IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 1, 2018, accepted June 29, 2018, date of publication July 16, 2018, date of current version August 28, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2856368*

# Behavior and Vulnerability Assessment of Drones-Enabled Industrial Internet of Things (IIoT)

**VISHAL SHARMA, (Member, IEEE), GAURAV CHOUDHARY, YONGHO KO, AND ILSUN YOU, (Senior Member, IEEE)**
Department of Information Security Engineering, Soonchunhyang University, Asan 31538, South Korea

Corresponding author: Ilsun You (ilsunu@gmail.com)

This work was supported in part by the Institute for Information & Communications Technology Promotion through the Korea Government (MSIT) (Rule Specification-based Misbehavior Detection for IoT-Embedded Cyber Physical Systems) under Grant 2017-0-00664 and in part by the Soonchunhyang University Research Fund.

**ABSTRACT** Accessibility to industrial processes and direct obtaining of the desired services are the major facilities of Industrial Internet of Things (IIoT). IIoT covers crucial aspects of smart systems, such as automation, keenly intellective setups, asset management, and user-industry collaboration. These user-industry setups are facilitated by modern era network technologies, which also include an immense dependence on drones as one of the on-demand components for amending the quality and maximizing the coverage. However, these kinds of network formations require precise operations of drones and their perpetual assessment. The existing studies have highlighted these issues but fail to provide the behavior as well as the vulnerability evaluations of drones enabled IIoT. In addition, the existing studies are unable to provide statewise verification of drones and do not recognize anomaly drones based on their behavior over varying properties. Furthermore, the existing solutions lack facilities for including security policies which help in assessing the vulnerabilities with a higher accuracy. This paper fills this gap by using a novel $N$-layered hierarchical context-aware aspect-oriented Petri net model that not only evaluates the drone behavior but also assesses it for potential vulnerabilities by the utilization of security policies. Statewise verification is performed for the proposed model along with a simulation study, which designates its paramountcy in providing low-complex and low-overhead-based solution with a detection rate higher than 95% and accuracy as high as 99.9%. The proposed approach increases the probability of selecting a correct drone by 81.71% even in the case of a high number of failures.

**INDEX TERMS** Behavior modeling, drones, HCAPN, IIoT, vulnerability assessment.

## I. INTRODUCTION

Industrial Internet of Things (IIoT) aims at connecting a large number of devices and sensors together while leveraging on the existing IoT technologies. IIoT facilitates the mutual exchange of information between the users and the industrial equipment for service provisioning and cost-effective solutions to user problems [1]. IIoT can be used for supporting different kinds of applications such as smart farming, smart city, smart parking systems, smart houses and smart factory [2]–[4]. All these applications focus on high-quality links between the user-side IoT, sensors and intermediate service providers, and demand reliability and security against different types of cyber attacks [5]–[9].

These networks are driven by communication protocols and policies for secure and effective transmissions. With the modernization of the network technologies, IIoT has also paved a way for using drones as one of the supporting entities for communication between the devices. Drones provide extensible support for connectivity through their on-demand links, which can be configured and operated as per the requirements of the networks [10]–[12]. One of the exemplary illustrations of such a scenario is shown in Fig. 1. Drones not only reduce the complexity of connectivity but also manage the load through effective resource allocation strategies [13]–[15].

In spite of huge market gains and technical support, there are certain issues related to the use of drones in the IIoT environment. From physical layer security to certification of devices, every component in such an environment requires continuous monitoring and tracking for potential vulnera-

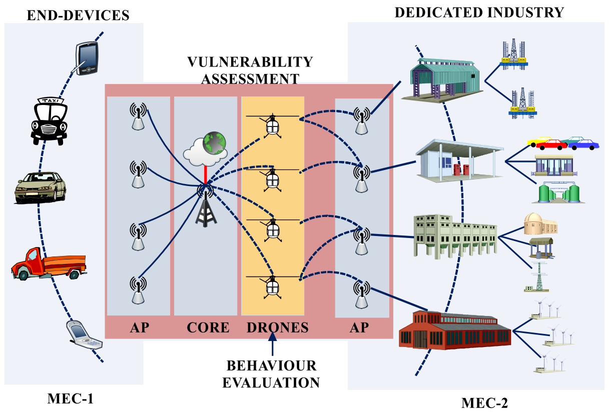


**FIGURE 1.** An illustration of the network scenario and problem statement.

bilities [16]. It is desired to have secure architecture and communication protocols which can cover all the necessary equipment and the drone enabled system without any alterations. Securing drone to drone links and drone to infrastructure links are other requirements of drone-enabled IIoT. The approaches emphasizing on such setups must provide a trustworthy communication while supporting on-demand and low-complex vulnerability assessment. Reliability of network entities and detection of changes in their features are the two other aspects of secure and efficient drone-enabled IIoT [17]. Besides, there is a need for an efficient and scalable solution which can form on-demand intrusion detection systems by simplification of its principles into identification rules.

The existing studies [17]–[22] have highlighted the requirements of behavior and vulnerability assessment to form a reliable and safe network. However, the majority of the existing solutions, which focus on the behavioral aspect of drones, fail to consider security policies which are must for vulnerability identification. In addition, there is a lack of threat modeling, dynamic-adaptation to failures and evaluation criteria in the existing studies, which needs to be resolved through effective strategies. State-wise evaluations are other missing aspects of the existing solutions, and the ones which focus on such a requirement are unable to resolve dependencies at low-complexity. Type of network, its metrics and the parameter of evaluation also play a crucial role in behavioral and vulnerability assessment of networked drones, however, the existing solutions fixate on limiting factors such as false positives, false negatives and communication links, which are certainly not enough for the evaluation of large-scale cyber-physical systems. Even the closely related study in [23], which primarily focuses on similar requirements, is unable to handle the complexities associated with the communication aspects of drones.

This paper considers the drone-enabled communication between the user-side IoT and IIoT for direct handling of the user requests by the intended equipment of a particular industry. In order to provide trustworthy and secure communications, this article aims at assessing the behaviour of drones and dynamically identifying any potential vulnerability, which may lead to critical attacks in near future. The proposed approach uses N-layered Hierarchical Context-Aware Aspect-Oriented Petri Net (HCAPN) modeling which helps in formally verifying the drone-enabled IIoT through an easy to deploy strategy. The proposed solution supports the dynamic assessment of drone behaviour as well as the implementation of the security policies to identify any potential threats amongst the drones.

The rest of the article is structured as follows: Section II provides problem statement and highlights of our contribution. Section III gives an insight of the related works. Section IV defines the network setup and conditions for behaviour modeling. Section V presents the proposed approach along with the novel HCAPN modeling for behaviour and vulnerability assessment. Section VI provides performance evaluations. Finally, Section VII concludes the article with a future scope.

## II. PROBLEM STATEMENT AND OUR CONTRIBUTION

Information in IIoT is sensitive and relies on the strength of connectivity among the end-users, the core and the IoT devices associated with each industry. The networks which are primarily supported by drones for connectivity with the core must ensure that each of the existing connections is free from any vulnerability. Further, it is required that the network devices are able to distinguish between the behaviors of drones and should be able to identify the legitimate aerial nodes for transmissions. Failure to do so may result in different types of threats, which may expose the operations of the entire network. Apart from these major assessments, the other crucial challenge is the selection of an approach, which can help to identify bad and good behaviour aerial nodes and also identify any occurrences of vulnerabilities and threats. In addition, the approach should also be able to verify its own correctness to make sure that its own operations are unaffected from the insider threats, which may violate its rule for falsification of the procedures used for evaluations. These three major concerns can be further mapped to the following requirements:

- R1: The approach should be able to identify whether an associated drone is legitimate or not.
- R2: The approach should be scalable and should be able to handle the additional load as well as support for multiple drones.
- R3: The vulnerability assessment should be predictive and must ensure a high degree of accuracy while estimating the results.
- R4: The approach should be able to justify the selection of drones with a control on behaviour monitoring.
- R5: The operations of the approach must follow low-complexity mechanism while monitoring the behaviour and assessing the potential vulnerabilities in the network.
- R6: The anomaly drone should be marked and this information should be shared with all.

**TABLE 1. A comparative evaluation of existing works.**

| Approach | Parameters | UAVs | Behaviour Monitoring | Vulnerability Assessment | Requirements |
|---|---|---|---|---|---|
| Adaptive IDS [17] | False negative rate, False positive rate | Yes | Yes | Yes | R1, R2, R3, R4, R6 |
| Safety assessment [24] | Percentage of failures, Firing transition rates of UAV | Yes | Yes | No | R2, R4, R7' (no self-evaluation) |
| Risk assessment model [25] | Communication links, Fail-safe states | Yes | No | Yes | R6, R7' (no self-evaluation) |
| Vulnerability assessment method [26] | Path search, Path cost, Loss flow | No | No | Yes | R3, R7' (no self-evaluation) |
| Mission-aware vulnerability assessment framework [27] | Attack paths, Mitigation plans | No | No | Yes | R1, R2, R6 |
| Behaviour modeling [22] | Predictability | Yes | Yes | No | R4, R7' (no self-evaluation) |
| UAVs monitoring system [23] | Detection rate, Wind effect on the waypoints and behaviour | Yes | Yes | Yes | R1, R2, R4, R6 |

- R7: The approach should also provide a state-wise verification of its procedures and identification process.

All of the above requirements are attainable through the proposed solution based on HCAPN modeling.

## III. RELATED WORKS

There have been limited works [17]–[22], to the best of authors' knowledge, on the behaviour and vulnerability assessment of drones-enabled networks especially focusing an IIoT environment. Despite that, there are certain solutions which can be used for such a requirement as discussed in Table 1.

Birnbaum *et al.* [23] proposed a monitoring system for UAVs. The authors used behaviour profiling to find behavioral anomaly by tracking the real-time behaviour and flight of multiple UAVs. The captured behaviors are compared with the initial behavioral model. This approach is based on the critical matching and derivation of behaviour rules, which can affect the operations if mapped incorrectly. Further, the model is highly complex and its implementation to the different application area, especially communication setups, is a concern. Rodriguez-Fernandez *et al.* [22] emphasized the behavioral modeling in UAV operation with the help of Double Chain Markov Models (DCMMs). This system includes process flow, predictability, and interpretability through DCMMs. The authors combine two higher order Markov chains into the same model to improve the predictive capabilities of the system.

Mitchell and Chen [17] proposed a specification-based IDS based on the behaviour rules. This scheme operates to find the attacker-type on the basis of false negative rates and false positive rates. In this scheme, the behaviour rule sets are derived from the UAV threat model. Although this is an effective solution, it fails to support the dynamic evaluations because of high resultant states. In addition, the approach operates on the strict satisfaction of all the rules, which sometimes are the adjustments of the drones during their flight. Such evaluations may result in ambiguous results.

Wang *et al.* [27] proposed a framework for mission-aware vulnerability assessment of the Cyber-Physical Systems (CPS). This framework provides profiling relationships among all CPS components by using a bottom-up approach. From their work, it is evident that the discovery of mission-critical components is a massive challenge to resolve in the cyber-physical world. The vulnerability assessment of UAVs is a complex task and it consists of the identification of threats to the mission. Hartmann and Steup [25] proposed an approach for a UAV specific risk assessment based on the component model of UAVs. The UAV components- communication system, data storage and sensor system are analyzed on the basis of existing vulnerabilities.

Wang *et al.* [26] proposed a vulnerability assessment method for IIoT. The proposed method relies on the formation of an attack graph. The attack risks are measured by a vulnerability scoring system. This method contributes to the vulnerability assessment model, vulnerability quantification method and a vulnerability algorithm based on maximum loss stream. In addition, the authors focused on the factors influencing the attack behaviour and relationship between network nodes. It is concluded that critical failures are responsible for the cancellation of the flight mission or an emergency landing.

Gonçalves *et al.* [24] formulated a safety assessment model for UAV by using Petri Nets (PNs). This model provides better reliability and safety of UAV operations and protection under failure conditions. However, it does not consider the vulnerability and security assessments, which are crucial for secure operations of IIoT.

