Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems pp. 211–218
DOI: 10.15439/2021F90
ISSN 2300-5963 ACSIS, Vol. 26

Decentralized Controller for Software Interconnected System Subject to Malicious Attacks

Pushkar Kishore
Dept. of C.S.E.
NIT Rourkela
Odisha, India
518CS1002@nitrkl.ac.in

Swadhin Kumar Barisal
1. Dept. of C.S.E., NIT Rourkela.
2. S'O'A Deemed to be University,
Odisha, India
swadhinbarisal@gmail.com

Durga Prasad Mohapatra
Dept. of C.S.E.
NIT Rourkela
Odisha, India
durga@nitrkl.ac.in

***Abstract*—This paper examines the decentralized controller for an interconnected software system subject to malicious attacks. The security of software interconnected systems (SIS) subject to malicious attacks is discussed using Event-Triggered Mechanism (ETM). We design a novel ETM with decentralized feedback for managing resources and keeping the system stable during attacks. We use Numenta-Hierarchical Temporal Memory (N-HTM) for monitoring the ETM values. In addition, numerical simulation of the service provider system is considered for illustrating our model's effectiveness. Experiments reveal that our model stabilizes the system after an average of 2s from the launch of the last attack. As a result, the average consumption of the resources is reduced by 70%.**

***Index Terms*—Decentralized control, Software interconnected systems, Event-triggered control, Numenta-Hierarchical Temporal Memory**

I. Introduction

Interconnected systems consist of a set of coupled subsystems that are physically distributed. We use decentralized control due to its better flexibility, scalability, and reliability compared to centralized control. In these years, decentralized control schemes have been used for dealing with complex interconnected systems [1].

Rapid development in computers and communication directed the rise of Software Control Systems (SCs). Most works on distributed system communication assumed that the quality of service of the communication would ensure stable communication. The objective of our model is to maintain prefixed controller performance during attacks. Different kinds of attacks are examined in security control domains such as denial-of-service (DoS) attacks [2], replay attacks, and deception attacks [3]. Liu et al. [2] concentrated on the stabilization problem for communication systems enduring intermittent DoS jamming attacks. During replay attacks, the data from the operator to the actuator is maliciously repeated. An et al. [4] examined a secure state estimation model based on an adaptive switched mechanism during deception attacks. Wang et al. [5] modeled deception attacks utilizing norm-bounded functions conditioned on the state of the system and developed a resilient control for neural control systems. Ding et al. [6] examined distributed recursive filter against deception attacks utilizing a gradient method. Unlike a simplistic communication system with just one independent controller, Software Interconnected System (SIS) has various subsystems and controllers, making it tough to analyze its performance during malicious attacks. In point-to-point communication, performance is hardly maintained by the controller during non-ideal data transmission. In these years, researchers are trying to improve the Quality of Service (QoS) of the software for a better Quality of Control (QoC). During the last decade, the event-triggered mechanism helped in balancing between QoS and QoC for control systems and sampled-data control systems [7]. Our model is different from the time-triggered mechanism (TTM) in terms of execution frequency of the Event-Triggered Mechanism (ETM). When the frequency of execution of ETM is reduced, resource consumption is controlled.

The existing state-of-the-art works highlight that ETC mainly relies on absolute error, relative error, and some additional measuring parameters. If the error is beyond a predefined threshold, then data-releasing is done. Fei et al. [8] investigated cloud-aided active suspension control where the ETM threshold depends on the bandwidth use. Tian et al. [9] designed a hybrid-triggered scheme which was based on random switching within TTM and ETM and achieved a commending tradeoff among QoC and QoS.

We measure the performance of ETM using a parameter termed as Data Releasing Rate (DRR). Data Release Sample Ratio (DRSR) is the ratio of the number of data releases to the number of data-sampling in a defined period. After reviewing the earlier state-of-the-art works, it was concluded that ETM could effectively reduce the DRR. Whenever an attack occurs on the software system, the controller needs more frequent data to stabilize the system again. We apply a technique, namely Numenta Hierarchical Temporal Memory (N-HTM) [10] which can find and spot anomalous patterns for data where simplistic techniques such as thresholds generate substantial false positives and false negatives. It helps set thresholds; otherwise, the delayed transmission will reduce the system life due to rapid temperature fluctuations. Until the ETM responds with a change in data, the service provider will get massive requests due to malicious attacks and heat the system. As the controller gets feedback from ETM, it reduces the load rapidly, leading to quick cooling. The cycle of rapid temperature changes deteriorates the system's life. A simple ETM will not be sufficient, and designing a resilient ETM for

SCS subject to malicious attacks is challenging. The above issues motivate us to design an efficient ETM.
This article introduces a decentralized controller for interconnected software systems subject to malicious attacks. The main contributions of this article are as follows:

1) Malicious attacks on the software are considered.
2) A novel ETM is proposed where the control unit receives the least amount of feedback defined using N-HTM and guarantees desired control performance during malicious attacks.
3) DRR during the run-time is maintained at a moderate level.
4) A decentralized security output feedback control technique is proposed for stabilizing the system during malicious attacks.

The remaining part of this paper is organized as follows: Section II discusses the integrated proposed model of the SIS, ETM, and malicious attacks. Section III manifests the design considerations of decentralized, resilient control for SIS subject to malicious attacks. Section IV signifies results, advantages and effectiveness of our proposed model using service provider model, Section V considers related state-of-the-art works, section VI highlights threats to validity of our model and Section VII concludes the paper.

II. Integrated proposed model of SIS, ETM, and malicious attacks

Before going in-depth, we summarize the notations that will be used in the paper. $\mathbb{R}^n$ is used for n-dimensional Euclidean space, $\mathbb{R}^{nxm}$ for a set of n x m matrices, $\|.\|$ represents Euclidean norm, P^T is the transpose of a matrix P, $E\{\beta\}$ evaluates the expectation of the stochastic variable β, $diag_N\{X_i\} = diag\{X_1, X_2, ..., X_N\}$, $col_n\{x_i\} = \left[x_1^T, ..., x_N^T\right]^T$ and $*$ represents the symmetric term in a matrix.

