$,

\[
G : \begin{cases} 
\dot{x}(t) = f(x(t), u(t)) \\
y(t) = h(x(t), u(t)),
\end{cases}
\]

where $f$ and $h$ are Lipschitz functions, $x(t) \in X \subset \mathbb{R}^n$, and $u(t) \in U \subset \mathbb{R}^m$, and $y(t) \in Y \subset \mathbb{R}^m$ are respectively the state, input and output of the system, and $X$, $U$ and $Y$ are the state, input and output spaces.
Definition 1. ([40]) The supply rate \( \omega(u(t), y(t)) \) is a well-defined supply rate, if for all \( t_0, t_1 \) where \( t_1 \geq t_0 \), and all solutions \( x(t) \in X, u(t) \in U, \) and \( y(t) \in Y \) of the dynamical system, we have,

\[
\int_{t_0}^{t_1} |\omega(u(t), y(t))| \, dt < \infty.
\]

Dissipativity and passivity are energy-based notions that characterize a dynamical system by its input/output behavior. A system is dissipative if the change in the system’s stored energy is upper-bounded by the energy supplied to the system. The energy supplied to the system is mathematically modeled by the supply function, and the energy stored in the system is mathematically modeled by the storage function.

Definition 2. ([40]) System \( G \) is dissipative with respect to the well-defined supply rate \( \omega(u(t), y(t)) \), if there exists a nonnegative storage function \( V(x) : X \rightarrow R^+ \) such that for all \( t_0, t_1 \) where \( t_1 \geq t_0 \), and all solutions \( x(t) \in X, u(t) \in U, \) and \( y(t) \in Y \) of the dynamical system,

\[
V(t_1) - V(t_0) \leq \int_{t_0}^{t_1} \omega(u(t), y(t)) \, dt,
\]

is satisfied. If the storage function is differentiable, we have,

\[
\dot{V}(t) \leq \omega(u(t), y(t)), \quad \forall t \geq 0.
\]

Definition 3. ([40]) As a special case of dissipativity, system \( G \) is called passive, if there exists a nonnegative storage function \( V(x) : X \rightarrow R^+ \) such that,

\[
V(t_1) - V(t_0) \leq \int_{t_0}^{t_1} u^T(t)y(t) \, dt
\]

is satisfied for all \( t_0, t_1 \) where \( t_1 \geq t_0 \), and all solutions \( x(t) \in X, u(t) \in U, \) and \( y(t) \in Y \) of the dynamical system.

Definition 4. ([40]) System \( G \) is considered to be Output Feedback Passive (OFP), if it is dissipative with respect to the well-defined supply rate, \( \omega(u, y) = u^T y - \rho y^T y \), for some \( \rho \in R \). Additionally, if the storage function is differentiable, we may have,

\[
\dot{V}(t) \leq u^T y - \rho y^T y.
\]
The above definition presents a more general form for the concept of passivity. Based on Definition 4, we can denote an output passive system based on its output passivity index. If $\rho < 0$ then the system has a shortage of passivity. A positive value for the passivity index $\rho$ indicates an excess in passivity. If $\rho > 0$, then the system is called output strictly passive (OSP).

**Definition 5.** ([42]) System $G$ is called finite-gain $L_2$-stable, if for the smallest possible positive gain $\gamma$, and $\forall u(t) \in U$, a $\beta$ exists such that over the time interval $[0, \tau]$ and for any positive $\tau$, we have,

$$
\|y_\tau\|_{L_2} \leq \gamma \|u_\tau\|_{L_2} + \beta.
$$

Here, $\|y_\tau\|_{L_2}$ and $\|u_\tau\|_{L_2}$ represent the $L_2$-norm of truncated signals over the time interval $[0, \tau]$. For instance,

$$
\|y_\tau\|_{L_2} = \sqrt{\int_0^\tau y^T(t)y(t)dt}.
$$

In probability theory, the expected value ($E[X]$) of a random variable $X$, intuitively, is the long-run average value of repetitions of the experiment it represents, in the continuous sense, this is defined as,

$$
E[X] = \int_{-\infty}^{+\infty} x f_{PDF}(x)dx.
$$

The notation $f_{PDF}(.)$ represents the probability density function (PDF) of a distribution. Expectation of the random variable $X$ conditioned on the hypothesis (or random distribution) $H$ is represented as $E[X|H]$. The complementary distribution function of the standard normal Gaussian distribution with zero mean ($\mu = 0$) and standard deviation $\sigma = 1$ is denoted as $Q(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{t^2}{2}} dt$.

The Gaussian distribution with mean $\mu$ and variance $\sigma^2$ is denoted as $\phi(x|\mu,\sigma^2) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$. Null and alternative hypotheses are represented as $H_0$ and $H_1$. Probability of an event is represented as $P$. Probability of false alarm (type 1 error) or accepting the alternative hypothesis and rejecting the null hypothesis mistakenly is shown as $P_{FA} = Pr(D = H_1|H_0)$ and probability of detection is $P_D = Pr(D = H_1|H_1)$.

### III. The Communication Graph Model

The communication flow between agents may be represented as a weighted directed graph. A graph is directed, if its edges have direction. We consider a finite positively weighted...
A directed graph $G := (V, E)$ with no loops and with the adjacency matrix $A$, where the entry $a_{i,j} \neq 0$, if there is a directed edge from vertex $i$ to vertex $j$, otherwise $a_{i,j} = 0$. The adjacency matrix $A$ represents both the link weights and the topology of the graph. $V$ is the vertex set including all vertices (all $N$ agents), $V = \{1, 2, ..., N\}$. $E$ is the edge set including all edges (communication links), $E \subset V \times V$. The agent $G_i$ can send information to agent $G_j$, if $(i, j) \in E$ and $a_{i,j} \neq 0$. The in-degree of a vertex $j$ is given by $d_{in}(j) = \sum_j a_{kj}$ and the out-degree of a vertex $j$ is given by $d_{out}(j) = \sum_j a_{jk}$ where $k$ respectively represents the in-neighbor ($V_{in}(j) = \{k \in V_{in}(j)|(k,j) \in E\}$) and out-neighbor ($V_{out}(j) = \{k \in V_{out}(j)|(j,k) \in E\}$) agents that have a communication link in common with agent $j$. We introduce the diagonal degree matrix $D_{N \times N}$ with $d_{jj} = d_{out}(j), \forall j \in V$. The weighted Laplacian matrix $L$ of the graph is defined as $L = D - A$. For the graph presented in Fig. 1, we have,

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}, \quad L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ 0 & 3 & -2 & 0 & -1 \\ 0 & 0 & 3 & 0 & -3 \\ 0 & 0 & -1 & 0 & 1 \\ -2 & -1 & 0 & -1 & 4 \end{bmatrix}.$$

**Definition 6.** [43] A vertex is balanced, if its in-degree is equal to its out-degree. A directed weighted graph is balanced, if all of its vertices are balanced.

It is important to note that the followings hold for a balanced directed graph, $1_N^T L = 0$ and $L^T 1_N = 0$, where $1_N = [1, ..., 1]^T$ is a vector of size $N$. The graph presented in Fig. 1 is balanced.
Definition 7. [43] A path of length $r$ in a directed graph is a sequence of $r + 1$ distinct vertices \( \{v_0, v_1, ..., v_r\} \) such that for every $i \in \{0, ..., r + 1\}$, \((v_i, v_{i+1})\) is an edge. A weak path is a sequence of $r + 1$ distinct vertices \( \{v_0, v_1, ..., v_r\} \) such that for every $i \in \{0, ..., r + 1\}$, either \((v_i, v_{i+1})\) or \((v_{i+1}, v_i)\) is an edge. A directed graph is weakly connected if any two vertices can be joined by a weak path.

Definition 8. [43] A directed graph is connected, if for any pair of distinct vertices $v_i$ and $v_j$, there is a weak path from $v_i$ to $v_j$. A directed graph is strongly connected, if for any pair of distinct vertices $v_i$ and $v_j$, there is a directed path from $v_i$ to $v_j$.

The connectivity measures of directed graphs are related to the algebraic properties of their Laplacian matrices [44].

Definition 9. [44] For a directed graph $G$ with the Laplacian matrix $L$, the algebraic connectivity is a real number defined as

\[
\lambda(G) := \min_{P} z^T L z,
\]

where $P = \{z \in \mathbb{R}^N : z \perp 1_N, ||z|| = 1\}$.

For a balanced connected graph $G$ with nonnegative weights and Laplacian matrix $L$, we have $\lambda(G) = \gamma_2(\frac{L+L^T}{2}) > 0$, where $\gamma_2$ is the second smallest eigenvalue of the matrix $\frac{L+L^T}{2}$ ($\gamma_1 = 0$) [44]. Lastly, we define $\mathcal{N}^{in}_j$ and $\mathcal{N}^{out}_j$. $\mathcal{N}^{in}_j$ denotes the set of all neighboring nodes that send information to agent $G_j$ including the weights associates with their communication graph topology. $\mathcal{N}^{out}_j$ denotes the set of all neighboring nodes that receive information from agent $G_j$ including the weights associates with their communication graph topology. For a balanced graph, the cardinality of these two are equal $|\mathcal{N}^{out}_j| = |\mathcal{N}^{in}_j|$. For instance, for the graph presented in Fig. 1 we have: $\mathcal{N}^{in}_5 = \{1G_2, 3G_3\}$ and $|\mathcal{N}^{in}_5| = |\mathcal{N}^{out}_5| = 4$.

IV. PROBLEM STATEMENT

We consider the problem of synchronization for a multi-agent system consisting of $N$ agents under an event-triggered network control framework. We assume that agents are output passive,

\[
\dot{V}_j(t) \leq u_j^T(t)y_j(t) - \rho_j y_j^T(t)y_j(t), \ \forall t > 0 \ \text{for} \ j = 1, ..., N.
\]

We consider an efficient event-based framework where agents communicate with each other only when necessary. In other words, agent $G_j$ sends new information to its neighboring agents
when the last information sent to other agents is outdated and requires a new modification based on $G_j$’s current dynamics and the event-triggering condition. This considerably decreases the communication load on the shared network. Consequently, it is assumed that the agents that will receive the new information from $G_j$ will update their control inputs accordingly. Each agent establishes a new communication attempt with its neighboring agents over a band-limited networks when its triggering condition is met. The triggering conditions are output-based and simple to design.

$$||e_j(t)||^2 > \delta_j||y_j(t)||^2.$$  (1)

The event-detector is located on the output of each agent to monitor the behavior of its output. An updated measure of $y_j$ is sent to the communication network when the error between the last information sent ($y_{tk}$) and the current one, $e_j(t) = y_j(t) - y_j(t_k)$ (for $t \in [t_k, t_{k+1})$) exceeds a predetermined threshold established by the designer based on the relation given in Eq. (1) and the design parameter $\delta_j$. At instances for which the triggering condition is met, and new information is successfully exchanged and the error is set back to zero, $e_j(t_{k+1}) = 0$. These simple triggering conditions will facilitate the design process by making it easier for the designer to understand and analyze the trade-offs amongst synchronization, performance and communication load. Each agent has its own respective sampler condition which is designed based on its passivity properties and its location in the underlying communication graph. This will be analytically presented in Section VII. Theorem 1 outlines the design condition for each $\delta_j$. The control input for each agent is represented by the summation of the differences between the agent’s output and the output of its neighboring agents multiplied by respective positive control gains,

$$u_j = \sum_{k \in N_j} a_k(y_k(t^n_k) - y_j(t^n_j)).$$  (2)

More specifically, the input $u_j$ for agent $G_j$ consists of the summation of $a_k(y_k(t^n_k) - y_j(t^n_j))$, where $y_j(t^n_j)$ represents agent $G_j$’s last output sent to its neighbors, and $y_k(t^n_k)$ represents the last received output from the neighboring agent $k$ where $k \in N_j$. $a_k > 0$ represents a control gain established by agent $G_j$ for each neighboring agent,

$$a_k \begin{cases} 
1 & \text{if agent } G_j \text{ receives information from agent } G_k \\
0 & \text{otherwise.}
\end{cases}$$

One can represent the underlying communication graph according to Section III in which case the control gains $a_k$ represent the arc weights in the graph. The assumption made here
is that during the initialization and design of the gains and communication links for the entire multi-agent, the underlying communication graph is connected and balanced. We denote the outputs of \( N \) agents by the vector \( Y = [y_1, y_2, ..., y_N]^T \). We define the matrix \( \Phi \in \mathbb{R}^{(N-1) \times N} \) as follows,

\[
\Phi = \begin{bmatrix}
-1 + (N - 1)\nu &= 1 - \nu & -\nu & ... & -\nu \\
-1 + (N - 1)\nu &= -\nu & 1 - \nu & \ddots & \vdots \\
& \vdots & \ddots & \ddots & -\nu \\
-1 + (N - 1)\nu &= -\nu & ... & -\nu & 1 - \nu \\
\end{bmatrix}
\]

(3)

where \( \nu = \frac{N - \sqrt{N}}{N(N-1)} \in \mathbb{R} \). Matrix \( \Phi \) exhibits the following properties: \( \Phi 1_N = 0 \), \( \Phi \Phi^T = I_{N-1} \), and,

\[
\Phi^T \Phi = \begin{bmatrix}
\frac{N-1}{N} &= -\frac{1}{N} & \cdots & -\frac{1}{N} \\
-\frac{1}{N} &= \frac{N-1}{N} & \ddots & \vdots \\
& \vdots & \ddots & -\frac{1}{N} \\
-\frac{1}{N} &= \cdots & -\frac{1}{N} & \frac{N-1}{N} \\
\end{bmatrix} = I_N - \frac{1}{N} 1_N 1_N^T.
\]

To measure synchronization mathematically, we define,

\[
\bar{Y} = \frac{1}{N} 1_N^T Y = \frac{1}{N} \sum_{i=1}^{N} y_i,
\]

(4)

and,

\[
Y_\Delta = (y_1 - \bar{Y}, y_2 - \bar{Y}, ..., y_N - \bar{Y})^T.
\]

(5)

\( Y_\Delta \) represents a measure for synchronization of agents. \( Y_\Delta = 0 \) only happens when all agents reach the same synchronized state \( y_1 = y_2 = ... = y_N = \bar{Y} \). We have \( \Phi^T \Phi Y = (I_N - \frac{1}{N} 1_N 1_N^T) Y = Y_\Delta \). Further,

\[
Y^T \Phi^T \Phi \Phi^T \Phi Y = Y_\Delta Y_\Delta.
\]

(6)

Lastly, we can show that,

\[
Y^T L^T Y = (Y_\Delta + \frac{1}{N} 1_N 1_N^T Y) L^T Y \\
= Y_\Delta L^T Y = Y_\Delta L^T (Y_\Delta + \frac{1}{N} 1_N 1_N^T Y) \\
= Y_\Delta^T L^T Y_\Delta \geq \lambda (G) Y^T \Phi^T \Phi Y = \lambda (G) Y^T Y - \frac{\lambda (G)}{N} Y^T 1_N 1_N Y,
\]

(7)
where $\lambda(G)$ represents the algebraic connectivity of the underlying communication graph and $L$ is the Laplacian matrix. In Section VII we represent the results for synchronization of the entire event-triggered multi-agent system and the design conditions for each event-detector based on the passivity properties of agents and algebraic properties of the communication graph.

V. SENSING, DETECTION AND FUSION FRAMEWORKS

The three most popular signal detection approaches for spectrum sensing are matched filtering detection method, feature detection method, and energy detection method [26]. Here, we adopt an energy-based detection approach for the detection center on each agent [45], [46]. The energy detector measures the energy in the input wave over a specific time interval. This means that our framework is based on detecting a deterministic signal over a noisy communication channel. The energy detection method, however, cannot differentiate between noise and signal, but at the same time does not need any prior knowledge about the signal’s distribution. It is assumed that the detection center makes decisions under a Neyman-Pearson (NP) set-up, and that the adversary is aware of it [47]. The local summary statistic of each agent is calculated from the received signal energy from the neighboring agents. As mentioned, at each triggering instance, each agent communicates with its neighbors. In our detection framework, this means that each communication attempt will update the summary statistic of neighboring agents. This process continues until the whole multi-agent system synchronizes to a steady-state. This steady-state represents the global test statistic at which the entire multi-agent system has reached synchronization. At each updating instance, each agent makes a decision whether the entire system has reached synchronization or not. As later defined, this process also decides if a neighboring agent is Byzantine or not. In order to fulfill the premise behind this framework, each agent is equipped with a detection unit that has access to the network topology in order to gain information [46]. We explain this in more details in this section.

The signals received by each agent’s detection unit are assumed to be unknown in details but deterministic. The band-limited communication environment in which signals travel is known. The noise is assumed to be Gaussian and additive with zero mean. Based on the assumption of a deterministic signal, we know that the input with signal present is Gaussian with a nonzero mean. For agent $G_j$, at time instant $\tau$, the sensed signal received from the neighboring agent
$G_k$, $y^*_k$ is given as,

$$y^*_k = \begin{cases} 
    n^*_k & \text{under } H_0 \\
    \tilde{h}_k s^* + n^*_k & \text{under } H_1,
\end{cases}$$

where $\tilde{h}_k$ represents the channel gain and $n^*_k$ represents the noise for the communication link from agent $G_k$ to agent $G_j$ ($H_1$ and $H_0$ here represent the hypotheses under which, the signal is present or not). The channel gain in the communication link between each two agents, models the effects of channel shadowing, channel loss and fading. $n^*_k$ is additive Gaussian noise with zero mean and variance $\sigma^2_k$ ($N(0, \sigma^2_k)$). It is assumed that the noise $n^*_k$ and signal $s^*_k$ are statistically independent. The channel gains $\tilde{h}_k$ and noise variances $\sigma^2_k$ for channels are readily available for each agent. These assumptions are justified by the fact that each detection unit can perform simple noise power estimation and channel gain estimation (by averaging the signal-to-noise ratio over a certain time interval) between consecutive sensing intervals to accurately obtain these values [48]. Additionally, we assume that $\tilde{h}_k$ is considered larger than the estimate value to compensate for any overhead [48].

It has been shown that control gain designs that compensate for the negative effects of the communication channel $\tilde{h}_k$ comparatively perform better [49]. As a result, one can design the optimal control gains $a_k$ (explained in details in Section IV) according to $a_k = \frac{K_k}{\tilde{h}_k}$ to compensate for channel effects. This is not a necessary rule to follow for the results presented in this paper. This weight design, however, will efficiently assign higher weights to channels with higher Signal-to-Noise ratio (more confidence in the received data) and vice-versa [49]. Lastly, the channel gains are assumed independent of each other, known and constant over each sensing period. This is justified by the slow-changing nature of the communication links where the delay requirement is short compared to the channel coherence time [50]. Each agent $G_j$ calculates a local summary statistic $T_k$ over a detection interval of $L$ samples, from the information received from its neighboring agent $G_k$,

$$T_k = \sum_{i=1}^{L} |y^*_k - y^*_j|^2.$$  \hspace{1cm} (8)

It can be assumed that $L = 2TW$ where $TW$ is an integer representing the time-bandwidth product of the energy detector with $T$ standing for the effective spectrum sensing time-interval and $W$ standing for the bandwidth of the sensing spectrum [51]. $y^*_j$ represents the last output sent from agent $G_j$ to its neighboring agents at instance $i$, which is also utilized in calculating
the local summary statistic $T_k$ over the detection interval of $L$. The energy in a finite number of samples for the local summary statistic can be approximated by the sum of squares of statistically independent Gaussian random variables having certain means ($|y_k^i - y_j^i|$) and equal variances. This sum has a Chi-Square distribution with $L$ degrees of freedom ($\chi^2_L$) in the absence of signal. In the presence of a deterministic signal ($H_1$ hypothesis), the sampling plan yields an approximation to the energy consisting of the sum of squares of random variables, where the sum has a non-central Chi-Square distribution with $L$ degrees of freedom with the non-centrality parameter $\eta_k$,

$$
\frac{T_k}{\sigma_k^2} \approx \begin{cases} 
\chi^2_L & \text{under } H_0 \\
\chi^2_L(\eta_k) & \text{under } H_1,
\end{cases}
$$

where $\eta_k = \frac{\sum_{i=1}^{L} (\tilde{h}_k y_k^i - y_j^i)^2}{\sigma_k^2}$.

