drones

MDPI

Article

An Intrusion Detection System for Drone Swarming Utilizing Timed Probabilistic Automata

Venkatraman Subbarayalu * and Maria Anu Vensuslaus

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
* Correspondence: venkats23@gmail.com or venkatraman.s@vit.ac.in

Abstract: Unmanned aerial vehicles (UAVs), commonly known as drones, have found extensive applications across diverse sectors, such as agriculture, delivery, surveillance, and military. In recent times, drone swarming has emerged as a novel field of research, which involves multiple drones working in collaboration towards a shared objective. This innovation holds immense potential in transforming the way we undertake tasks, including military operations, environmental monitoring, and search and rescue missions. However, the emergence of drone swarms also brings new security challenges, as they can be susceptible to hacking and intrusion. To address these concerns, we propose utilizing a timed probabilistic automata (TPA)-based intrusion detection system (IDS) to model the normal behavior of drone swarms and identify any deviations that may indicate an intrusion. This IDS system is particularly efficient and adaptable in detecting different types of attacks in drone swarming. Its ability to adapt to evolving attack patterns and identify zero-day attacks makes it an invaluable tool in protecting drone swarms from malicious attacks.

Keywords: drones; intrusion detection; probabilistic automata; timed automata; swarming; behavioral model

Citation: Subbarayalu, V.; Vensuslaus, M.A. An Intrusion Detection System for Drone Swarming Utilizing Timed Probabilistic Automata. *Drones* **2023**, *7*, 248. https://doi.org/10.3390/drones7040248

Academic Editor: Carlos Tavares Calafate

Received: 23 February 2023
Revised: 16 March 2023
Accepted: 21 March 2023
Published: 3 April 2023

1. Introduction

A drone swarm is a group of unmanned aerial vehicles (UAVs) or flying robots that work as a team to achieve a specific goal. A drone swarm has several advantages over a single drone. The entire system is robust, meaning that the failure or loss of a single UAV does not affect the performance of the entire system. The flexibility of the drone swarm is extensively increased by dynamically adapting the various configuration styles and standards. Communication plays a significant role in UAV swarm control and coordination. The communication architecture characterizes how the data are exchanged between UAVs or between UAVs and the central control center. Due to the adaptation of UAV drone swarming technology, one of the main considerations is to monitor the drones in the open space and their states in both spatial and temporal aspects.

The worldview of a multitude of advanced robotics expects it to rise above the limits of a single robot by empowering the collaboration of bigger groups. This is enlivened by the collective of animals, where creatures and bugs have been seen to join powers toward a shared objective that is excessively perplexing. Depending on the application paradigm, an entire drone swarm that is highly scalable, is one in which the number of drones in the collection that can be increased or decreased [1]. The manufacturing cost of unmanned aerial vehicles (UAVs) is becoming cheaper and UAVs are available to a larger extent, and utilization of this technology keeps on increasing, and this has opened several research challenges. UAVs are adopted for applications, including agriculture, military rescue operations, supply chain management, inventory control, emergency operations, and surveillance [2,3].

During natural calamities, such as floods, fires, earthquakes, and storms, it is difficult to access locations, and there is a delay in performing rescue operations [4,5]. Rescue

operations are crucial for mankind, as they involve the lives of living beings. UAVs utilized for rescue operations can speed up rescue operations. These mini-flying robots are mounted with many sensors, such as cameras and night vision cameras, that are useful to make disaster estimations, and find and locate flood survivors. Furthermore, they capture and send real-time aerial images to the ground station for better clarity and visualization. Some UAVs are designed to carry a few kilograms to supply the essential items most needed during emergency situations. With the use of drone swarms, search and rescue operations are sped up. In the affected areas, there is no hope for communication due to damage to mobile towers. Using a UAV swarm, temporary communication channels are built to help survivors communicate with rescue teams [6].

The drone swarm model in Figure 1 refers to the design and organization of a group of drones to work together in a coordinated manner. It typically involves a central control system that communicates with each individual drone and directs their behavior based on a set of predefined rules or algorithms. The architecture may include different types of drones, such as leader and follower drones, and may use a variety of communication protocols to ensure efficient and reliable communication between the drones and the control system. Some drone swarm architectures also incorporate artificial intelligence and machine learning algorithms to enable the drones to adapt and learn from their environment and improve their performance over time. Overall, the goal of drone swarm architecture is to create a system that performs complex tasks more efficiently and effectively than a single drone, while also providing greater flexibility and scalability for a wide range of applications, including military, industrial, and civilian use cases.