A. System Description

Consider an interconnected system S depicted in Figure 1:

$$\dot{x}(t) = Ax(t) + Bu(t) + f(t, x(t)) \tag{1}$$

$$y(t) = Cx(t) \tag{2}$$

where A, B and C are matrices, $f(t, x(t))$ represents the coupling between interconnected systems, state of the system: $x(t) = col_N\{x_i(t)\}$, input to the system: $u(t) = col_N\{u_i(t)\}$, output of the system: $y(t) = col_N\{y_i(t)\}$, $x_i(t)\epsilon\mathbb{R}^{n_i}(\sum_{i=1}^{N} n_i = n)$, $y_i(t)\epsilon\mathbb{R}^{p_i}(\sum_{i=1}^{N} p_i = p)$, $u_i(t)\epsilon\mathbb{R}^{m_i}(\sum_{i=1}^{N} m_i = m)$, $p < n$ and C is row full rank. Here it is assumed that coupling function satisfies Equation 3:

$$||f_i(t, x(t))|| \leq \delta_i^2 \|F_i x(t)\| \tag{3}$$

where δ_i is scalar and F_i is a known matrix.

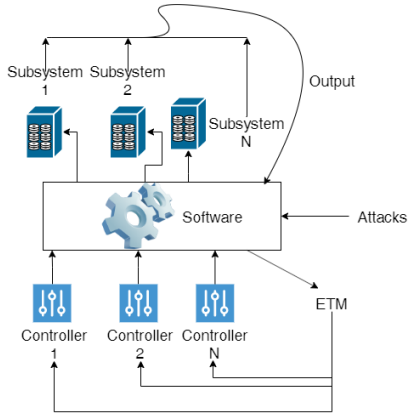


Fig. 1: The framework of software decentralized control system

B. Novel ETM

Figure 1 shows the output of the subsystem provided to the software and monitored by the ETM. The ETM monitors the output and accordingly decides whether to inform the controller or not. We use N-HTM for calculating the raw anomaly score of outputs of subsystems, as anomaly score indicates the deviation between actual and predicted output. First, an anomaly score is computed from the intersection between predicted and actual sparse vectors. Then, we compute the anomaly likelihood value from the window of the last W raw anomaly scores. N-HTM models this distribution as a rolling normal.

$$W = \sum S_t \tag{4}$$

where, t = 1, 2, 3, 4, ... The sample mean and variance are continuously calculated and updated using Equations 5 and 6.

$$\mu_t = \frac{\sum_{i=0}^{i=w-1} s_{t-i}}{k} \tag{5}$$

$$\delta_t^2 = \frac{\sum_{i=0}^{i=w-1} (s_{t-i} - \mu_t)^2}{k-1} \tag{6}$$

where, w is the last window length and k is the number of instances. Then an average of recent anomaly scores is evaluated using Equation 7, and the anomaly is confirmed by applying a threshold to the Gaussian tail probability.

$$L_t = 1 - Q(\frac{\mu_t{}' - \mu_t}{\delta_t}) \tag{7}$$

where $\mu_t' = \frac{\sum_{i=0}^{i=w-1} s_{t-i}}{j}$ and $j < k$. Anomalous behavior will be reported if $L_t \geq 1$.
Remark 1: We use different output values, i.e., y from other ETC [11]. Anomalous data is searched within a sequence of outputs, and an alert is sent to the controller for altered output. Anomalous finder like N-HTM tackles fluctuations caused by noises and disturbances.
Remark 2: Compared to the conventional ETM, our model sends packets sporadically to the controller when the system is under attack or sensing external disturbance. The above

process will ensure better QoC, and DRR average value will be lower through runtime.

C. Malicious Attack Model

Decentralized control will break the controller problems by utilizing multiple controllers. In our work, the decentralized scheme of the subsystems is defined using Equation 8.

$$u_i(t) = K_i y_i(t) \tag{8}$$

where, K_i is the local controller gain of the subsystems.
Remark 3: The centralized output feedback control is defined using Equation 9.

$$u_i^c(t) = \sum_{j=1}^{N} k_{ij} y_j(t) \tag{9}$$

The control information transmitted via software is vulnerable to attack. The control input to the subsystem during the malicious attack is defined using Equation 10.

$$\breve{u}_i(t) = u_i(t_k) \pm a_i(t) u_i(t_k) \tag{10}$$

where, $t\varepsilon\left[t_k + \tau_k, t_{k+1} + \tau_{k+1}\right]$, $a_i(t_k)$ is the attack that tampers the control input at time t_k, τ_k is the software processing delay having value between η and expectation of τ. Gu et al. [11] had proved that malicious attack will remain undetected when $\|a_i(t)\| \leq \varsigma^3/N$, where ς is a scalar value.
Remark 4: The malware which attacks intermittently is effective due to these reasons.

1) The probability of the continuous attack being detected is more prominent than random intermittent attacks. E.g., Trojan is tougher to detect due to its stealth and intermittent attacking behavior, while worms are easily detected due to continuous attack [1].
2) The attack is obstructed due to the underlying operating system on which the software is running.
3) The objective of the attack is to destroy the control system; thus, the malicious instances need to alter the input so that the dropout of the attack signal looks like a standard transmission.

Remark 5: $a_i(t)$ varies between 0 to 1 depending on whether the data transmitted through the system is benign or malicious. The intensity of attacks happening on each subsystem is defined using the number of malicious samples active at any point.
From the above discussions, the control input during malicious attacks is finalized using Equation 11.

$$\breve{u}(t) = Ky(t_k) \pm a(t) Ky(t_k) \tag{11}$$

where, $K = diag_N\left\{K_i\right\}$.