A. Decision Making Step

Each agent $G_j$ makes local decisions as to whether the entire multi-agent system has reached synchronization or not. The summary statistic for synchronization, given the entire system, may be represented as $T^*_j = \sum_{i \in N} T^*_j$, where $T^*_j = \sum_{k \in N_j} T^*_j$ for $j = 1, ..., N$. This then can be compared against a threshold $\gamma$ in order to decide if the system has synchronized. If this holds for the entire event-triggered multi-agent system then the entire multi-agent network has synchronized. Each agent $G_j$, however, makes its own decision on the synchronization hypothesis using the predefined threshold $\gamma_j$,

$$
\text{Decision}_{\text{syn}} = \begin{cases} 
H_0 & \text{if } T^*_j < \gamma_j \\
H_1 & \text{otherwise.}
\end{cases}
$$

Where, $\gamma_j = \sum_{k \in N_j} L \sigma_k^2 + \lambda$ (see (20)). $\lambda$ is a positive constant representing the allowed margin of error (or our confidence in the process). The exact choice of $\lambda$ depends on the desired detection and false alarm rates and is beyond the scope of work presented here. This is explained in more details in Section VIII. We assume the threshold $\gamma_j$ has already been selected based on performance, detection and false alarm criteria. The relation in (9) means that if sums of differences between an honest agent’s output and the outputs of all its neighboring agents is small enough, then the honest agent may decide that the multi-agent system has reached synchronization. In other words, the entire event-triggered multi-agent system has reached synchronization, if $T^*_j < \gamma_j$ for $j = 1, ..., N$. 

VI. BYZANTINE ATTACK

Multi-agent systems are vulnerable to attacks due their strong reliance on secure communication links and legitimate exchange of information. One of the most common type of such attacks is named Byzantine. Originally, proposed in [25], a Byzantine attack may take different forms [52], [53], our focus in this paper remains with intelligent data-falsification and weight manipulation attacks [27], [28]. The main goals of Byzantine attackers is to first decrease the detection probability and increase the probability of false alarms, and then to degrade the multi-agent system’s performance. This makes the problem of the Byzantine attack and defending against it very challenging and complicated. For the Byzantine agents, we adopt an approach that leaves the attacker with more power than usually allowed in practice. This leads to a conservative assessment of security risks but helps with analytical tractability. In this vein, we assume that Byzantine agents in fact know the true hypothesis and they use this knowledge to construct the most effective fictitious data in order to confuse the synchronization goal. This assumption obviously is difficult (but not impossible) to satisfy in practice. For this to be possible, the attackers should have a separate network for the cooperation amongst themselves.

As we will show in Section VII for the entire event-triggered multi-agent network system to reach synchronization, a connected balanced communication graph is required. In Section VII, we also quantify the negative effects of weight manipulation resulting in an unbalanced underlying communication graph. We assume that Byzantine agents attack the multi-agent system from two different angles. First, the Byzantine agents disturb the underlying premise behind the convergence of the multi-agent system by introducing new weights that will undermine the balanced property of the underlying communication graph. Second, the Byzantine agents falsify their own information sent to other honest agents in order to conceal their identity and also to coerce the entire multi-agent system into following their desired behavior. The attack model, we consider is extremely general and covers several different Byzantine plots. To be more specific, if the event-triggered multi-agent network system is designed and initialized according to Theorem 1 by the designer to reach synchronization, then we assume that at the initialization instance, the Byzantine agents \( \mathcal{N}_B \) introduce the following fictitious weights \( a'_k \) into the underlying communication graph,

\[
a'_k = a_k + \omega_j \quad \forall G_j \in \mathcal{N}_B, \text{ and } \forall G_k \in \mathcal{N}^{\text{in}}_j.
\]
Additionally, at each communication instance, we assume that the Byzantine agents falsify their information according to,

\[ \tilde{y}_j = y_j \pm \Delta_j \quad \forall G_j \in \mathcal{N}_B. \]

Where \( \Delta_j \) may represent the power of the attack inflicted by the Byzantine agent \( G_j \). The model presented above allows the Byzantine nodes to manipulate their weights and falsify their information in a completely arbitrary manner based on their desire. As a result, the Byzantine agents are able to conceal themselves while degrading the performance of the entire system.

A. Modeling of the Data Falsification Attack

The main goal of the Byzantine agents is to manipulate the sensing results in a stealthy way and to reverse the synchronization status. In the presence of a synchronized state, the goal is to “vandalize” and move the multi-agent’s state back to the state of lack of synchronization (\( H_0 \rightarrow H_1 \)), and in the absence of synchronization, the goal is to “exploit” and to move the current state to the state of the presence of synchronization at the desired value set by the Byzantine agents (\( H_1 \rightarrow H_0 \)). This type of data injection attack is adaptive and extremely general. Each Byzantine agent may perform a stealthy manipulation of sensing data independently. The attack is “adaptive”, in the sense that the data-falsification is based on the neighbors’ states, and with the assumption that the adversary has prior knowledge on the detection algorithm. The attack is “covert”, in the sense that the adversary manipulates the sensing data without being detected. Outsider attackers can be effectively expelled from the network with an authentication mechanism. In this work, we focus on insider attackers that reside in legitimate nodes.

Based on the assumption that Byzantine agents are intelligent and know the true hypothesis, we analyze the worst case detection performance of data-falsifications and define the attack devised by the agent \( G_i \) as follows,

\[ \tilde{y}_i = \begin{cases} 
  y_i + \Delta_i & \text{with probability } P_i \quad \text{under } H_0 \\
  y_i & \text{with probability } 1 - P_i \quad \text{under } H_0,
\end{cases} \]

and,

\[ \tilde{y}_i = \begin{cases} 
  y_i - \Delta_i & \text{with probability } P_i \quad \text{under } H_1 \\
  y_i & \text{with probability } 1 - P_i \quad \text{under } H_1,
\end{cases} \]

where \( P_i \) is the attack probability and \( y_i \) is the Byzantine agent’s true time-variant output. \( \Delta_i \) is a constant value that represents the strength of the attack. \( \Delta_i \) is set by the Byzantine agent based
on the information it receives from its neighbors and may be positive or negative to fulfill the "exploitation" and "vandalism" objectives. For example, under the hypothesis $H_0$, we may define

$$\eta_i = \frac{\sum_{k=1}^L |\tilde{h}_i y^k - y^k_j|^2}{\sigma_i^2} \approx 0.$$  

The Byzantine agent by utilizing the attack parameter $\Delta_i > 0$ or $\Delta_i < 0$ may commit vandalism ($\eta'_i = \frac{\sum_{k=1}^L |\tilde{h}_i(y^k_i + \Delta_i) - y^k_j|^2}{\sigma_i^2} \approx L\tilde{h}_i^2\Delta_i^2, H_0 \rightarrow H_1$).

Under the hypothesis $H_1$, we may define the mean values $\mu_j = \frac{1}{L} \sum_{k=1}^L y^k_j$, $\mu_i = \frac{1}{L} \sum_{k=1}^L \tilde{h}_i y^k_i$, and $\eta_i = \frac{\sum_{k=1}^L |\tilde{h}_i y^k - y^k_j|^2}{\sigma_i^2}$ for an honest communication from agent $G_i$ to the host agent $G_j$ and $\eta'_i = \frac{\sum_{k=1}^L |\tilde{h}_i(y^k_i - \Delta_i) - y^k_j|^2}{\sigma_i^2}$ for a Byzantine communication from agent $G_i$ to the host agent $G_j$ over the detection interval $L$. One can see that, $\eta'_i = \eta_i + \frac{\sum_{k=1}^L (\tilde{h}_i^2\Delta_i^2 + 2\tilde{h}_i\Delta_i y^k_j - 2\tilde{h}_i^2\Delta_i^2 y^k_i)}{\sigma_i^2}$, hence the Byzantine agent with the selection of $\Delta_i > \frac{2(\mu_i - \mu_j)}{\tilde{h}_i}$ may commit an exploitative attack ($H_1 \rightarrow H_0$). Lastly, the Byzantine agent can adaptively estimate the relationship between its true output and its neighboring outputs based on the information it receives and accordingly set the value of $\Delta_i$.

This modeling of Byzantine attacks is quite common in literature and covers a vast domain of adversary models [26]. The above inequalities show the basic principle in terms of the amount of changes an attacker has to inject in order to fulfill "exploitation" and "vandalism" objectives, respectively. Lastly, as shown later, Byzantine agents will use large values for $\Delta_i$’s so that the magnitude of the local test statistics are dominated by the Byzantine agents’ outputs and the degradation of the detection performance and the overall system’s performance is maximized. This is, however, in odds with the Byzantine agents’ other objective to conceal themselves. As a result, the Byzantine agents will have to choose their parameters wisely in order to fulfill both concealment and performance degradation objectives.

VII. MAIN RESULTS

A. Synchronization Results

**Theorem 1.** Consider the event-triggered multi-agent system described in Section IV where each sub-system $G_j$ is output passive with the output passivity index $\rho_j$ and is controlled by the input mechanism given in (2). If the underlying connected communication graph is balanced, the communication time-delays and disturbances are negligible, and the communication attempts amongst all agents $G_j$ where $j = 1, ..., N$, are governed by the triggering conditions,

$$||e_j(t)||_2^2 > \delta_j ||y_j(t)||_2^2,$$


where the design parameters $\delta_j$ are chosen such that,

$$0 < \delta_j \leq \frac{2}{\mathcal{N}^n_{j}}(\lambda(G) + \rho_j) - \frac{1}{\alpha} - \frac{1}{\beta},$$

where $\alpha > 0$ and $\beta > 0$ are design variables and $\lambda(G)$ is the connectivity of the underlying communication graph, then the entire event-triggered multi-agent system achieves output synchronization asymptotically.

Proof. Each agent $G_j$ is output passive with the storage function (Lyapunov function) $V_j$ where,

$$\dot{V}_j(t) \leq u_j^T(t)y_j(t) - \rho_jy_j^T(t)y_j(t), \ \forall t > 0,$$

where the output passivity level is indicated by $\rho_j \in \mathbb{R}$. $u_j, y_j \in \mathbb{R}^m$ are the inputs and outputs of appropriate dimensions for the agent $G_j$. The error of the triggering condition for agent $j$ is defined as $e_j(t) = y_j(t) - y_j(t^n_j)$ for triggering instances $n = 0, 1, 2, \ldots$. Accordingly, for each agent, we have $e_j^T(t)e_j(t) \leq \delta_j y_j^T(t)y_j(t)$ between each two triggering instances. Given the control input in (2), and the framework described in Section IV, the input to the agent $G_j$ is defined as,

$$u_j = \sum_{k \in \mathcal{N}_{j}^n} a_k(y_k(t^n_k) - y_j(t^n_j)) = \sum_{k \in \mathcal{N}_{j}^n} a_k[(y_k(t) - e_k(t)) - (y_j(t) - e_j(t))],$$

where $n = 0, 1, 2, \ldots$ are the triggering instances. The relationship for the storage function of agent $G_j$ becomes,

$$\dot{V}_j \leq \sum_{k \in \mathcal{N}_{j}^n} a_k[(y_k(t) - e_k(t)) - (y_j(t) - e_j(t))]^T y_j(t) - \rho_jy_j^T(t)y_j(t)

= \sum_{k \in \mathcal{N}_{j}^n} a_k[(y_k(t) - y_j(t)) - (e_k(t) - e_j(t))]^T y_j(t) - \rho_jy_j^T(t)y_j(t).$$

In order to show synchronization for all $N$ agents, we consider the following storage function for the entire multi-agent system,

$$\dot{S} = \sum_{j=1}^{N} \dot{V}_j \leq \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_{j}^n} a_k[(y_k(t) - y_j(t)) - (e_k(t) - e_j(t))]^T y_j(t) - \sum_{j=1}^{N} \rho_jy_j^T(t)y_j(t).$$
As we explained in Section III and Section IV, the flow of information amongst agents may be represented by the Laplacian of the underlying communication graph $L$. Moreover, if we define the matrix $E = [e_1^T, e_2^T, ..., e_N^T]^T$, then we have,

$$\dot{S} = \sum_{j=1}^N \dot{V}_j \leq -Y^T L^T Y + Y^T L^T E - \sum_{j=1}^N p_j y_j^T(t)y_j(t)$$

(10)

$$\leq -\lambda(G) Y^T Y + Y^T L^T E - \sum_{j=1}^N p_j y_j^T(t)y_j(t),$$

(11)

where $\lambda(G) > 0$ represents the algebraic connectivity of the underlying connected communication graph. Next, we may show the following,

$$Y^T L^T E = E^T L Y = \sum_{j=1}^N \sum_{k \in \mathcal{N}_j^n} a_k(y_j(t) - y_k(t))^T e_j(t)$$

$$= \sum_{j=1}^N \sum_{k \in \mathcal{N}_j^n} a_k y_j^T(t)e_j(t) - \sum_{j=1}^N \sum_{k \in \mathcal{N}_j^n} a_k y_k^T(t)e_j(t).$$

(12)

For all $j$ and $k$, we can have: $y_j^T(t)e_j(t) \leq \frac{\alpha e_j^T(t)e_j(t)}{2} + \frac{y_j^T(t)y_j(t)}{2\alpha}$ and $y_j^T(t)e_j(t) \leq \frac{\beta e_j^T(t)e_j(t)}{2} + \frac{y_j^T(t)y_k(t)}{2\beta}$ where $\alpha, \beta > 0$. Utilizing these relationships in (12), we have,

$$Y^T L^T E \leq \sum_{j=1}^N \sum_{k \in \mathcal{N}_j^n} a_k \left[ \frac{\alpha e_j^T(t)e_j(t)}{2} + \frac{y_j^T(t)y_j(t)}{2\alpha} \right]$$

$$+ \sum_{j=1}^N \sum_{k \in \mathcal{N}_j^n} a_k \left[ \frac{\beta e_j^T(t)e_j(t)}{2} + \frac{y_j^T(t)y_k(t)}{2\beta} \right].$$

This can be further simplified to have,

$$Y^T L^T E \leq \sum_{j=1}^N |\mathcal{N}_j^n| \left[ \frac{(\alpha + \beta)e_j^T(t)e_j(t)}{2} + \frac{y_j^T(t)y_j(t)}{2\alpha} \right]$$

$$+ \sum_{j=1}^N \sum_{k \in \mathcal{N}_j^n} a_k \left[ \frac{y_j^T(t)y_k(t)}{2\beta} \right].$$

Further, we know that between any two triggering instances, one can show $e_j^T(t)e_j(t) \leq \delta_j y_j^T(t)y_j(t)$. This further gives us,

$$Y^T L^T E \leq \sum_{j=1}^N |\mathcal{N}_j^n| \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha} \right] y_j^T(t)y_j(t)$$

$$+ \sum_{j=1}^N \sum_{k \in \mathcal{N}_j^n} a_k \left[ \frac{y_j^T(t)y_k(t)}{2\beta} \right].$$
We have assumed that the underlying communication graph is balanced. This property implies that \(\sum_{j=1}^{N} \sum_{k \in \mathcal{N}_{in}^{j}} a_k \left[ \frac{y_j^T(t)y_k(t)}{2\beta} \right] = \sum_{j=1}^{N} \left| \mathcal{N}_{in}^{j} \right| \left[ \frac{y_j^T(t)y_j(t)}{2\beta} \right] \). This leads to,

\[
Y^T L^T E \leq \sum_{j=1}^{N} \left| \mathcal{N}_{in}^{j} \right| \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha} + \frac{1}{2\beta} \right] y_j^T(t)y_j(t). \tag{13}
\]

Utilizing (13) in (11), we have,

\[
\dot{S} = \sum_{j=1}^{N} \dot{V}_j \leq -\lambda(G)Y^TY - \sum_{j=1}^{N} \rho_j y_j^T(t)y_j(t) \\
+ \sum_{j=1}^{N} \left| \mathcal{N}_{in}^{j} \right| \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha} + \frac{1}{2\beta} \right] y_j^T(t)y_j(t). \tag{14}
\]

We introduce the square diagonal matrix \(\Theta \in \mathbb{R}^{N \times N}\), where

\[
[\Theta]_{j,i} = \begin{cases} 
+\lambda(G) + \rho_j - \left| \mathcal{N}_{in}^{j} \right| \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha} + \frac{1}{2\beta} \right] & \text{if } j = i \\
0 & \text{otherwise.}
\end{cases}
\]

Given (6) and \(\Theta\), (14) becomes,

\[
\dot{S} \leq -Y_\Delta^T \Theta Y_\Delta.
\]

If the event-triggered multi-agent system is designed according to the theorem such that for each node \(G_j\), we have: \(\delta_j \leq \frac{\lambda(G)^{-1}(\lambda(G) + \rho_j) - \frac{1}{\alpha + \beta}}{\alpha + \beta} \) then matrix \(\Theta\) is semi-positive. Moreover, for the storage function \(S\) we have: \(S \geq 0\) and \(\dot{S} \leq 0\) for \(\forall y \in \mathbb{R}^m\) and \(\forall t \geq 0\). This implies \(\dot{S} \to 0\) as \(t \to \infty\) according to Barbalat’s Lemma [42]. Consequently, \(Y_\Delta\) converges to the limit set \(D = \{x|Y_\Delta = 0, x \in \mathbb{R}^{mN}\}\) for all states of all agents,

\[
0 \leq Y_\Delta^T \Theta Y_\Delta \leq -\dot{S}
\]

This also means that the entire multi-agent system synchronizes asymptotically.

\[\square\]

Remark 1. The triggering conditions show that agents that are more passive with higher output passivity indices can have larger triggering intervals and will be required to send their information to the network less frequently.

Remark 2. Graph connectivity has a relation with the communication rate amongst agents as well. The higher the connectivity of the underlying communication graph for the multi-agent system is, the larger the triggering intervals may be (less frequent communication attempts).
Remark 3. The result presented in Theorem 1 also shows that agents with a higher number of neighbors will be required to send their information to the network more frequently compared to others. In other words, agents with a high number of neighbors play a more crucial part in the synchronization process of the entire multi-agent system. This is due to the fact that the triggering conditions show a reciprocal relationship between triggering intervals and number of neighbors. If an agent is responsible for sending its information to a higher number of neighbors (a higher number of neighboring agents rely on its information), then the agent will have to update its neighbors more frequently.

Remark 4. It is important to note that we did not consider the effects of external disturbances and time-delays in Theorem 1. It is assumed that these effects are negligible. However, if the delays are large enough, or external disturbances are strong enough, then they may affect the performance of the entire system.

Remark 5. The results in Theorem 1 are quite lenient. More specifically, they may hold for non-passive systems as well. For agent $G_j$, the triggering instance $\delta_j$ should be chosen such that $+\lambda(G) + \rho_j - |N_{in}^m|\left[\frac{(a+b)\delta_j}{2} + \frac{1}{2a} + \frac{1}{2b}\right] > 0$. It is clear that for a non-passive system with a shortage of output passivity $\rho_j$, one can still design a multi-agent system that will synchronize as long as $\delta_j$ is chosen such that, $0 < \delta_j \leq \frac{\sqrt{\lambda(G) + \rho_j - \frac{1}{a} - \frac{1}{b}}}{a + b}$.

B. Zeno-Behavior Analysis

In practical settings, it may be necessary to guarantee a lower-bound on the time-intervals between triggering instances. The main motivation behind this problem is to avoid Zeno-behavior for the triggering conditions. Zeno-behavior happens when an infinite number of triggering conditions are met in a finite time-interval defeating the purpose of the event-triggered control framework. In order to avoid this behavior, we introduce a small positive constant $c$ to the triggering conditions to guarantee a positive lower-bound. We have shown before that the triggering conditions given in Theorem 1 do guarantee a positive lower-bound for inner-event time instances [54]. Here, we show that our synchronization results does hold for the triggering condition $||e_j(t)||_2^2 > \delta_j||y_j(t)||_2^2 + c$ for $j = 1, ..., N$ as well. As a result, in practical applications one can use this triggering condition to secure a positive lower-bound, if necessary.
**Corollary 1.** Consider the event-triggered multi-agent system described in Section IV, where each sub-system $G_j$ is output passive with the output passivity index $\rho_j$ and is controlled by the input given in 2. If the underlying connected communication graph is balanced, the communication time-delays and disturbances are negligible, and the communication attempts amongst all agents $G_j$ where $j = 1, \ldots, N$, are governed by the triggering conditions,

$$||e_j(t)||^2 > \delta_j||y_j(t)||^2 + c,$$

where the design parameters $\delta_j$ are chosen such that,

$$0 < \delta_j \leq \frac{2}{|\mathcal{N}^m_j|}(\lambda(G) + \rho_j) - \frac{1}{\alpha} - \frac{1}{\beta},$$

then the entire event-triggered network system achieves output synchronization asymptotically.