## IV. NETWORK SETUP

The proposed behaviour and vulnerability assessment solution supports IIoT that uses drones as its key communicating entity. Fig. 1 shows an exemplary illustration of the network scenario and requirements of more drones for load balancing, and Table 2 gives the summary of notations used in the system modeling. The network comprises a core which serves two sets of Mobile Edge Computing (MEC) (MEC-1 and MEC-2). MEC-1 includes the end-users, while MEC-2 serves the IIoT. The network components include a set of end-devices in MEC-1 denoted by a set $\mathcal{E}$. Sets of Access Points (APs), denoted by $\mathcal{A}_1$ and $\mathcal{A}_2$, support connectivity to end-user and IIoT, respectively. The network uses multiple drones, denoted by a set $\mathcal{D}$, for connectivity between the core and $\mathcal{A}_2$. The devices beyond $\mathcal{A}_2$, which operate in the periphery of an industry are collectively considered through a set $\mathcal{C}$. The key aspect of the work proposed in this paper is to analyze the behaviour of drones and to ensure that passes between the elements of set $\mathcal{D}$ and the elements of set $\mathcal{A}_2$ are free from vulnerabilities and potential threats.[1] Optionally, $\mathcal{D}$ can be divided into two sets, $\mathcal{D}_1$ and $\mathcal{D}_2$ for user-side and industry

[1] This paper assumes that the drones are sufficiently equipped for handling heavy payload for connecting core to the APs.

**TABLE 2.** Summary of symbols and their descriptions.

| Symbol | Description |
|---|---|
| $N$ | Number of layered PNs |
| $\mathcal{G}$ | Antenna characteristics |
| $\tau_{AB}$ | Absent duration |
| $\tau_O$ | Active time after the occurrence of failure |
| $\mathcal{A}_m$ | Area under Maneuvering |
| $\mathcal{E}_a^{(t)}, \mathcal{M}_a^{(t)}$ | Available values of energy and memory |
| $\lambda$ | Average number of failures for each subset |
| $\mathcal{P}_{avg}$ | Average probability of selecting a correct drone |
| $\mathcal{B}$ | Behaviour index |
| $\mathcal{T} = \langle \kappa, \mathcal{B}, \mathcal{D}_{\mathcal{B}\kappa} \rangle$ | Behaviour Triplet |
| $\psi$ | Bernoulli constant |
| $Z$ | Channel coefficient |
| $\mathcal{D}_{\mathcal{B}\kappa}$ | Decision marked by the system |
| $\Theta$ | Direction of movement |
| $\mathcal{E}_r^{(t)}, \mathcal{M}_r^{(t)}$ | Energy and memory requirements of the network |
| $\tau_{NF}$ | Flyby time of the new drone to the desired location |
| $H$ | Flying altitude |
| $P_r$ | Initial power received at distance $d_0$ |
| $\mathcal{R}_o$ | Initial resources before the occurrence of a failure |
| $\frac{1}{\gamma}$ | Mean lifetime of the drone |
| $N_o$ | Noise |
| $\mathcal{F}$ | Number of behaviour conditions |
| $\mathcal{W}$ | Number of channels |
| $\lvert\mathcal{D}'\rvert$ | Number of failed drones |
| $\mathcal{F}_x$ | Number of failures in $x$th subset |
| $n$ | Number of service switchings |
| $s$ | Number of subsets |
| $\mathcal{V}$ | Number of vertices |
| $C_L$ | Observational value of a behaviour at the $L$th state |
| $\frac{\vartheta}{\mu}$ | Offered rate/Attainable rate |
| $\mathcal{O}_R$ | Offloading delay |
| $\mathcal{C}_{R,L}^{(1)}, \mathcal{C}_{R,L}^{(2)}$ | Choice outputs |
| $\delta$ | Path loss exponent |
| $\delta_{LoS}, \delta_{NLoS}$ | Path loss exponents |
| $\mathcal{R}_{ps}$ | Per second resource consumption |
| $\mathcal{P}_{1,L-1}, \mathcal{P}_{2,L-1}$ | Priori probabilities for L number of states |
| $\epsilon$ | Random variables |
| $\rho_R$ | Received power |
| $\mathcal{R}_p$ | Resource depletion rate |
| $\tau_R$ | Response time |
| $\kappa$ | Responsiveness of the system w.r.t. behaviour rules |
| $\mathcal{D}$ | Set of drones |
| $\mathcal{E}$ | Set of end-devices in MEC-1 |
| $\mathcal{C}$ | Set of Industrial devices |
| $\mathcal{A}_1, \mathcal{A}_2$ | Sets of APs |
| $v$ | Speed |
| $\beta$ | System bandwidth |
| $\eta_x$ | The number of drones in the $x$th subset |
| $r$ | The radius of the area under consideration |
| $\tau_m$ | The time consumed in performing a maneuver |
| $\tau_F$ | Time after which a failure is encountered |
| $\tau_{LoS}$ | Time consumed in attaining LoS |
| $\tau_{regain}$ | Time consumed to regain the connectivity |
| $\tau_{SW}$ | Time for service switching |
| $\tau_{\mathcal{A}}$ | Time taken by a drone to cover an assigned area |
| $\mathcal{C}_T$ | Total conditions |
| $t$ | Total instances |
| $\mathcal{T}_p$ | Transmission power |
| $\mathcal{C}_{US}$ | Unfavourable conditions |

side connectivity, respectively. These services help to connect users to the dedicated industry. The model can be formulated in the selection of an available drone considering that out of $|\mathcal{D}|$ number of drones, $|\mathcal{D}'|$ number of drones may fail. The system is further extended such that if there is a $s$ number of subsets, then each subset can have at least one or more drone failures. For this, let $\eta_x$ be the number of drones in the $x$th subset, such that $\sum_{x=1}^{s} \eta_x = |\mathcal{D}|$, and $\lambda$ be the average number of failures for each subset, such that $\lambda = \frac{1}{s} \sum_{x=1}^{s} \mathcal{F}_x$, where $\mathcal{F}_x$ is the number of failures in $x$th subset. From this, the number of subsets without failure can be calculated as $1 - \frac{\lambda}{\sum_{x=1}^{s} \eta_x}$. This entire condition can be obtained over the Bernoulli constant $\psi$, which is the number of failures below which the network cannot sustain such that the probability that a defected or failed drone gets selected at a given instance is expressed as:

$$P(\eta, \lambda, \psi) = \frac{\eta!}{\psi!\,(\eta - \psi)!} \left(\frac{\lambda}{\eta}\right)^{\psi} \left(1 - \frac{\lambda}{\eta}\right)^{\eta - \psi},$$
$$\eta = \sum_{x=1}^{s} \eta_x. \tag{1}$$

For the total instances, $t$, the average probability of selecting a correct drone can be calculated as:

$$\mathcal{P}_{avg} = 1 - \frac{1}{t} \sum_{i=1}^{t} P(\eta, \lambda, \psi). \tag{2}$$

This is the observed value, however, the prediction using it is difficult to attain as the failures are unpredictable and discrete events. In addition, the predictions can be performed by continuously observing the system. Let $\tau_{\mathcal{A}}$ be the time taken by the drone to cover an assigned area and it is assumed that the signal quality remains unaffected in the assigned zone for each drone. Now, considering such a deployment, the failure can lead to three possibilities. First is that the traffic will be shifted to neighboring drone, which has to cover multiple areas that may affect the quality of the signal. The second includes sending more drones to a place that can result in overheads due to waiting, and the third one can be the combination of first and second perspectives, which may cause lesser overheads. These can be understood through the following formulations. Let $\tau_F$ be the time after which a failure is encountered, then for shifting the traffic to a neighboring drone as per the first condition, the time after which the connectivity is regained is given as:

$$\tau_{connect} = \tau_F + \tau_{LoS} + \tau_{SW}, \tag{3}$$

and the useful time is calculated as:

$$\tau_{useful} = \tau_F + \tau_O - \underbrace{(\tau_{LoS} + \tau_{SW})}_{\text{positioning overheads}}, \tag{4}$$

where $\tau_{LoS}$ is the time consumed in attaining Line of Sight (LoS), $\tau_{SW}$ is the time for service switching, and $\tau_O$ is the active time after the occurrence of failure. Now, for the second condition, (3) and (4) changes to

$$\tau_{connect} = \tau_F + \tau_{NF} + \tau_{SW}, \tag{5}$$

and

$$\tau_{useful} = \tau_F + \tau_O - \underbrace{(\tau_{NF} + \tau_{SW})}_{\text{allocation overheads}}, \tag{6}$$

where $\tau_{NF}$ is the flyby time of the new drone to the desired location. The applicability of the third case is subject to a condition of the time difference, i.e., if $\tau_{NF} >> \tau_{LoS}$, then the third case will be operated by selecting the first and second case in a sequential manner. By the time, a new drone arrives at the desired location, the load is balanced with the help of existing drones, and the services can be shifted with lesser delays. The only overhead in this scenario is because of the dual service switching, which is affordable compared with the excessive waiting time. However, $\tau_{NF} << \tau_{LoS}$, then only option 2 is followed provided that delay due to $\tau_{NF}$ is at a minimum.

In any of the above case, let $\tau_{regain}$ be the time to regain the connectivity, such that

$$\tau_{regain} = \tau_{LoS} | \tau_{NF} + n\tau_{SW}, \tag{7}$$

where $n$ is the number of times the services are switched while reconnecting the drones. Following this, it can be identified that a certain amount of resources deplete during this regaining period, which can be calculated as:

$$\mathcal{R}_p = \mathcal{R}_o e^{-\gamma \tau_{regain}}, \tag{8}$$

where $\mathcal{R}_o$ is the initial resources before the occurrence of a failure, and $\frac{1}{\gamma}$ is the mean lifetime of the drone with the given resources and consumption rate. If the network has to attain a rate of $\frac{\vartheta}{\mu}$ at all the instances, such that per second resource consumption is given by $\mathcal{R}_{ps}$, then for entire duration, $\mathcal{R}_p \geq \mathcal{R}_{ps}$ at a given $t$, where $t \leq \tau_{regain}$. If this condition remains unsatisfied, the network is at failure as it cannot afford the desired rate for the given amount of resources.

It is assumed that the core part of the network has a significant amount of energy and memory for the entire duration of the network, whereas the drones may be isolated because of high depletion rate. Thus, it becomes necessary to consider the energy and memory requirements of the network and at any given instance,

$$\mathcal{E}_r^{(t)} \not< \mathcal{E}_a^{(t)}, \text{and } \mathcal{M}_r^{(t)} \not< \mathcal{M}_a^{(t)}, \tag{9}$$

where $\mathcal{E}_r^{(t)}$ and $\mathcal{M}_r^{(t)}$ are the energy and the memory requirements of the network, and $\mathcal{E}_a^{(t)}$ and $\mathcal{M}_a^{(t)}$ are their available values, respectively. For the previously given conditions on $\mathcal{R}_p$, the variables can be replaced by formulations in (9) for making it an energy or memory dominant system. Along with these conditions, the network is subject to deviation constraints, according to which:

$$\min \left( \sqrt{\frac{1}{\alpha_1}\sum_{i=1}^{\alpha_1} (\mathcal{K}_i - \overline{\mathcal{K}})^2} - \sqrt{\frac{1}{\alpha_2}\sum_{i=1}^{\alpha_2} (\mathcal{K}_i - \overline{\mathcal{K}})^2} \right), \quad \alpha_1 \leq \tau_{useful}, \alpha_2 \leq \tau_{regain}, \tag{10}$$

where $\mathcal{K}$ can either be operated for $\mathcal{E}_r$ or $\mathcal{M}_r$ or both. The above deviation helps to understand resource consumption behaviour of the drones as well as the network.