[1]https://www.websecurity.digicert.com/security-topics/difference-between-virus-worm-and-trojan-horse

D. Overall Model

Combining Equations 1, 2 and 8, we define SIS using Equation 12.

$$\dot{x}(t) = Ax(t) + B(I + a(t))Ky(t_k) + f(t) + \sum_{i=1}^{N} (\bar{a}_i - a_i(t))BL_iKy(t_k) \tag{12}$$

where, $L_i = diag\left\{0..0I0..0\right\}$; $I's$ location depends on the value of i. We define $e(t_k, l) = x(t_k) - x(t_k + l)$ and $\eta_t = t - (t_k + l)$. Then, Equation 12 is transformed to Equation 13.

$$\dot{x}(t) = Ax(t) + B(I + a(t))KC\left[e(t_k, l) + x(t - \eta(t))\right] + f(t) + \sum_{i=1}^{N}(\bar{a}_i - a_i(t))BL_iKC[e(t_k, l) + x(t - \eta_t))] \tag{13}$$

A SIS is termed stable whenever $x^T(t)Px(t) \leq \varsigma^2$, $\forall t \geq t_0 + T$ for $\|a_i(t)\| \leq \varsigma^3/N$. Here, $P > 0$ and $T > 0$. The primary purpose of our work is to build a controller and ETM such that SIS is stable in the presence of malicious attacks.

III. Controller Design

In this section, we develop the controller together with ETM for SIS when malicious attacks are happening. Conditions will be formed and represented in terms of a set of linear matrix inequalities. At first, we define $\zeta = \begin{bmatrix} x^T(t) & x^T(t-\eta_1) & x^T(t-\eta(t)) & x^T(t-\eta_2) & a^T(t) \end{bmatrix}$. $\Im_i$ represents a compatible row-matrix with the i^{th} block as identity matrix and other as zero matrices, e.g. $\Im_3 = \begin{bmatrix} 0 & 0 & I & 0 & 0 \end{bmatrix}$. Later on, we discuss some lemmas which are applied in our designs.
Lemma 1 [11]: Let $\eta(t)\varepsilon\left[\eta_1, \eta_2\right]$, $x(t)\varepsilon\mathbb{R}^n$ be some positive matrices like $R_1\varepsilon\mathbb{R}^{nxn}$, $R_2\varepsilon\mathbb{R}^{nxn}$ and matrix $U\varepsilon\mathbb{R}^{nxn}$. Then the inequalities are given in Equation 14:

$$\begin{aligned} -\eta_1 \int_{t-\eta_1}^{t} \dot{x}^T(s) R_1 \dot{x}(s) ds &\leq \zeta^T(t) \Re_1 \zeta(t) \\ -(\eta_2 - \eta_1) \int_{t-\eta_2}^{t-\eta_1} \dot{x}^T(s) R_2 \dot{x}(s) ds &\leq \zeta^T(t) \Re_2 \zeta(t) \end{aligned} \tag{14}$$

Where, $\Re_1 = -(\Im_1 - \Im_2)^T R_1 (\Im_1 - \Im_2)$ and $\Re_2$ is solved using Equation 15.

$$\Re_2 = -\begin{bmatrix} \Im_2 - \Im_3 \\ \Im_3 - \Im_4 \end{bmatrix}^T \begin{bmatrix} R_2 & * \\ U & R_2 \end{bmatrix} \begin{bmatrix} \Im_2 - \Im_3 \\ \Im_3 - \Im_4 \end{bmatrix} \tag{15}$$

Lemma 2: For some given constants, η_1, η_2, ϱ, ς, σ, k, with ETM and SIS is stable whenever there exists matrices like $P > 0$, $\psi > 0$, $Q_1 > 0$, $Q_2 > 0$, $R_1 > 0$, $R_2 > 0$, a matrix U, a positive scalar ϵ fulfilling:

$$\begin{bmatrix} \Gamma_1 & * & * \\ ZA_0 & -Z & * \\ \Gamma_2 & 0 & -\Gamma_3 \end{bmatrix} < 0 \tag{16}$$

Where, Γ_1 is defined as:

$$\Gamma_1 = \begin{bmatrix} \Gamma_{11} & * & * & * & * \\ R_1 & \Gamma_{22} & * & * & * \\ \Gamma_{31} & \Gamma_{32} & \Gamma_{33} & * & * \\ 0 & -U & \Gamma_{43} & \Gamma_{44} & * \\ \Gamma_{51} & 0 & \Gamma_{53} & 0 & \Gamma_{55} \end{bmatrix} \tag{17}$$

$\Gamma_{11} = PA + A^T P + Q_1 + Q_2 - R_1 + \epsilon\delta^2 F^T F + \varsigma P$
$\Gamma_{22} = -Q_1 - R_1 - R2$
$\Gamma_{31} = C^T K^T B^T P(I + a)$
$\Gamma_{32} = R_2 + U$
$\Gamma_{33} = -2R_2 - U - U^T + 4\sigma\psi$
$\Gamma_{43} = \Gamma_{32}$
$\Gamma_{44} = -Q_2 - R_2$
$\Gamma_{51} = \Gamma_{31}$
$\Gamma_{53} = 2\sigma(1 + \varrho)\psi C$
$\Gamma_{55} = -(1 - \sigma - 2\varrho\sigma)\psi$
$A_0 = \begin{bmatrix} A & 0 & B(I + a)KC & 0 & Ba \end{bmatrix}$
$A_{2i} = \begin{bmatrix} 0 & 0 & BL_iKC & 0 & -2BL_i \end{bmatrix} \forall i \epsilon \{1, n\}$

$$\Gamma_2 = \begin{bmatrix} ZA_{21} \\ . \\ . \\ . \\ ZA_{2N} \end{bmatrix}$$

$Z = \eta_1^2 R_1 + (\eta_2 - \eta_1)^2 R_2$
$\Gamma_3 = diag\{Z, ..., Z\}$ N times
$\psi = C^T \tilde{\psi} C$

Now, we evaluate the values of gain and parameters of ETM.