**Proof:** The proof follows the same line of reasoning as the proof given for Theorem 1. Following the same steps, one can show that,

$$Y^T \Delta E = \sum_{j=1}^{N} |\mathcal{N}^m_j||\frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha}y_j^T(t)y_j(t)$$

$$+ \sum_{j=1}^{N} \sum_{k \in \mathcal{N}^m_j} a_k|\frac{y_k^T(t)y_k(t)}{2\beta}| + \sum_{j=1}^{N} |\mathcal{N}^m_j||\frac{(\alpha + \beta)c}{2}|.$$

Further, one sees,

$$\dot{S} = \sum_{j=1}^{N} \dot{V}_j \leq -\lambda(G)Y^TY - \sum_{j=1}^{N} \rho_jy_j^T(t)y_j(t)$$

$$+ \sum_{j=1}^{N} |\mathcal{N}^m_j||\frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha} + \frac{1}{2\beta}y_j^T(t)y_j(t) + \sum_{j=1}^{N} |\mathcal{N}^m_j||\frac{(\alpha + \beta)c}{2}|.$$  

By introducing the same matrix given in Theorem 1, $\Theta$, one has,

$$\dot{S} \leq -Y^T \Delta \Theta Y + \sum_{j=1}^{N} |\mathcal{N}^m_j||\frac{(\alpha + \beta)c}{2}|.$$

For small values of $c$, and if the event-triggered multi-agent system is design according to the corollary such that for each node $G_j$, we have: $\delta_j \leq \frac{2}{|\mathcal{N}^m_j|}(\lambda(G) + \rho_j) - \frac{1}{\alpha} - \frac{1}{\beta}$ then matrix $\Theta$ is semi-positive. Moreover, for the storage function $S$ we have: $S \geq 0$ and $\dot{S} \leq 0$ for $\forall y \in R^m$ and $\forall t \geq 0$. This implies $\dot{S} \rightarrow 0$ as $t \rightarrow \infty$ according to Barbalat’s Lemma [42]. Consequently, $Y^\Delta$ converges to the limit set $D = \{x|Y^\Delta = 0, x \in R^{mN}\}$ for all states of all agents, which proves the corollary. □
Remark 6. Corollary 7 shows a trade-off between communication rate and performance. It is clear that for very large values of $c$ (very low communication rate), the synchronization state degrades quickly. In other words, synchronization is upper-bounded according to the relation, 
$Y_{\Delta}^{T}\Theta Y_{\Delta} < \sum_{j=1}^{N} |N_{j}\cap[(\alpha+\beta)c]| \cdot \sum_{j=1}^{N} N_{j}\cap[(\alpha+\beta)c]$. As a result, the designer should consider this trade-off before selecting the design parameter $c$. However, synchronization is possible based on the assumption that $c$ is chosen to be a very small positive number, and as a result the selection of $c$ is not consequential for the synchronization of the overall system.

C. Effects of Byzantine Agents on Synchronization

We assume that amongst $N$ agents, there are $N_B$ Byzantine nodes with the attack model described in Section VI and $N_H$ honest nodes ($N_H + N_B = N$). $N_H$ and $N_B$ represent the set of honest and Byzantine agents, respectively. We represent the honest and Byzantine neighboring agents for $G_j$ by $N_{j}^{inH}$ and $N_{j}^{inB}$ ($N_{j}^{inH} \cap N_{j}^{inB} = \emptyset$, $N_{j}^{inH} \cup N_{j}^{inB} = N_{j}^{in}$). $|N_{j}^{in}|$ represents the same cardinality definition as given in Theorem I. We define the cardinality of $N_{j}^{inB}$, $|N_{j}^{inB}|$ as only the number of neighbors for the Byzantine agents excluding their communication weights. $|N_{j}^{inB}|$ is zero for honest agents. The set of all Byzantine agents is represented by $N_B$ and the set of all honest agents is represented by $N_H$. For the honest agent $G_j^H$, the input under both hypotheses may be presented as,

$$u_j^H = \sum_{k \in N_{j}^{in}} a_k(y_k(t_n^k) - y_j(t_n^j)) = \sum_{k \in N_{j}^{in}} a_k[(y_k(t) - y_j(t)) - (e_k(t) - e_j(t))],$$

where,

$$e_k'(t) = \begin{cases} e_k(t) \pm \Delta_k & \text{if } G_k \in N_{j}^{inB} \\ e_k(t) & \text{otherwise.} \end{cases}$$

For the Byzantine agent $G_j^B$, the input may be presented as,

$$u_j^B = \sum_{k \in N_{j}^{in}} a_k^B(y_k(t_n^k) - y_j(t_n^j)) = \sum_{k \in N_{j}^{in}} a_k^B[(y_k(t) - y_j(t)) - (e_k'(t) - e_j(t))],$$

where, $a_k^B = a_k + \omega_j$. The Lyapunov storage function for the entire multi-agent event-triggered network system becomes,

$$\dot{S} = \sum_{j=1}^{N} \dot{V}_j \leq \sum_{j=1}^{N} \sum_{k \in N_{j}^{in}} a_k^B[(y_k(t) - y_j(t)) - (e_k'(t) - e_j(t))]^T y_j(t) - \sum_{j=1}^{N} \rho_j y_j^T(t)y_j(t).$$
where,
\[
a'_k = \begin{cases} 
  a_k + \omega_j & \text{if } G_j \in \mathcal{N}^B \\
  a_k & \text{otherwise.}
\end{cases}
\]

It is important to note that \( \omega_j = 0 \) for honest agents. First, it can be shown that,
\[
\sum_{j=1}^{N} \sum_{k \in \mathcal{N}^n_j} a'_k (y_k(t) - y_j(t))^T y_j(t) \\
= \sum_{j \in \mathcal{N}_B} \sum_{k \in \mathcal{N}^n_j} (a_k + \omega_j)(y_k(t) - y_j(t))^T y_j(t) + \sum_{j \in \mathcal{N}_H} \sum_{k \in \mathcal{N}^n_j} a_k (y_k(t) - y_j(t))^T y_j(t) \\
= \sum_{j \in \mathcal{N}_B} \sum_{k \in \mathcal{N}^n_j} \omega_j(y_k(t) - y_j(t))^T y_j(t) + \sum_{j=1}^{N} \sum_{k \in \mathcal{N}^n_j} a_k (y_k(t) - y_j(t))^T y_j(t) \\
\leq \sum_{j \in \mathcal{N}_B} \sum_{k \in \mathcal{N}^n_j} \frac{\omega_j y_k^T(t)y_k(t)}{4} + \sum_{j=1}^{N} \sum_{k \in \mathcal{N}^n_j} a_k (y_k(t) - y_j(t))^T y_j(t).
\]

As a result, we have,
\[
\dot{S} = \sum_{j=1}^{N} \dot{V}_j \leq -Y^T L^T Y + Y^T L'^T E' - \sum_{j=1}^{N} \rho_j y_j^T(t)y_j(t) \\
+ \sum_{j \in \mathcal{N}_B} \sum_{k \in \mathcal{N}^n_j} \frac{\omega_j y_k^T(t)y_k(t)}{4},
\]

where,
\[
E' = \begin{cases} 
  e_j(t) \pm \Delta_j & \text{if } G_j \in \mathcal{N}^B \\
  e_j(t) & \text{otherwise,}
\end{cases}
\]

and \( L' \) is the Laplacian matrix of the new underlying communication graph consisting of \( a'_k \)'s and is defined as,
\[
[L']_{j,i} = \begin{cases} 
  \sum_{k \in \mathcal{N}^n_j} a'_k & \text{if } j = i \\
  -a'_k & \text{if there is an arc from } G_i \text{ to } G_j \text{ with the gain } a'_k.
\end{cases}
\]

From Section [Ⅳ], we remember,
\[
\bar{Y} = \frac{1}{N} 1_N^T Y = \frac{1}{N} \sum_{i=1}^{N} y_i,
\]
and the measure for synchronization for the multi-agent system,
\[
Y_{\Delta} = (y_1 - \bar{Y}, y_2 - \bar{Y}, ..., y_N - \bar{Y})^T.
\]
We may follow the same steps as given in Theorem 1 and get to the following,

\[
\dot{S} = \sum_{j=1}^{N} \dot{V}_j \leq -Y^T L^T Y + \sum_{j=1}^{N} (|N_{j}^{\text{in}}| \left[ \frac{1}{2\alpha} + \frac{1}{2\beta} \right] + |N_{j}^{\text{out}}| \frac{\omega_j}{2\alpha}) y_j^T(t)y_j(t) \\
+ \sum_{j=1}^{N} \sum_{k \in N_j^{\text{in}}} a_k' \frac{(\alpha + \beta)}{2} |e_j^\tau(t)| e_j^\tau(t) \dot{e}_j(t) - \sum_{j=1}^{N} \rho_j y_j^T(t)y_j(t) \\
+ \sum_{j \in N_B} \sum_{k \in N_j^{\text{in}}} \left( \frac{1}{4} + \frac{1}{2\beta} \right) \omega_j y_k^T(t)yk(t)
\]

\[
\leq -\lambda(G) Y^T Y + \sum_{j=1}^{N} (|N_{j}^{\text{in}}| \left[ \frac{1}{2\alpha} + \frac{1}{2\beta} \right] + |N_{j}^{\text{out}}| \frac{\omega_j}{2\alpha}) - \rho_j y_j^T(t)y_j(t) \\
+ \sum_{j=1}^{N} (|N_{j}^{\text{in}}| + |N_{j}^{\text{out}}|^\omega_j \frac{(\alpha + \beta)}{2} |e_j^\tau(t)| e_j^\tau(t) + \sum_{j \in N_B} \sum_{k \in N_j^{\text{in}}} (\frac{1}{4} + \frac{1}{2\beta} \omega_j y_k^T(t)yk(t), (15)
\]

where \(\alpha\) and \(\beta\) are the same parameters as given in Theorem 1. One can clearly quantify the negative effects, Byzantine nodes introduce to the entire framework by comparing (15) and (14). The error term in (15) can be expanded as well by utilizing \((e_j(t) \pm \Delta_j)^T (e_j(t) \pm \Delta_j) \leq 2(e_j^T(t)e_j(t) + \Delta_j^2)\), leading to the following,

\[
\dot{S} \leq -\lambda(G) Y^T Y + \sum_{j=1}^{N} (|N_{j}^{\text{in}}| \left[ \frac{1}{2\alpha} + \frac{1}{2\beta} \right] + |N_{j}^{\text{out}}| \frac{\omega_j}{2\alpha}) \rho_j y_j^T(t)y_j(t) \\
+ (\alpha + \beta) \sum_{j=1}^{N} (|N_{j}^{\text{in}}| + |N_{j}^{\text{out}}|^\omega_j e_j^T(t)e_j(t) + \sum_{j \in N_B} (|N_{j}^{\text{in}}| + |N_{j}^{\text{out}}|^\omega_j) \Delta_j^2) \\
+ \sum_{j \in N_B} \sum_{k \in N_j^{\text{in}}} (\frac{1}{4} + \frac{1}{2\beta} \omega_j y_k^T(t)yk(t). (16)
\]

We introduce the same square diagonal matrix \(\Theta \in R^{N \times N}\), where,

\[
[\Theta]_{j,i} = \begin{cases} 
+\lambda(G) + \rho_j - |N_{j}^{\text{in}}| \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha} + \frac{1}{2\beta} \right] & \text{if } j = i \\
0 & \text{otherwise.}
\end{cases}
\]

Given (6), (16) and \(\Theta\), we have,

\[
\dot{S} \leq -Y_\Delta^T \Theta Y_\Delta + \sum_{j \in N_B} |N_{j}^{\text{in}}| \frac{\omega_j}{2\alpha} y_j^T(t)y_j(t) \\
+ (\alpha + \beta) \sum_{j=1}^{N} (\frac{|N_{j}^{\text{in}}|}{2} + |N_{j}^{\text{out}}|^\omega_j e_j^T(t)e_j(t) + \sum_{j \in N_B} (|N_{j}^{\text{in}}| + |N_{j}^{\text{out}}|^\omega_j) \Delta_j^2) \\
+ \sum_{j \in N_B} \sum_{k \in N_j^{\text{in}}} (\frac{1}{4} + \frac{1}{2\beta} \omega_j y_k^T(t)yk(t). (17)
\]
Given the assumption that the multi-agent system was initially designed according to Theorem 1, we have $\Theta > 0$. After simplifying, and given $\dot{S} \to 0$ as $t \to \infty$, we have,

$$0 < -Y^T_\Delta \Theta Y_\Delta + \sum_{j \in \mathcal{N}_B} |\mathcal{N}^\text{in}_{jb}| \frac{\omega_j}{2\alpha} y^T_j(t) y_j(t)$$

$$+ (\alpha + \beta) \sum_{j=1}^N \left( \frac{|\mathcal{N}^\text{in}_j|}{2} + |\mathcal{N}^\text{in}_{jb}| \omega_j \right) e^T_j(t) e_j(t) + \sum_{j \in \mathcal{N}_B} \left( |\mathcal{N}^\text{in}_j| + |\mathcal{N}^\text{in}_{jb}| \omega_j \Delta^2_j \right)$$

$$+ \sum_{j \in \mathcal{N}_B} \sum_{k \in \mathcal{N}^\text{in}_j} \left( \frac{1}{4} + \frac{1}{2\beta} \right) \omega_j y^T_k(t) y_k(t).$$

(18)

(18) quantifies the effects of weight distortions and number of neighboring Byzantine agents on the convergence of the entire multi-agent system. More specifically, if one assumes the system is designed according to Theorem 1 but initialized with the presence of Byzantine nodes then one can show an upper-bound for the effects of Byzantine agents on synchronization,

$$0 < Y^T_\Delta \Theta Y_\Delta \leq \sum_{j \in \mathcal{N}_B} |\mathcal{N}^\text{in}_{jb}| \frac{\omega_j}{2\alpha} y^T_j(t) y_j(t)$$

$$+ (\alpha + \beta) \sum_{j=1}^N \left( \frac{|\mathcal{N}^\text{in}_j|}{2} + |\mathcal{N}^\text{in}_{jb}| \omega_j \right) e^T_j(t) e_j(t) + \sum_{j \in \mathcal{N}_B} \left( |\mathcal{N}^\text{in}_j| + |\mathcal{N}^\text{in}_{jb}| \omega_j \Delta^2_j \right)$$

$$+ \sum_{j \in \mathcal{N}_B} \sum_{k \in \mathcal{N}^\text{in}_j} \left( \frac{1}{4} + \frac{1}{2\beta} \right) \omega_j y^T_k(t) y_k(t).$$

(19)

It is obvious that in the presence of Byzantine agents, $Y_\Delta \neq 0$, as the honest agents will only be able to synchronize to a value that is based on the wrong data receiving from Byzantine agents ($y \pm \Delta$). In the best scenario, $y_H, y_B \to \bar{Y}' = \frac{Y_B + Y_H}{N}$, where $Y_B$ represents the Byzantine outputs of all Byzantine agents sent to their neighbors and $Y_H$ represents the true outputs of honest agents. More specifically, if one assumes that the multi-agent system is designed according to Theorem 1 but initialized with the presence of Byzantine agents then one can show a lower-bound and an upper-bound for the effects of Byzantine agents on synchronization and outputs of agents. The lower-bound happens when all agents synchronize to the wrong value $Y'$. However, even the synchronization to this false value is not guaranteed anymore due to the positive upper-bound given in (19). The positive upper-bound given in (19) characterizes the worst possible outcome inflicted upon the multi-agent system by the Byzantine agents. For honest agents, (19) demonstrates an upper-bound for the largest possible deviation caused by Byzantine agents between the honest agent’s output and the correct synchronized value. Number of Byzantine nodes has a direct relation to this effect on synchronization. A larger number of Byzantine nodes
can have a larger effect on deviating the multi-agent system from its true synchronized value. Additionally, there is the same direct effect between the Byzantine weights and synchronization, namely, the larger the Byzantine weights are, the larger deviations from the true synchronized value are. To sum up, there is a direct relationship between the upper-bound of deviations from the synchronized point (output over-shoot) and number of Byzantine neighbors and their associated Byzantine weights. The positive term involving $\Delta_j^2$ shows the relationship between the data falsification parameters and synchronization. The same direct relationship holds here as well. Falsifying the data by increasing the magnitudes of $\Delta_j$ directly weakens the synchronization of the entire multi-agent system. These results also show that an honest agent with a single Byzantine neighbor may never reach synchronization as the Byzantine neighbor by establishing the value of $\Delta_j$ and $\omega_j$ may consistently distract the honest node from reaching the synchronized value. Lastly, the positive term involving $\Delta_j^2$ guarantees a positive upper-bound for all agents, no matter if an agent has a Byzantine neighbor or not.

As an example, let’s look at the multi-agent system given in Fig. 2. We assume that in the five agent system $G_1, G_4, G_5 \in N_H$ and the rest are Byzantine agents. The honest agent $G_1$ has only one Byzantine neighbor $G_2$ with parameters $\Delta$ and $\omega$. We assume that the data falsifications for $G_2$ and $G_3$ have a positive addition sign, $y_{B_2}(t) = y_2(t) + \Delta$ and $y_{B_3}(t) = y_3(t) + \Delta$. $y_B(t)$ is the output of Byzantine agent after data falsification. We have assumed that agent $G_2$ itself has another Byzantine neighbor $G_3$ ($|N_{3B}^{in}| = 0, |N_{3B}^{in}| = 1$ with parameters $\Delta$ and $\omega$). $G_4$ and $G_5$ are honest nodes. If $|N_{1B}^{in}| = 1, |N_{1B}^{in}| = 2$ for $G_1$, $\alpha = 1$, $\beta = 1$, $\lambda(G) = 1$, $\rho_1 = 1.9$, $\delta_1 = 0.4$, $N = 5$. Given that $|N_{2B}^{in}| = 1, |N_{2B}^{in}| = 1$ and with the assumption that $\rho_2 = 0.8$, $\delta_2 = 0.3$ for $G_2$ (Notice that $\delta_1$ and $\delta_2$ meet the synchronization conditions given in Theorem 1).
and $G_2$ at their respective triggering instances $t_k$ ($c_1^T(t_k)c_1(t_k) = 0, c_2^T(t_k)c_2(t_k) = 0$), we have,

$$0 < 0.1(y_1(t_k) - \bar{Y}(t_k))^2 \leq (2 + \omega)\Delta^2 + 0.5\omega y_3^T(t_k)y_2(t_k)$$

$$+ 0.75\omega (y_1^T(t_k)y_1(t_k) + y_3^T(t_k)y_3(t_k)),$$

$$0 < 0.5(y_2(t_k) - \bar{Y}(t_k))^2 \leq (2 + \omega)\Delta^2 + 0.5\omega y_2^T(t_k)y_2(t_k)$$

$$+ 0.75\omega (y_1^T(t_k)y_1(t_k) + y_3^T(t_k)y_3(t_k)),$$

where $\bar{Y}(t_k) = \frac{y_1(t_k) + y_2(t_k) - \Delta + y_3(t_k) - \Delta + y_4(t_k) + y_5(t_k)}{5}$. This means that a single Byzantine agent can overtake an honest agent and control its behavior by determining the parameters $\omega$ and $\Delta$. Additionally, the Byzantine agents can affect $\bar{Y}$ through agent $G_1$. This also means that if each honest agent in the network has only one Byzantine neighborhood and if all Byzantine neighborhoods work together by utilizing the same Byzantine parameters $\Delta$ and $\omega$ then the Byzantine agents can coerce the entire multi-agent system into following their desired behavior. However, for this attack to be meaningful, the values $\Delta$ and $\omega$ should be large enough to disrupt the overall performance of the multi-agent system. As we will see in the following sections, larger values of $\Delta$ and $\omega$ are easy to detect given our proposed detection framework. As a result a single Byzantine neighbor can be identified easily and its negative effects can be easily mitigated. In order for the Byzantine attack to be successful, each honest node should have more than only one Byzantine neighbor and $\mathcal{N}_{j}^{\text{in}}$ should be large enough. Our aim is to characterize the relationship between the number of Byzantine neighbors, detection performance, and synchronization. We will show the minimum number of Byzantine neighbors, an honest agent should have before the detection process is entirely blinded and the Byzantine agents become undetectable. We characterize the most efficient Byzantine attack. Lastly, we propose a more resilient algorithm for synchronization of the multi-agent systems.