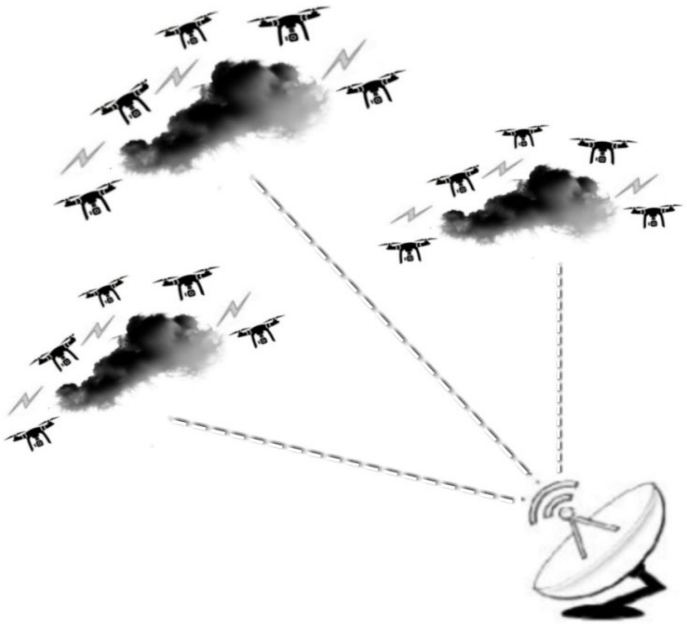

Figure 1. Drone Swarm Model.

Drones in swarming configurations are vulnerable to various types of attacks that compromise their operation and mission. Some common types of attacks in drone swarming include:

Jamming Attacks: In a jamming attack, an intruder sends a strong radio signal that interferes with the drone's wireless communication, causing it to lose communication with the rest of the swarm. This results in the loss of coordination and control of the drone swarm.

Spoofing Attacks: In a spoofing attack, an intruder sends false signals to the drone, tricking it into accepting fake information. This causes the drone to deviate from its intended path, potentially putting it and others at risk.

Hijacking Attacks: In a hijacking attack, an intruder gains unauthorized access to the drone's control system and takes control of the drone. This results in the drone being used for malicious purposes, such as for espionage or to carry out attacks on targets.

Man-in-the-Middle Attacks: In a man-in-the-middle attack, an intruder intercepts and manipulates the communication between the drone and the swarm, causing the drone to behave differently from its intended behavior.

Denial of Service Attacks: In a denial-of-service attack, an intruder floods the drone's communication channel with false data, causing it to become overwhelmed and unable to process legitimate data.

These attacks have significant consequences, such as compromising the confidentiality, integrity, and availability of the drone swarm's mission and data. It is crucial to implement effective security measures, including intrusion detection systems (IDSs), to protect against these and other types of attacks. IDSs are designed to detect and respond to unauthorized access attempts on computer systems, networks, or applications. In the context of drone swarming, IDSs are used to detect unauthorized access attempts by intruders who seek to compromise the swarm's mission or steal sensitive information. There are two main types of intrusion detection systems: signature-based and anomaly-based. Signature-based IDSs use a database of known attack patterns, also known as signatures, to detect intrusion attempts. Anomaly-based IDSs, however, monitor normal system behaviors and flag any activity that deviates from the norm as suspicious.

In the context of drone swarming, anomaly-based IDSs are preferred as they detect unknown and evolving attacks that signature-based IDSs cannot. Implementing IDSs in drone swarming presents several challenges, including: Resource Constraints: Drones have limited resources, such as power and computational capacity, which limit the implementation of IDSs. Network Latency: Drone swarms rely on wireless communication for coordination, and network latency impacts the accuracy of the intrusion detection. Interference: Interference from other devices, such as other drones, impacts the accuracy of the intrusion detection. Dynamic Environment: Drone swarms operate in dynamic environments, and the system's normal behavior changes rapidly, making it challenging to accurately detect intrusions.

The main contributions of this system are as follows:

(i) TPA is a mathematical model used to represent probabilistic systems with timing constraints. By utilizing TPA, the proposed IDS is able to accurately model and analyze the behavior of drone swarms;
(ii) The proposed IDS coordinates the behavior of multiple drones by monitoring the behavior of the individual drones in the swarm. This enables the system to detect and respond to potential security threats in real-time;
(iii) The proposed IDS monitors the behavior of individual drones in the swarm, which allows for the detection of anomalous behavior that may indicate a security breach;
(iv) The proposed IDS provides real-time monitoring and detection of anomalous behaviors. This allows for rapid response and mitigation of potential security threats, making drone swarming a safer technology.

The article is structured as follows: Section 2 reviews existing intrusion detection systems (IDSs) in the context of drone swarming. Section 3 introduces the fundamentals of timed probabilistic automata (TPA) and their relevance to the proposed IDS. Section 4 presents the architecture of the proposed intrusion detection system for drone swarming scenarios. Section 5 discusses the results of the experimental evaluations of the proposed IDS. Finally, Section 6 summarizes the contributions of this work and discusses potential future research directions in this field.

2. Literature Survey

Alfeo. et al. [7] developed an optimized model for coordinating drone swarms in a target space. The model was developed and tested with various simulations and real-time scenarios. Asbach. et al. [8] defined a method to devise a plan for exploration during

natural disasters based on interesting measures. It allows the drone swarm to travel in the optimized path and find the survivors without further delays. Queralta et.al [9] proposed a layered architecture for reconfigurability in heterogeneous flying robots. The real-time application of UAV swarms is in view of a mix of ideas and procedures from the advanced mechanic's space, multi-specialist frameworks area, and edge-cloud processing space. This work clarified how to build a reconfigurable drone swarm and the diverse equipment and programming required to make reconfigurability and flexibility conceivable.