Theorem 1: For some given constants, η_1, η_2, ϱ, ς, σ and ϵ, we have a stable SIS with ETM during malicious attacks when $\tilde{\psi} > 0$, $\tilde{Q}_j > 0$, $\tilde{R}_1 > 0$, $\tilde{R}_2 > 0$, Y, $\tilde{U}$ and V.

$$\Gamma = \begin{bmatrix} \tilde{\Gamma}_1 & * & * & * \\ \tilde{A}_1 & -\tilde{Z}_0 & * & * \\ \tilde{\Gamma}_2 & 0 & \tilde{\Gamma}_3 & * \\ \tilde{\Gamma}_4 & 0 & 0 & -\epsilon I \end{bmatrix} < 0 \tag{18}$$

$$CX = VC \tag{19}$$

Where,

$$\tilde{\Gamma}_1 = \begin{bmatrix} \tilde{\Gamma}_{11} & * & * & * & * \\ \tilde{R}_1 & \tilde{\Gamma}_{22} & * & * & * \\ \tilde{\Gamma}_{31} & \tilde{\Gamma}_{32} & \tilde{\Gamma}_{33} & * & * \\ 0 & -\tilde{U} & \tilde{\Gamma}_{43} & \tilde{\Gamma}_{44} & * \\ \tilde{\Gamma}_{51} & 0 & \tilde{\Gamma}_{53} & 0 & \tilde{\Gamma}_{55} \end{bmatrix} \tag{20}$$

$\tilde{\Gamma}_{11} = AX + A^T X + \tilde{Q}_1 + \tilde{Q}_2 - \tilde{R}_1 + \varsigma X$
$\tilde{\Gamma}_{22} = -\tilde{Q}_1 - \tilde{R}_1 - \tilde{R}2$
$\tilde{\Gamma}_{31} = C^T Y^T B^T (I + a)$
$\tilde{\Gamma}_{32} = \tilde{R}_2 + \tilde{U}$
$\tilde{\Gamma}_{33} = -2\tilde{R}_2 - \tilde{U} - \tilde{U}^T + 4\sigma\tilde{\psi}$
$\tilde{\Gamma}_{43} = \tilde{\Gamma}_{32}$
$\tilde{\Gamma}_{44} = -\tilde{Q}_2 - \tilde{R}_2$
$\tilde{\Gamma}_{51} = \tilde{\Gamma}_{31}$
$\tilde{\Gamma}_{53} = 2\sigma(1 + \rho)\tilde{\psi} C$
$\tilde{\Gamma}_{55} = -(1 - \sigma - 2\rho\sigma)\tilde{\psi}$
$\tilde{\Gamma}_4 = \begin{bmatrix} \epsilon\delta FX & 0 & 0 & 0 & 0 \end{bmatrix}$
$\tilde{A}_0 = \begin{bmatrix} AX & 0 & B(I + a)YC & 0 & Ba \end{bmatrix}$
$\tilde{A}_{2i} = \begin{bmatrix} 0 & 0 & BL_iYC & 0 & -WBL_i \end{bmatrix} \forall i \epsilon \{1, n\}$

$$\tilde{\Gamma}_2 = \begin{bmatrix} \tilde{A}_{21} \\ . \\ . \\ . \\ \tilde{A}_{2N} \end{bmatrix}$$

$\tilde{\Gamma}_3 = diag\left\{\tilde{Z}_1, ..., \tilde{Z}_N\right\}$
$\tilde{Z}_i = -2\alpha_i X + \alpha_i^2 \tilde{Z}$

The gain is defined using Equation 21 and weight matrix using 22.

$$K = YV^{-1} \tag{21}$$

$$\bar{\psi} = (CC^T)^{-1} CX^{-1} \tilde{\psi} X^{-1} C^T (CC^T)^{-1} \tag{22}$$

Theorem 2: For some given constants, η_1, η_2, ϱ, ς, σ, ϵ and φ , we have a stable SIS with ETM during malicious attacks when $\tilde{\psi} > 0$, $\tilde{Q}_j > 0$, $\tilde{R}_1 > 0$, $\tilde{R}_2 > 0$, Y, $\tilde{U}$ and V. The linear inequalities holding at this stage is:

$$\Gamma < 0 \tag{23}$$

$$\begin{bmatrix} -\varphi I & * \\ CX - VC & -I \end{bmatrix} \tag{24}$$

IV. Results and Effectiveness

In this section, we discuss the service provider model, which manifests the advantages and effectiveness of our model. Figure 2 represents the architecture of the service provider system. The above model comprises four subsystems with a

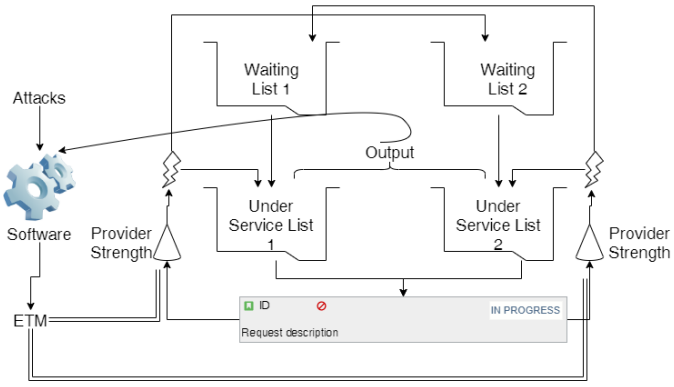


Fig. 2: Architecture of Service Provider System

waiting list containing requests to be served later and **under service list** processing the requests. Requests are stored in a pool and sent to the waiting list or **under service list**. Requests passing through right provider strength are passed to **under service list 2** and **waiting list 1**. Similarly, requests passing through left provider strength will be sent to the **under service length 1** and **waiting list 2**. Requests are transferred to the **under service** list from their respective waiting list at a fixed rate. Our objective is to keep the number of requests at 50% of the capacity of the list size such that the server keeps running smoothly. The limit of 50% can be varied according to our server processing power. Provider Strength is varied across