**Passivity and Effects of Byzantine Agents:** Passivity to some extent can compensate and mitigate the negative effects of Byzantine agents. We later illustrate this through an example. One can determine from [19], that for larger diagonal entries of $\Theta$, the effects of agents’ output overshoot, or the size of the largest possible deviation from $\bar{Y}$ may be mitigated and decreased. From the definition of $\Theta$, one can see that the excess of passivity may lead to a larger entry for agent $G_j$ in $\Theta$ and a better worst case scenario in terms of deviations from the desired value $\bar{Y}$. For all agent $G_j$ where $j = 1, ..., N$, we can represent $\rho_j = \rho'_j + \rho''_j > 0$, where
\[ \epsilon' - \rho_j' = \lambda(G) - |\mathcal{N}^{in}_j|^\frac{(\alpha+\beta)\Delta}{2} + \frac{1}{2\alpha} + \frac{1}{2\beta} \], with \( \epsilon' > 0 \). Then (19) becomes,

\[
0 < Y^T_{\Delta} \Theta^* Y_{\Delta} \leq \sum_{j \in \mathcal{N}_B} |\mathcal{N}^{in}_j| \frac{\omega_j}{2\alpha} y_j^T(t) y_j(t)
+ (\alpha + \beta)[\frac{|\mathcal{N}^{in}_j|}{2} + |\mathcal{N}^{in}_{jB}| \omega_j \epsilon_j^T(t) \epsilon_j(t) + \sum_{j \in \mathcal{N}_B} (|\mathcal{N}^{in}_j| + |\mathcal{N}^{in}_{jB}| \omega_j^2)]
+ \sum_{j \in \mathcal{N}_B} \sum_{k \in \mathcal{N}^{in}_j} \left( \frac{1}{4} + \frac{1}{2\beta} \right) \omega_j y_k^T(t) y_k(t) - Y^T_{\Delta} \Theta_{\rho_{\Delta}} Y.
\]

where

\[
[\Theta_{\rho_{\Delta}}]_{j,i} = \begin{cases} 
\rho_{\Delta}^j & \text{if } j = i \\
0 & \text{otherwise.}
\end{cases}
\]

\[
[\Theta^*]_{j,i} = \begin{cases} 
\epsilon' & \text{if } j = i \\
0 & \text{otherwise.}
\end{cases}
\]

This means that an excess of passivity in agents can compensate for the negative effects of Byzantine weight and data manipulation. A design-based interpretation of this result tells us that one can design the triggering conditions (see 1) such that \( \rho_{\Delta}^j > 0 \) for all agents, in order to increase the resilience of the entire event-triggered multi-agent system. However, this results in a decrease in the size of triggering intervals and a higher communication rate amongst agents. But the conclusion is that more passive multi-agent systems are also more resilient toward this type of Byzantine attack.

D. Simulation Example

**Example 1.** We consider a multi-agent event-triggered network system consisting of five agents \( (i = 1, ..., 5) \) with the underlying balanced communication topology given in Fig. 3. We assume the following simple dynamics for sub-systems,

\[
G_i = \begin{cases} 
\dot{x}_i(t) = -c_i x_i(t) + u_i(t) \\
y_i(t) = x_i(t),
\end{cases}
\]

with \( c_1 = 1.2, \ c_2 = 2.2, \ c_3 = 2.4, \ c_4 = 0.6, \ c_5 = 4. \) One can verify that all agents are dissipative with the storage function \( V_i(x) = \frac{1}{2} x_i^T(t) x_i(t) \). This results in output passivity indices,
\( \rho_1 = 1.2, \rho_2 = 2.2, \rho_3 = 2.4, \rho_4 = 0.6, \rho_5 = 4. \) The Laplacian matrix of the underlying communication graph amongst agents is,

\[
L = \begin{bmatrix}
2 & -1 & -1 & 0 & 0 \\
0 & 3 & -2 & 0 & -1 \\
0 & 0 & 3 & 0 & -3 \\
0 & -1 & 0 & 1 & 0 \\
-2 & -1 & 0 & -1 & 4
\end{bmatrix}.
\]

with the connectivity measure: \( \lambda(G) = 1.234. \) Based on Theorem 1, one can design the following triggering conditions,

\[
||e_1(t)||_2^2 > 0.21||y_1(t)||_2^2,
||e_2(t)||_2^2 > 0.14||y_2(t)||_2^2,
||e_3(t)||_2^2 > 0.20||y_3(t)||_2^2,
||e_4(t)||_2^2 > 0.60||y_4(t)||_2^2,
||e_5(t)||_2^2 > 0.29||y_5(t)||_2^2,
\]

by selecting \( \alpha_i = 1, \beta_i = 1 \) for \( i = 1, \ldots, 5 \). For initial conditions, \( y_1(0) = 5, y_2(0) = 10, y_3(0) = -5, y_4(0) = 1, y_5(0) = -3 \), Fig. 4 shows that the system synchronizes. Fig. 5 shows the evolution of triggering condition for each agent and shows that Zeno-behavior does not happen and the event-triggered premise is met.

Now, we show the effects of a Byzantine attack on the multi-agent system to affirm our results in the previous sub-section. We consider the case where agent \( G_1 \) is compromised. First, we
Fig. 4. The Outputs of the Multi-Agent System.

Fig. 5. The Inner-Event Time Intervals of the Multi-Agent System.
assumes that $G_1$ has not manipulated its weight but that it only manipulates the information it is sending to other agents. $G_1$ sends $\tilde{Y}_1 = Y_1 + \Delta$ where $\Delta = 10$ instead of sending its true value. Fig 6 shows the effects of the Byzantine agent on synchronization. As expected, the convergence deviates from the correct synchronized value by a positive magnitude which depends on $\Delta$, additionally, the error propagates through the network and affects other honest agents. Lastly, we consider the effects of weight manipulation and data falsification together. We assume that agent $G_5$ has changed its input weight $a_1 = 2$ to $a'_1 = 8$ and also sends the same false data $\tilde{Y}$ to its neighbors. As shown in Fig. 7, a single Byzantine agent is able to deceive the entire multi-agent system into following its desired behavior by manipulating its weight and falsifying its data. Moreover, as expected, comparing Fig 6 and Fig. 7, we see that by combining weight manipulation and data falsification, the upper-bound for all outputs of all agents has increased.

It is important to note that in our analysis of attack parameters, we did not consider the attack probability ($P$). This was to characterize the worst possible effect the Byzantine attack may have on the entire system. In the following sections we will expand the above work to consider the
probability of attack. Next, we will analyze the performance of the detection unit.

VIII. AN ANALYSIS ON THE PERFORMANCE OF THE DETECTION FRAMEWORK

A. Transient Performance Analysis of the Detection Algorithm

In this subsection, we analyze the transient performance of the detection framework. This analysis is based on characterization of the probability of correctly detecting that the multi-agent system is under attack and detecting the true signal (probability of detection) and the probability of false alarm; incorrectly determining that the multi-agent system has not reached synchronization when indeed it has (failing to detect the attack). As mentioned in Section VI, Byzantine nodes attempt to replace the hypothesis $H_0$ with $H_1$ and vice versa. The probability of detection characterizes the ability of the detection unit to discover the Byzantine agents’ attempt to ”exploit” ($H_1 \rightarrow H_0$). The false alarm probability characterizes the probability that the detection unit does not detect the Byzantine agents’ objective to ”vandalize” ($H_0 \rightarrow H_1$) and falsely decides the non-existence of an attack. This means that even-though the system has reached synchronization, the detection unit follows the falsified information received from the Byzantine agents and decides against synchronization. The transient analysis of the detection unit is based on the hypothesis testing presented in (9). We remember that each agent $G_j$ independently
calculates its local statistics with each neighbor \( G_k \), according to \( T^j_k = \sum_{i=1}^{L} |y^i_k - y^j_i|^2 \) over the time-interval of length \( L \). The overall local statistics with all neighbors of \( G_j \), which is utilized in the decision making process given in (9) at time-instance \( t \), becomes \( \wedge^j_t = (\sum_{k \in \mathcal{N}^{\text{in}}_j} T^j_k)^t \). We also remember that according to the framework presented in Section VI, the Byzantine agents are intelligent, know the true hypothesis, and will attempt to exploit and vandalize the synchronization process by confusing the detection framework and diverging the synchronization agents are intelligent, know the true hypothesis, and will attempt to exploit and vandalize the synchronization process by falsifying their data. In this subsection, we characterize the transient degradation of the detection performance in the presence of Byzantine neighbors. The local test statistic for the detection unit of an honest agents in the presence of honest and Byzantine neighbors at time-instance \( t \) is \( (\sum_{k \in \mathcal{N}^{\text{in}}_j} T^j_k)^t = (\sum_{k \in \mathcal{N}^{\text{in}}_j} T^j_k)^t + (\sum_{k \in \mathcal{N}^{\text{in}}_B} \bar{T}^j_k)^t \). For sufficiently large number of local test statistics of length \( L \), the distribution of the test statistics with a Byzantine neighbor \( G_k \), \( \bar{T}^j_k \), given the hypothesis \( H_i \) \( (i = 0, 1) \) is a Gaussian mixture of \( \mathcal{N}(\mu_0, k, \sigma_{o0}^2) \) with probability \( (1 - P_k) \) and \( \mathcal{N}(\mu_1, k, \sigma_{o1}^2) \) with probability \( P_k \). The distribution of the test statistics from an honest neighbor \( G_k \), \( T^j_k \), given the hypothesis \( H_i \) \( (i = 0, 1) \) is a Gaussian distribution \( \mathcal{N}(\mu_0, k, \sigma_{o0}^2) \), where,

\[
(\mu_0)_k = L\sigma^2_k, \quad (\mu_1)_k = L\sigma^2_k + L\bar{h}^2_k \Delta_k \sigma^2_k
\]

\[
(\mu_1)_k = (L + \eta_k)\sigma^2_k, \quad (\mu_1)_k = (L + \eta'_k)\sigma^2_k
\]

\[
(\sigma^2_{o0})_k = 2L\sigma^4_k, \quad (\sigma^2_{o1})_k = 2(L + 2L\bar{h}^2_k \Delta^2_k)\sigma^4_k
\]

\[
(\sigma^2_{o1})_k = 2(L + 2\eta_k)\sigma^4_k, \quad (\sigma^2_{o1})_k = 2(L + 2\eta'_k)\sigma^4_k
\]

As a result the probability distribution of \( \bar{T}^j_k \) becomes,

\[
f_{PDF}(\bar{T}^j_k|H_i) = (1 - P_k)\phi((\mu_0)_k, (\sigma^2_{o0})_k) + P_k\phi((\mu_1)_k, (\sigma^2_{o1})_k), \quad i = 0, 1.
\]

\( \phi(\mu, \sigma^2) \) is the probability distribution function of \( X \sim \mathcal{N}(\mu, \sigma^2) \). In order to attain a closed form for the transient probability distribution of the detection center, first we start with a simple example and then expand the results to the general case. For the sake of convenience, we assume that \( P_k = P \) for all \( k \in \mathcal{N}^{\text{in}}_j \). If we assume that agent \( G_j \) has 2 Byzantine neighbors \( (G_1, G_2) \) and 2 Honest neighbors \( (G_3, G_4) \), at time-instance \( t \), we have \( \wedge^j_t = (\bar{T}^j_1)^t + (\bar{T}^j_2)^t + (T^j_3)^t + (T^j_4)^t \). \( \wedge^j_t \) is the result of the summation of independent random variables. Consequently, the distribution of \( \wedge^j_t \) is the result of the convolution of the distribution of these independent random variables,

\[
f_{PDF}(\wedge^j_t|H_k) = f_{PDF}((\bar{T}^j_1)^t|H_k) * f_{PDF}((\bar{T}^j_2)^t|H_k) * f_{PDF}((T^j_3)^t|H_k) * f_{PDF}((T^j_4)^t|H_k), \quad k = 0, 1.
\]
Further we have,
\[
f_{PDF}(\land_j^i | H_k) = [(1 - P_1)\phi((\mu_{k0}), (\sigma_{k0}^2)) + P_1\phi((\mu_{k1}), (\sigma_{k1}^2))] \\
\ast [(1 - P_2)\phi((\mu_{k0}), (\sigma_{k0}^2)) + P_2\phi((\mu_{k1}), (\sigma_{k1}^2))] \\
\ast \phi(\mu_3, \sigma_3^2) * \phi(\mu_4, \sigma_4^2),
\]
\[
= (1 - P_1)(1 - P_2)\phi((\mu_{k0}), (\sigma_{k0}^2)) * \phi((\mu_{k1}), (\sigma_{k1}^2)) * \phi(\mu_3, \sigma_3^2) * \phi(\mu_4, \sigma_4^2) \\
+ (1 - P_1)P_2\phi((\mu_{k0}), (\sigma_{k0}^2)) * \phi((\mu_{k1}), (\sigma_{k1}^2)) * \phi(\mu_3, \sigma_3^2) * \phi(\mu_4, \sigma_4^2) \\
+ P_1(1 - P_2)\phi((\mu_{k1}), (\sigma_{k1}^2)) * \phi((\mu_{k0}), (\sigma_{k0}^2)) * \phi(\mu_3, \sigma_3^2) * \phi(\mu_4, \sigma_4^2) \\
+ P_1P_2\phi((\mu_{k1}), (\sigma_{k1}^2)) * \phi((\mu_{k0}), (\sigma_{k0}^2)) * \phi(\mu_3, \sigma_3^2) * \phi(\mu_4, \sigma_4^2), \ k = 0, 1.
\]

Given the fact that the convolution of two normal distributions is also a normal distribution with a mean and variance resulting from the summation of the means and variances of the initial normal distributions, and that \(P_k = P\) for all \(k \in N^{B}_{j}\), we have,
\[
f_{PDF}(\land_j^i | H_k) = (1 - P)(1 - P)\phi((\mu_{k0}), (\sigma_{k0}^2)) + (\mu_{k2}) + \mu_3 + \mu_4, (\sigma_{k0}^2) + (\mu_{k1}), (\sigma_{k1}^2) + (\mu_{k2}) + \mu_3 + \mu_4, (\sigma_{k0}^2) + (\mu_{k1}), (\sigma_{k1}^2) + (\sigma_{k0}^2) + (\sigma_{k1}^2) + (\sigma_{k0}^2) + (\sigma_{k1}^2) + (\sigma_{k0}^2) + (\sigma_{k1}^2) + (\sigma_{k0}^2) + (\sigma_{k1}^2), \ k = 0, 1.
\]

Byzantine agents behave probabilistically in the sense that their states change from Byzantine to honest and vice versa with a probability that depends on \(P\). We define the set \(Z_B\) as the combination of Byzantine states for Byzantine agents such that for this example, we have, \(Z_B = \{H_1, H_2, B_1, B_2, H_1, B_2, B_1, B_2\}\), where the presence of \(B_i\) or \(H_i\) in the combinations of states indicates that the Byzantine agent \(G_i\) is behaving as a Byzantine or honest agent, respectively. We denote \(Z_B^{ind}\) as indices of Byzantine agents in \(Z_B\) states, \(Z_B^{ind} = \{Z_1 = \{\}, Z_2 = \{1\}, Z_3 = \{2\}, Z_4 = \{1, 2\}\}\). As a result we have the complement set, \(C(Z_B^{ind}) = \{Z_1^c = \{1, 2\}, Z_2^c = \{2\}, Z_3^c = \{1\}, Z_4^c = \{\}\}\). Needless to say, we have the following cardinality relationship, \(|Z_i| + |N_H| = N|\).

**Lemma 1.** The probability distribution function of the local test statistic for agent \(G_j\) with \(N_B\) Byzantine neighbors and \(N_H\) honest agents with the detection time-interval \(L\), at time-instance
t, given the hypothesis $H_i$ ($i = 0, 1$), $\land_j^t$ is a Gaussian mixture determined by,

$$f_{PDF}(\land_j^t|H_k) = \sum_{z_i \in Z_i^{ind}} P|z_i|(1 - P)^{|Z_i|}Q(\phi(\mu, \sigma^2)),$$

where,

$$\mu = \sum_{i \in Z_i^c} (\mu_{k0}) + \sum_{i \in Z_i} (\mu_{k1}),$$

$$\sigma^2 = \sum_{i \in Z_i^c} (\sigma_{k0}^2) + \sum_{i \in Z_i} (\sigma_{k1}^2),$$

and $k = 0, 1$.

Consequently, the transient performance of the detection unit of agent $G_j$ may be characterized by the probability of detection and false alarm as follows,

**Proposition 1.** The probability of detection and false alarm of the detection unit for agent $G_j$ at time-instance $t$ may be characterized as,

$$(P_D^j)^t = Pr(D = H_1|H_1)$$

$$= \sum_{z_i \in Z_i^{ind}} P|z_i|(1 - P)^{|Z_i|}Q\left(\frac{\gamma_j - \sum_{i \in Z_i^c}(\mu_{10}) - \sum_{i \in Z_i}(\mu_{11}) - \sum_{i \in N_H}(\mu_{10})}{\sqrt{\sum_{i \in N}(\sigma_{10}^2)}}\right),$$

$$(P_{FA}^j)^t = Pr(D = H_1|H_0)$$

$$= \sum_{z_i \in Z_i^{ind}} P|z_i|(1 - P)^{|Z_i|}Q\left(\frac{\gamma_j - \sum_{i \in Z_i^c}(\mu_{00}) - \sum_{i \in Z_i}(\mu_{01}) - \sum_{i \in N_H}(\mu_{00})}{\sqrt{\sum_{i \in N}(\sigma_{00}^2)}}\right).$$

**Proof.** $(P_D^j)^t$ can be easily derived from calculating $Pr(D = H_1|H_1)$ from a combination of Gaussian distributions with the following means and variances, $(\mu_{10}) = (L + \eta_k)\sigma_k^2$, $(\mu_{11}) = (L + \eta_k')\sigma_k^2$ and $(\sigma_{10}) = 2(L + 2\eta_k)\sigma_k^2$, $(\sigma_{11}) = 2(L + 2\eta_k')\sigma_k^2$ for all neighbors $k \in N_{in}$ (honest and Byzantine agents $G_j$) according to Lemma 1. It is important to note that for a Gaussian distribution $Y$ with the mean $\mu$ and variance $\sigma^2$, $X = \frac{Y - \mu}{\sigma}$ is a standard normal distribution and $P(Y > y) = P(X > x) = Q(\frac{y - \mu}{\sigma^2}) = Q(x)$. $(P_{FA}^j)^t$ or $Pr(D = H_1|H_0)$ may be calculated in a similar manner given the means and variances, $(\mu_{00}) = L\sigma_k^2$, $(\mu_{01}) = L\sigma_k^2 + Lh_k^2g_k\sigma_k^2$ and $(\sigma_{00}) = 2L\sigma_k^4$, $(\sigma_{01}) = 2(L + 2Lh_k^2g_k^2)\sigma_k^4$ for all $k \in N_{in}$ (honest and Byzantine agents $G_j$). It is important to note that one may first establish a desired rate of false alarm by deciding $\gamma_j$ and then determine the detection performance. 

It is important to note that the probability of detection indicates the probability that agent $G_j$ detects the Byzantine attack which is trying to distort performance by persuading the detection unit that the entire network has reached synchronization or by forcing the honest agent to follow
the falsified data, and consequently decides against it based on the distribution of the true signal. Additionally, under (9), the probability of false alarm indicates the probability that Byzantine neighbors will succeed in coercing agent \( G_j \) into mistakenly deciding that the entire multi-agent network has not reached synchronization when indeed it has, thereby fulfilling its adversarial objective to move the system from \( H_0 \rightarrow H_1 \).