The behaviour modeling of drones in the given IIoT setup is subject to different metrics which play a crucial role in sustaining communications between the end users and its dedicated responsible industry. In this paper, the behaviour modeling is mathematically derived by providing governing conditions for a total of fifteen metrics, namely, Area under maneuvering $\mathcal{A}_m$, $|\mathcal{D}|$, location $(x, y, z)$ w.r.t. to a reference (0,0,0), Signal-to-Interference-plus-Noise Ratio (SINR), received power ($\rho_R$), direction of movement ($\Theta$) w.r.t. reference point, speed ($v$), $\mathcal{R}_p$, $\tau_{\mathcal{A}}$, $\mathcal{E}_r$, $\mathcal{M}_r$, response time ($\tau_R$), offloading delay ($\mathcal{O}_R$), flying altitude ($H$), and absent duration ($\tau_{AB}$). In the given system model, the interference observed by a $j$th drone from $\eta - 1$ drones and other communicating network entities can be expressed as:

$$SINR = \frac{\mathcal{T}_p \mathcal{G} H^{-\delta}}{\sum_{i=1, i \neq j}^{\eta} \sum_{k=1}^{|\mathcal{A}_1|+|\mathcal{A}_2|} \sum_{q=1}^{|\mathcal{E}|+|\mathcal{C}|} \mathcal{T}_p \mathcal{G} H^{-\delta} + N_o}, \tag{11}$$

where $\mathcal{T}_p$ is the transmission power, $\mathcal{G}$ denotes the antenna characteristics, $\delta$ is the path loss exponent, and $N_o$ is the noise. The model can be extended for LoS and NLoS (Non-Line of Sight) mode by considering a separate path loss exponent for each of them denoted by $\delta_{LoS}$ and $\delta_{NLoS}$. Using these, the received power for a $j$th drone for its distance from a defined point of reference can be calculated as:

$$\rho_R^{(LoS)} = P_r^{(LoS)} - 10\delta_{LoS} \log\left(\frac{H_j}{d_o}\right) + \epsilon_j^{(LoS)}, \tag{12}$$

and

$$\rho_R^{(NLoS)} = P_r^{(NLoS)} - 10\delta_{NLoS} \log\left(\frac{H_j}{d_o}\right) + \epsilon_j^{(NLoS)}, \tag{13}$$

where $P_r^{(NLoS)}$ and $P_r^{(NLoS)}$ are the initial powers received at a distance $d_o$ w.r.t. reference point, and $\epsilon^{(LoS)}$ and $\epsilon^{(NLoS)}$ are the random variables for received signal strength in LoS and NLoS scenarios, respectively. Note that $\mathcal{O}_R$ is similar to the time spent in switching the services across the drones, and can be expressed as the offloading delay [28]. However, it is the responsibility of the system to make sure that offloading delays are well below the limits. In the given network, offloading delay [28] can be expressed as:

$$\mathcal{O}_R = \frac{\mathcal{W}}{\beta \log\left(1 + Z\, \mathcal{T}_p\right)}, \tag{14}$$

where $\mathcal{W}$ is the number of channels, $Z$ is the channel coefficient, and $\beta$ is the system bandwidth. These network evaluations and the previously described metrics are used to formulate behaviour problem by considering the constraints in Table 3. In this table, $\tau_m$ is the time consumed in performing a maneuver and $r$ is the radius of the area under consideration.

The above conditions for the behaviour assessments are used in making policies and definitions. These formulations

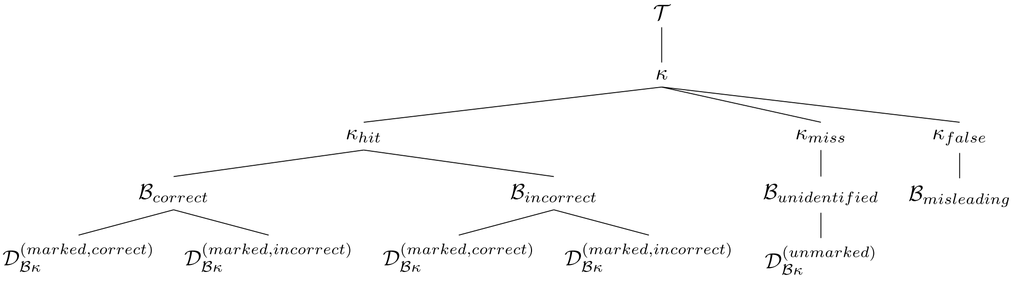

$\mathcal{T}$
$\kappa$
$\kappa_{hit}$
$\kappa_{miss}$
$\kappa_{false}$
$\mathcal{B}_{correct}$
$\mathcal{B}_{incorrect}$
$\mathcal{B}_{unidentified}$
$\mathcal{B}_{misleading}$
$\mathcal{D}_{\mathcal{B}\kappa}^{(marked,correct)}$
$\mathcal{D}_{\mathcal{B}\kappa}^{(marked,incorrect)}$
$\mathcal{D}_{\mathcal{B}\kappa}^{(marked,correct)}$
$\mathcal{D}_{\mathcal{B}\kappa}^{(marked,incorrect)}$
$\mathcal{D}_{\mathcal{B}\kappa}^{(unmarked)}$

**FIGURE 2.** Behaviour Triplet ($\mathcal{T} = \langle\kappa, \mathcal{B}, \mathcal{D}_{\mathcal{B}\kappa}\rangle$) hierarchy.

**TABLE 3.** Conditions for behavioral modeling.

| Sr. No. | Parameter | Conditions |
|---|---|---|
| $P'1$ | $\mathcal{A}_m$ | $(x, y, z)$ *lies in Assigned* $(\mathcal{A}_m)$ |
| $P'2$ | $\lvert\mathcal{D}\rvert$ | $Adjacency(D)_{[][]}^{(j)} \neq NULL$ |
|  | $\lvert\mathcal{D}\rvert$ | $Incidence(D)_{[][]}^{(j)} \neq NULL$ |
| $P'3$ | $(x, y, z)$ | $(x, y, z) \leq \max(x, y, z)$ |
| $P'4$ | $SINR$ | $SINR \geq mean(SINR)$ |
|  | $SINR$ | $SINR \geq SINR', SINR' = SINR\ at\ \frac{\vartheta}{\mu}$ |
| $P'5$ | $\rho_R$ | $\rho_R \geq \min(\rho_R)$ |
| $P'6$ | $\Theta$ | $\Theta = \pm 45\,\text{deg}$ |
| $P'7$ | $v$ | $v \leq permissible(v)$ |
| $P'8$ | $\mathcal{R}_p$ | $\frac{1}{\gamma} \geq \min(\frac{1}{\gamma})$ |
| $P'9$ | $\tau_{\mathcal{A}}$ | $\tau_{\mathcal{A}} \leq (\frac{2\pi r}{v} + \tau_m)$ |
| $P'10$ | $\mathcal{E}_r$ | $\mathcal{E}_r^{(t)} \nless \mathcal{E}_a^{(t)}$ |
| $P'11$ | $\mathcal{M}_r$ | $\mathcal{M}_r^{(t)} \nless \mathcal{M}_a^{(t)}$ |
| $P'12$ | $\tau_R$ | $\tau_R \leq \tau_{NF}$ |
| $P'13$ | $\mathcal{O}_R$ | $\mathcal{O}_R \leq (\tau_{NF} + \tau_{SW})$ |
| $P'14$ | $H$ | $H \leq permissible(H)$ |
| $P'15$ | $\tau_{AB}$ | $\tau_{AB} \leq (\tau_{NF} + \tau_{SW})$ |

are inspired by detection theory [29] with a modification that identification of correct behaviour is marked as hit irrespective of the nature of detection. Formally, this can be understood from the following definition:

*Definition 1:* The behaviour of a system can be expressed as a triplet $\mathcal{T} = \langle\kappa, \mathcal{B}, \mathcal{D}_{\mathcal{B}\kappa}\rangle$, where $\kappa$ is the responsiveness of the system w.r.t. behaviour rules, $\mathcal{B}$ is the behaviour index, and $\mathcal{D}_{\mathcal{B}\kappa}$ is the decision marked by the system on the given conditions.

*Explanation:* The above definition can be explained by considering four outputs for the behaviour index, $\mathcal{B}_{correct}$, $\mathcal{B}_{incorrect}$, $\mathcal{B}_{misleading}$, and $\mathcal{B}_{unidentified}$, where $\mathcal{B}_{correct}$ is marked if the rules suggest that a given entity is working appropriately, $\mathcal{B}_{incorrect}$ is marked if the entity is working inappropriately as per the rules. Note that both these indexes are legitimate outputs and must be treated as "Hit" if compared with detection theory. $\mathcal{B}_{misleading}$ is similar to the false alarms of detection theory and $\mathcal{B}_{unidentified}$ is the failure to decide on the basis of given rules. These behaviour indexes are identified for three values of $\kappa$- $\kappa_{hit}$, $\kappa_{miss}$, and $\kappa_{false}$, such that $\kappa_{hit} = \{\mathcal{B}_{correct}, \mathcal{B}_{incorrect}\}$, $\kappa_{miss} = \{\mathcal{B}_{unidentified}\}$, $\kappa_{false} = \{\mathcal{B}_{misleading}\}$. Based on these, the triplet is completed by taking a decision for each of these mappings, such that three observations are made for $\mathcal{D}_{\mathcal{B}\kappa}$- $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,correct)}$, $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,incorrect)}$, and $\mathcal{D}_{\mathcal{B}\kappa}^{(unmarked)}$. $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,correct)}$ and $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,incorrect)}$ show whether the given entity behaves correctly or incorrectly, and $\mathcal{D}_{\mathcal{B}\kappa}^{(unmarked)}$ shows that the system misses to decide on the basis of given rules. The details of the defined triplet can be followed from Fig. 2. Further, the false alarms are ignored by these decision variables and treated as unmarked for the final output. Now, on the basis of $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,correct)}$and $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,incorrect)}$, the system can be modeled into time-based hypothetical Maximum a Posteriori Testing (MAP) based on detection theory [29]. For this, let $\mathcal{P}_{1,L-1}$ and $\mathcal{P}_{2,L-1}$ be the priori probabilities for $L$ number of states denoting outcomes, $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,correct)}$and $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,incorrect)}$, respectively, while operational for the total useful time, where

$$\mathcal{P}_{1,L-1} = \frac{\mathcal{C}_{S,(L-1)}}{\mathcal{C}_T}, \tag{15}$$

and

$$\mathcal{P}_{2,L-1} = \frac{\mathcal{C}_{US,(L-1)}}{\mathcal{C}_T}. \tag{16}$$

Here, $\mathcal{C}_S$ denotes the behavioral conditions that are in favor of performance, $\mathcal{C}_{US}$ denotes the unfavorable conditions, $\mathcal{C}_T$ are the total conditions. The value of these metrics are obtained from Table 3. Now, from the definition of MAP,

$$\mathcal{C}_{R,L}^{(1)} = \frac{\mathcal{P}\left(C_L|\mathcal{C}_S\right) \cdot \mathcal{P}_{1,L-1}}{\mathcal{P}\left(C_L|\mathcal{C}_S\right) \cdot \mathcal{P}_{1,L-1} + \mathcal{P}\left(C_L|\mathcal{C}_{US}\right) \cdot \mathcal{P}_{2,L-1}}, \tag{17}$$

and

$$\mathcal{C}_{R,L}^{(2)} = \frac{\mathcal{P}\left(C_L|\mathcal{C}_{US}\right) \cdot \mathcal{P}_{2,L-1}}{\mathcal{P}\left(C_L|\mathcal{C}_S\right) \cdot \mathcal{P}_{1,L-1} + \mathcal{P}\left(C_L|\mathcal{C}_{US}\right) . \cdot \mathcal{P}_{2,L-1}}, \tag{18}$$

where $\mathcal{C}_{R,L}^{(1)}$ and $\mathcal{C}_{R,L}^{(2)}$ are the outputs to make a choice between $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,correct)}$and $\mathcal{D}_{\mathcal{B}\kappa}^{(marked,incorrect)}$ depending whether $\mathcal{C}_{R,L}^{(1)}$ ($>$ or $\leq$) $\mathcal{C}_{R,L}^{(2)}$. Here, $C_L$ is the observational value of a behaviour at the $L$th state and can be operated over the probability $\frac{\mathcal{C}_{US,L}}{\mathcal{C}_T}$. In general, the condition $\mathcal{C}$ on the given parameters can be written as $\big(\mathcal{A}_m^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge |\mathcal{D}|^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge (x, y, z)^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge SINR^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge \rho_R^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge \Theta^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge v^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge \mathcal{R}_p^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge \tau_{\mathcal{A}}^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge \mathcal{E}_r^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge \mathcal{M}_r^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge \tau_R^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge \mathcal{O}_R^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge H^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \wedge \tau_{AB}^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}}\big)$ for all