run-time to keep **under service lists'** performance at their best. The outputs from the **under service** list are sent to the software prone to malicious attacks. After getting the feedback from the software, ETM decides the following vector of parameters for provider strengths to ensure fulfilling our objective.
The system can be modelled using following equations:

$$\begin{aligned}\frac{dr_{usr1}}{dt} &= (c_{usr1}/C_{usr1})(r_{usr1})^2 + (c_{wl1}/C_{usr1})(r_{wl1})^2 + \frac{\eta_1}{C_{usr1}}v_1 \\ \frac{dr_{usr2}}{dt} &= (c_{usr2}/C_{usr2})(r_{usr2})^2 + (c_{wl2}/C_{usr2})(r_{wl2})^2 + \frac{\eta_2}{C_{usr2}}v_2 \\ \frac{dr_{wl1}}{dt} &= (c_{wl1}/C_{wl1})(r_{wl1})^2 + \frac{(1-\eta_1)}{C_{wl1}}v_1 \\ \frac{dr_{wl2}}{dt} &= (c_{wl2}/C_{wl2})(r_{wl2})^2 + \frac{(1-\eta_2)}{C_{wl2}}v_2 \\ y_{usr1} &= k_c r_{usr1}, y_{usr2} = k_c r_{usr2} \end{aligned} \quad (25)$$

where, $C_{usr1} = C_{wl1} = C_{usr2} = C_{wl2} = 1000$ representing number of requests going in the sub-system, $c_{usr1} = c_{wl1} = c_{wl2} = c_{usr2} = 50$ representing numbers of requests going out of the sub-system, $k_c = 0.5$, r_* shows the maximum holding capacity of the subsystems in terms of number of requests and η_* along with v_* represents the provider strength. We keep the operating range of the system as:

$$0 \leq r_{usr1} \leq 800, 0 \leq r_{usr2} \leq 800, 0 \leq r_{wl1} \leq 500, 0 \leq r_{wl2} \leq 500 \quad (26)$$

For the minimal case, we consider following parameters: r_{usr1} = 600, r_{usr2} = 600, r_{wl1} = 200, r_{wl2} = 200, v_1 = 2, v_2 = 2, η_1 = 0.7 and η_2 = 0.3. Then, we obtain the following system equations:
$x_{usr1} = r_{usr1}$ - 600, $x_{usr2} = r_{usr2}$ - 600, $x_{wl1} = r_{wl1}$ - 200, $x_{wl2} = r_{wl2}$ - 200, $u_1 = v_1$ - 2 and $u_2 = v_2$ - 2.
Then Equation 25 is modified as follows:

$$\begin{aligned}\dot{x}_{usr1} &= 0.05(x_{wl1}+200)^2 - 0.05(x_{usr1}+600)^2 + 0.0007(u_1+2) \\ \dot{x}_{usr2} &= 0.05(x_{wl2}+200)^2 - 0.05(x_{usr2}+600)^2 + 0.0003(u_2+2) \\ \dot{x}_{wl1} &= -0.05(x_{wl1}+200)^2 + 0.0003(u_1+2) \\ \dot{x}_{wl2} &= -0.05(x_{wl2}+200)^2 + 0.0007(u_2+2) \end{aligned} \quad (27)$$

From 26, we define the range of the variables as:

$$-600 \leq x_{usr1} \leq 200, -600 \leq x_{usr2} \leq 200, -200 \leq x_{wl1} \leq 300, -200 \leq x_{wl2} \leq 300 \quad (28)$$

From Equation 27 and 28, we redefine the system model as:

$$\begin{aligned}\dot{x}_{usr1} &= 0.05x_{wl1}^2 + 20x_{wl1} - 0.05x_{usr1}^2 - 60x_{usr1} + 0.0007u_1 \\ \dot{x}_{usr2} &= 0.05x_{wl2}^2 + 20x_{wl2} - 0.05x_{usr2}^2 - 60x_{usr2} + 0.0003u_2 \\ \dot{x}_{wl1} &= 0.05x_{wl1}^2 - 20x_{wl1} + 0.0003u_1 \\ \dot{x}_{wl2} &= 0.05x_{wl2}^2 - 20x_{wl2} + 0.0007u_2 \\ y_{usr1} &= 0.5x_{usr1}, y_{usr2} = 0.5x_{usr2} \end{aligned} \quad (29)$$

From Equations 28 and 29, we restate Equation 1 as:

$$A = \begin{bmatrix} -60 & 20 & 0 & 0 \\ 0 & -20 & 0 & 0 \\ 0 & 0 & -60 & 20 \\ 0 & 0 & 0 & -20 \end{bmatrix}$$
$$B = \begin{bmatrix} 0.0007 & 0 \\ 0 & 0 \\ 0 & 0.0003 \\ 0 & 0 \end{bmatrix} F = \begin{bmatrix} 39 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 \\ 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 0.06 \end{bmatrix} \quad (30)$$

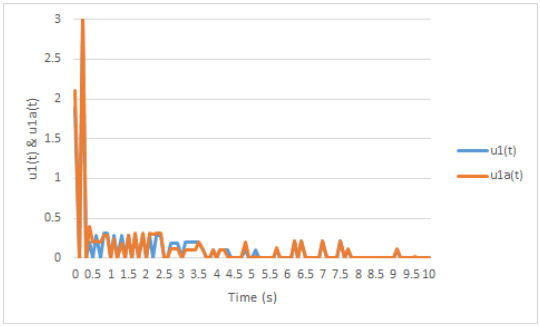
We define $x_i = [C_{usri}, C_{wli}]^T$. The subsystem S can be described using Equation 1 and satisfies