**Example 2.** Consider agent \( G_2 \) in the event-triggered multi-agent system given in Example 1 with three neighbors. We consider the same underlying communication graph and dynamics for the entire event-triggered multi-agent system. We assume that \( G_5 \) is a Byzantine neighbor and \( G_1 \) and \( G_4 \) are honest neighbors for agent \( G_2 \). We will analyze the transient detection and false alarm probability distributions for the local test statistic \( \Lambda_j^2 \) for agent \( G_2 \)'s detection center and quantify the harmful effects of the Byzantine neighbor \( G_5 \) on the detection performance. We consider the same dynamics for the agents and the initial conditions, \( y_1(0) = 3.5, \ y_2(0) = 4, \ y_3(0) = 0.5, \ y_4(0) = 3, \ y_5(0) = 2 \). The Byzantine agent manipulates its weight to \( a_5' = a_5 + 1 \). We consider the channel gains \( \tilde{h}_1 = 0.92, \ \tilde{h}_4 = 0.95, \ \tilde{h}_5 = 0.96 \) and assume \( \sigma_i^2 = 1 \) for all the communication links, \( i = 1, ..., 5 \). We plot the transient performance of the detection unit for a set of attack strengths \( \Delta_1 = 0.8, \ \Delta_1 = 0.9, \ \Delta_1 = 1, \ \Delta_1 = 1.2 \) and \( \Delta_1 = 1.6 \). In all cases, we assume the probability of attack \( P_5 = 0.5 \). Lastly, the detection interval for the detection unit is \( L = 15 \) and \( \lambda = 15 \) where, \( \gamma_j = \sum_{k \in N_j} L \sigma_k^2 + \lambda \) is chosen based on the desired false alarm rate (see (9)). Fig. 8 depicts the transient detection performance for different attack strengths. As seen, the Byzantine neighbor can considerably harm the detection performance. Similarly, the false alarm rate for the same set-up for when the entire multi-agent has synchronized (under \( H_0 \)) is shown in Fig. 9. It is clear that the Byzantine neighbor can considerably increase the false alarm rate by appropriately selecting the attack parameters. This means that agent \( G_2 \) will mistakenly continue its communication with its neighbors based on the false belief that the multi-agent system has not reached synchronized. Similar to the steady-state analysis of the detection framework, this example shows that the Byzantine agents can considerably degrade the transient performance of the detection unit.

**B. Steady-State Performance Analysis of the Detection Algorithm**

Our detection platform is based on Neyman-Pearson theorem. A Neyman-Pearson based detector measures the signal-to-noise ratio for the unknown signal \( y \) over a certain time-interval.
Fig. 8. Probability of Detection (Detection Interval $L = 15$, Attack Parameters: $P_5 = 0.5$, $\Delta_5$ and $a'_5 = a_5 + 1$).

Fig. 9. Probability of False Alarm (Detection Interval $L = 15$, Attack Parameters: $P_5 = 0.5$, $\Delta_5$ and $a'_5 = a_5 + 1$).
and detects the presence of the deterministic signal $s$ in $y$ at points, where the signal-to-noise ratio is maximized. It is known that the Neyman-Pearson theorem provides the optimal decision criterion, where the likelihood ratio is compared with a threshold $\gamma$, previously computed to minimize a given false alarm probability \[55\]. The selection of the optimal global threshold $\gamma$ is beyond the scope of the work presented here, and we assume that $\gamma$ in (9) has already been selected based on some performance and application criteria.

We characterize the steady-state performance of our proposed detection platform performance by examining its deflection coefficient against intelligent Byzantine attacks. The deflection coefficient of the test statistic is defined as,

$$D(\wedge) = \frac{E[\wedge|H_1] - E[\wedge|H_0]}{E[(\wedge - E[\wedge|H_0])^2|H_0]},$$

(25)
or the difference of the means (expectations) of the test statistic under the two independent and identically distributed hypothesis distributions, $H_0$ and $H_1$ (with the same variance), divided by the variance of the test statistic under $H_0$. The deflection coefficient formula given above can characterize the steady-state performance of the detection framework and analyze the limitations of the detection procedure by quantifying the distributions under both null and alternative hypotheses based on the number of Byzantine neighbors and attack parameters. This quantification can characterize the distance between the expectations of these two hypotheses to show a limit-case probability of correctly detecting the attacks. Since, the deflection coefficient is directly related to the area of overlapped regions between two distributions, it can efficiently characterize the decision performance in a binary hypothesis testing environment [48], [50], [56]. Moreover, the deflection coefficient can be obtained by only calculating the mean and variance from the observed data set without modeling the exact distributions. There is a direct relationship between the detection performance and positive values of the deflection coefficient. Given the event-triggered multi-agent network system design proposed in previous sections and designed according to Theorem [1] and the proposed detection framework in Section V, here we characterize the minimum number of Byzantine neighboring agents required to make the detection coefficient for test statistics equal to zero. First, we characterize the relationship between the number of Byzantine neighbors of an honest agent and the performance of its detection unit. Second, we characterize the minimum number of Byzantine agents that can entirely blind the detection unit of a single honest agent.
Consider an event-triggered multi-agent system designed according to Theorem 2. We define, $\eta$ between the honest agent $G$ and its neighbor $G_k$. For an honest agent with $N_h$ honest and $N_B$ Byzantine neighbors, the condition for the detection unit to become entirely blinded or to make the deflection coefficient zero over the detection interval $L$ is,

$$
\sum_{k=1}^{N_B} LP_k [2\hat{h}_k \Delta_k (\mu_k - \mu_j) + \hat{h}_k^2 \Delta_k^2 (\sigma_k^2 - 1)] = \sum_{k=1}^{N} \eta_k \sigma_k^2,
$$

where $\eta_k = \frac{\sum_{i=1}^{L} [\hat{h}_k s_k - y_j (t_i)]^2}{\sigma_k^2}$, $\mu_j = \frac{1}{L} \sum_{i=1}^{L} y_j^i$, and $\mu_k = \frac{1}{L} \sum_{i=1}^{L} \hat{h}_k y_k^i$.

**Proof.** As mentioned in Section V, each local test statistic $T_k$ over the detection time-interval $L$ between the honest agent $G_j$ and its neighbor $G_k$ may follow a central or non-central chi-square distribution. We define, $\eta_k = \frac{\sum_{i=1}^{L} [\hat{h}_k s_k - y_j (t_i)]^2}{\sigma_k^2}$ for an honest communication from agent $G_k$ to the honest agent $G_j$ over the detection interval $L$. We define, $\eta'_k = \frac{\sum_{i=1}^{L} [\hat{h}_k s_k - y_j (t_i)]^2}{\sigma_k^2}$ for a Byzantine communication from agent $G_k$ to the honest agent $G_j$ over the detection interval $L$. We can see that, $\eta'_k = \eta_k + \frac{1}{\sigma_k^2} \sum_{i=1}^{L} [\hat{h}_k s_k - y_j (t_i)]^2$. Further, the true mean $(\mu_{kj})$ and variance $(\sigma_{kj}^2)$ under the null hypothesis $H_0$ and alternative hypothesis $H_1$ for honest communications are as follows,

$$
\mu_{kj} = \begin{cases} 
L \sigma_k^2 & \text{Under } H_0 \\
(L + \eta_k) \sigma_k^2 & \text{Under } H_1,
\end{cases}
$$

$$
\sigma_{kj}^2 = \begin{cases} 
2L \sigma_k^4 & \text{Under } H_0 \\
2(L + 2\eta_k) \sigma_k^4 & \text{Under } H_1.
\end{cases}
$$

Above, $\eta_k \approx 0$ is implied under $H_0$ or the hypothesis that the two agents have synchronized according to (9). For the honest agent $G_j$ with $N$ neighbors where, $N_B$ of them are Byzantine and $N_h$ of them are honest, we may have,

$$
E[\land H_0] = \sum_{k=1}^{N_h} L \sigma_k^2 + \sum_{k=1}^{N_B} [P_k (L + L \hat{h}_k \Delta_k) \sigma_k^2 + (1 - P_k)L \sigma_k^2]
$$

$$
E[\land H_1] = \sum_{k=1}^{N_h} (L + \eta_k) \sigma_k^2 + \sum_{k=1}^{N_B} [P_k ((L + \eta_k) \sigma_k^2) + (1 - P_k)(L + \eta_k) \sigma_k^2],
$$

$$
E[(\land - E[\land H_0])^2 | H_0] = \sum_{k=1}^{N_h} 2L \sigma_k^4 + \sum_{k=1}^{N_B} [P_k L^2 \hat{h}_k \Delta_k^4 \sigma_k^4 - P_k^2 L^2 \hat{h}_k \Delta_k^4 \sigma_k^4 + 2L \sigma_k^4].
$$
Utilizing the above definitions into $E[\land|H_1] - E[\land|H_0]$ and simplifying further we have,

$$E[\land|H_1] - E[\land|H_0] = \sum_{k=1}^{N_H} \eta_k \sigma_k^2 + \sum_{k=1}^{N_B} \eta_k \frac{\left[P_k \left[ \sum_{i=1}^{L} (\tilde{h}_k^2 \Delta_k^2 + 2 \tilde{h}_k \Delta_k y_i^j - 2 \tilde{h}_k^2 \Delta_k y_i^j) \right] \right]}{\sigma_k^2} - \tilde{h}_k^2 \Delta_k \sigma_k^2 + \eta_k \sigma_k^2$$

We denote the means of the output signals of agents $G_j$ and $G_k$ over the detection time-interval $L$ as $\mu_j = \frac{1}{L} \sum_{i=1}^{L} y_i^j$ and $\mu_k = \frac{1}{L} \sum_{i=1}^{L} \tilde{h}_k y_i^k$. For the Byzantine agents to be able to blind the detection unit ($D(\land) = 0$), they need to enforce $E[\land|H_0] = E[\land|H_1]$. This means that,

$$\sum_{k=1}^{N_B} LP_k [2 \tilde{h}_k \Delta_k (\mu_k - \mu_j) + \tilde{h}_k^2 \Delta_k^2 (\sigma_k^2 - 1)] = \sum_{k=1}^{N} \eta_k \sigma_k^2,$$

(29)

where $\eta_k = \frac{\sum_{i=1}^{L} |\tilde{h}_k y_i^k - y_0(t_j^j)|^2}{\sigma_k^2}$. This quantifies the steady-state effects of the number of neighboring Byzantine agents, attack strengths and attack probabilities on the detection unit of an honest agent and also proves the theorem.

If we assume that $\Delta_k = \Delta$, $P_k = P$, and $\tilde{h}_k = \tilde{h}$ for all $k = 1, ..., N_B$ and $\eta_k = \eta$ and $\sigma_k = \sigma$ for all $k = 1, ..., N$, and quantify the distances between the means of Byzantine agents’ outputs and the honest agent’s output, $\mu_k - \mu_j = d_k = D$ for $k = 1, ..., N_B$, then the condition given in Theorem 2 simplifies to

$$\frac{N_B}{N} = \frac{LP |2 \tilde{h} \Delta + h^2 \Delta^2 (\sigma^2 - 1)|}{N_B},$$

where $N$ represents the number of neighbors for agent $G_j$. This relation shows that an intelligent Byzantine attack can blind the entire detection framework by an appropriate selection of $P$ and $\Delta$. This also means that blinding the detection framework is still possible even in cases that the Byzantine nodes are in the minority in the neighborhood of the honest agent $G_j$. Moreover, this reveals the trade-off that if the Byzantine agents are in the minority in the neighborhood of agent $G_j$, then they will need to select larger attack parameters ($P$ and $\Delta$), in order to blind the detection unit, this, however, in return makes the job of detection easier for the honest agents. For the honest node, this shows the importance of quick detection of rogue agents. For the Byzantine agents, this shows the importance of quickly occupying the neighborhood of $G_j$ in order to maintain their inconspicuous state and fulfill their adversarial objectives. Lastly, this relationship shows the importance of the distance between the means of outputs of Byzantine agents and the honest agent. In case of a Byzantine attack, the larger the distances between the means of outputs of the occupied agents and the mean of the output of the honest agent are, the easier it is for the Byzantine agents to degrade the performance of the multi-agent systems. In other words, the Byzantine agents will require to exert less effort (smaller values for $\Delta$ and $P$) to blind the detection unit. This also helps the Byzantine agents to stay hidden. However, the job of
selecting attack parameters for Byzantine agents becomes more complicated as the multi-agent system gets closer to the synchronized state. In other words, this relation also reveals the trade-off between degrading the performance by the Byzantine attack and the desire to stay hidden from the detection unit.

**Example 3.** Consider agent $G_2$ in the event-triggered multi-agent system given in Example 1 with three neighbors. We consider the same underlying communication graph and initial conditions for the entire event-triggered multi-agent system. We assume that $G_4$ is a Byzantine neighbor and $G_1$ and $G_5$ are honest neighbors of agent $G_2$. We consider the following channel gains for the communication links between agents $G_2$ and its neighbors: $\tilde{h}_1 = 0.8$, $\tilde{h}_4 = 0.90$, $\tilde{h}_5 = 0.72$ and assume $\sigma_i^2 = 1.2$ for all the noise in all the communication links i.e. $i = 1,...,5$. The detection units rely on the detection time-interval $L = 20$. The deflection coefficient for agent $G_2$’s detection unit is depicted in Fig. [10]. The contour plot (Fig. [11]) shows the underneath of the three-dimensional shaded surface in Fig. [10] and clarifies the relationship amongst the attack strength parameter $\Delta_4$, the attack probability $P_4$ and the deflection coefficient of the detection unit located on agent $G_2$ i.e. $D_2(\wedge)$. The larger values of $\Delta_4$ in general, makes the job of detection easier, if $P_4$ is kept small enough. For the Byzantine agent to be able to blind the detection unit, a certain balance between $\Delta_4$ and $P_4$ is required. Lastly, Fig [10] shows that it is possible for a single Byzantine agent to blind the agent $G_2$’s detection unit, even though the majority of $G_2$’s neighbors are honest nodes. Indeed, the fact that $\frac{1}{3}$ of the $G_2$’s neighborhood is occupied by a Byzantine agent has severely degraded the detection performance.

Next, we will propose two different learning-based control approaches to mitigate the negative effects mentioned above, imposed by Byzantine agents on the multi-agent system.

**IX. A LEARNING-BASED CONTROL METHOD FOR MITIGATING THE EFFECTS OF THE BYZANTINE ATTACK**

A. *Distributed Weight Assignments for Mitigating the Effects of Weight Manipulations*

In this subsection, we propose a robust distributed weight design that will achieve synchronization and in the case of an attack mitigate the adversarial effects of Byzantine agents. First, we will deal with the problem of weight manipulation. Common in the literature, it is assumed that the feedback control framework given in (2) is designed by the agent itself. This will leave the entire event-triggered multi-agent framework extremely vulnerable to adversarial attempts
such as node capture or Byzantine attacks as the nodes themselves can independently have a
great influence on the synchronization efforts (in our case, this great influence was quantified
in Subsection VII-C). Here, we propose a synchronization algorithm in which the weights for
the feedback control for agent $G_j$ are assigned by its neighbors $G_k$, where $k \in \mathcal{N}_j^{in}$. In other
words, with sending its first information, the neighbor $G_k$ also sends its desired feedback weight
which will be used to initiate the feedback control of agent $G_j$ in order to reach synchronization.
Under this framework, we assume that each agent $G_j$ is aware of its $d_{in}(j)$ (defined in Section

Fig. 10. Deflection Coefficient for agent $G_2$ in Example 1 as a function of Attack Probability $P$ and Attack Strength $\Delta$.

Fig. 11. Contour plot of the Deflection Coefficient for agent $G_2$ as a function of Attack Probability $P$ and Attack Strength $\Delta$. 
similar to before the communication framework is balanced \((d_{in}(j) = d_{out}(j))\) and that each agent \(G_j\) has only authority over designing its own triggering condition by selecting its design parameter \(\delta_j\) —feedback weights are assigned by neighbors. As a result, a Byzantine agent, instead of being able to diverge the entire behavior of an overtaken agent in the event-triggered multi-agent network system and consequently mislead the entire multi-agent system, is only able to partly distract the proper behavior of its neighboring agents. An honest agent will only be taken over entirely, if the majority of its neighbors are Byzantine. This is highly unlikely in the presence of a detection framework. This will extremely lighten the burden of the mitigation process and improve the overall performance.

**Theorem 3.** Consider the event-triggered multi-agent network system described in Section IV, where each sub-system \(G_j\) is output passive with the output passivity index \(\rho_j\) and is controlled by the input given in (2). Consider that the feedback weights in (2) for each sub-system \(G_j\) are assigned by its neighbors \(G_k\), where \(k \in N_j^{in}\). If the underlying connected communication graph resulting from weight assignments is balanced, the communication time-delays and disturbances are negligible, and the communication attempts amongst all agents \(G_j\) where \(j = 1, ..., N\), are governed by the triggering conditions,

\[
||e_j(t)||_2^2 > \delta_j||y_j(t)||_2^2,
\]

where the design parameters \(\delta_j\) are chosen such that,

\[
0 < \delta_j \leq \frac{2N_j^{in}(\lambda(\tilde{G}) + \rho_j) - \frac{1}{\alpha} - \frac{1}{\beta}}{\alpha + \beta},
\]

where \(\alpha > 0\) and \(\beta > 0\) are design variables and \(\lambda(\tilde{G})\) is the connectivity of the underlying communication graph, then the entire event-triggered multi-agent network system achieves output synchronization asymptotically.

**Proof.** Proof is similar to Theorem [1] as we assume that all agents are honest and initiate the neighbors’ feedback control based on their respective \(d_{in}\) according to the assumption that the resulting underlying communication graph is balanced. One can define a modified Laplacian matrix for the communication graph \(\tilde{L} = \tilde{D} - \tilde{A}\), where \(\tilde{D}\) is an \(N \times N\) diagonal matrix with \(\tilde{d}_{j,j} = d_{in}(j)\), representing the sum of assigned weights to agent \(G_j\) and \(\tilde{A}\) is the \(N \times N\) adjacency matrix with \(\tilde{a}_{i,j} \neq 0\) representing the gain assigned by agent \(G_i\) to agent \(G_j\), and \(\tilde{a}_{i,j} = 0\) when there is no communication link between two agents \(G_i\) and \(G_j\). Since the communication link
is balanced \((d_{in}(j) = d_{out}(j) \text{ for } \forall j = 1, \ldots, N)\), \(\bar{L} = L\), similarly, \(\lambda(\bar{G}) = \lambda(G)\) then one can simply represent the new framework based on the previous one and show synchronization for the entire event-triggered multi-agent network system following the same steps presented in Theorem \(\square\)

Next, we analyze the effects of a Byzantine attack where a Byzantine agent \(G_k\) will disturb the balanced communication graph through weight manipulation by assigning \(a_k^B = a_k + \omega_k\) to its neighbors, where \(\omega_k > 0\). For analytical tractability, we do not consider the data falsification and will show that the new approach will mitigate the negative effects of weight manipulation. In the next subsection, we will discuss the mitigation process for the data falsification part of the Byzantine attack. Similar to the previous sections, we assume that amongst the \(N\) agents, there are \(N_B\) Byzantine nodes with the attack model described in Section \(\square\) and \(N_H\) honest nodes \((N_H + N_B = N)\). \(\mathcal{N}_H\) and \(\mathcal{N}_B\) represent the set of honest and Byzantine agents, respectively. We represent the honest and Byzantine neighboring agents for \(G_j\) by \(\mathcal{N}_{jn}^H\) and \(\mathcal{N}_{jn}^B\) \((\mathcal{N}_{jn}^H \cap \mathcal{N}_{jn}^B = \emptyset, \mathcal{N}_{jn}^H \cup \mathcal{N}_{jn}^B = \mathcal{N}_{jn}^B)\). \(|\mathcal{N}_{jn}^B|\) represents the same cardinality definition as given in Theorem \(\square\) The set of all Byzantine agents is represented by \(\mathcal{N}^B\) and the set of all honest agents is represented by \(\mathcal{N}^H\). It is important to note that for the case where the feedback weights are assigned by the neighbors, the Byzantine neighbor \(G_k^B\) assigns \(a_k^B = a_k + \omega_k\) to the feedback control for agent \(G_j\), otherwise, \(a_k^H = a_k\) is assigned. The Lyapunov storage function for the entire event-triggered multi-agent network system becomes,

\[
\dot{S} = \sum_{j=1}^{N} \dot{V}_j \leq \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_{jn}^B} a_k [(y_k(t) - y_j(t)) - (e_k(t) - e_j(t))]^T y_j(t) \\
+ \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_{jn}^B} \omega_k [(y_k(t) - y_j(t)) - (e_k(t) - e_j(t))]^T y_j(t) - \sum_{j=1}^{N} \rho_j y_j^T(t) y_j(t).
\]

It is important to note that \(\omega_k = 0\) for honest neighbors. First, it can be shown that,

\[
\sum_{j=1}^{N} \sum_{k \in \mathcal{N}_{jn}^B} \omega_k (y_k(t) - y_j(t))^T y_j(t) \leq \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_{jn}^B} \frac{\omega_k y_k^T(t) y_k(t)}{4}.
\]
We follow the same approach as before and end up with,

\[
\dot{S} = \sum_{j=1}^{N} \dot{V}_j \leq -Y^T L^T Y + Y^T L^T E - \sum_{j=1}^{N} \rho_j y_j^T(t) y_j(t) \\
+ \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j^{inB}} \omega_k y_k^T(t) y_k(t) \leq -\sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j^{inB}} \omega_k [(e_k(t) - e_j(t))]^T y_j(t) \\
= -Y^T L^T Y + Y^T L^T E - \sum_{j=1}^{N} \rho_j y_j^T(t) y_j(t) + \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j^{inB}} \frac{\omega_k y_k^T(t) y_k(t)}{4},
\]