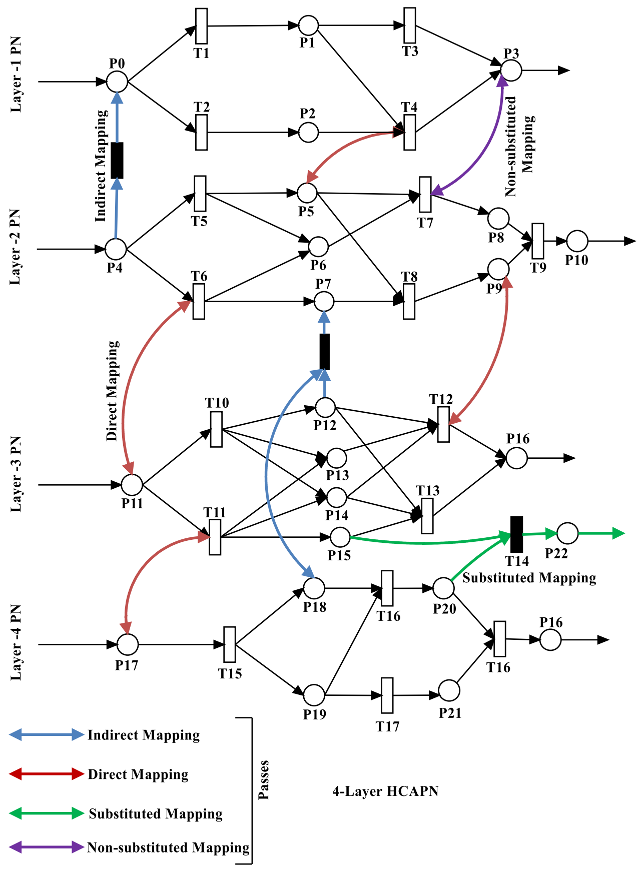


**FIGURE 3.** An illustration of a 4-layer HCAPN with passes.

mandatory and $\big(\mathcal{A}_m^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee |\mathcal{D}|^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee (x, y, z)^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee SINR^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \rho_R^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \Theta^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \nu^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \mathcal{R}_p^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \tau_{\mathcal{A}}^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \mathcal{E}_r^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \mathcal{M}_r^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \tau_R^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \mathcal{O}_R^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee H^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}} \vee \tau_{AB}^{(\mathcal{D}_{\mathcal{B}\kappa})^{\mathcal{O}'}}\big)$ for limited parameters. Here, $\mathcal{O}'$ denotes the output for the given behaviour, i.e., marked-correct or marked-incorrect. The above modeling can be used for checking false positives or false negatives and difficulty in judgment can be plotted to understand the behaviour of the system.

## V. PROPOSED APPROACH

This paper uses the behaviour modeling presented in the previous section for defining the operational rules which help to determine the operational capabilities of a system, especially drones-enabled IIoT. The proposed approach is based on the formation of an N-layered HCAPN which is inspired by Aspect-Oriented PNs, Hierarchical PNs and Cultured PNs [30]–[34], and initially presented as an ideology for secure localization in Sharma *et al.* [35].[2] This paper considers an altogether different definition of "hierarchy", which is discussed in the next section.

### A. HIERARCHICAL CONTEXT-AWARE ASPECT-ORIENTED PETRI NETS (HCAPN)

HCAPN functions similar to a regular PN and uses similar rules but with a changed ideology, definition, and workflow. At first, this paper formally introduces the concept of HCAPN followed by its mathematical representations, flow, and verifications. Then, these are used for analyzing the behaviour and assessing the vulnerability of the considered network model. The terminologies and definitions are provided below:

- Hierarchy: In the formulated HCAPN, the hierarchy is defined as overlapping layered architecture attained by merging of the $N$ number of context-aware aspect-oriented PNs into a single PN model. Further, the formulated PN is unbounded in terms of merging but bounded in terms of rules of a general PN. The merging can have a maximum of $\frac{\mathcal{V}}{2}\,(\mathcal{V}-1)$ extra arcs between the places and the transitions, where $\mathcal{V}$ is the total number of vertices (places and transitions) in $N$ number of context-aware aspect-oriented PNs. These extra arcs are termed as passes. Note that existing Hierarchical PNs follow a parent-child model, where a sequential procedure is used for merging and is likely to result in a low-complex structure but with limited applications. However, unbounded passing in the HCAPN provides a high degree of freedom and a wide range of applications that too at low-complexity and lower overheads. Figure 3 presents an exemplary illustration of 4-layer HCAPN with four different types of passes. These passes help to determine the flow as well as manage the connectivity between the PNs by following an unbounded hierarchy. The details of these passes are as follows:
  1) Indirect Mapping: For a given HCAPN, if new transitions or places are introduced while passing contextual information, the type of passing is termed as indirect mapping.
  2) Direct Mapping: For a given HCAPN, if the existing transitions are connected to the existing places and vice versa, the type of passing is termed as direct mapping.
  3) Substituted Mapping: For a given HCAPN, if the merging results in a new place as a formal or informal output, the type of passing is termed as substituted mapping.
  4) Non-substituted Mapping: For a given HCAPN, if the merging is performed with the existing output place, the type of passing is termed as non-substituted mapping.
- Context-Awareness: Contextual awareness is defined on the basis of the number of passes available in an HCAPN, which is a resultant of merging of two or more PNs, and the number of passes that are actually available for usability to get the desired output. The contextual-awareness is achieved based on the inputs from the previous state, place or transition depending on the nature of passes used for modeling a given system. Also, this

[2] The previously presented version discusses the applicability of HCAPN and does not provide any details on its mechanisms and flow.

helps to maintain the state information for the entire PN model. In the proposed HCAPN, the context is passed as information alongside the tokens through parameters used for defining a particular system. In general HCAPN, the contextual-awareness is obtained through the following parameters:

1) Complexity and operational overheads: This refers to keeping a check on the operational cost of the designed HCAPN and managing a particular pass on the basis of the level of complexity it can cause on a particular place or transition. A particular pass is used only for reducing the operational cost of the HCAPN while producing an output.
2) Delay: It is required that a newly included pass should not cause an excessive delay while firing in a particular direction. The delay can be caused due to excessive dependence on the previous entity, which requires many computations for generating an output for a pass.
3) Reliability: It is to be ensured that HCAPN must be reliable in generating aspects for each of the newly generated arcs without causing any redundancy or overlapping. Mutual consents are other factors to be achieved in HCAPN as this helps to attain reliable connectivity between the places and transitions of two or more PNs.
4) Reachability: Inclusion of new passes may cause a loop in the model, which may affect the operations of the entire system. Thus, it is necessary that context-awareness is supported in terms of reachability, which can be observed through the firing of all transitions and non-negative values of tokens in case of time-dependent operations.
5) Dependability: While merging two or more PNs, it is likely to observe a dependency between two or more places or transitions, however, it is required that irrespective of the dependencies, the model should be operable through unbounded control leading to a correct output.
6) Detection rate: The context in HCAPN should also support detection of freshly induced or active passes which can reduce the complexity and operational cost of the system. The detection rate of any place or transition for new rules leading to fresh passes must be high. In addition, the detection of inaccurate context should also be identified in HCAPN with high accuracy and zero tolerance.

- Aspect-Oriented: Aspect-oriented feature in HCAPN is inspired by the properties defined by Xu and Nygard [30]. This feature helps to accommodate the concepts of aspect-oriented programming in the developed unbounded hierarchical PNs. The aspects help to define the roles on the basis of properties and context formed over the available parameters of any system. This helps to maintain the flow of model towards the intended direction while fixating on a place, which will provide the final output.

### *B. DEFINITION AND RULES*

The proposed behaviour and vulnerability assessment approach depends on the accurate formation of HCAPN for detecting any false entity in the network. Thus, it is required to formally introduce the HCAPN and its rules as given below:

*Definition 2:* The developed HCAPN can be defined mathematically as $\mathbb{H} = (\mathbb{P}, \mathbb{T}, \mathbb{A}, \mathbb{Q}, \mathbb{E}, \mathbb{C}, \mathbb{S}, \mathbb{L}, \mathbb{O})$, where $\mathbb{P}$ denotes the set of places, $\mathbb{T}$ denotes the set of transitions, $\mathbb{A}$ denotes the set of arcs between $\mathbb{P}$ and $\mathbb{T}$, $\mathbb{Q}$ is the set of number of passes between layers, $\mathbb{E}$ is the set of the types of passes that connect places and transitions, $\mathbb{C}$ denotes the set of context on each $\mathbb{A}$, $\mathbb{S}$ denotes the set having aspect for each layer, $\mathbb{L}$ denotes the number of layers and the subset $\mathbb{U}_s = (\mathbb{P}, \mathbb{T}, \mathbb{A}, \mathbb{C}, \mathbb{S})$, such that $\mathbb{L} = N$, and $\mathbb{O}$ denotes the set of output places.

#### 1) RULE OF PASSES

The number of passes in HCAPN governs the flow of the model and affects the reachability of the system. An incorrect number of passes may result in a loop, which causes overheads and lots of resources get wasted without producing any desirable output.

**Rule V-B.1.1:** In a given $\mathbb{H} = (\mathbb{P}, \mathbb{T}, \mathbb{A}, \mathbb{Q}, \mathbb{E}, \mathbb{C}, \mathbb{S}, \mathbb{L}, \mathbb{O})$, if $|\mathbb{Q}| \leq |\mathbb{P}| + |\mathbb{T}|$, the number of possible intermediate routes increases by $\frac{\mathcal{V}}{2}(\mathcal{V} - 1)$, which will cause excessive overheads and huge delay in producing an output. Even if only places or transitions are mapped through the passes, i.e. $|\mathbb{Q}| = |\mathbb{P}|$ or $|\mathbb{Q}| = |\mathbb{T}|$, the maximum passes may result in a similar number of conflicts.

**Rule V-B.1.2:** From the definitions and previous remarks, it is clear that the number of passes drives the flow in HCAPN, thus, it is necessary to formulate the number of passes in HCAPN for its smooth operations. The number of passes can be given for bounded and unbounded hierarchy in HCAPN. The number of passes for a bounded HCAPN is given by $\left(\sum |\mathbb{P}_a| - \sum |\mathbb{P}_b| + 1\right)(|\mathbb{L}| - 1)$, where $|\mathbb{P}_a|$ and $|\mathbb{P}_b|$ are the number of places in two joining PNs. The similar formulation can be operated for the number of transitions also. Similarly, the number of passes for unbounded HCAPN is given by $\left(\sum |\mathbb{P}_a| - \sum |\mathbb{P}_b| + 1\right) \frac{|\mathbb{L}|}{2} (|\mathbb{L}| - 1)$.

#### 2) RULE OF PLACES

The number of places in an HCAPN is governed by the number of layers combined to generate a final PN. However, the number of places also affects the output of the HCAPN. Thus, there are certain rules of places which are to be followed for accurate formation of HCAPN as given below:

**Rule V-B.2.1:** The layers can be combined by following any type of passes out of the proposed four combinations. However, HCAPN cannot have any new intermediate places between the combining layers of two or more PNs. Only transitions can be introduced that too for the ‘‘indirect mapping’’