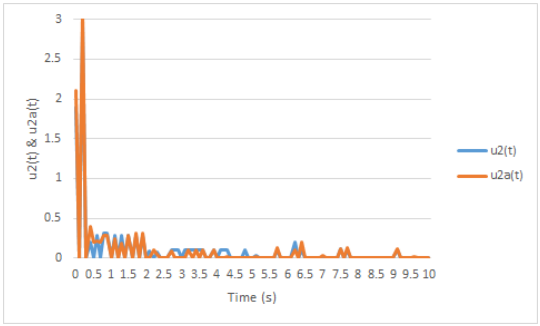
$$||f_i(t, x(t))|| \leq 0.01 \, ||F_i x(t)|| \quad (31)$$

The measured output of the subsystem S_i is:

$$y_i(t) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} x_i(t) \forall i = 1, 2 \quad (32)$$

Here, the system is a nonself-regulating due to the presence





(a) Sequence of attacks on subsystem wl2 and usr2

Fig. 3: Sequence of attacks done on the subsystem

of two positive provider units in the system. In Figure 2,

TABLE I: Malware dataset

Sl. No.	Malware family	Number of samples
1	Backdoor	1352
2	Worm	559
3	Trojan	2394
4	Virus	809
Total number of samples		5114

control signal is passed through a software. We choose software processing delay η = 1 ms and expectation of τ = 10 ms. We select the parameters of ETM as σ = 0.1 and ϱ = 0.2.

Suppose our software is attacked by malicious instances described in Table I. These samples are provided by VirusTotal [2]. In case of $\|a_i(t)\| \leq \varsigma^3/N$, ς is 0.1. Figure 3 depicts the sequences of attacks done on the in/out puts of subsystems.

From Equation 21 and 22, we can obtain the gain and weight matrix as described below:

$$k = \begin{bmatrix} -0.8 & 1.2 \\ 1.2 & -0.8 \end{bmatrix} \tag{33}$$

$$\bar{\psi} = \begin{bmatrix} 0.000704 & 0.000384 \\ 0.000064 & -0.000256 \end{bmatrix} \tag{34}$$

As per the attacks depicted in Figure 3, and with initial values determined from r_{usr1}, r_{wl1}, r_{usr2} and r_{wl2} = [100, 150, 150, 100], we can get the state response of each subsystem using the parameters described above and depicted in Figure 4. It is observed that subsystems achieve stability after 6s even under attack conditions. The data release sequence is depicted in Figure 5. It is observed that the average DRR is around 30% of the total time. Thus, it is clear that our proposed model effectively provides service even when the software is under attack. The reduction of DRR achieved using ETM assist in saving computational resources.

Now, we will illustrate the beneficial aspect of our proposed model using:

1) Provider performance.
2) DRR.

We alter the execution of the subsystems by attacking software from 6s to 10s. Now, we will study the response of the subsystem under two cases:

1) *Case 1*: The model proposed by Gu et al. [11], where σ = 0.1 and ϱ = 0.2 and dependent on ETM.
2) *Case 2*: The model proposed by us with similar parameters and dependent on N-HTM based ETM.

Our proposed model's ETM is a bit less sensitive to variable external disturbances compared to other ETM (as shown in case 1). The sensitivity is measured by the number of transmissions sent to the providers. The average DRR is low, but instantaneous higher DRR is required sometimes to defy and counter the effects of the attack. From Figure 6 and 7, we

[2]https://www.virustotal.com/

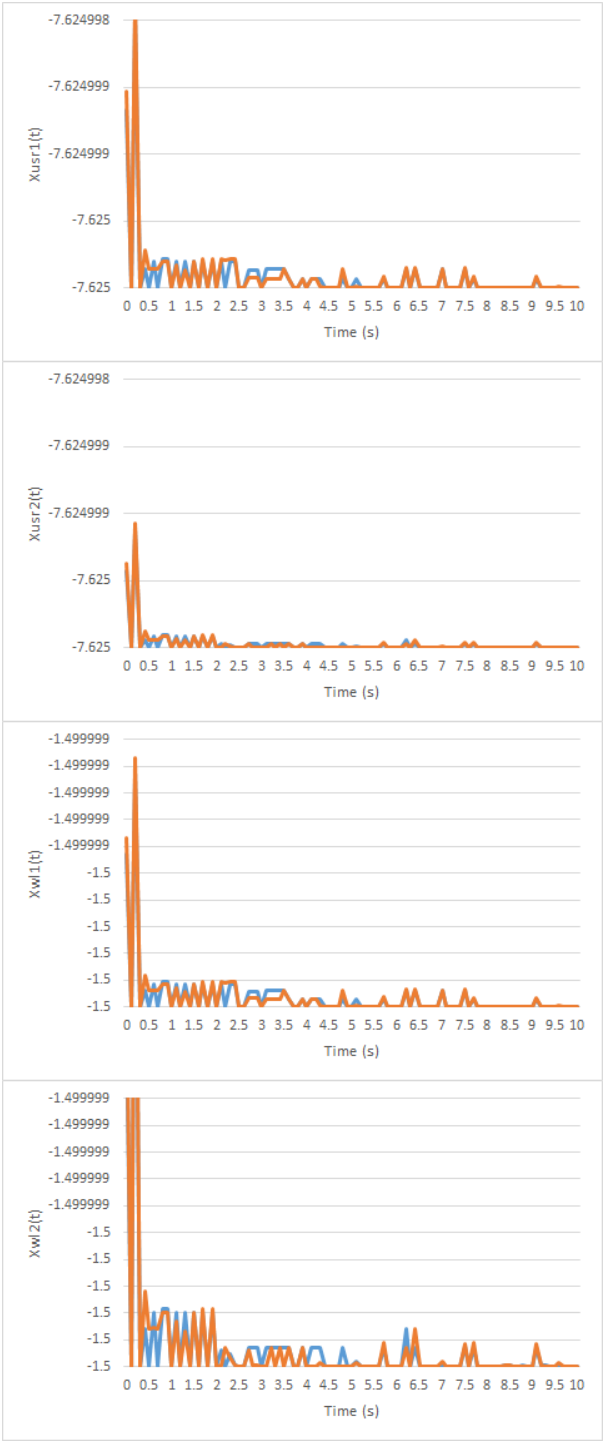


Fig. 4: State response of each subsystem

conclude that the average DRR is higher in Case 1 compared to Case 1 during the extreme attack duration of 6-10s. It is also concluded that the system providers with proposed N-HTM based ETM receive less information in case of the attack on software than other ETM. However, the anomaly score returned by the ETM covers up the lower DRR and saves