(31)

where \(\alpha\) and \(\beta\) are the same parameters as given in Theorem 1. We denote, \(|\mathcal{W}_j|\) as the sum of the weight manipulations that were assigned to the agent \(G_j\) from its Byzantine neighbors. \(L'\)

is the Laplacian matrix of the new underlying communication graph consisting of \(a'_k\)'s and is defined as,

\[
[L']_{j,i} = \begin{cases} 
\sum_{k \in \mathcal{N}_j^{in}} a'_k & \text{if } j = i \\
-a'_k & \text{if there is an arc from } G_i \text{ to } G_j \text{ with the gain } a'_k,
\end{cases}
\]

where \(a'_k\) are defined as before. We may follow the same steps as given in Theorem 1 and get to the following,

\[
\dot{S} = \sum_{j=1}^{N} \dot{V}_j \leq -Y^T L^T Y + \sum_{j=1}^{N} (|\mathcal{N}_j^{in}| + |\mathcal{W}_j|) \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha}\right] y_j^T(t) y_j(t) \\
+ \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j^{in}} \omega_k \left[ \frac{y_k^T(t) y_k(t)}{2\beta} \right] - \sum_{j=1}^{N} \rho_j y_j^T(t) y_j(t) + \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j^{inB}} \frac{\omega_k y_k^T(t) y_k(t)}{4} \\
\leq -\lambda(G) Y^T Y + \sum_{j=1}^{N} (|\mathcal{N}_j^{in}| + |\mathcal{W}_j|) \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha}\right] y_j^T(t) y_j(t) \\
+ \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j^{in}} \omega_k \left[ \frac{y_k^T(t) y_k(t)}{2\beta} \right] - \sum_{j=1}^{N} \rho_j y_j^T(t) y_j(t) + \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j^{inB}} \frac{\omega_k y_k^T(t) y_k(t)}{4},
\]

(32)

We introduce the same square diagonal matrix \(\Theta \in \mathbb{R}^{N \times N}\), where,

\[
[\Theta]_{j,i} = \begin{cases} 
+\lambda(G) + \rho_j - |\mathcal{N}_j^{in}| \left[ \frac{\alpha + \beta}{2} + \frac{1}{2\alpha} + \frac{1}{2\beta} \right] & \text{if } j = i \\
0 & \text{otherwise},
\end{cases}
\]
Given (32) and \( \Theta \), we have,

\[
\dot{S} \leq -Y^T \Theta Y + \sum_{j=1}^{N} \left| W_j \right| \left[ \frac{(\alpha + \beta) \delta_j}{2} + \frac{1}{2\alpha} y_j^T(t) y_j(t) + \sum_{j=1}^{N} \sum_{k \in N_j^{\alpha B}} \omega_k \left( \frac{1}{4} + \frac{1}{2\beta} \right) y_k^T(t) y_k(t) \right].
\]

(33)

Given the assumption that the multi-agent system was initially designed according to Theorem 1, we have \( \Theta > 0 \). After simplifying, and given \( \dot{S} \to 0 \) as \( t \to \infty \), we have,

\[
0 < Y^T \Theta Y \leq \sum_{j=1}^{N} \left| W_j \right| \left[ \frac{(\alpha + \beta) \delta_j}{2} + \frac{1}{2\alpha} y_j^T(t) y_j(t) + \sum_{j=1}^{N} \sum_{k \in N_j^{\alpha B}} \omega_k \left( \frac{1}{4} + \frac{1}{2\beta} \right) y_k^T(t) y_k(t) \right].
\]

(34)

Comparing (34) with (19), one can see that assignments of the agents’ weights by their neighbors can greatly decrease the magnitude of the upper-bound on the deviations from the synchronized state. Additionally, by not allowing the Byzantine agents design their own weights, we are diversifying the negative effects caused by the weight manipulations. We are dividing the weight manipulation attack into two parts. One part is still related to the Byzantine agents and their outputs and cannot be mitigated without a direct access to the corrupt agents (second part of the summation given in (34)). The first part of the upper-bound shown in (34), however, may be mitigated by the honest agents given their passivity indices and their design of the triggering conditions. This greatly helps with the synchronization process and lowers the upper-bound of the deviations. Moreover, if a Byzantine agent only has honest neighbors, through this mitigation process, the output of the Byzantine agent will eventually reach synchronization as the information and weights received by the Byzantine agent from its honest neighbors will follow the requirements given in Theorem 3. Consequently, the honest agents will be able to entirely mitigate the negative effects of weight manipulations – This will be illustrated in Example 4. This combined with the detection framework presented in the next section for dealing with data falsifications, can completely eradicate the negative effects of the Byzantine attack. We will explain this in more details next.

**Mitigating the effects of Weight Manipulation by utilizing the Passivity Properties of Agents:** As it was characterized before passivity can ameliorate the effects of a Byzantine attack. For all agents \( G_j \) where \( j = 1, \ldots, N \), we can represent the passivity indices with, \( \rho_j = \)
\[ \rho_j^\prime + \rho_j^\Delta > 0, \text{ where } \rho_j^\prime > 0 \text{ and } \rho_j^\Delta > 0. \] We assume that the triggering conditions are designed according to Theorem 3, where,

\[ 0 < \delta_j \leq \frac{2}{|N_{in}^j|} \left( \lambda(G_j) + \rho_j \right) - \frac{1}{\alpha} - \frac{1}{\beta}, \]

Simplifying this relation based on \( \rho_j \), and annotating \( \rho_j^\prime = \lambda(G) - |N_{in}^j| \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha} + \frac{1}{2\beta} \right] \), we may have, \( \rho_j - \rho_j^\Delta = \rho_j^\prime \). As a result, if the triggering conditions are designed such that,

\[ \rho_j^\Delta > \sum_{k \in N_{in}^j} \omega_k \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha} \right] \]

where \( \omega_k \)'s are weight manipulations caused by the Byzantine neighbors, then the effects of weight manipulations committed by the Byzantine agents are assuaged. To extrapolate this result to the entire event-triggered multi-agent system, we assume that the triggering conditions are designed such that the above relation holds for each agent, we introduce the positive definite diagonal \( N \times N \) matrix \( \Theta^\Delta \in \mathbb{R}^{N \times N} \), where,

\[
[\Theta^\Delta]_{j,i} = \begin{cases} 
\rho_j^\Delta - \sum_{k \in N_{in}^j} \omega_k \left[ \frac{(\alpha + \beta)\delta_j}{2} + \frac{1}{2\alpha} \right] > 0 & \text{if } j = i \\
0 & \text{otherwise.} 
\end{cases}
\]

As a result, (34) becomes,

\[ 0 < Y_T^\Delta \Theta^\Delta Y \leq -Y_T^\Delta \Theta^\Delta Y + \sum_{j=1}^{N} \sum_{k \in N_{in}^j} \omega_k \left( \frac{1}{4} + \frac{1}{2\beta} \right) y_k^T(t)y_k(t). \] (35)

Defining the positive definite matrix \( \Theta' = \Theta + \Theta^\Delta \), and simplifying further we have,

\[ 0 < Y_T^\Delta \Theta' Y \leq \sum_{j=1}^{N} \sum_{k \in N_{in}^j} \omega_k \left( \frac{1}{4} + \frac{1}{2\beta} \right) y_k^T(t)y_k(t). \] (36)

By comparing (36) with (34) and consequently (18), one can see that the new method of distributed weight assignments in conjunction with the utilization of the passivity qualities of sub-systems can greatly mitigate the effects of a Byzantine attack’s weight manipulations. It is important to note that by giving the authority to the honest agents to be able to adjust their triggering conditions according to their assessment of the magnitude of weight manipulations committed by their neighboring Byzantine agents, one can entirely mitigate this part of the Byzantine attack. This is done based on the fact that the honest agent \( G_j \) is aware of its \( d_{in}(j) \) and can estimate the weight manipulations by observing the difference between its \( d_{in}(j) \) and the actual weights assigned to \( G_j \) by its neighbors. Moreover, by decreasing the magnitude
of the design variable $\delta_j$ (shortening the triggering intervals - increasing the communication rate), $G_j$ may increase $\rho_j^\Delta$ and compensate for the negative effects of weight manipulations. This will entirely eradicate the negative effects of weight manipulations committed by isolated Byzantine agents (with no Byzantine neighbors). However, in order to mitigate the negative effects of weight manipulations in cases where the Byzantine agents have Byzantine neighbors will require further attention. The mitigation method offered in the next section combined with the detection framework will attempt to further mitigate these negative effects. Lastly, one can initiate the design of the event-triggered multi-agent system by selecting smaller values for $\delta_j$, $j = 1, ..., N$ (a more conservative event-triggered design). This will generally result in a more resilient event-triggered multi-agent network system against Byzantine attacks.

**Example 4.** We consider an event-triggered multi-agent network system consisting of four agents ($i = 1, ..., 4$) with the underlying balanced communication topology given in Fig. [72] We assume the following dynamics for agents,

$$
G_i = \begin{cases} 
\dot{x}_i(t) = -c_i x_i(t) + u_i(t) \\
y_i(t) = x_i(t),
\end{cases}
$$

with $c_1 = 1.2, c_2 = 1.8, c_3 = 2.6, c_4 = 0.80$ and initial conditions, $y_1(0) = 2, y_2(0) = -10, y_3(0) = 1, and y_4(0) = -2$. One can verify that all agents are dissipative with the storage function $V_i(x) = \frac{1}{2}x_i^T(t)x_i(t)$. This results in output passivity indices $\rho_1 = 1.2, \rho_2 = 1.8, \rho_3 = 2.6, \rho_4 = 0.8$ for the agents. The Laplacian matrix of the underlying communication graph amongst agents before Byzantine attack is balances as follows,

$$
L = \begin{bmatrix}
1 & -1 & 0 & 0 \\
0 & 2 & -2 & 0 \\
0 & 0 & 2 & -2 \\
-1 & -1 & 0 & 2
\end{bmatrix}.
$$

with the connectivity measure, $\lambda(G) = 2$. We assume that the event-triggered multi-agent network
system is designed based on Theorem 3 with the following triggering conditions,

\[
\begin{align*}
||e_1(t)||_2^2 &> 0.80||y_1(t)||_2^2, \\
||e_2(t)||_2^2 &> 0.64||y_2(t)||_2^2, \\
||e_3(t)||_2^2 &> 0.60||y_3(t)||_2^2, \\
||e_4(t)||_2^2 &> 0.35||y_4(t)||_2^2,
\end{align*}
\]

by selecting \(\alpha_i = 1, \beta_i = 1\) for \(i = 1, \ldots, 4\). We assume \(G_2\) and \(G_4\) are Byzantine agents and instead of assigning correct weights \(a_2 = 1\) and \(a_4 = 2\) to agents \(G_1\) and \(G_3\), they assign \(a_2 + \omega_2\) and \(a_4 + \omega_4\) to agents \(G_1\) and \(G_3\) where \(\omega_2 = 2.5\) and \(\omega_4 = 2\). At time \(t = 0.4s\), in order to mitigate the attack, honest agents \(G_1\) and \(G_3\) increase \(\rho_1^{\Delta}\) and \(\rho_3^{\Delta}\), and their communication rate by shortening their triggering intervals. They alter their triggering parameters from \(\delta_1 = 0.80\) and \(\delta_3 = 0.60\) to \(\delta_1 = 0.40\) and \(\delta_3 = 0.15\). Fig. 13 shows that the system synchronizes as a consequence of this mitigation attempt. One can see clearly in Fig. 13 that due to the Byzantine attack the agents diverge at first and it is only after the honest agents mitigate the attack by following the steps given in Sub-Section IX-A that the multi-agent system takes some corrective steps and eventually synchronizes.

B. A Learning-Based Distributed Algorithm for Mitigating the Effects of Data Falsification

Identifying the Byzantine Agents: Here, we propose an algorithm based on which each agent \(G_j\) is able to identify each of its neighbors \(G_k\), where \(k \in \mathcal{N}_j^{in}\) as an honest or Byzantine neighbor. The identification of each neighboring agent is done in order to realize whether the
receiving information is trustworthy or not. This is necessary for the next step of the mitigation process. We already know that based on a sufficiently large number of detection data-points of length \( L \) coming from the neighbor agent \( G_k \), one can postulate that if the neighbor agent \( G_k \) is honest, then the data points will follow a normal distribution conditioned on the hypothesis \( H_i (i = 0, 1) \), namely \( f_{PDF}(T_k | H_i) = \mathcal{N}((\mu_{i0})_k, (\sigma_{i0}^2)_k) \) where \( i = 0, 1 \) and \( k \in \mathbb{N}_j \). The exact form of \( (\mu_{i0})_k \) and \( (\sigma_{i0}^2)_k \) are given in (20), (21), (22) and (23), and the test statistics \( T_k \) is defined in Sub-section VIII-A. \( f_{PDF}(T_k | H_i) \) is the probability density function (PDF) of test statistics under each hypothesis \( H_i, i = 0, 1 \). Similarly, if the neighbor agent \( G_k \) is Byzantine, then the data coming from this neighbor is a Gaussian mixture from \( \mathcal{N}((\mu_{i1})_k, (\sigma_{i1}^2)_k) \) with the probability of \( 1 - P_k \) and \( \mathcal{N}((\mu_{i1})_k, (\sigma_{i1}^2)_k) \) with the probability of \( P_k \), where \( i = 0, 1 \) and \( k \in \mathbb{N}_j \). As a result, the decision making process will be based on a hypothesis testing framework, where it is decided that the neighboring agent \( G_k \) is honest under the hypothesis \( H_i \), i.e. \( (Dec_i^k)_0 \), if the receiving data can be justified by the Gaussian distribution expected under the hypothesis \( H_i \). Otherwise, if a Gaussian mixture justifies the data points better, \( (Dec_i^k)_1 \) under the hypothesis \( i \) is decided. This means that it is decided that \( G_k \) is Byzantine, if the receiving data from \( G_k \)
follows the distribution of the expected Gaussian mixture under the hypothesis $H_i$. This decision may be made using the maximum likelihood decision rule \[47\],
\[
\frac{f_{PDF}(T_k|(Dec_{0i}))}{f_{PDF}(T_k|(Dec_{1i}))} \overset{H}{\geq} 1. 
\]
However, these distributions are unknown to the honest agents and the detection unit should learn these distributions’ respective parameters. Next, we will cover the estimation process based on the proposed framework.

Learning the Distributions’ Parameters: The parameters in \[37\] are unknown and should be estimated. The formulation given in \[37\] is parametric and consequently, the framework may be looked at as a parametric statistical estimation problem \[47\]. Accordingly, under the hypothesis $H_i (i = 0, 1)$, for the honest neighboring agents $G_k$ where $k \in N_j^{inH}$, the parameters to be estimated are $\theta_0 = ((\mu_{i0})_k, (\sigma^2_{i0})_k)$ and for the Byzantine neighboring agents $G_k$ where $k \in N_j^{inB}$, the parameters to be estimated are the ones in the set $\theta_0$ and $\theta_1 = ((\mu_{i1})_k, (\sigma^2_{i1})_k, P_k)$. Here, we propose a learning-based algorithm for estimating the parameter sets $\theta_0$ and $\theta_1$. We annotate the estimation of the means and variances as, $(\tilde{\mu}_{i0})_k$, $(\tilde{\mu}_{i1})_k$, $(\tilde{\sigma}^2_{i0})_k$, $(\tilde{\sigma}^2_{i1})_k$, and $\tilde{P}_k$ for $H_i (i = 0, 1)$ and $k \in N_j^{inH}$ or $k \in N_j^{inB}$. Let us assume a detection time-interval of length $L$ where, the time-interval includes $L$ triggering instances that may be utilized as the sample points for the estimation \[57\]. For example, between neighboring agents $G_j$ and $G_k$ at the discrete time-instance $i$, the test statistics attained by agent $G_j$ in regard to the neighbor $G_k$ becomes $t_k^i = \sum_{n=L-i}^L |y_{kn}^n - y_{jn}^n|^2$, and the $L$ number of samples $y_{kn}^n$ and $y_{jn}^n$ depend on the most recent outputs of the respective event-detectors at the time-instance $n$ during the detection interval $L$.

In order to estimate the parameters in the set $\theta_0$ for the case of honest agents under the hypotheses $H_0$ and $H_1$, we can simply utilize the method of moments \[47\]. For the honest neighbors, we know that the data should preferably follow a normal distribution with the means and variances given in \[20\], \[21\], \[22\] and \[23\]. We assume the learning iterations of length $L_p$ (each learning iteration consists of $L_p$ data points). Each learning iteration may consist of one or two sets of data belonging to the hypothesis $H_0$ and $H_1$, respectively. At the learning iteration $l$, we may have $T_k^{(l)} = [(t_k^{L_0})^0, (t_k^{L_1})^0, ..., (t_k^{L_0})^1, (t_k^{L_1})^1, ..., (t_k^{L_1})^1]$, where $L_0 + L_1 = L_p$. Given $L_i$ ($i = 0, 1$) data points, for a normal distribution and for the learning iteration $l$ with $L_p$ total number of data points, the first and second moments theoretically may be represented as,

$$(m_1)_k^i = \frac{1}{L_i} \sum_{j=1}^{L_i} t_k^i,$$
and,

\[(m_2)_k^l = \frac{1}{L_i} \sum_{j=1}^{L_i} t_k^j,\]

for \(H_i(i = 0, 1)\) and \(k \in \mathcal{N}_j^{inh}\). Consequently, \((m_1)_k^l = (\mu_{i0})_k\) is the estimator for the sample mean for \(H_i(i = 0, 1)\) and \(k \in \mathcal{N}_j^{inh}\) at the learning iteration \(l\). And for the variances, we have, \((\hat{\sigma}^2_{i0})_k + (\hat{\mu}_{i0})^2_k = (m_2)_k^l\) for \(H_i(i = 0, 1)\) with \(k \in \mathcal{N}_j^{inh}\). As a result, we may have,

\[(\hat{\sigma}^2_{i0})_k + \frac{1}{L_i} \sum_{j=1}^{L_i} t_k^j - (\hat{\mu}_{i0})_k^2 = \frac{1}{L_i} \sum_{j=1}^{L_i} (t_k^j - (\hat{\mu}_{i0})_k)^2,\]

for \(H_i(i = 0, 1)\) and \(k \in \mathcal{N}_j^{inh}\). To sum up, the learned parameter set \(\theta_0\) at the learning iteration \(l\) for an honest communication between two honest agents becomes,

\[\tilde{\theta}_0 = ((\hat{\mu}_{i0})_k, (\hat{\sigma}^2_{i0})_k) = \left(\frac{1}{L_i} \sum_{j=1}^{L_i} t_k^j, \frac{1}{L_i} \sum_{j=1}^{L_i} (t_k^j - (\hat{\mu}_{i0})_k)^2\right),\]

(38)

for \(H_i(i = 0, 1)\) and \(k \in \mathcal{N}_j^{inh}\).