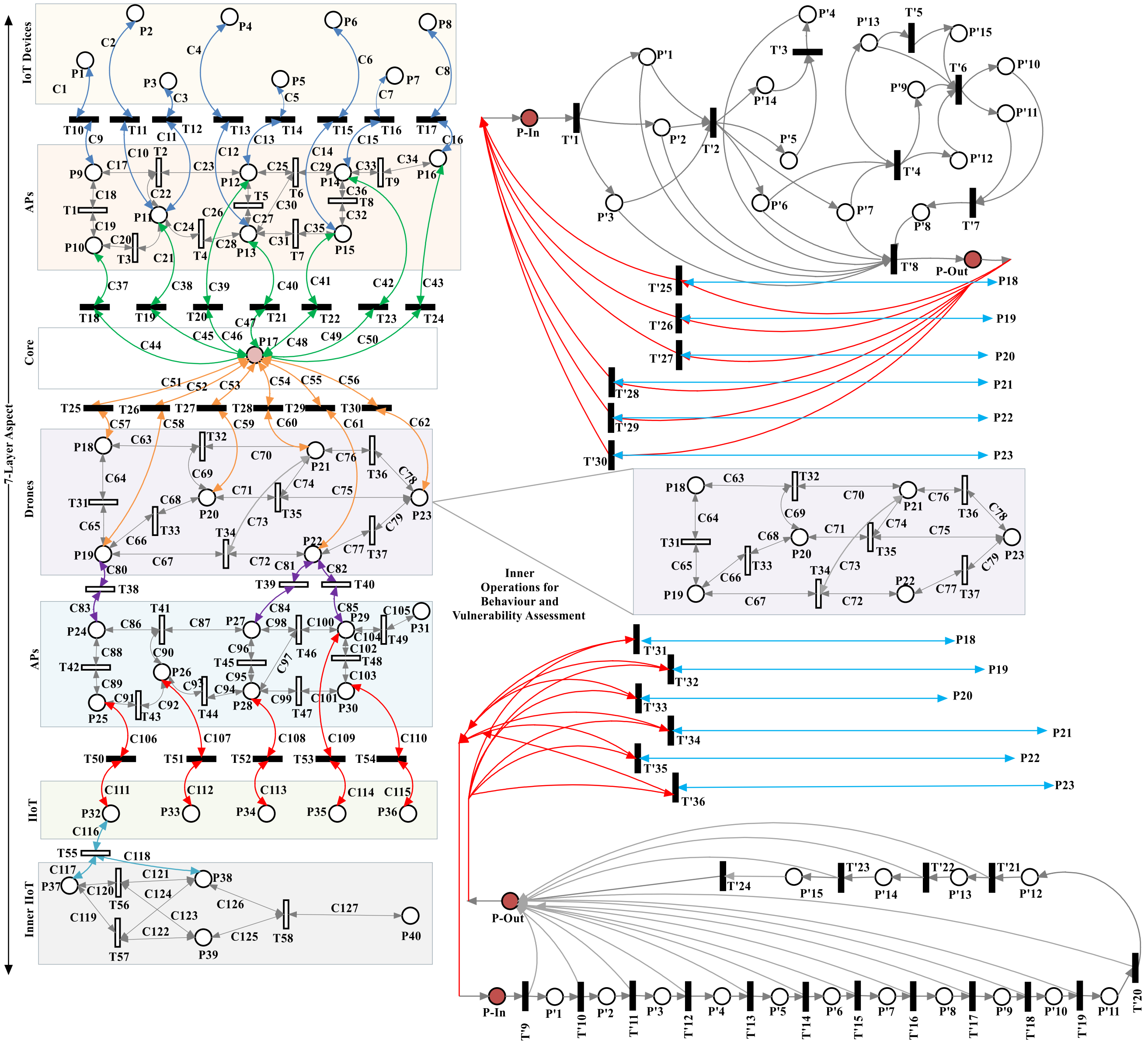


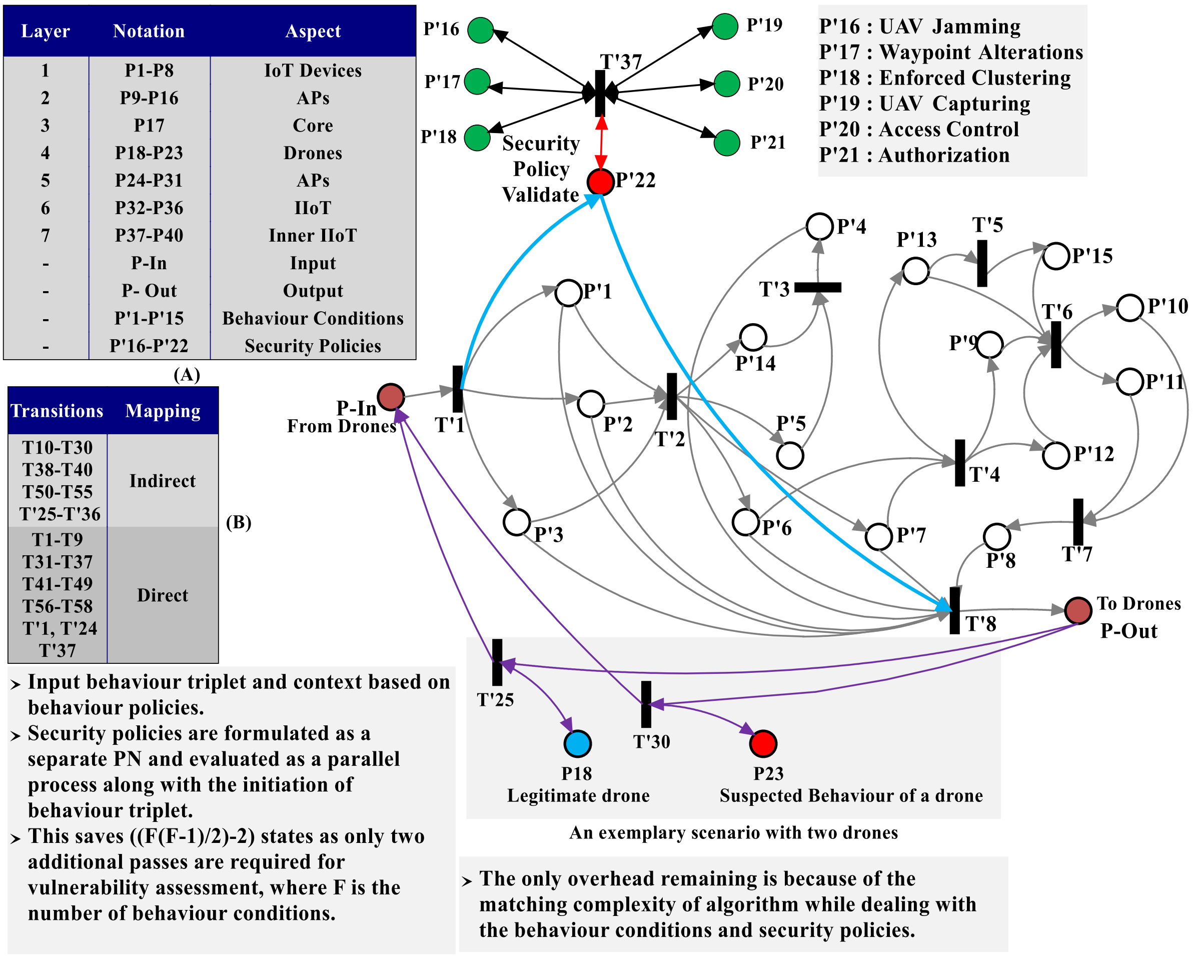
**FIGURE 4.** A complete overview of HCAPN for the defined network model along with its context, aspect and flow during behaviour and vulnerability assessment.

**TABLE 4.** The details of context used by the proposed HCAPN model.

| Arc | Context | Reverse |
|---|---|---|
| $\mathbb{C}1-\mathbb{C}16$ | $< ID, SINR, (x,y,z), v >$ | TRUE |
| $\mathbb{C}17-\mathbb{C}36$ | $< ID, < \mathbb{C}1-\mathbb{C}16 : ID >, SINR, \mathcal{R}_p, \mathcal{E}_r, \mathcal{M}_r, \rho_R, Load >$ | TRUE |
| $\mathbb{C}37-\mathbb{C}50$ | $< ID, < \mathbb{C}17-\mathbb{C}36 : * >, ADJ[\mathbb{C}17-\mathbb{C}36][\mathbb{C}17-\mathbb{C}36], INC[\mathbb{C}17-\mathbb{C}36][\mathbb{C}17-\mathbb{C}36] >$ | TRUE |
| $\mathbb{C}51-\mathbb{C}62$ | $< ID, < \mathbb{C}37-\mathbb{C}50 : ID, ADJ[\mathbb{C}17-\mathbb{C}36][\mathbb{C}17-\mathbb{C}36], INC[\mathbb{C}17-\mathbb{C}36][\mathbb{C}17-\mathbb{C}36] >, \mathcal{A}_m, Load, \theta >$ | TRUE |
| $\mathbb{C}63-\mathbb{C}79$ | $< Table\ 3 >$ | TRUE |
| $\mathbb{C}80-\mathbb{C}85$ | $< ID, < \mathbb{C}63-\mathbb{C}79 : ID, Load, \theta >, (x,y,z), SINR, \mathcal{R}_p >$ | TRUE |
| $\mathbb{C}86-\mathbb{C}105$ | $< ID, < \mathbb{C}1-\mathbb{C}16 : ID >, SINR, \mathcal{R}_p, \mathcal{E}_r, \mathcal{M}_r, \rho_R, Load >$ | TRUE |
| $\mathbb{C}106-\mathbb{C}127$ | $< ID, SINR, (x,y,z), handling\_device\_ID >$ | TRUE |

of passes. This helps to prevent any loop-back failures in the model as well as prevents the failures due to non-reachability and excessive overheads. Thus, except for newly introduced output places, the total number of places and transitions in a final HCAPN is given as $|\mathbb{P}| + |\mathbb{T}| + |\mathbb{T}'|$, where $|\mathbb{T}'|$ is the set of newly introduced transitions for mapping intermediate places of two or more PNs.

**Rule V-B.2.2:** The second condition is applicable for output nodes only and it denotes the number of output places that can be available in the final HCAPN. According to this rule,



| Layer | Notation | Aspect |
|---|---|---|
| 1 | P1-P8 | IoT Devices |
| 2 | P9-P16 | APs |
| 3 | P17 | Core |
| 4 | P18-P23 | Drones |
| 5 | P24-P31 | APs |
| 6 | P32-P36 | IIoT |
| 7 | P37-P40 | Inner IIoT |
| - | P-In | Input |
| - | P- Out | Output |
| - | P'1-P'15 | Behaviour Conditions |
| - | P'16-P'22 | Security Policies |

| Transitions | Mapping |
|---|---|
| T10-T30<br>T38-T40<br>T50-T55<br>T'25-T'36 | Indirect |
| T1-T9<br>T31-T37<br>T41-T49<br>T56-T58<br>T'1, T'24<br>T'37 | Direct |

- Input behaviour triplet and context based on behaviour policies.
- Security policies are formulated as a separate PN and evaluated as a parallel process along with the initiation of behaviour triplet.
- This saves ((F(F-1)/2)-2) states as only two additional passes are required for vulnerability assessment, where F is the number of behaviour conditions.

- The only overhead remaining is because of the matching complexity of algorithm while dealing with the behaviour conditions and security policies.

**FIGURE 5.** A complete overview of HCAPN for behaviour and vulnerability assessment along with the details of transitions, places and security policies.

for bounded HCAPN, the number of output places can be given by $|\mathbb{L}|+(|\mathbb{L}|-1)$. For unbounded HCAPN, the number of output places becomes $|\mathbb{L}| + \frac{|\mathbb{L}|}{2}(|\mathbb{L}|-1)$. From both the rules, the final rule of combination can be formulated according to which, the number of places and transitions in bounded and unbounded HCAPN can be given as $|\mathbb{H}| = |\mathbb{P}|+|\mathbb{T}|+|\mathbb{T}'|+|\mathbb{L}|+(|\mathbb{L}|-1)-|\mathbb{L}|, = |\mathbb{P}|+|\mathbb{T}|+|\mathbb{T}'|+(|\mathbb{L}|-1)$ , and $|\mathbb{H}| = |\mathbb{P}| + |\mathbb{T}| + |\mathbb{T}'| + |\mathbb{L}| + \frac{|\mathbb{L}|}{2}(|\mathbb{L}|-1) - |\mathbb{L}|, = |\mathbb{P}| + |\mathbb{T}| + |\mathbb{T}'| + \frac{|\mathbb{L}|}{2}(|\mathbb{L}|-1)$, respectively.

## C. BEHAVIOUR AND VULNERABILITY ASSESSMENT WITH HCAPN

The proposed approach helps to assess the behaviour and identify any potential vulnerability in a drone enabled IIoT. This reduces the probability of the network to undergo any attack in the lateral stage of their operations. The proposed approach relies on forming a novel HCAPN model which serves as a state evaluator as well as a dynamic assessment tool for different entities involved in communications. However, at the present stage, the evaluations are presented only for the drone layer which connects the core of the network to the APs that serve the IIoT. Following this, a complete HCAPN model is presented in Fig. 4 with details of context in Table 4. The figure contains seven aspects formed out of the layered view of the entire network. The drone aspect (layer 5) is operated by another PN which forms an individual HCAPN with it by using the behaviour conditions. Once the network is initiated, the context from the drones is passed to the PN of behaviour conditions, which checks for any false instances while evaluating the triplet ($\mathcal{T} = \langle \kappa, \mathcal{B}, \mathcal{D}_{\mathcal{B}_\kappa} \rangle$). This triplet can be operated through "AND" clause or "OR" clause depending on the requirements and relaxations induced in a network. The "AND" clause is strict and can identify a misbehavior in a single instance, whereas for "OR" clause the outputs at the final stage are further evaluated while matching the flying conditions imposed on the drones at the start of the network.