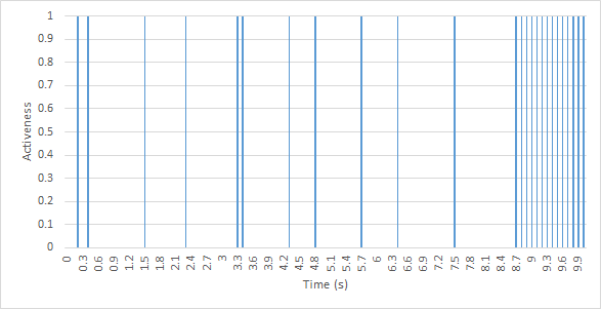


Fig. 5: Release by ETM

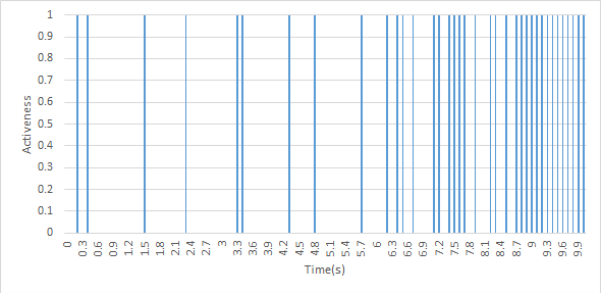


Fig. 7: Release by ETM (Case 1)

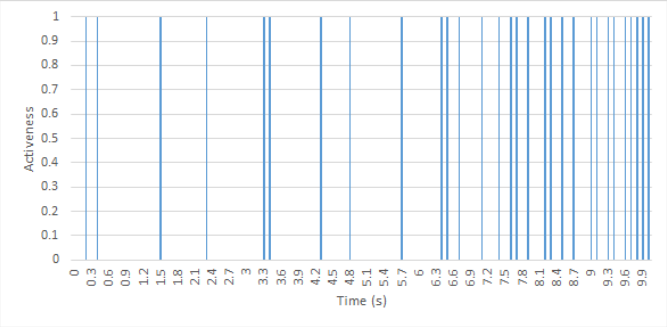


Fig. 6: Release by ETM (Case 2)

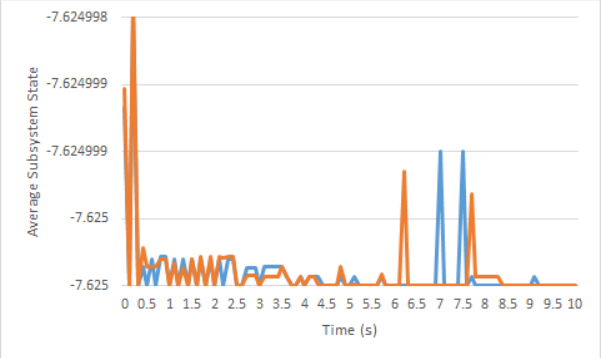


Fig. 8: Average state response (Case 2)

resources. An anomaly score provides exact rectification for the inputs such that the system reaches stability or remains stable. However, the ETM in case 1 needs to send many data for higher-order attacks and needs much time for stabilization. Therefore, the above practice will deplete a lot of resources and inefficient for longer running time. Thus, we can achieve better performance using our proposed model. From Figure 8, it is concluded that on average, the subsystems will stabilize after 8^{th} second, i,e., 2 seconds after the extensive attack had occurred. However, for Case 1, the system stabilizes at the last moment, i.e., 9^{th} second. However, the software may not always fail from an attack as it depends on penetration level. For the above case, N-HTM will not create any threat alert.

V. Related State-of-the-art Works

Liang et al. [12] studied the quantized cooperative control problem for multiagent systems with unknown gains. They designed a speed function and observed that the tracking errors converge to a prescribed compact set in a given finite time. Niu et al. [1] proposed an adaptive neural-network-based dynamic surface control (DSC) method for stochastic interconnected nonlinear non-strict-feedback systems. The proposed controllers guarantee that the closed-loop stochastic interconnected system is probably semi-globally bounded stable.

Liu et al. [2] estimated the security of distributed state for nonlinear networked systems against denial-of-service attacks. An event-triggered scheme and a quantization mechanism were employed to reduce network burden. Lyapunov stability theory was used for ensuring the exponential stability of the estimation error systems. A numerical example was considered for testing the feasibility of their proposed method.

Tang et al. [13] investigated the tracking control of mobile robots under the presence of malicious denial-of-service attacks. Some explicit characterizations were considered for the duration and frequency property of malicious DOS attacks. They developed a set of event-triggering conditions for ensuring tracking convergence. A practical experiment is conducted by tracking the control of an amigobot mobile robot under the presence of malicious attacks. Ye et al. [3] investigated the detection problem of false data injection attacks in cyber-physical systems (CPSs) with white noise. For ensuring the stability of CPSs during false data-injection attacks, a summation (SUM) detector was proposed. The SUM detector utilized the current compromised as well as historical information for identifying the threat. An improved false data-injection attack with a time-variable increment coefficient was also developed. Some simulations were conducted for ascertaining the effectivity of the SUM detector.

Gu et al. [11] studied the security of NIS in the presence of cyber-attacks based on a new ETM. They designed a novel ETM and a decentralized output feedback control (DOFC) scheme to keep NIS stable in the presence of cyber-attacks. They reduced the data-release rate, consequently reducing network bandwidth, battery supply, and computation. Numerical simulations were done to illustrate the effectiveness of their technique.