Peng et al. [10] discussed multi-dimensional programming for UAV complex communication. The communication boundaries and strategies vary based on the application. This method uses recurrent neural networks to communicate during jamming conditions. Fabra et al. [11] specified methods for an efficient take-off with coordination. Chen et al. [12] surveyed various network topologies and their communications. They proposed various routing protocols in drones. Hildmann et al. [13] executed algorithms, such that drones cover the maximum area for real-time monitoring applications. Kusyk et al. [14] applied game theory to control drones, and this leads to the way to produce autonomous drones. Arnold et al. [15] devised algorithms for the behavior of drones. These behaviors are made intelligent through artificial intelligence-based techniques, such that the drones do not collide in mid-air. There are numerous research studies have been carried out in the field of wireless sensor networks (WSNs) and their relative challenges, such as energy saving, energy efficiency, portability, and interoperability. This research direction is very much related to the functionality and operation of drones.

Olfati-Saber [16] provides an explanation for flocking and swarming algorithms. Flocking behavior is not directed towards a specific objective, and therefore lacks a high level of control, whereas swarming can offer clear means of control to guide the swarm towards accomplishing a particular task, such as navigating towards a designated tree while avoiding obstacles along the path. Lawton et.al [17] proposed three strategies for behavior control, namely: formation control using coupled dynamics, formation control using coupled dynamics with inter-robot damping based on passivity, and saturated control. A heterogeneous system in [18] makes use of the parallelism, redundancy, and distributed solutions of swarming coordination. Additionally, such a system can incorporate mission specifications because each agent has different skills and payloads. [19] Ramadan et.al surveyed intrusion detection systems for the internet of drones, also known as FANETs (flying ad hoc networks) using RNN-LSTM. Jiang et al. [20] studied types of attacks in UAVs by considering datasets available to the public. They even concluded that the limitations of the model were due to insufficient datasets.

Based on Table 1, an IDS technique is a crucial component in protecting drone swarming systems from various types of attacks. The IDS technique detects different types of attacks by monitoring and analyzing the behavior of the drones and their communication within the swarm. By detecting anomalies and suspicious activity, the IDS technique alerts the operators and prevents potential harm to the swarm. The IDS technique implemented uses various algorithms, such as rule-based systems, machine learning, and statistical analysis. Each of these algorithms has its own strengths and weaknesses, and the choice of algorithm depends on the specific requirements of the drone swarming system. In addition, the IDS technique is further improved by integrating it with other security measures, such as encryption, authentication, and access control. This provides a more comprehensive defense against attacks and increases the overall security of the drone swarming system. Overall, the IDS technique plays a vital role in ensuring the security and reliability of drone swarming systems. By detecting and preventing attacks, it helps to maintain the integrity and functionality of the swarm, which is essential for successful drone operations.

Table 1. Advanced IDS strategies for combating drone swarming attacks.

Attacks	Detection Techniques	Drone Swarm Size	Sensor Configuration	Data Rate	Detection Accuracy	Advantage	Disadvantage
Man-in-the-Middle (MITM) attacks	Encryption Analysis [21]	Small	Encryption sensors	High	Medium	Detect encryption anomalies	Limited range, susceptible to false alarms
	Packet Sniffing [22]	Medium	Network sensors	High	High	Detect unusual packet patterns	More expensive, limited field of view
	Signature Analysis [23]	Large	Multiple sensors	Medium	Low	Detect unusual communication signatures	High false alarm rate, limited accuracy
Denial of Service (DoS) attacks	Network Traffic Analysis [24]	Small	Network sensors	High	Medium	Detect unusual network traffic patterns	Limited range, susceptible to false alarms
	Resource Utilization Monitoring [25]	Medium	Resource utilization sensors	High	High	Detect resource utilization anomalies	More expensive, limited field of view
	Pattern Recognition [26]	Large	Multiple sensors	Medium	Low	Detect unusual behavior patterns	High false alarm rate, limited accuracy
Hijacking attacks	Radio Frequency Interference Detection [27]	Small	Radio frequency sensors	High	Medium	Detect unauthorized control signals	Limited range, susceptible to false alarms
	GPS Spoofing Detection [28]	Medium	GPS sensors	High	High	Detect GPS spoofing attacks	More expensive, limited field of view
	Video Stream Analysis [29]	Large	Cameras	Medium	Low	Detect visual changes in the drone's environment	High false alarm rate, limited accuracy

3. Background

The timed probabilistic automata (TPA) [30] mathematical model combines the concepts of probabilistic automata and timed automata. TPAs extend the traditional probabilistic automata by adding the notion of time to the model, allowing the transitions between states to be associated with a time delay. In TPAs, the time delay is either deterministic or probabilistic, meaning that it has a fixed or a random value. This allows TPAs to model systems with time-sensitive behavior, such as communication protocols, performance and reliability of systems, and the behavior of complex systems. The time delays in TPAs are either discrete or continuous, depending on the application. Discrete time delays represent the time in terms of time steps or clock ticks, while continuous time delays represent the time in real-time units. TPAs are used to analyze various properties of systems, such as reachability, stability, and performance. TPAs are used to model complex systems, such as distributed systems, network protocols, and control systems. In summary, timed probabilistic automata is a powerful mathematical model that allows the for modelling and the analysis of systems with time-sensitive behavior. It combines the concepts of probabilistic automata and timed automata to provide a comprehensive framework for modeling and analyzing complex systems.