### 1) HCAPN VERIFICATION

It is to be noted that the proposed HCAPN model allows dynamically adjustment of the rules, which supports continuous and dedicated evaluation of the network metrics. In general state verifiers, the number of states may be as high as $2^{\mathcal{V}}$, which cause excessive overheads, however, in the HCAPN, the behaviour PN operates parallel in all the involved entities allowing verification with complexity less than equal to that of forming a graph. Such an evaluation resolves the overall complexity of the proposed approach and makes it a competent solution.

Once behaviour conditions are set, the input and the output of this PN are operated with security policies, which take into account the present state and provides a decision that helps to finalize the identification of a legitimate and a vulnerable drone as shown in Fig. 5. From this figure, it can be noticed that there are six major security policies for drones that support the vulnerability identification. These conditions are fed in form of argument to the function that calls the security policy PN. The security policy PN is fired through a single transition and violation of any one of these policies results in the identification of a potentially vulnerable drone. The only overhead apart from the graph formation involved in this entire behaviour modeling and vulnerability assessment is that of the matching algorithm, which can be neglected considering the importance of such a system. It is further to be marked that the proposed HCAPN can simultaneously operate in both the directions allowing 2-way verification through single PN. These verifications can further be classified as a centralized mechanism that takes place at the core and all the instructions are passed from the core to the drones or as a distribution mechanism where every drone is responsible for transmissions with the legitimate drone.

The success of all the operations depends on the accuracy of HCAPN formation and correct flow of context across the network. The proposed HCAPN is verified for its correctness by following the earlier described rule of places and rule of passes along with conflict evaluation properties of a general PN. Further, the reachability is accessed while evaluating the dependency of a transition on such a place that cannot immediately pass the required context for making a decision.

### 2) N-STATE FLOW VERIFICATION

The proposed HCAPN can identify any vulnerability by saving $\left(\frac{\mathcal{F}}{2}(\mathcal{F}-1)\right)-2$ states as only input and output places are operated with the security policies along with behaviour assessment, where $\mathcal{F}$ is the number of behaviour conditions. Irrespective of the value of $\mathcal{F}$, vulnerability assessment is performed just by consuming two times firing of a transition, which reduces the computational complexity of the overall system. In order to understand the operational flow of the proposed HCAPN, two exemplary HCAPNs are presented in Figs. 6 and 7. The figures show that only layer-1 is a complexity causing stage, which is negligible as it is based on an individual entity, whereas the inner aspects operate only once and are used for all the individual IoT as well as IIoT. These generalizations show that the architecture of the network is responsible for the complexity of HCAPN. Further, the number of entities in each layer also causes a lesser impact as evaluations are not dependent on the sequential flow. In addition, the proposed HCAPN can also be used

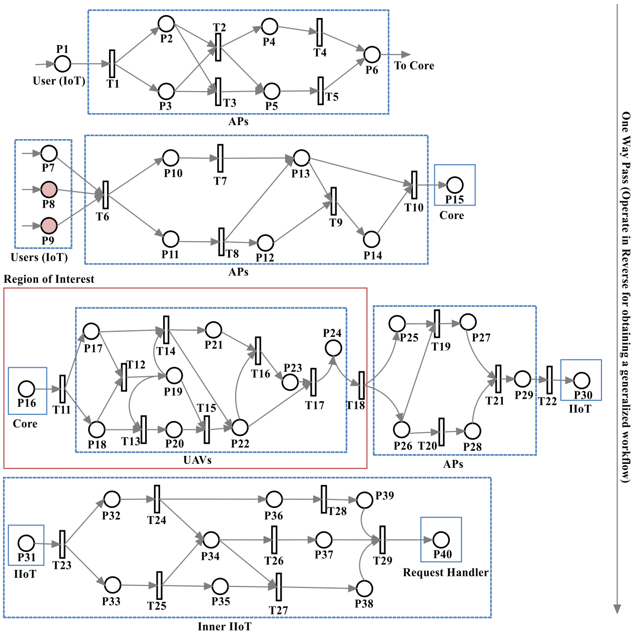


**FIGURE 6. An exemplary illustration of combinatorial PNs for different entities of the defined network model using a single IoT device.**

for balancing the network load and identifying the shortest path between the users and its intended handler in IIoT. The number of layers does impact the performance of the HCAPN as the number of passes increases the dependency on the previous layer for successful operations. However, for the passes between the entities, the ones with the relevant context are fired immediately providing a support for fast processing. Thus, to effectively utilize the HCAPN, it is required to handle the race-conditions between the places of different layers, which are not in the scope of this article and will be presented in our future reports. Irrespective of that, the proposed HCAPN allows effective assessment of drone behaviour as well as identification of potential vulnerabilities through security policies.[3] All the processes explained in the proposed section while considering the system modeling are provided in Algorithm 1. The algorithm helps to understand the flow and alterations which can be dynamically made without disturbing the regular operations of the network. Further, the algorithm also helps to understand the crucial points in the proposed solution, where different checkpoints can be marked for obtaining intermediate as well as the final results. Note that choice of communication patterns and protocols are not considered at this part of our work and shall be presented in future reports.

## VI. PERFORMANCE EVALUATIONS

The proposed approach is evaluated for its capacity to identify the behaviour of drones used for supporting communications between the core and the IIoT. The evaluations are intended towards the identification of vulnerable drones which can

[3] The security policies are vendor/service provider specific. At present, a generalized set of security policies is used to form a PN as shown in Fig. 5.

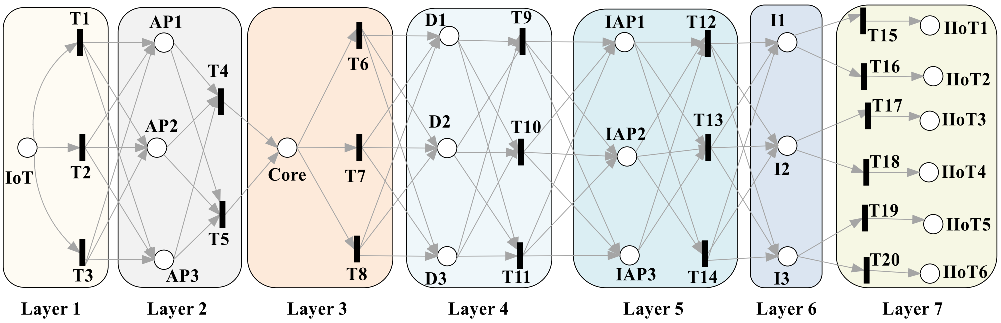


**FIGURE 7.** N-State flow verification for the proposed scenario using a subset of HCAPN.

**Algorithm 1** Behaviour and Vulnerability Marking

```
Input: System model, behaviour conditions, security policies
Output: Behaviour and vulnerability markings
while (Transmission!=NULL) do
    if (New_Entity==TRUE && Modifications==TRUE) then
        Initiate the network
        Decide on centralized, distributive or both mode of HCAPN
        Formulate behaviour model and ensure security policies as PN
        Set ($\mathcal{T} = \langle \kappa, \mathcal{B}, \mathcal{D}_{\mathcal{B}_\kappa} \rangle$)
        Check for entities and formulate PN
        Combine PNs to form HCAPN
    else
        Validate HCAPN, security policy PN and behaviour PN
        Fetch values and fill ($\mathcal{T} = \langle \kappa, \mathcal{B}, \mathcal{D}_{\mathcal{B}_\kappa} \rangle$) using behaviour PN
        Merge HCAPN and behaviour PN
        Load security PN
        Operate transitions and keep tracking
        if (Transitions_Firing== TRUE) then
            Continue marking and select nodes for transmissions
        else
            Mark and correct HCAPN
            Validate and continue
        end if
    end if
    Maintain logs
end while
Update network, keep track and store results
```

**TABLE 5.** Parameter configurations.

| Symbol | Value | | |
|---|---|---|---|
| $\mathcal{E}$ | 1000 | $\mathcal{M}_a$ | 1.2 GB |
| $\mathcal{A}_1$ | 10 | $d_o$ | 10 m |
| $\mathcal{A}_2$ | 10 | $\beta$ | 30 Hz |
| $\mathcal{D}$ | 5-20 | $Z$ | 0.1-1 dBm$^{-1}$ |
| $\mathcal{C}$ | 1000 | $\mathcal{W}$ | 7, 9 |
| $s$ | 2-4 | $\epsilon^{(NLoS)}$ | 3 |
| $\mathcal{F}_x$ | 1-2 | $\epsilon^{(LoS)}$ | 3.5 |
| $\psi$ | 1 | $P_r^{(NLoS)}$ | 25 dBm |
| $t$ | 3600 s | $P_r^{(LoS)}$ | 30 dBm |
| $v$ | 25 Kmph | $\delta_{NLoS}$ | 3 |
| $\mathcal{A}_m$ | 1 sq.km. | $\delta_{LoS}$ | 2.5 |
| $\tau_F$ | 100 s interval | $N_o$ | -174 dBm/Hz |
| $\tau_{LoS}$ | 5-10 s | $\mathcal{H}$ | 100-200 m |
| $\tau_{SW}$ | 1-5s | $\mathcal{G}$ | -11 dB |
| $\tau_{NF}$ | 10-20s | $\mathcal{T}_p$ | 30 dBm |
| $n$ | $1-10$ | $\mathcal{E}_r$ | 450-1000J |
| $\frac{1}{\gamma}$ | 500-1200 s | $\mathcal{M}_r$ | 512-1024 MB |
| $\frac{\vartheta}{\mu}$ | 256 kbps | $\mathcal{E}_a$ | 2000 J |

harm the network as well as can be the potential threats. Behaviour conditions and security policies help to attain such a requirement through HCAPN. The results are attained by formulating the entire system through Petri.Net Simulator[4] along with numerical evaluations as input to the system. The values of parameters used for analyzing the proposed solution are provided in Table 5.

The network is operated with 20 drones which facilitate data as an intermediate entity and the system model plays its role while operating parallel to the general operations of the drone. The HCAPN formed helps to determine the legitimate drones which can be selected for transmissions and also helps to identify drones which have a potential vulnerability that can lead to a certain type of threat. To present this, the entire network of drones is considered as the region of interest and all the drones are operated in four sets. All these sets are evaluated for behaviour conditions through HCAPN and the final decision is taken on the detection of a drone as shown in Table 6.[5] From this table, the role of each behaviour condition and the probabilistic model can be followed, and it can be noticed that for the given 10 states for all the drones, the drone with ID-6 is marked as a potential anomaly in its second state. This information is passed to all other drones

[4] https://github.com/larics/Petri.Net, last accessed April 2018.

[5] The results are presented for two sets only. Note that only 1 set has vulnerable drones.