For the generation of anomaly scores in sequential data,

Ahmad et al. [10] developed a technique termed N-HTM. It is predominantly used for real-time applications dedicated to finding out the anomalies in data streams of their respective domains. They had demonstrated that their system was efficient, produced accurate results in the presence of noisy data, detected subtle temporal anomalies and minimized false positives, and adaptable to statistical change in the data. Kishore et al. [14] proposed an incremental malware detection model for meta-feature API and system call sequence. They used the N-HTM for generating the anomaly score of each element in the sequence of system and API calls. The detection accuracy of 95.2% achieved using N-HTM was also the motivating factor for using N-HTM.

VI. Threats to Validity

In this section, we identify some possible threats to the validity of our approach. First, the probability of attacks on each of the sub-systems is kept constant. It is done to ensure that N-HTM learns the patterns properly and provides anomaly scores accurately. If the attack probability is different, then the N-HTM will be inaccurate due to improperly learning multiple different distribution patterns at once [10]. However, we can make the model work by considering other attention mechanism based models. Malicious samples that can attack only network channels will be ineffective. While studying the response of sub-systems, we define stability at the negative values. It seems unusual, but the state of all sub-systems is maintained at a negative point from the beginning. Due to these reasons, values at base of the state response graph represent stable state of the sub-systems. For the values provided in the ETM release graph, 1 represents active, and 0 represents no release. These results depend on the response of the software if an attack occurs on any of the sub-systems.

At last, the centralized controller will be least prone to attacks and maintain stability during the duration of attack [11]. However, the continuous attack will block the service providers from responding when the decision-maker is under continuous attack. This problem can be covered using a decentralized controller with the least time required to attain stability and reduce resource consumption with ETM guided by N-HTM.

VII. Conclusions and Future work

This paper proposes a model that stabilizes the Software InterConnected Model (SIS), having decentralized controllers during malicious attacks. A new Event-Triggered Mechanism (ETM) is designed, having Numenta-Hierarchical Temporal Memory at its back-end. N-HTM seems effective in reducing the average Data Release Rate (70% reduced) and even stabilizes the system by providing anomalous scores for guiding the following input to the providers. Due to the reduction in average DRR, we reduce the power usage of battery supply, disk utilization, and processor computation time. Controller gain and ETM parameters assist in stabilizing the model based on the response of ETM obtained during the presence of attacks. Controller gain and ETM parameters are obtained using stochastic analysis and Lyapunov stability theory. Lastly, we choose a service provider system to check the effectiveness of our proposed model and observe that subsystems stabilize after 2s from the launch time of the last attack. N-HTM based ETM helped in reducing DRR and resource consumption by 70%.

As future work, we will try with other anomaly detectors like Long Short term memory, Bidirectional Long Short-term memory as the base of ETM. The above configuration of the system resembles an edge-computing computation where a network-based communication channel is not required.

References

[1] B. Niu, H. Li, Z. Zhang, J. Li, T. Hayat, and F. E. Alsaadi, "Adaptive neural-network-based dynamic surface control for stochastic interconnected nonlinear nonstrict-feedback systems with dead zone," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1386–1398, 2018. doi: 10.1109/TSMC.2018.2866519

[2] J. Liu, W. Suo, L. Zha, E. Tian, and X. Xie, "Security distributed state estimation for nonlinear networked systems against dos attacks," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 3, pp. 1156–1180, 2020. doi: 10.1002/rnc.4815

[3] D. Ye and T.-Y. Zhang, "Summation detector for false data-injection attack in cyber-physical systems," *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2338–2345, 2019. doi: 10.1109/tcyb.2019.2915124

[4] L. An and G.-H. Yang, "Secure state estimation against sparse sensor attacks with adaptive switching mechanism," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2596–2603, 2017. doi: 10.1109/tac.2017.2766759

[5] K. Wang, E. Tian, J. Liu, L. Wei, and D. Yue, "Resilient control of networked control systems under deception attacks: a memory-event-triggered communication scheme," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 4, pp. 1534–1548, 2020. doi: 10.1002/rnc.4837

[6] D. Ding, Z. Wang, D. W. Ho, and G. Wei, "Distributed recursive filtering for stochastic systems under uniform quantizations and deception attacks through sensor networks," *Automatica*, vol. 78, pp. 231–240, 2017. doi: 10.1201/9780429465901-16

[7] J. Xu, Y. Tang, W. Yang, F. Li, and L. Shi, "Event-triggered minimax state estimation with a relative entropy constraint," *Automatica*, vol. 110, p. 108592, 2019. doi: 10.1016/j.automatica.2019.108592

[8] Z. Fei, X. Wang, M. Liu, and J. Yu, "Reliable control for vehicle active suspension systems under event-triggered scheme with frequency range limitation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019. doi: 10.1109/tsmc.2019.2899942

[9] E. Tian, Z. Wang, L. Zou, and D. Yue, "Probabilistic-constrained filtering for a class of nonlinear systems with improved static event-triggered communication," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 5, pp. 1484–1498, 2019. doi: 10.1002/rnc.4447

[10] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017. [Online]. Available: https://doi.org/10.1016/j.neucom.2017.04.070

[11] Z. Gu, J. H. Park, D. Yue, Z.-G. Wu, and X. Xie, "Event-triggered security output feedback control for networked interconnected systems subject to cyber-attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020. doi: 10.1109/tsmc.2019.2960115

[12] H. Liang, Y. Zhang, T. Huang, and H. Ma, "Prescribed performance cooperative control for multiagent systems with input quantization," *IEEE Transactions on cybernetics*, vol. 50, no. 5, pp. 1810–1819, 2019. doi: 10.1109/tcyb.2019.2893645

[13] Y. Tang, D. Zhang, D. W. Ho, W. Yang, and B. Wang, "Event-based tracking control of mobile robot with denial-of-service attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 9, pp. 3300–3310, 2018. doi: 10.1109/tsmc.2018.2875793

[14] P. Kishore, S. K. Barisal, and D. P. Mohapatra, "An incremental malware detection model for meta-feature api and system call sequence," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2020. doi: 10.15439/2020f73 pp. 629–638.