TPA is a mathematical model used to represent probabilistic systems with timing constraints. Formally, a TPA is defined as a tuple (Q, Σ, Δ, q0, F, E), where:

Q is a finite set of states.

Σ is a finite set of input symbols.

Δ is a finite set of real-valued time intervals.

q0 ∈ Q is the initial state.

F ⊆ Q is the set of accepting states.

E is a set of edges, where each edge e = (q, a, Δ', q', p) represents a transition from state q to state q' on input symbol a with a probability p, where p is a value between 0 and 1, and Δ' is a set of time intervals that must elapse before the transition can be taken.

A TPA operates as follows: at each step, the TPA reads input symbol a and determines which transition to take probabilistically based on the probabilities associated with each outgoing edge from the current state. In addition, the TPA keeps track of the amount of time that has elapsed since the last transition, and the timing constraints specified on the outgoing edges determine when the next transition can be taken. The TPA accepts a given input sequence if there exists a path from the initial state to an accepting state that satisfies the timing constraints on each transition along the path. TPAs are useful for modeling and analyzing real-time systems that exhibit probabilistic behavior, such as communication protocols, sensor networks, and control systems. They can be analyzed using formal verification techniques to ensure that the system meets certain performance or safety requirements.

Surveillance drones are designed to gather and transmit information from the air. The operational behaviors of surveillance drones include the following: Takeoff and Landing: The drone must be able to take off and land safely and efficiently, often autonomously. Navigation: The drone must be able to navigate to specific locations and fly along predetermined flight paths, either autonomously or under human control. Sensing: The drone must be equipped with various sensors, such as cameras, microphones, and environmental sensors, to gather information about the environment. Data Transmission: The drone must be able to transmit the data gathered by its sensors in real-time to a ground control station or to a remote cloud-based server. Power Management: The drone must be able to manage its power consumption to ensure that it has enough power to complete its mission, either by using rechargeable batteries or by refueling in flight. Obstacle Avoidance: The drone must be able to avoid obstacles in its path, such as trees, buildings, and other objects, to ensure safe and efficient flight. Mission Management: The drone must be able to manage its mission, including starting and stopping missions, changing its flight path, and responding to external commands, either autonomously or under human control.

Security: The drone must be able to implement security measures to protect against unauthorized access and tampering, such as encryption and secure communication protocols. Maintenance: The drone must be designed for easy maintenance, including regular inspections, cleaning, and replacement of components, to ensure its continued operation. Finally, the operational behaviors of surveillance drones are diverse and complex, requiring a combination of hardware, software, and control systems to ensure their effective and efficient operation. The use of drones in swarming applications presents new security challenges, and intrusion detection systems play a crucial role in securing these systems. However, implementing IDSs in drone swarming is challenging due to resource constraints, network latency, interference, and the dynamic environment in which these systems operate. Despite these challenges, continued research in this area is critical to ensure the secure deployment of drone swarms for various applications.

4. Proposed Model

The design of TPAs for intrusion detection in drone swarming involves the following steps:

Modeling the Normal Behavior: The first step in designing a TPA for intrusion detection is to model the normal behavior of the drone swarm. This involves specifying the states, transitions, and probabilistic time constraints that describe the swarm's normal behavior. Specifying the Intrusion Behavior: Once the normal behavior of the drone swarm has been modeled, the next step is to specify the intrusion behavior that the TPA should detect. This involves defining the states, transitions, and time constraints that describe the behavior of the swarm in the event of an intrusion. Probabilistic Analysis: Once the normal and intrusion behaviors have been modeled, the next step is to perform a probabilistic analysis to determine the probability of the drone swarm's behavior deviating from the normal behavior and entering the intrusion behavior. This analysis is used to determine the false positive and false negative rates of the intrusion detection system, which is used to fine-tune the TPA to meet the desired performance requirements.

Integration with the Drone Swarm: The final step in designing a TPA for intrusion detection in drone swarming is to integrate the TPA with the drone swarm. This involves implementing the TPA on the drone swarm's onboard computer system and configuring it to monitor the swarm's behavior in real-time, and detect deviations from the normal behavior that may indicate an intrusion. In assumption, TPAs are an effective tool for intrusion detection in drone swarming by combining temporal logic with probabilistic analysis to model and analyze the behavior of the drone swarm. The design of TPAs for intrusion detection involves modeling the normal and intrusion behavior, performing probabilistic analysis, and integrating the TPA with the drone swarm.

Let S be a set of states, where s_i represents the state of drone i. Let T be a set of transitions between the states, where t_ij represents the transition from state s_i to state s_j. Let R be a set of probabilistic rates, where r_ij represents the probability of transition t_ij occurring.

Each state s_i defined by a vector (x_i, y_i, z_i, θ_i, v_i, w_i), where:

x_i, y_i, and z_i are the drone's position coordinates in 3D space.

θ_i is the drone's heading angle.

v_i is the drone's velocity.

w_i is the drone's angular velocity.