**TABLE 6.** State-wise results for two different sets of drones out of four sets through behaviour modeling over HCAPN using "∨" operations.

| Set | $\vert\mathcal{D}\vert$ | Slots | $L$ | $\mathcal{C}_S$ | $\mathcal{C}_{US}$ | $\mathcal{P}_{1,L-1}$ | $\mathcal{P}_{2,L-1}$ | $\mathcal{P}(C_L\vert\mathcal{C}_S)$ | $\mathcal{P}(C_L\vert\mathcal{C}_{US})$ | $\mathcal{C}^{(1)}_{R,L}$ | $\mathcal{C}^{(2)}_{R,L}$ | $\mathcal{D}^{(marked,correct)}_{\mathcal{B},\kappa}$ | $\mathcal{D}^{(marked,incorrect)}_{\mathcal{B},\kappa}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 (ID-1) | 360 | 1 | 15 | 0 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1 (Yes) | 0 (No) |
| | | 720 | 2 | 14 | 1 | 0.93 | 0.07 | 0.91 | 0.09 | 0.85 | 0.01 | 1 (Yes) | 0 (No) |
| | | 1080 | 3 | 13 | 2 | 0.87 | 0.13 | 0.82 | 0.18 | 0.71 | 0.02 | 1 (Yes) | 0 (No) |
| | | 1440 | 4 | 12 | 3 | 0.80 | 0.20 | 0.73 | 0.27 | 0.58 | 0.05 | 1 (Yes) | 0 (No) |
| | | 1800 | 5 | 12 | 3 | 0.80 | 0.20 | 0.73 | 0.27 | 0.58 | 0.05 | 1 (Yes) | 0 (No) |
| | | 2160 | 6 | 12 | 3 | 0.80 | 0.20 | 0.73 | 0.27 | 0.58 | 0.05 | 1 (Yes) | 0 (No) |
| | | 2520 | 7 | 11 | 4 | 0.73 | 0.27 | 0.64 | 0.36 | 0.47 | 0.10 | 1 (Yes) | 0 (No) |
| | | 2880 | 8 | 11 | 4 | 0.73 | 0.27 | 0.64 | 0.36 | 0.47 | 0.10 | 1 (Yes) | 0 (No) |
| | | 3240 | 9 | 11 | 4 | 0.73 | 0.27 | 0.64 | 0.36 | 0.47 | 0.10 | 1 (Yes) | 0 (No) |
| | | 3600 | 10 | 11 | 4 | 0.73 | 0.27 | 0.64 | 0.36 | 0.47 | 0.10 | 1 (Yes) | 0 (No) |
| 2 | 5 (ID-6) | 360 | 1 | 10 | 5 | 0.67 | 0.33 | 0.50 | 0.50 | 0.33 | 0.17 | 1 (Yes) | 0 (No) |
| | | 720 | 2 | 9 | 6 | 0.60 | 0.40 | 0.40 | 0.60 | 0.24 | 0.24 | 0 (No) | 1 (Yes) |
| | | 1080 | 3 | 9 | 6 | 0.60 | 0.40 | 0.40 | 0.60 | 0.24 | 0.24 | 0 (No) | 1 (Yes) |
| | | 1440 | 4 | 8 | 7 | 0.53 | 0.47 | 0.30 | 0.70 | 0.16 | 0.33 | 0 (No) | 1 (Yes) |
| | | 1800 | 5 | 5 | 10 | 0.33 | 0.67 | 0.00 | 1.00 | 0.00 | 0.67 | 0 (No) | 1 (Yes) |
| | | 2160 | 6 | 7 | 8 | 0.47 | 0.53 | 0.20 | 0.80 | 0.09 | 0.43 | 0 (No) | 1 (Yes) |
| | | 2520 | 7 | 6 | 9 | 0.40 | 0.60 | 0.10 | 0.90 | 0.04 | 0.54 | 0 (No) | 1 (Yes) |
| | | 2880 | 8 | 6 | 9 | 0.40 | 0.60 | 0.10 | 0.90 | 0.04 | 0.54 | 0 (No) | 1 (Yes) |
| | | 3240 | 9 | 5 | 10 | 0.33 | 0.67 | 0.00 | 1.00 | 0.00 | 0.67 | 0 (No) | 1 (Yes) |
| | | 3600 | 10 | 5 | 10 | 0.33 | 0.67 | 0.00 | 1.00 | 0.00 | 0.67 | 0 (No) | 1 (Yes) |

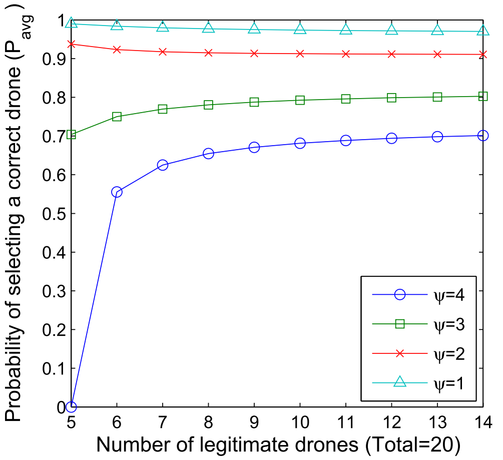


**FIGURE 8.** Average probability of selecting a correct drone with a variation in the number of failures and the available number of legitimate drones.

and communication with it is stopped and it is ensured that the network remains aloof from the potential threats of the drone with ID-6. Using these results each drone can be traced for the entire duration depending on the time stamps. The accuracy and detection rate attain 100% results on strict conditions. However, in ambiguous scenarios, the accuracy of the model is between 90% to 99.9% with an error of ±1% and the detection rate higher than 95% for four sets of drones.

In addition to the above behavioral and vulnerability assessments, evaluations are performed on the overall tracking behaviour of the system using the defined configurations. At first, the results are presented for analyzing the average probability of selecting a correct drone w.r.t. the variations in the number of failures and the number of legitimate drones as shown in Fig. 8. The results show that the probability of continuing with the given setup increases as the number of correct drones increases, and the number of failures impose a direct impact on the network. Using these probabilistic values, the HCAPN accommodate for updated policies while involving more drones to support transmissions. The results show that in the presence of vulnerable drones, the probability of selecting a correct drone increases by 81.71% along with an increase in correctly performing drones.

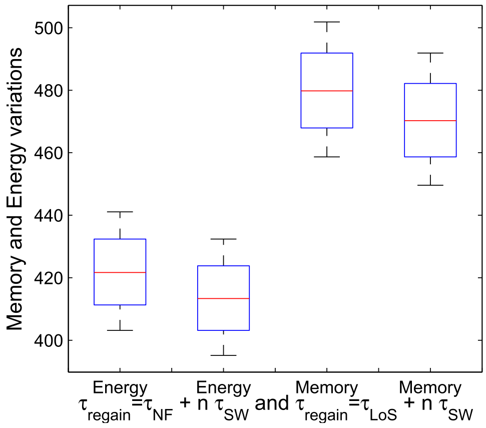


**FIGURE 9.** Resource (Energy and Memory) extension rate with a variation in the time to regain the connectivity.

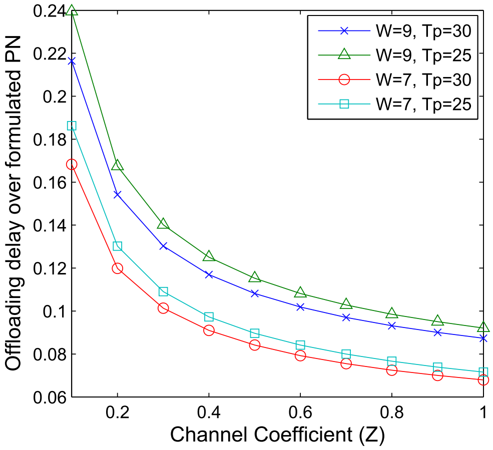


**FIGURE 10.** Network offloading delay with a variation in the number of channels and the transmit power.

**TABLE 7.** Behaviour dominance from resources in HCAPN.

| Place | Resource Utilization | Place | Resource Utilization |
|---|---|---|---|
| $P'1$ | 42.1% | $P'8$ | 26.3% |
| $P'2$ | 42.1% | $P'9$ | 63.2% |
| $P'3$ | 42.1% | $P'12$ | 63.2% |
| $P'14$ | 42.1% | $P'13$ | 10.5% |
| $P'5$ | 42.1% | $P'15$ | 5.3% |
| $P'4$ | 15.8% | $P'10$ | 63.2% |
| $P'6$ | 63.2% | $P'11$ | 63.2% |
| $P'7$ | 63.2% | | |

In the second part, the evaluations are performed to analyze the resource variation to fit in the lifetime module of the proposed system setup. The results are presented for the energy and memory utilization of 20 drones while regaining connectivity when an attacker or a vulnerable drone is identified as shown in Fig. 9. The results suggest that the energy range of the system varies between 413.50 and 421.85 J, whereas the memory varies between 470.47 and 479.97 MB for connection regaining period depending on the flyby time and the LoS attaining time of the new and the existing drones. The variation in energy and memory consumption over the defined configurations for two different conditions in (6) is of the order of 1.98% only. These suggest that at a given rate, the network is able to sustain the load due to the excessive requirements of the energy and memory resources.

The delay in identification of the vulnerable drone and selection of the next drone to support the transmission has considerable effects on the performance of the system. A network operating with HCAPN must not cause excessive overheads as this may violate the operational conditions of the network. Thus, it is important that the defined operations should be operated within the stipulated duration. Results in Fig. 10 suggest that mean offloading delay caused by the proposed approach are in the order of 0.13s, 0.12s, 0.10s, 0.09s while varying the transmit power and the number of channels on each drone. The higher number of channels and more transmit power causes a lower offloading delay while the scenarios with a lower value for any of these two parameters causes higher delays. But, even the highest values observed in the evaluations are negligible which justifies the efficiency of the system in providing quick decidability on the behaviour of drones.

Results presented in Fig. 11 and Table 7 suggest the oscillations and the dominance of the conditions in assessing the behaviour as well as the potential vulnerability amongst the given number of drones. From these results, it can noticed that conditions, $\Theta$, $\tau_{\mathcal{A}}$, $\tau_R$, $\mathcal{E}_r$, $\mathcal{M}_r$ and $v$ play the decisive role in marking a drone as abnormal (incorrect) or normal (correct) during its operations.

Note that for 15 conditions, the PN for behavioral modeling in HCAPN operates only for four extra instances to take a final decision, which shows the low-complex nature of the proposed approach. With an average response time of 0.11 s between each place, the complexity of operations is presented for places, layers and the number of passes on the overall system to perform self-verification for the end to end connectivity, as shown in Fig. 12. The results show that the overall time for verification of HCAPN is very low even for a high number of layers, which presents its significance

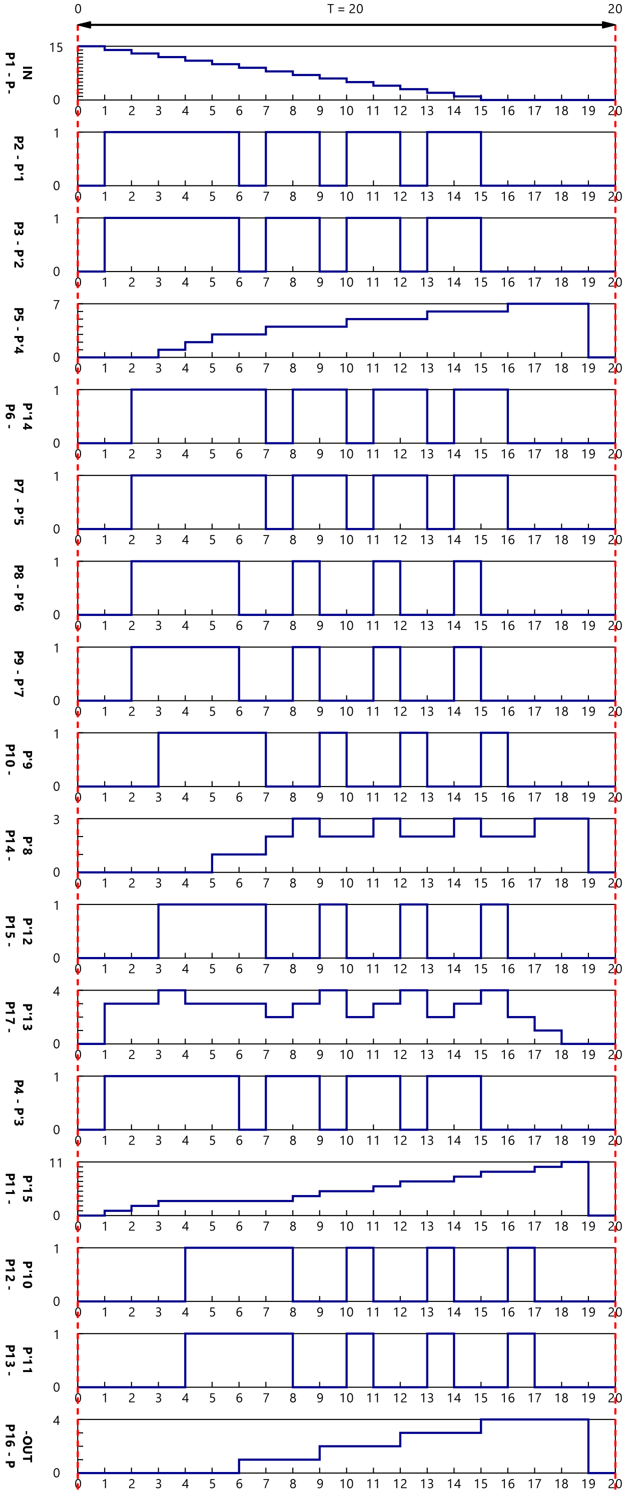


**FIGURE 11.** The oscillations for 15 behavioral conditions operated for each of the drones while forming the complete HCAPN. The states include outputs of P-In and P-Out. The diagrams help to understand the periodic token content of each behavioral condition for the input parameters of a single drone.