Next, we define the complete learning process for an honest neighboring agent based on the above estimators. Each learning phase consists of \(L_p\) detection time-intervals of length \(L\) (where the parameters are estimated as above) or \(L_p \times L\) data points. One can recall from Sub-Section \(\text{V-A}\) that each agent is consistently deciding whether the system has reached synchronization \((\text{Decision}_{syn})\), i.e. \(H_0\) is the correct hypothesis, or otherwise, i.e. \(H_1\) is the correct hypothesis. As a result, each data point \(l_i\) after the detection time-interval of length \(L\) under the set of the \(L_p\) data points comes with an index indicating if the estimation is happening under the hypothesis \(H_0\) or \(H_1\). We annotated the number of these estimation as \(L_i\) given the hypothesis \(H_i\) \((i = 0, 1)\). Needless to say \(L_0 + L_1 = L_p\). As an application related side-note, one can set a required lower-bound for the number of data points under hypothesis \(H_i\) \((i = 0, 1)\) before the estimation (learning process) for the parameters under the hypothesis starts. For instance, one can require, \(L_1 \geq \tau_1\), before the learning process starts for the parameters under \(H_1\). For the learning process that starts at time \(l^t_i + 1\) respectively for \((i = 0, 1)\) and when the next \(L_p\) data points are available, we already have, \((\tilde{\mu}_{00})_k = [(\tilde{\mu}_{00})^0_k, \ldots, (\tilde{\mu}_{00})^L_k]\), \((\tilde{\sigma}^2_{00})_k = [(\tilde{\sigma}^2_{00})^0_k, \ldots, (\tilde{\sigma}^2_{00})^L_k]\) and similarly, \((\tilde{\mu}_{10})_k = [(\tilde{\mu}_{10})^0_k, \ldots, (\tilde{\mu}_{10})^L_k]\), \((\tilde{\sigma}^2_{10})_k = [(\tilde{\sigma}^2_{10})^0_k, \ldots, (\tilde{\sigma}^2_{10})^L_k]\). We can define our so-far estimates as \((\tilde{\mu}_{00})_k(l^0) = (\tilde{\sum}^g_{i=1}(\tilde{\mu}_{00})^i_k)/L, (\tilde{\mu}_{10})_k(l^1) = (\tilde{\sum}^t_{i=1}(\tilde{\mu}_{10})^i_k)/L\) as the (current) estimate for the means under each hypothesis and \((\tilde{\sigma}^2_{00})_k(l^0) = (\tilde{\sum}^g_{i=1}(\tilde{\sigma}^2_{00})^i_k)/L\) and \((\tilde{\sigma}^2_{10})_k(l^1) = (\tilde{\sum}^t_{i=1}(\tilde{\sigma}^2_{10})^i_k)/L\) as the current estimates for the variances under each hypothesis. These values also play the rule of the
initial points for the next learning iteration. As a result, in a recursive manner, the next learned values at \( l^i + 1 \) for the means may be determined as follows,

\[
(\hat{\mu}_{00})_k(l^0 + 1) = \frac{l^0}{l^0 + 1} (\hat{\mu}_{00})_k(l^0) + \frac{1}{(l^0 + 1)L_0} \sum_{i=1}^{L_0} (t^i)^0_k, \tag{39}
\]

\[
(\hat{\mu}_{10})_k(l^1 + 1) = \frac{l^1}{l^1 + 1} (\hat{\mu}_{10})_k(l^1) + \frac{1}{(l^1 + 1)L_1} \sum_{i=1}^{L_1} (t^i)^1_k, \tag{40}
\]

where \((t^i)^0_k\) and \((t^i)^1_k\) respectively represent the next set of test statistic data points received under the hypothesis \( H_0 \) or \( H_1 \) for the next learning interval \( l = 1, \ldots, L_p \) from agent \( G_k \). This recursive algorithm will require the estimation framework to only record the true values of the last \( l^0 \) or \( l^1 \) estimates respectively in a queue, and by calculating the new estimate at \( l^i + 1 \), the first element of the respective queue may be discarded and the new estimate may be added to the queue. This means that the learning process will make use of \( l^i \times L_p \times L \) data points while only storing \( l^i \) data points for each hypothesis \( H_i \) \((i = 0, 1)\), respectively. Similarly, the process for learning the variances becomes,

\[
(\hat{\sigma}^2_{00})_k(l^0 + 1) = \frac{l^0}{l^0 + 1} [(\hat{\sigma}^2_{00})_k(l^0)] + \frac{1}{(l^0 + 1)L_0} \sum_{i=1}^{L_0} ((t^i)^0_k - (\hat{\mu}_{00})_k(l^0 + 1))^2, \tag{41}
\]

\[
(\hat{\sigma}^2_{10})_k(l^1 + 1) = \frac{l^1}{l^1 + 1} [(\hat{\sigma}^2_{10})_k(l^1)] + \frac{1}{(l^1 + 1)L_1} \sum_{i=1}^{L_1} ((t^i)^1_k - (\hat{\mu}_{10})_k(l^1 + 1))^2, \tag{42}
\]

where \((t^i)^0_k\) and \((t^i)^1_k\) respectively represent the next set of test statistic data points received under the hypothesis \( H_0 \) or \( H_1 \) for the next learning interval \( l = 1, \ldots, L_p \) from agent \( G_k \). Lastly, as mentioned, as a design matter, one can put performance criteria such as \( L_0 > \tau_1 \) and \( L_1 > \tau_2 \) as quantities to be met first before the learning process for each of parameter sets under each hypothesis starts in order to make sure that the learning data-set is large enough for a more precise estimation. This two-level estimation process, will achieve a very good learning-based estimates while maintaining low memory requirements, as only the values of the last \( l^i \) estimates (containing the information for \( L \times L_p \) data points) and their respective hypothesis keys are required to be memorized in a feedback, recursive manner.

For Byzantine agents, we take another common approach, previously utilized in control literature \cite{58}, \cite{59}, called maximum likelihood method (MLE) of parameter estimation. This is due to the fact that additional to the means and variances of the Byzantine data, one needs to also estimate the latent variable \( P_k \) \((k \in \mathcal{N}_j^{m \alpha})\) or the probability of the attack. MLE, developed
by Fisher [60], has many desirable theoretical properties, such as consistency, efficiency and unbiasedness under certain conditions [47]. The likelihood is the joint probability of a set of observations, conditioned on a choice for the parameters \( \tilde{\theta}_1 \), \( L i k(\tilde{\theta}_1, y) = P(y|\tilde{\theta}_1) \), where \( y \) represents the data sample points, \( \tilde{\theta}_1 \) is the set of parameters to be estimated, and \( P \) is the probability distribution. According to this relation, the parameter set \( (\tilde{\theta}_1^{MLE}) \) that maximizes the likelihood of the observed data gives the best estimator. This value is called the maximum likelihood estimate (MLE),

\[
\tilde{\theta}_1^{MLE} = \arg\max_{\tilde{\theta}_1} L i k(\tilde{\theta}_1, y).
\]

Each learning phase \( t \) consists of \( L_p \) data-points. We denote the test statistic between the agent \( G_j \) and the Byzantine neighbor \( G_k \) during the learning phase of length \( L_p \) as, \( T_k^{(t)} = [(t_1^0)_k, (t_2^0)_k, \ldots, (t_{L_0}^0)_k, (t_1^1)_k, (t_2^1)_k, \ldots, (t_{L_1}^1)_k] \). Similar to before \( L_0 + L_1 = L_p \). Additionally, similar to before, one can start the learning process for each set of parameters under each hypothesis \( H_i \) \( (i = 0, 1) \), once \( L_0 > \tau_1 \) and \( L_1 > \tau_2 \). We utilize the estimates resulting from each learning phase as initial values for the next round of estimations. Additionally, we utilize the Expectation-Maximization (EM) algorithm for the learning process. The Expectation-Maximization (EM) algorithm is an iterative learning-based method for finding \( \tilde{\theta}_1^{MLE} \) [47]. The EM algorithm alternates between two steps, an expectation step which calculates the expectation of the log-likelihood given the current estimates for the parameters, and a maximization step which computes the parameters that maximize the expected log-likelihood found in the first step—This step involves derivations with respect to unknown parameters (means and variances) and substitutions for the latent parameter set \( Z \). These new values then initialize the next expectation step. We annotate the latent parameters as \( Z = [z_0, z_1] \) (in our case, the latent variables represent the attack probabilities for the Byzantine agent \( G_k \), i.e. \( \tilde{P}_k \) and \( 1 - \tilde{P}_k \)). For Byzantine neighboring agent \( G_k \), we have,

\[
L i k(\tilde{\theta}_1, y) = L i k((\tilde{\mu}_{ij})_k, (\tilde{\sigma}^2_{ij})_k; T_k, Z) = p(T_k, Z|(\tilde{\mu}_{ij})_k, (\tilde{\sigma}^2_{ij})_k),
\]

where \( T_k \) represents our data points, \( p(.) \) is the joint PDF of the data points and latent variables conditioned on the parameters and \( j = 0, 1 \). Further, we can describe the above based on a marginal and a conditional distribution, this gives us,

\[
p(T_k, Z|(\tilde{\mu}_{ij})_k, (\tilde{\sigma}^2_{ij})_k) = p(z_j|T_k, (\tilde{\mu}_{ij})_k, (\tilde{\sigma}^2_{ij})_k)p(T_k|(\tilde{\mu}_{ij})_k, (\tilde{\sigma}^2_{ij})_k) = \pi_k p(T_k|(\tilde{\mu}_{ij})_k, (\tilde{\sigma}^2_{ij})_k),
\]

(43)
where \(0 \leq \pi_k^j \leq 1\) represents the distribution for the latent variable which is the probability of attack for the Byzantine agent \(G_k\). Also, \(j = 0, 1\) and \(\pi_k^0 + \pi_k^1 = 1\). To sum all this up, for estimating the Byzantine parameters of agent \(G_k\), and in order to describe the above relationship based on single data points, we expand (43) to have,

\[
Lik((\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k; T_k, Z) = p(T_k, Z| (\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k) = \prod_{n=1}^{L_p} \prod_{j=0}^{1} \pi_k^j p(t^n_k)^j|((\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k).
\]

(44)

Under the EM algorithm, for the first step, the expectation is calculated based on the log-likelihood function of the distributions. Given (44), the expectation step (Q-function) based on the current estimate set \(\tilde{\theta}_1^{(l)}\), for agent \(G_k\) under the hypotheses \(H_i\) \((i = 0, 1)\), becomes,

\[
Q(\tilde{\theta}_1|\tilde{\theta}_1^{(l)}) = E_{z|T_k, \tilde{\theta}_1^{(l)}}[\log Lik((\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k; T_k, Z)]
\]

\[
= E_{z|T_k, \tilde{\theta}_1^{(l)}}[\log p(T_k, Z| \tilde{\theta}_1)|T_k, \tilde{\theta}_1^{(l)}]
\]

\[
= \sum_{n=1}^{L_o} \log \left( \prod_{j=0}^{1} \pi_k^j p((t^n_k)^j|((\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k)\right) \right)
\]

\[
+ \sum_{n=1}^{L_1} \log \left( \prod_{j=0}^{1} \pi_k^j p((t^n_k)^j|((\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k)\right) \right), \]

(45)

where \(\tilde{\theta}_1^{(l)} = ((\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k, (\tilde{\sigma}_{ij}^2)_k, \pi_k^j)\) for \(j = 0, 1\), are the current estimates for the neighboring Byzantine neighbor \(G_k\). It is also well-known in the literature that given the current estimate \(\theta_1^{(l)}\), the conditional distribution of \(Z\), i.e. \(p(z_j|(t^n_k)_r, (\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k)\), under hypothesis \(H_r\), \((r = 0, 1)\) respectively, for each summation in (45), is determined by Bayes’ Theorem as,

\[
p(z_j|(t^n_k)_r, (\mu_{ij})_k, (\sigma_{ij}^2)_k) = \frac{\pi_k^j p((t^n_k)_r|((\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k)\right) \right)}{\sum_{j=0}^{1} \pi_k^j p((t^n_k)_r|((\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k)\right) \right)}.
\]

(46)

For the maximization step, we should maximize \(Q(\tilde{\theta}_1|\tilde{\theta}_1^{(l)})\), by taking derivatives with respect to the parameters, i.e. \(\tilde{\theta}_1^{(l+1)} = \arg \max_{\tilde{\theta}_1} Q(\tilde{\theta}_1|\tilde{\theta}_1^{(l)})\). Simplifying further, and utilizing Jensen’s inequality and given the fact that log likelihood is a concave function \(47\), we have,

\[
\tilde{\theta}_1^{(l+1)} = \arg \max_{\tilde{\theta}_1} Q(\tilde{\theta}_1|\tilde{\theta}_1^{(l)})
\]

\[
\equiv \arg \max_{\tilde{\theta}_1} \sum_{n=1}^{L_0} \sum_{j=0}^{1} [p(z_j|(t^n_k)_0, (\mu_{ij})_k, (\sigma_{ij}^2)_k)\right) \right)(\log \pi_k^j - \frac{(t^n_k)_0 - (\tilde{\mu}_{ij})_k)^2}{2(\tilde{\sigma}_{ij}^2)_k} - \frac{\log(\pi_k^j)}{2}]
\]

\[
+ \sum_{n=1}^{L_1} \sum_{j=0}^{1} [p(z_j|(t^n_k)_1, (\tilde{\mu}_{ij})_k, (\tilde{\sigma}_{ij}^2)_k)\right) \right)(\log \pi_k^j - \frac{(t^n_k)_1 - (\tilde{\mu}_{ij})_k)^2}{2(\tilde{\sigma}_{ij}^2)_k} - \frac{\log(\pi_k^j)}{2}]].
\]

(47)
This should be done subject to the constraint that $\sum_{j=0}^{1} \pi_k^j = 1$ for the Byzantine agent $G_k$. Similar to common approaches in literature in regard to EM-based estimation of Gaussian mixtures [61], we utilize a Lagrangian multiplier for maximization, hence we have,

$$\max J = Q(\tilde{\theta}_1, \tilde{\theta}_1^{(l)}) + \lambda \left( \sum_{j=0}^{1} \pi_k^j - 1 \right)$$

$$\equiv \max \left[ \sum_{n=1}^{L_0} \sum_{j=0}^{1} \left[ p(z_j | (t^n_k)_0^0, (\tilde{\mu}_{0j})_k^{(l)}), (\tilde{\sigma}_{0j}^2)_k^{(l)} \right] (\log \pi_k^j) - \frac{((t^n_k)_0^0 - (\tilde{\mu}_{0j})_k^{(l)})^2}{2(\tilde{\sigma}_{0j}^2)_k^{(l)}} - \frac{\log(\tilde{\sigma}_{0j}^2)_k^{(l)}}{2} \right]$$

$$+ \sum_{n=1}^{L_1} \sum_{j=0}^{1} \left[ p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}), (\tilde{\sigma}_{1j}^2)_k^{(l)} \right] (\log \pi_k^j) - \frac{((t^n_k)_1^1 - (\tilde{\mu}_{1j})_k^{(l)})^2}{2(\tilde{\sigma}_{1j}^2)_k^{(l)}} - \frac{\log(\tilde{\sigma}_{1j}^2)_k^{(l)}}{2} \right]$$

$$+ \lambda \left( \sum_{j=0}^{1} \pi_k^j - 1 \right) \right]. \quad (48)$$

In order to maximize the above, one should solve for the equations resulting from the derivative of each parameter by equating them with zero. As an example, we have (for $j = 0, 1$),

$$\frac{d}{d\pi_k^j} J = \lambda + \sum_{n=1}^{L_0} p(z_j | (t^n_k)_0^0, (\tilde{\mu}_{0j})_k^{(l)}, (\tilde{\sigma}_{0j}^2)_k^{(l)}) \pi_k^j + \sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)}) \pi_k^j = 0, \quad (49)$$

which gives us,

$$-\pi_k^j \lambda = \sum_{n=1}^{L_0} p(z_j | (t^n_k)_0^0, (\tilde{\mu}_{0j})_k^{(l)}, (\tilde{\sigma}_{0j}^2)_k^{(l)}) + \sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)}), \quad (50)$$

or $-\pi_k^j \lambda = (L_0 + L_1) \pi_k^j$. And we have, $\lambda = -L_p$. In a similar manner, we can take derivatives and simplify further to find the following recursive estimations for the Byzantine parameters for agent $G_k$,

$$(\pi_k^j)^{(l+1)} = \frac{\sum_{n=1}^{L_0} p(z_j | (t^n_k)_0^0, (\tilde{\mu}_{0j})_k^{(l)}, (\tilde{\sigma}_{0j}^2)_k^{(l)}) + \sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)})}{L_p}, \quad (51)$$

$$(\tilde{\mu}_{0j})_k^{(l+1)} = \frac{\sum_{n=1}^{L_0} p(z_j | (t^n_k)_0^0, (\tilde{\mu}_{0j})_k^{(l)}, (\tilde{\sigma}_{0j}^2)_k^{(l)}) (t^n_k)_0^0 + \sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)}) (t^n_k)_1^1}{\sum_{n=1}^{L_0} p(z_j | (t^n_k)_0^0, (\tilde{\mu}_{0j})_k^{(l)}, (\tilde{\sigma}_{0j}^2)_k^{(l)})}, \quad (52)$$

$$(\tilde{\sigma}_{0j}^2)_k^{(l+1)} = \frac{\sum_{n=1}^{L_0} p(z_j | (t^n_k)_0^0, (\tilde{\mu}_{0j})_k^{(l)}, (\tilde{\sigma}_{0j}^2)_k^{(l)}) ((t^n_k)_0^0 - (\tilde{\mu}_{0j})_k^{(l)})^2 + \sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)}) ((t^n_k)_1^1 - (\tilde{\mu}_{1j})_k^{(l)})^2}{\sum_{n=1}^{L_0} p(z_j | (t^n_k)_0^0, (\tilde{\mu}_{0j})_k^{(l)}, (\tilde{\sigma}_{0j}^2)_k^{(l)})}, \quad (53)$$

$$(\tilde{\mu}_{1j})_k^{(l+1)} = \frac{\sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)}) (t^n_k)_1^1 + \sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)}) (t^n_k)_1^1}{\sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)})}, \quad (54)$$

$$(\tilde{\sigma}_{1j}^2)_k^{(l+1)} = \frac{\sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)}) ((t^n_k)_1^1 - (\tilde{\mu}_{1j})_k^{(l)})^2 + \sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)}) ((t^n_k)_1^1 - (\tilde{\mu}_{1j})_k^{(l)})^2}{\sum_{n=1}^{L_1} p(z_j | (t^n_k)_1^1, (\tilde{\mu}_{1j})_k^{(l)}, (\tilde{\sigma}_{1j}^2)_k^{(l)})}. \quad (55)$$
Similar to the previous algorithm, one can design independent performance criteria such as $L_0 > \tau_1$ and $L_1 > \tau_2$ as quantities to be met first before the learning process for each of the parameter sets starts. This is to make sure that the learning data-set is large enough for a precise estimation. The algorithm may be solved recursively. At discrete time instance $l$ when enough information is received the recursive learning algorithm starts and the estimates at the end of each expectation-maximization run may be used as initial values for the next learning iteration $l + 1$ that uses a new set of $L_p$ data points. After the learning process has ended, the honest agent may classify its neighbor $G_k$ as a Byzantine or honest agent following the likelihood-based hypothesis testing,

$$\frac{\tilde{f}_{PDF}(T_k|Dec_k^0)}{\tilde{f}_{PDF}(T_k|Dec_k^1)} \geq_B 1,$$

where $\tilde{f}_{PDF}(\cdot)$ represents the probability distribution function attained based on the best estimates of the parameters available to the agent.