Each transition t_ij can be defined by a time interval [t_start, t_end], where t_start represents the time at which the transition begins, and t_end represents the time at which the transition ends. The probability of transition t_ij occurring is defined by function r_ij(t), where r_ij(t) is the probability of the transition occurring at time t. Using these definitions, our proposed TPA algorithm for drone swarming is as follows:

1. Initialize the system with an initial state set S_0;
2. Compute the set of possible transitions T_i that can be made from each state s_i in S_0;
3. Compute the transition probability r_ij(t) for each transition t_ij in T_i;

4. Calculate the expected value E_ij of the transition time as: E_ij = ∫t_startˆt_end t * r_ij(t) dt;
5. Compute the next state S_1 as the set of all possible next states reachable from any transition in T_i;
6. Repeat steps 2–5 for each state in S_1 to determine possible transitions and their probabilities;
7. Continue the process for a specified number of time steps or until a desired condition is met.

Our proposed TPA algorithm is used to model the behavior of a swarm of drones, where each drone has its own state and can transition between states based on probabilistic rates. The algorithm was adapted to include additional factors, such as sensor data, communication delays, and environmental conditions to model more complex swarm behaviors.

A state transition diagram is a graphical representation of the behavior of a system that shows the possible states that the system is in and the transitions between these states. A state transition diagram for surveillance drone operations may include the following states. Standby: The drone is in standby mode, waiting for a command to start a mission. Takeoff: The drone is taking off and ascending to its operating altitude. Navigation: The drone is flying to its designated location and following its predetermined flight path. Data Collection: The drone is collecting data using its sensors and transmitting this data to a ground control station or remote server. Obstacle Avoidance: The drone is avoiding obstacles in its path and rerouting its flight path as necessary. Emergency Landing: The drone is descending to the ground and landing in response to an emergency.

A state transition diagram for surveillance drone operations may include the following transitions: Start Mission: The transition from Standby to Takeoff, triggered by a command to start a mission. Complete Takeoff: The transition from Takeoff to Navigation, triggered by the drone reaching its operating altitude. Start Data Collection: The transition from Navigation to Data Collection, triggered by the drone reaching its designated location. Encounter Obstacle: The transition from Navigation or Data Collection to Obstacle Avoidance, triggered by the drone detecting an obstacle in its path. Avoid Obstacle: The transition from Obstacle Avoidance to Navigation or Data Collection, triggered by the drone successfully avoiding the obstacle and resuming its flight path. Emergency: The transition from Navigation, Data Collection, or Obstacle Avoidance to Emergency Landing, triggered by the drone detecting an emergency. Complete Emergency Landing: The transition from Emergency Landing to Standby, triggered by the drone successfully landing on the ground.

A state transition diagram is a useful tool for representing the complex behavior of surveillance drones and for understanding the transitions between the various states that the drone is in during its operations.

Consider Figure 2. Timed Probabilistic Automaton of a drone, that models a simple system with nine states: “OFF”, “ON”, “Standby”, “Takeoff”, “Navigation”, “DataCollection”, “ObstacleAvoidance”, “EmergencyLanding”, and “ReturntoLaunch”. The TPA has a clock variable “X” that represents the time elapsed since a particular event. In the transition from State “Takeoff” to State “Navigation”, there is a clock constraint of “60 <=X <= 300 | seconds”, meaning that the transition will occur between 60 s to 300 s after the Takeoff event. The probability of this transition is 0.95, indicating that there is a 95% chance that the transition will occur between 60 s to 300 s. When the transition arrives at the destination state, the clock variable “X” is reset.

Figure 3 illustrates the design approach of an intrusion detection system (IDS) using timed probabilistic automata (TPA) based automata controller strategy for identifying malicious activities in a drone swarm system. The TPA approach considers the operational constraints of resource-limited drone systems, and establishes a set of normal behavior activities for the drone devices. The TPA acts as an event-driven operator for each drone device, while the automata controller/monitor serves as the adaptive knowledge engine that defines the agile interplay between the basic TPA instances. The proposed IDS using TPA detects trends of deviation in the sequential occurrence of activities, and the system

uses coordinated automaton to maintain the duration for all potential combinations of drone swarm scenarios. Event orchestration in a drone swarm environment involves extracting a new service or activity by integrating atomic events produced by drone devices during execution and discovery. However, detecting intrusions results in high energy consumption and significant computing overhead, which uncovers policy breaches and anomalies in the drone swarm environment. To address these vulnerabilities, we propose an efficient solution to orchestrate different intrusion detection patterns that perform functions similar to safety and protection policies.