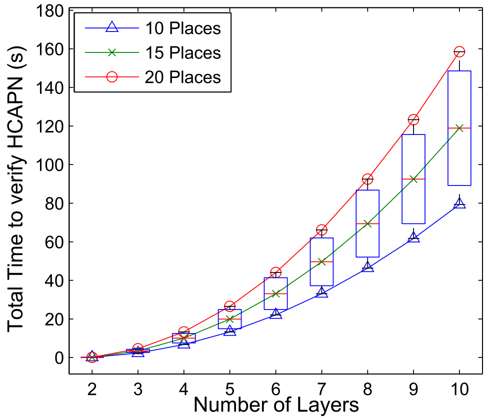


**FIGURE 12.** Operational complexity of verification procedures for the entire HCAPN with a varying number of places and transitions.

in assessing the behaviour and vulnerability of networks with different sets and number of entities.

## VII. CONCLUSION

This paper proposes a novel N-layered Hierarchical Context-Aware Aspect-Oriented Petri Net (HCAPN) model which helps to evaluate the drone behaviour and identifies any potential vulnerability by the utilization of security policies. The proposed HCAPN model ensures identification of drones which may violate the operational conditions of the network and may expose the entire network to different types of cyber-threats. The evaluations suggest that the proposed approach provides low-complex and low-overheads based behavioral and vulnerability assessment model with a detection rate higher than 95% and accuracy as high as 99.9%. The proposed approach also increases the probability of selecting a correct drone by 81.71% even in the case of a high number of failures. In addition, the results are presented for resource (memory and energy) extensions while connecting end users to IIoT devices, network offloading delays, state-wise outputs for all the drones, and oscillations of HCAPN behaviour conditions. From the methodology and results, it is evident the proposed approach can be used as a benchmark for assessing networks which involve drones as a crucial entity.

In future, the proposed approach will be extended for direct inclusion of security aspects with the HCAPN model and focus will be given to the automation and the inclusion of communication procedures.

## REFERENCES

[1] C. Zhu, J. J. P. C. Rodrigues, V. C. M. Leung, L. Shu, and L. T. Yang, "Trust-based communication for the industrial Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 16–22, Feb. 2018.

[2] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1351–1360, Jun. 2018.

[3] B. M. Lee and H. Yang, "Massive MIMO for industrial Internet of Things in cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2641–2652, Jun. 2018.

[4] L. Lyu, C. Chen, S. Zhu, and X. Guan, "5G enabled codesign of energy-efficient transmission and estimation for industrial IoT systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2690–2704, Jun. 2018.

[5] J. Li *et al.*, "Secure distributed deduplication systems with improved reliability," *IEEE Trans. Comput.*, vol. 64, no. 12, pp. 3569–3579, Dec. 2015.

[6] Z. Huang, S. Liu, X. Mao, K. Chen, and J. Li, "Insight of the protection for data security under selective opening attacks," *Inf. Sci.*, vols. 412–413, pp. 223–241, Oct. 2017.

[7] V. Sharma and R. Kumar, "Teredo tunneling-based secure transmission between UAVs and ground ad hoc networks," *Int. J. Commun. Syst.*, vol. 30, no. 7, p. e3144, 2017.

[8] Q. Do, B. Martini, and K.-K. R. Choo, "A data exfiltration and remote exploitation attack on consumer 3D printers," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 10, pp. 2174–2186, Oct. 2016.

[9] N. D. W. Cahyani, B. Martini, K.-K. R. Choo, and A. Al-Azhar, "Forensic data acquisition from cloud-of-things devices: Windows smartphones as a case study," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 14, p. e3855, 2017.

[10] V. Sharma, I. You, G. Pau, M. Collotta, J. D. Lim, and J. N. Kim, "LoRaWAN-based energy-efficient surveillance by drones for intelligent transportation systems," *Energies*, vol. 11, no. 3, p. 573, 2018.

[11] J. P. G. Sterbenz, "Drones in the smart city and IoT: Protocols, resilience, benefits, and risks," in *Proc. 2nd Workshop Micro Aerial Vehicle Netw., Syst., Appl. Civilian Use*, 2016, p. 3.

[12] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, Feb. 2017.

[13] V. Dey, V. Pudi, A. Chattopadhyay, and Y. Elovici, "Security vulnerabilities of unmanned aerial vehicles and countermeasures: An experimental study," in *Proc. 17th Int. Conf. Embedded Syst. (VLSID) VLSI Design*, Jan. 2018, pp. 398–403.

[14] V. Sharma, F. Song, I. You, and H.-C. Chao, "Efficient management and fast handovers in software defined wireless networks using UAVs," *IEEE Netw.*, vol. 31, no. 6, pp. 78–85, Nov./Dec. 2017.

[15] V. Sharma, R. Kumar, and R. Kaur, "UAV-assisted content-based sensor search in IoTs," *Electron. Lett.*, vol. 53, no. 11, pp. 724–726, May 2017.

[16] Y. He, Y. Peng, S. Wang, D. Liu, and P. H. W. Leong, "A structured sparse subspace learning algorithm for anomaly detection in UAV flight data," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 1, pp. 90–100, Jan. 2018.

[17] R. Mitchell and I.-R. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 5, pp. 593–604, May 2014.

[18] R. N. Akram *et al.*, "Security, privacy and safety evaluation of dynamic and static fleets of drones," in *Proc. IEEE/AIAA 36th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2017, pp. 1–12.

[19] H. N. Saha *et al.*, "A cloud based autonomous multipurpose system with self-communicating bots and swarm of drones," in *Proc. 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2018, pp. 649–653.

[20] F. Aznar, M. Pujol, R. Rizo, and C. Rizo, "Modelling multi-rotor UAVs swarm deployment using virtual pheromones," *PLoS ONE*, vol. 13, no. 1, p. e0190692, 2018.

[21] I. You, V. Sharma, M. Atiquzzaman, and K.-K. R. Choo, "GDTN: Genome-based delay tolerant network formation in heterogeneous 5G using inter-UA collaboration," *PLoS ONE*, vol. 11, no. 12, p. e0167913, 2016.

[22] V. Rodriguez-Fernandez, A. Gonzalez-Pardo, and D. Camacho, "Modelling behaviour in UAV operations using higher order double chain Markov models," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 28–37, Nov. 2017.

[23] Z. Birnbaum, A. Dolgikh, V. Skormin, E. O'Brien, D. Müller, and C. Stracquodaine, "Unmanned aerial vehicle security using behavioral profiling," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Jun. 2015, pp. 1310–1319.

[24] P. Gonçalves, J. Sobral, and L. A. Ferreira, "Unmanned aerial vehicle safety assessment modelling through Petri nets," *Rel. Eng. Syst. Safety*, vol. 167, pp. 383–393, Nov. 2017.

[25] K. Hartmann and C. Steup, "The vulnerability of UAVs to cyber attacks—An approach to the risk assessment," in *Proc. 5th Int. Conf. Cyber Conflict (CyCon)*, Jun. 2013, pp. 1–23.

[26] H. Wang, Z. Chen, J. Zhao, X. Di, and D. Liu, "A vulnerability assessment method in industrial Internet of Things based on attack graph and maximum flow," *IEEE Access*, vol. 6, pp. 8599–8609, 2018.

[27] X. Wang, M. Davis, J. Zhang, and V. Saunders, "Mission-aware vulnerability assessment for cyber-physical systems," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, vol. 1, Aug. 2015, pp. 1148–1153.

[28] S. Barbarossa, P. Di Lorenzo, and S. Sardellitti, "Computation offloading strategies based on energy minimization under computational rate constraints," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2014, pp. 1–5.

[29] T. A. Schonhoff and A. A. Giordano, *Detection and Estimation Theory and Its Applications*. London, U.K.: Pearson College Division, 2006.

[30] D. Xu and K. E. Nygard, "Threat-driven modeling and verification of secure software using aspect-oriented Petri nets," *IEEE Trans. Softw. Eng.*, vol. 32, no. 4, pp. 265–278, Apr. 2006.

[31] M. Mascheroni, "Hypernets: A class of hierarchical Petri nets," Ph.D. dissertation, Facoltà Scienze Matematiche, Fisiche Naturali, Dept. Inform. Sistemistica Comunicazione, Univ. Milano-Bicocca, Milan, Italy, 2010.

[32] K. Jensen, "Coloured Petri nets," in *Proc. IEE Colloq. Discrete Event Syst., New Challenge Intell. Control Syst.*, 1993, pp. 1–5.

[33] P. Huber, K. Jensen, and R. M. Shapiro, "Hierarchies in coloured Petri nets," in *Proc. Int. Conf. Appl. Theory Petri Nets*. Bonn, Germany: Springer, 1989, pp. 313–341.

[34] R. Fehling, "A concept of hierarchical Petri nets with building blocks," in *Proc. Int. Conf. Appl. Theory Petri Nets*. Gjern, Denmark: Springer, 1991, pp. 148–168.

[35] V. Sharma, D. N. K. Jayakody, I. You, R. Kumar, and J. Li, "Secure and efficient context-aware localization of drones in urban scenarios," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 120–128, Apr. 2018, doi: 10.1109/MCOM.2018.1700434.

**VISHAL SHARMA** (S'13–M'17) received the B.Tech. degree from Punjab Technical University in 2012 and the Ph.D. degree in computer science and engineering from Thapar University in 2016. In 2016, he was a Lecturer with Thapar University. From 2016 to 2017, he was a joint Post-Doctoral Researcher with the MobiSec Lab, Department of Information Security Engineering, Soonchunhyang University, South Korea, and Soongsil University, South Korea. He is currently a Research Assistant Professor with the Department of Information Security Engineering, Soonchunhyang University. He has authored and co-authored over 60 journal/conference articles and book chapters. His areas of research and interests are 5G networks, UAVs, estimation theory, and artificial intelligence. He was a TPC Member of ITNAC-IEEE. TCBD'17. He is serving as a TPC Member of ICCMIT'18, CoCoNet'18, and ITNAC-IEEE TCBD'18. He serves as the Program Committee Member for the *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*. He is a Professional Member of ACM. He is the past Chair of the ACM Student Chapter-TU Patiala. He was a PC Member of MIST'16. He received three best paper awards from the IEEE International Conference on Communication, Management and Information Technology, Warsaw, Poland, in 2017, CISC-S'17, South Korea, in 2017, and IoTaas, Taiwan, in 2017. He was the Track Chair of MobiSec'16 and AIMS-FSS'16. He was a Reviewer of MIST'17. He serves as a reviewer for various IEEE Transactions and other journals.

**GAURAV CHOUDHARY** received the B.Tech. degree in computer science and engineering from Rajasthan Technical University in 2014 and the master's degree in cybersecurity from the Sardar Patel University of Police, Security and Criminal Justice in 2017. He is currently pursuing the Ph.D. degree with the Department of Information Security Engineering, Soonchunhyang University, Asan, South Korea. His areas of research and interests are UAVs, mobile and Internet security, Internet of Things security, network security, and cryptography.

**YONGHO KO** received the B.S. degree in information security engineering from Soonchunhyang University, Asan, South Korea, where he is currently pursuing the joint master-Ph.D. degree with the Department of Information Security Engineering. His current research interests include mobile Internet security, Internet of Things security, and formal security analysis.

**ILSUN YOU** (SM'13) received the M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, South Korea, in 1997 and 2002, respectively, and the Ph.D. degree from Kyushu University, Japan, in 2012. From 1997 to 2004, he was a Research Engineer with Thin Multimedia, Inc., Internet Security Co., Ltd., and Hanjo Engineering Co., Ltd.. He is currently an Associate Professor with the Department of Information Security Engineering, Soonchunhyang University. His main research interests include Internet security, authentication, access control, and formal security analysis. He is a fellow of the IET. He has served or is currently serving as the main Organizer of international conferences and workshops, such as MobiWorld, MIST, SeCIHD, AsiaARES, and so forth. He is the Editor-in-Chief of the *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*. He is on the Editorial Board of *Information Sciences*, the *Journal of Network and Computer Applications*, the *International Journal of Ad Hoc and Ubiquitous Computing*, *Computing and Informatics*, the *Journal of High Speed Networks*, *Intelligent Automation & Soft Computing*, and *Security and Communication Networks*.