**Mitigating the Effects of Data Falsification:** Once the Byzantine agents are identified based on the above algorithm. One can utilize this information to mitigate the effects of the attack. Unlike most approaches in the literature that rely on excluding the Byzantine agents, we utilize the Byzantine information against rogue agents in order to benefit the entire event-triggered multi-agent system. As the first step, for the decision making step, we define a new local summary statistic based on the information received from only the honest agents, i.e. $(T^*_j)^H = \sum_{k \in N^H_j} T^j_k$. Similar to before, each agent will make its own decision on the synchronization hypothesis using the predefined threshold $\gamma_j^H$,

$$Decision_{syn} = \begin{cases} H_0 & \text{if } (T^*_j)^H < \gamma_j^H \\ H_1 & \text{otherwise.} \end{cases}$$

Under the hypothesis $H_0$, at the learning iteration $l + 1$, the honest agent would estimate the means $((\tilde{\mu}_{00})^{(l+1)}_k$ and $(\tilde{\mu}_{01})^{(l+1)}_k$) based on the received information from the Byzantine neighbor $G_k$ or estimate only $(\tilde{\mu}_{00})^{(l+1)}_k$ based on the received information from the honest neighbor $G_k$ according to the algorithm given in the previous sub-section. These estimates follow the form given in [20]. As a result, the honest agent may closely estimate the attack parameter $\tilde{\Delta}_{l+1}^{k}$ after each learning iteration for the Byzantine neighbor $G_k$ as follows,

$$\tilde{\Delta}_{l+1}^{k} \approx \sqrt{\frac{(\tilde{\mu}_{01})^{(l+1)}_k - (\tilde{\mu}_{00})^{(l+1)}_k}{L \tilde{\sigma}^2_k \tilde{h}^2_k}}.$$ (56)
In a similar manner, under the hypothesis $H_1$, an analogous estimation process may be undertaken by the honest agent. Here, it is important to note that,

$$
(\tilde{\mu}_{11})^{(l+1)} - (\tilde{\mu}_{10})^{(l+1)} = L\tilde{h}_k^2\Delta_k^2 + 2L\tilde{h}_k\Delta_k(\mu_j - \tilde{\mu}_k),
$$

(57)

where $\mu_j = \frac{1}{L}\sum_{i=1}^{L} y_i^j$. Since $\mu_k$ is not available to us, it is prudent to use the estimation $\tilde{\mu}_k = \frac{1}{L}\sum_{i=1}^{L} \mu_i^k$ where $\mu_i^k = \frac{1}{|N^m|}\sum_{k\in N^m} y_k^i$ at time-instance $i$ during the detection interval of length $L$. $\tilde{\mu}_k$ can provide us with a good initial value as it represents the general state the entire multi-agent system is at. Later, at the learning iteration $l + 1$, one can replace $\mu_k$ with $\frac{1}{L}\sum_{i=1}^{L}(\tilde{y}_k^i - \tilde{\Delta}_k^i)$. Based on (57), we also have,

$$
\tilde{h}_k\Delta_k^2 + 2\Delta_k(\mu_j - \tilde{\mu}_k) = (\sqrt{\tilde{h}_k}\Delta_k + \frac{(\mu_j - \tilde{\mu}_k)}{\sqrt{\tilde{h}_k}})^2 - \frac{(\mu_j - \tilde{\mu}_k)^2}{\tilde{h}_k}.
$$

(58)

Finally, by utilizing (57) and (58), we have,

$$
\tilde{\Delta}_{k+1}^l \approx \frac{1}{\tilde{h}_k} \sqrt{\frac{1}{L} [(\tilde{\mu}_{11})^{(l+1)} - (\tilde{\mu}_{10})^{(l+1)}] + \frac{(\mu_j - \tilde{\mu}_k)^2}{\tilde{h}_k} - \frac{(\mu_j - \tilde{\mu}_k)}{\sqrt{\tilde{h}_k}}}. 
$$

(59)

In the above relations, we have assumed that $\tilde{\sigma}_k$ and $\tilde{h}_k$ for the communication links are available to the agents. As mentioned before, these assumptions are justified by the fact that each detection unit can perform simple noise power estimation and channel gain estimation (by averaging the signal-to-noise ratio over a certain time interval) between consecutive sensing intervals to accurately obtain these values. Finally, instead of excluding the Byzantine agent $G_k$ in the process of mitigating the attack, one can utilize the false information after the estimation and mitigate the negative adversarial effects after the learning iteration $l + 1$ by replacing the Byzantine agent’s output information with $\tilde{y}_k = y_k + \tilde{\Delta}_{k+1}^l$ under the hypothesis $H_i$ ($i = 0, 1$), respectively. Next, we will demonstrate our approach with an example.

**Example 5.** We consider a multi-agent event-triggered network system consisting of four agents $(i = 1, ..., 4)$ with the underlying balanced communication topology given in Fig. 14. We assume the following dynamics for agents,

$$
G_i = \begin{cases} 
\dot{x}_i(t) = -c_i x_i(t) + u_i(t) \\
y_i(t) = x_i(t),
\end{cases}
$$

with $c_1 = 1.2, c_2 = 2.2, c_3 = 2.4, c_4 = 0.60$ and initial conditions, $y_1(0) = 5, y_2(0) = 10, y_3(0) = 4, y_4(0) = 1$. One can verify that all agents are dissipative with the storage function $V_i(x) = \frac{1}{2}x_i^T(t)x_i(t)$. This results in output passivity indices $\rho_1 = 1.2, \rho_2 = 2.2, \rho_3 = 2.4, \rho_4 = 0.6$. 
The Laplacian matrix of the underlying communication graph amongst agents before Byzantine attack is balanced as follows,

\[
L = \begin{bmatrix}
1 & 0 & -1 & 0 \\
0 & 1 & -1 & 0 \\
-1 & 0 & 2 & -1 \\
0 & -1 & 0 & 1 \\
\end{bmatrix},
\]

with the connectivity measure, \( \lambda(G) = 2 \). We assume that the multi-agent network system is designed based on Theorem 3 with the following triggering conditions,

\[
||e_1(t)||_2^2 > 0.21||y_1(t)||_2^2, \\
||e_2(t)||_2^2 > 0.14||y_2(t)||_2^2, \\
||e_3(t)||_2^2 > 0.20||y_3(t)||_2^2, \\
||e_4(t)||_2^2 > 0.45||y_4(t)||_2^2.
\]

An additive Gaussian noise with zero mean and variance \( \sigma_k^2 = 1.22 (N(0, \sigma_k^2)) \) is assumed in the communication links. The channel gains are \( \tilde{h}_k = 1 \) for \( k = 1, ..., 4 \). We assume \( G_1 \) is a Byzantine agent. Under the hypothesis \( H_0 \), at time \( t = 3s \), \( G_1 \) attacks the network with the attack parameters \( P_1 = 0.70 \) and \( \Delta_1 = 8 \). One can see that the behavior of the multi-agent system drastically deviates from its desired synchronized behavior as a result of the attack (Fig. 15 and Fig. 16). The honest agent \( G_3 \) detects the Byzantine agent and starts the process of learning the Byzantine agent’s behavior using the proposed mitigation algorithm. The learning parameters are \( L = 12 \) with 20 learning iterations \( (l = 20) \) of length 20 \( (L_p = 20) \) which takes advantage of an overall of 400 data points. At time \( t = 6s \), the honest agent \( G_3 \) estimates the attack parameters as \( \tilde{P}_1 = 0.68 \) (Fig. 17) and \( \tilde{\Delta}_1 = 6.35 \) using the proposed algorithm given in the previous sub-section and the relation given below,

\[
\tilde{\Delta}_{l+1} \approx \frac{1}{\sqrt{h_1}} \left( \frac{(\mu_1)(l+1) - (\tilde{\mu}_1)(l+1)}{Lh_1} + \frac{(\mu_3 - \tilde{\mu}_1)^2}{\tilde{\mu}_1} - \frac{(\mu_3 - \tilde{\mu}_1)}{\sqrt{h_1}} \right).
\]

The estimations at each learning iteration are given in Fig. 17, Fig. 18 and Fig. 19. The mitigation process starts at time \( t = 8 \) where the information received from agent \( G_1 \) by agent \( G_3 \) is replaced with \( \tilde{y}_1 = y_1 - \tilde{\Delta}_1^{l+1} \). One can see the positive effects of this mitigation process in Fig. 15 and Fig. 16 toward the end of the experiment where the multi-agent system enhances its performance and reaches synchronization again.
Fig. 14. The Underlying Communication Graph for the Multi-Agent System Presented in Example 5.

Fig. 15. The Outputs of the Multi-Agent System in the Presence of the Byzantine Attack (Attack Parameters: $P_1 = 0.70$ and $\Delta_1 = 8$).
Fig. 16. The Deviations of the Outputs from the Synchronized State of the Multi-Agent System in the Presence of the Byzantine Attack (Attack Parameters: $P_1 = 0.70$ and $\Delta_1 = 8$).

Fig. 17. The Estimated Probability of Attack Using the Proposed Algorithm.
Fig. 18. The Estimated Mean $(\tilde{\mu}_{00})^{(t+1)}_1$, under $H_0$ Using the Proposed Algorithm.

Fig. 19. The Estimated Mean $(\tilde{\mu}_{01})^{(t+1)}_1$, under $H_1$ Using the Proposed Algorithm.
X. Concluding Remarks

The work presented in this paper may be divided into two parts. The first part consists of a comprehensive event-triggered control design proposal that can guarantee synchronization for a network of multi-agent systems based on their passivity properties. This proposed control design is capable of reducing the communication load amongst sub-agents while maintaining synchronization and desired performance criteria. Under this part of the work, we also show the lack of Zeno behavior for the event-triggered conditions. The second part of our work concerns security. Under this section, we introduced a general powerful model for Byzantine attacks containing both data falsification and weight manipulation. Additionally, we introduced a detection framework, through which, the honest agents will attempt to detect and mitigate the effects of the attack. We gave a full performance analysis of the detection unit based on both transient and steady-state characteristics of the framework. Lastly, we presented two learning-based mitigation processes. The first one was based on the passivity properties of the agents and intended to mitigate the negative effects of weight manipulation. The second proposed learning-based control framework dealt with the problem of data falsification. Under this framework, the honest agents attempt to estimate their neighbor’s states and consequently learn the attack parameters for Byzantine neighbors. After learning these parameters then the honest agents utilize this information to eradicate the negative effects of adversarial attempts and enhance the performance and synchronization of the entire event-triggered multi-agent network system.
REFERENCES

[1] M. Schneider-Fontan and M. J. Mataric, “Territorial Multi-Robot Task Division,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 815–822, 1998.

[2] J. A. Fax and R. M. Murray, “Information Flow and Cooperative Control of Vehicle Formations,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[3] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, “Novel Type of Phase Transition in a System of Self-Driven Particles,” *Physical Review Letters*, vol. 75, no. 6, p. 1226, 1995.

[4] R. W. Beard, J. Lawton, and F. Y. Hadaegh, “A Coordination Architecture for Spacecraft Formation Control,” *IEEE Transactions on Control Systems Technology*, vol. 9, no. 6, pp. 777–790, 2001.

[5] J. Xiang, Y. Li, and D. J. Hill, “Cooperative Output Regulation of Linear Multi-Agent Network Systems with Dynamic Edges,” *Automatica*, vol. 77, pp. 1–13, 2017.

[6] S. Y. Shafi and M. Arcak, “Adaptive Synchronization of Diffusively Coupled Systems,” *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 131–141, 2015.

[7] J. Wang, K. Chen, and F. L. Lewis, “Coordination of Multi-Agent Systems on Interacting Physical and Communication Topologies,” *Systems & Control Letters*, vol. 100, pp. 56–65, 2017.

[8] W. Ni and D. Cheng, “Leader-Following Consensus of Multi-Agent Systems under Fixed and Switching Topologies,” *Systems & Control Letters*, vol. 59, no. 3, pp. 209–217, 2010.

[9] Y. Zheng, Y. Zhu, and L. Wang, “Consensus of Heterogeneous Multi-Agent Systems,” *IET Control Theory & Applications*, vol. 5, no. 16, pp. 1881–1888, 2011.

[10] J. R. Klotz, Z. Kan, J. M. Shea, E. L. Pasiliao, and W. E. Dixon, “Asymptotic Synchronization of a Leader-Follower Network of Uncertain Euler-Lagrange Systems,” *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 174–182, 2015.

[11] T. Liu, D. J. Hill, and J. Zhao, “Output Synchronization of Dynamical Networks with Incrementally-Dissipative Nodes and Switching Topology,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 9, pp. 2312–2323, 2015.

[12] H. Yu and P. J. Antsaklis, “Output Synchronization of Networked Passive Systems with Event-Driven Communication,” *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 750–756, 2014.

[13] L. Scardovi, M. Arcak, and E. D. Sontag, “Synchronization of Interconnected Systems with Applications to Biochemical Networks: An Input-Output Approach,” *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1367–1379, 2010.

[14] B. Wang, J. Wang, L. Zhang, B. Zhang, and X. Li, “Cooperative Control of Heterogeneous Uncertain Dynamical Networks: An Adaptive Explicit Synchronization Framework,” *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1484–1495, 2017.

[15] X. Liu and S. Li, “Cluster Synchronization for Linearly Coupled Nonidentical Systems with Delays via Aperiodically Intermittent Pinning Control,” *IEEE Access*, vol. 5, pp. 4179–4189, 2017.

[16] T. Liu, D. J. Hill, and J. Zhao, “Synchronization of Dynamical Networks by Network Control,” *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1574–1580, 2012.

[17] X. Liu, K. Zhang, and W.-C. Xie, “Consensus Seeking in Multi-Agent Systems via Hybrid Protocols with Impulse Delays,” *Nonlinear Analysis: Hybrid Systems*, vol. 25, pp. 99–108, 2017.

[18] W. Zhang, Y. Tang, Q. Miao, and J.-A. Fang, “Synchronization of Stochastic Dynamical Networks under Impulsive Control with Time Delays,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1758–1768, 2014.

[19] R. Lu, W. Yu, J. Lü, and A. Xue, “Synchronization on Complex Networks of Networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2110–2118, 2014.
[20] Z. Chen, “Pattern Synchronization of Nonlinear Heterogeneous Multiagent Networks with Jointly Connected Topologies,” IEEE Transactions on Control of Network Systems, vol. 1, no. 4, pp. 349–359, 2014.
[21] S. Su, Z. Lin, and A. Garcia, “Distributed Synchronization Control of Multiagent Systems with Unknown Nonlinearities,” IEEE Transactions on Cybernetics, vol. 46, no. 1, pp. 325–338, 2016.
[22] W. Lu, Y. Han, and T. Chen, “Synchronization in Networks of Linearly Coupled Dynamical Systems via Event-Triggered Diffusions,” IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 12, pp. 3060–3069, 2015.
[23] G. Wen, M. Z. Chen, and X. Yu, “Event-Triggered Master-Slave Synchronization with Sampled-Data Communication,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 63, no. 3, pp. 304–308, 2016.
[24] X. Niu, Y. Liu, and Y. Man, “Adaptive Leader-Following Consensus for Uncertain Nonlinear Multi-Agent Systems,” Asian Journal of Control, vol. 19, no. 3, pp. 1189–1196, 2017.
[25] L. Lamport, R. Shostak, and M. Pease, “The Byzantine Generals Problem,” ACM Transactions on Programming Languages and Systems, vol. 4, no. 3, pp. 382–401, 1982.
[26] L. Zhang, G. Ding, Q. Wu, Y. Zou, Z. Han, and J. Wang, “Byzantine Attack and Defense in Cognitive Radio Networks: A Survey,” IEEE Communications Surveys & Tutorials, vol. 17, no. 3, pp. 1342–1363, 2015.
[27] A. G. Fragkiadakis, E. Z. Tragos, and I. G. Askoxyalakis, “A Survey on Security Threats and Detection Techniques in Cognitive Radio Networks,” IEEE communications surveys and tutorials, vol. 15, no. 1, pp. 428–445, 2013.
[28] A. Vempaty, K. Agrawal, P. Varshney, and H. Chen, “Adaptive Learning of Byzantines’ Behavior in Cooperative Spectrum Sensing,” in Wireless Communications and Networking Conference. IEEE, 2011, pp. 1310–1315.
[29] A. S. Rawat, P. Anand, H. Chen, and P. K. Varshney, “Collaborative Spectrum Sensing in the Presence of Byzantine Attacks in Cognitive Radio Networks,” IEEE Transactions on Signal Processing, vol. 59, no. 2, pp. 774–786, 2011.
[30] S. Marano, V. Matta, and L. Tong, “Distributed Inference in the Presence of Byzantine Sensors,” in Asilomar Conference on Signals, Systems and Computers. IEEE, 2006, pp. 281–284.
[31] M. Abdelhakim, L. E. Lightfoot, and T. Li, “Reliable Data Fusion in Wireless Sensor Networks under Byzantine Attacks,” in Military Communications Conference. IEEE, 2011, pp. 810–815.
[32] R. Chen, J.-M. Park, and K. Bian, “Robust Distributed Spectrum Sensing in Cognitive Radio Networks,” in The IEEE Conference on Computer Communications. IEEE, 2008, pp. 1876–1884.
[33] S. Marano, V. Matta, and L. Tong, “Distributed Detection in the Presence of Byzantine Attacks,” IEEE Transactions on Signal Processing, vol. 57, no. 1, pp. 16–29, 2009.
[34] F. R. Yu, H. Tang, M. Huang, Z. Li, and P. C. Mason, “Defense Against Spectrum Sensing Data Falsification Attacks in Mobile ad hoc Networks with Cognitive Radios,” in IEEE Military Communications Conference. IEEE, 2009, pp. 1–7.
[35] S. Liu, H. Zhu, S. Li, X. Li, C. Chen, and X. Guan, “An Adaptive Deviation-Tolerant Secure Scheme for Distributed Cooperative Spectrum Sensing,” in IEEE Global Communications Conference. IEEE, 2012, pp. 603–608.
[36] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. T. Hou, “Vulnerability and Protection for Distributed Consensus-Based Spectrum Sensing in Cognitive Radio Networks,” in IEEE Proceedings. IEEE, 2012, pp. 900–908.
[37] X. Wang and M. D. Lemmon, “Event-Triggering in Distributed Networked Control Systems,” IEEE Transactions on Automatic Control, vol. 56, no. 3, pp. 586–601, 2011.
[38] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, “Distributed Event-Triggered Control for Multi-Agent Systems,” IEEE Transactions on Automatic Control, vol. 57, no. 5, pp. 1291–1297, 2012.
[39] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, “Event-Based Broadcasting for Multi-Agent Average Consensus,” Automatica, vol. 49, no. 1, pp. 245–252, 2013.
[40] J. C. Willems, “Dissipative Dynamical Systems Part I: General Theory,” Archive for Rational Mechanics and Analysis, vol. 45, no. 5, pp. 321–351, 1972.
[41] J. Bao and P. L. Lee, *Process Control: the Passive Systems Approach*. Springer Science & Business Media, 2007.

[42] H. K. Khalil, *Nonlinear Systems*. Pearson, 3rd edition, 2002, vol. 9.

[43] C. Godsil and G. F. Royle, *Algebraic Graph Theory*. Springer Science & Business Media, 2013, vol. 207.

[44] C. W. Wu, “Algebraic Connectivity of Directed Graphs,” *Linear and Multilinear Algebra*, vol. 53, no. 3, pp. 203–223, 2005.

[45] F. F. Digham, M.-S. Alouini, and M. K. Simon, “On the Energy Detection of Unknown Signals over Fading Channels,” *IEEE Transactions on Communications*, vol. 55, no. 1, pp. 21–24, 2007.

[46] H. Urkowitz, “Energy Detection of Unknown Deterministic Signals,” *Proceedings of the IEEE*, vol. 55, no. 4, pp. 523–531, 1967.

[47] P. J. Bickel and K. A. Doksum, *Mathematical Statistics: basic ideas and selected topics*. CRC Press, 2015, vol. 2.

[48] B. Shen, S. Ullah, and K. Kwak, “Deflection Coefficient Maximization Criterion based Optimal Cooperative Spectrum Sensing,” *AEU-International Journal of Electronics and Communications*, vol. 64, no. 9, pp. 819–827, 2010.

[49] F. Visser, G. J. Janssen, and P. Pawelczak, “Multinode Spectrum Sensing based on Energy Detection for Dynamic Spectrum Access,” in *IEEE Vehicular Technology Conference*. IEEE, 2008, pp. 1394–1398.

[50] I. H. Arka, M. Ismail, and A. A. El-Saleh, “Selective Weight Setting Algorithm in Cognitive Radio Network under Resource Limitation,” in *IEEE International Conference on Space Science and Communication*. IEEE, 2013, pp. 313–317.

[51] W. Zhang, Z. Wang, Y. Guo, H. Liu, Y. Chen, and J. Mitola III, “Distributed Cooperative Spectrum Sensing based on Weighted Average Consensus,” in *IEEE Global Telecommunications Conference*. IEEE, 2011, pp. 1–6.

[52] D. Dolev, “The Byzantine Generals Strike Again,” *Journal of algorithms*, vol. 3, no. 1, pp. 14–30, 1982.

[53] T. Ho, B. Leong, R. Koetter, M. Médard, M. Effros, and D. R. Karger, “Byzantine Modification Detection in Multicast Networks using Randomized Network Coding,” in *Proceedings of International Symposium on Information Theory*. IEEE, 2004, p. 144.

[54] A. Rahnama, M. Xia, and P. J. Antsaklis, “Passivity-Based Design for Event-Triggered Networked Control Systems,” *IEEE Transactions on Automatic Control*, to be published.

[55] S. M. Kay, “Fundamentals of Statistical Signal Processing: Detection Theory,” 1998.

[56] D. Alonso-Román and B. Beferull-Lozano, “Adaptive Consensus-based Distributed Detection in WSN with Unreliable Links,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 4438–4442.

[57] J. A. Gubner, *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, 2006.

[58] B. David and G. Bastin, “A Maximum Likelihood Parameter Estimation Method for Nonlinear Dynamical Systems,” in *Proceedings of the 38th IEEE Conference on Decision and Control (CDC)*, vol. 1. IEEE, 1999, pp. 612–617.

[59] Y. Bresler and A. Macovski, “Exact Maximum Likelihood Parameter Estimation of Superimposed Exponential Signals in Noise,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1081–1089, 1986.

[60] R. Fisher, “On an Absolute Criterion for Fitting Frequency Curves,” *Statistical Science*, vol. 12, no. 1, pp. 39–41, 1997.

[61] S. Y. Kung, M.-W. Mak, and S.-H. Lin, *Biometric Authentication: a Machine Learning Approach*. Prentice Hall Professional Technical Reference Upper Saddle River, 2005.