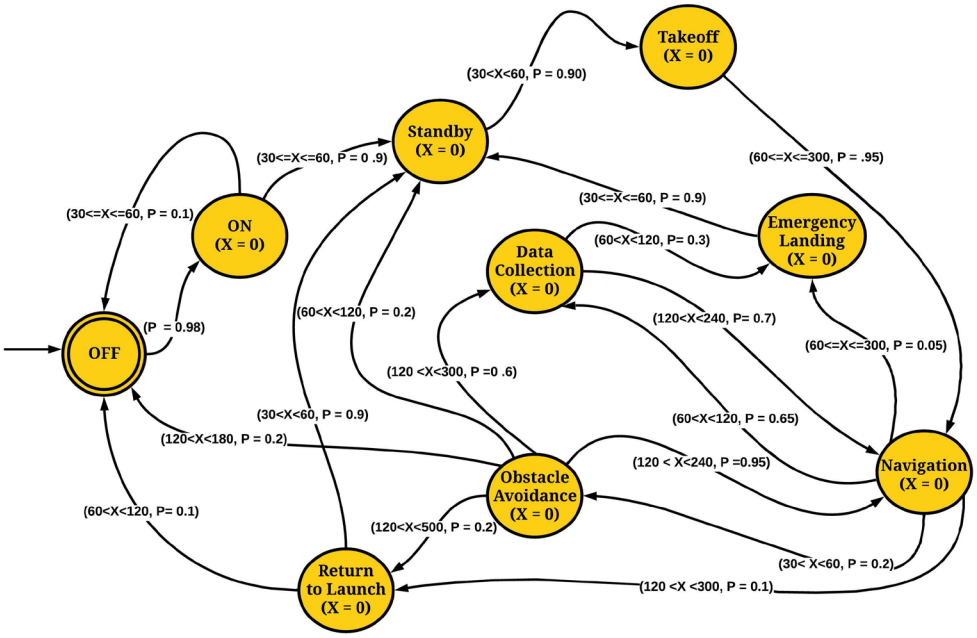

Figure 2. Timed probabilistic automaton of a surveillance drone.

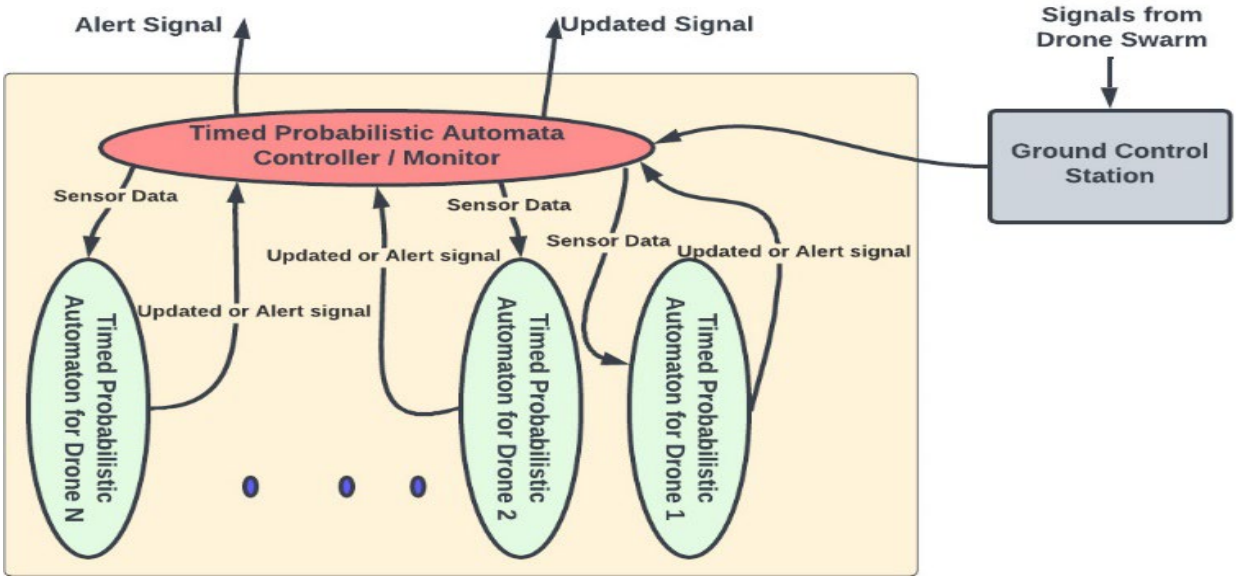

Figure 3. Timed probabilistic automata (TPA) based controller.

Detection of Denial-of-Service Attacks:

To detect a Denial-of-Service (DoS) attack based on the number of incoming packets in a given time window. Assuming that the number of incoming packets received by

the ith drone within a time window of T is represented by Ni(T), the probability of a Denial-of-Service (DoS) attack can be calculated as follows:

$$\text{Probability of DoS in drone swarm} = \Pr[\text{Ni}(T) > \text{T\_max}]$$

where T_max is the maximum number of packets each drone in the drone swarm can handle within the time window T. If the calculated probability exceeds a certain threshold, then conclude that a DoS attack is likely to be occurring.

Detection of Hijack Attacks:

A hijack attack occurs when a malicious entity takes control of one or more drones in the swarm. To detect a hijack attack based on the deviation of drone behavior from the expected behavior. Consider the expected behavior of a drone i at time t as E_i(t). This behavior is modeled using TPA that takes into account the drone's mission, environment, and other factors. The actual behavior of the drone at time t is denoted by A_i(t), then calculate the probability of a hijack attack as follows:

$$\text{Probability of Hijack attack} = \Pr[\exists i: |\text{E\_i}(t) - \text{A\_i}(t)| > \varepsilon]$$

where ε is a threshold value that represents the maximum acceptable deviation from the expected behavior. If the calculated probability exceeds a certain threshold, then conclude that a hijack attack is likely to be occurring.

Detection of Replay Attacks:

A replay attack occurs when an attacker intercepts and re-transmits a legitimate message between two drones in the swarm. To detect a replay attack, with the use of a time-based approach that takes into account the freshness of messages. We denote the time at which message m is received by drone i as t_i(m) then maintain set R_i of recently received messages by drone i. If a new message m' is received by drone i, then check whether m' has already been received by i or any other drone within a certain time window:

$$\text{Probability of Replay attack} = \Pr[\exists i, j: \text{t\_i}(m') - \text{t\_j}(m) \leq \Delta]$$

where Δ is the maximum allowable time difference between the reception of m and m'. If the calculated probability exceeds a certain threshold, then conclude that a replay attack is likely to be occurring.

The use of formal methods [31,32] in intrusion detection systems for drone swarming is a logical approach that provides a high level of assurance in the system's correctness and security. Formal methods refer to mathematical techniques for modeling, analyzing, and verifying computer systems. These methods involve rigorous mathematical reasoning and logic, which helps ensure the correctness and completeness of the system's behavior. In the case of intrusion detection systems for drone swarming, formal methods are particularly useful because they allow for the specification of complex behaviors and interactions among multiple drones. Formal methods can help identify and prevent potential vulnerabilities, such as attacks on the communication channels or manipulation of drone behavior, that could compromise the security of the swarming system.

5. Experimental Analysis

Energy efficiency refers to the ratio of the amount of useful work performed by a system to the amount of energy consumed by the system. In the context of drone swarming, energy efficiency is defined as the ability of the swarm to achieve its objectives while minimizing the energy consumption of individual drones. Detection DOS, hijack, and replay attacks are security threats that compromise the energy efficiency of a drone swarm by causing individual drones to consume more energy than necessary or by causing the swarm to fail to achieve its objectives. Detection DOS, hijack, and replay attacks are security threats that can compromise the energy efficiency of a drone swarm by causing individual drones to consume more energy than necessary or by causing the swarm to fail to achieve

its objectives. One way to model the impact of these attacks on energy efficiency is to use a cost function that takes into account the energy consumption of individual drones and the success rate of the swarm in achieving its objectives. The cost function is expressed as follows:

$$C = E + \alpha\ (1 - S)$$

where C is the cost of the swarm, E is the total energy consumption of individual drones, S is the success rate of the swarm in achieving its objectives, and α is a weight factor that balances the importance of energy consumption and success rate. Detection of DoS, hijack, and replay attacks are modeled by increasing the energy consumption of individual drones when they are forced to perform extra computations to detect and mitigate the attacks. This is modeled as follows:

$$E = E + \beta\_DoS + \beta\_hijack + \beta\_replay$$

where β_DoS, β_hijack, and β_replay are the extra energy consumption due to doS, hijack and replay attacks respectively.

The proposed TPA-based IDS system was tested using a drone swarm setup in Gazebo9 simulator. The drone rules were written in C++ to create plugins that extended Gazebo's functionality, while XML was used to define the simulation environment. The swarm consisted of small and medium drones with a ground control station, installed with the proposed TPA-based IDS. The approach had two modules: the first module analyzed the packet header details to classify packets as malicious or not, while the second module used an automata controller (AC) to check the operational behaviors of all drone devices. The drone swarm environment was accessed by legitimate drone pilots via a wireless ground station, where few legitimate and intruder drones generated different types of malicious events. Table 2 shows the various types of anomalous traffic generated by two drone systems equipped with remote controlled (RC) transmitters. These systems were used to simulate malicious clients that disrupt the network by sending and receiving messages. Additionally, the proposed TPA-based IDS also accounts for malicious activities, such as replay, insert, and modify that are generated by a few legitimate drone systems. The impact of an attack-generating model on the performance of a proposed IDS depends on several factors, including the quality of the attack-generating model, the nature and complexity of the attacks it generates, and the effectiveness of the TPA-based IDS. If the attack-generating model is of high quality and generates realistic and diverse attacks, it can help identify vulnerabilities in the TPA-based IDS and highlight areas for improvement.

Table 2. Test dataset attributes.

Test Dataset (TD)	Legal Events	False/Anomaly Events	Overall Events	False Events Ratio (%)
TD1	3800	800	4600	17.39
TD2	10,000	3500	13,500	25.92
TD2	16,750	4900	21,650	22.63

Our results are evaluated using performance metrics that include precision, recall, F-measure, and accuracy, which are commonly used in assessing malicious behavior. Precision, recall, F-measure, and accuracy are defined in Equations (1)–(4), respectively, as follows:

$$\text{Precision} = (TP/(TP + FP)) \times 100 \quad (1)$$

$$\text{Recall} = (TP/(TP + FN)) \times 100 \quad (2)$$

$$\text{F-Measure} = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}) \quad (3)$$

$$\text{Accuracy} = (TN + TP)/((TN + FP) + (TP + FN)) \times 100 \quad (4)$$

where TP (True Positive) and TN (True Negative) represent the number of events correctly classified as malicious events and normal events, respectively. FN (False Negative) and FP (False Positive) refer to the number of malicious payloads misclassified as legitimate events and legitimate events misclassified as malicious events, respectively.

Comparative Analysis of the Performance Metrics for Our TPA-Based IDS

The evaluation of three test datasets uses three different intrusion detection systems (IDS) based on different automata models. The performance of each system is measured in terms of precision, recall, F-measure, and accuracy for different numbers of drones.

Figure 4 shows the Timed Automata-based IDS: The system performs reasonably well with all three numbers of drones, achieving an average F-measure of 85.18% and an accuracy of 85.46%. The system shows a slightly better performance with 10 and 30 drones. Probabilistic Automata-based IDS: This system outperforms the other two systems with a significantly higher F-measure of 94.11% and an accuracy of 93.98%. The performance of the system slightly decreases with the increase in the number of drones. Timed Probabilistic Automata-based IDS: The system performs the best among all three IDSs with an F-measure of 99.10% and an accuracy of 99.14%. However, the system shows a significant decrease in performance with 30 drones.

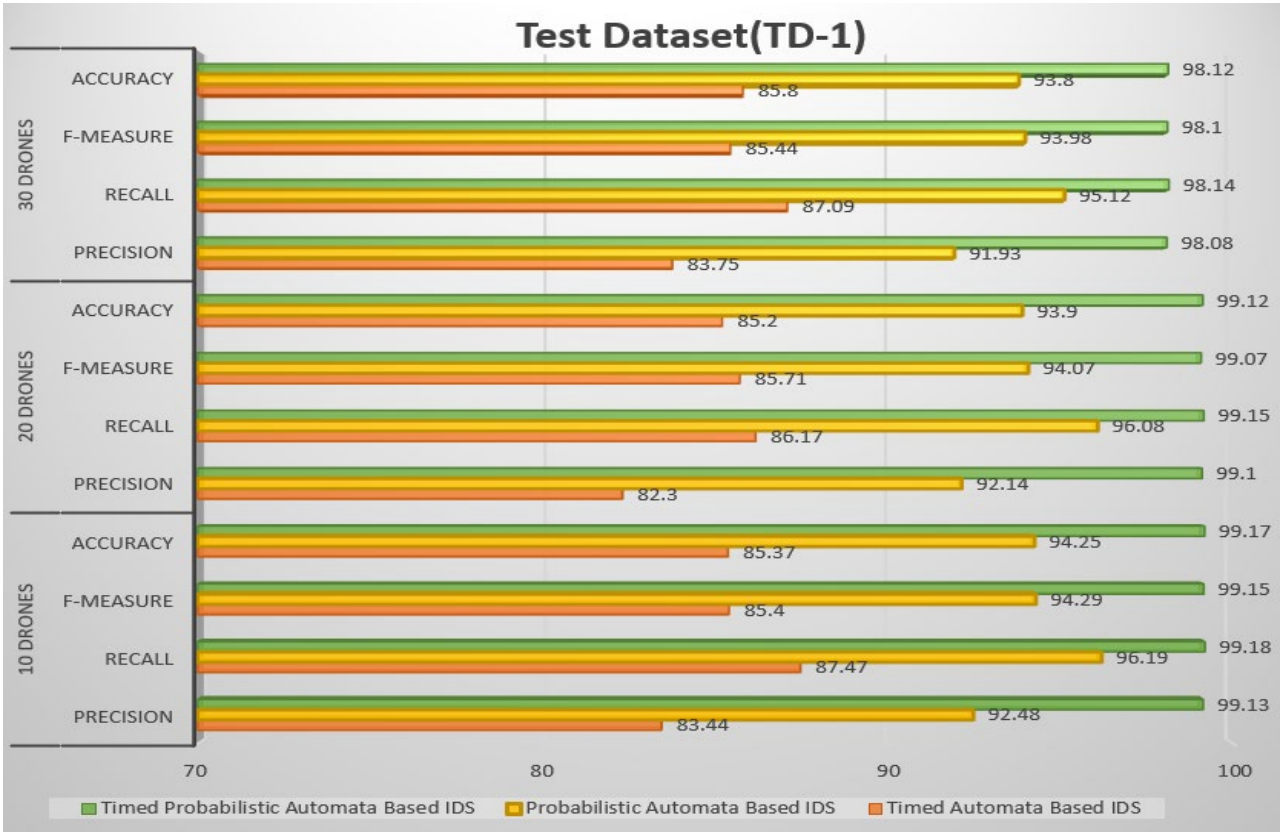

Figure 4. Performance metrics of our TPA-based IDS for test dataset-1.

Figure 5 shows the Timed Automata-based IDS: The system performs moderately well with an average F-measure of 75.70% and an accuracy of 75.79%. The performance of the system slightly decreases with an increase in the number of drones. Probabilistic Automata-based IDS: The system shows a better performance than the Timed Automata-based IDS, achieving an F-measure of 87.33% and an accuracy of 87.22%. The performance of the system slightly decreases with the increase in the number of drones. Timed Probabilistic Automata-based IDS: The system shows the best performance among all three IDSs with

an F-measure of 98.60% and an accuracy of 98.64%. The performance of the system slightly varies with the number of drones, with the best performance at 20 drones.

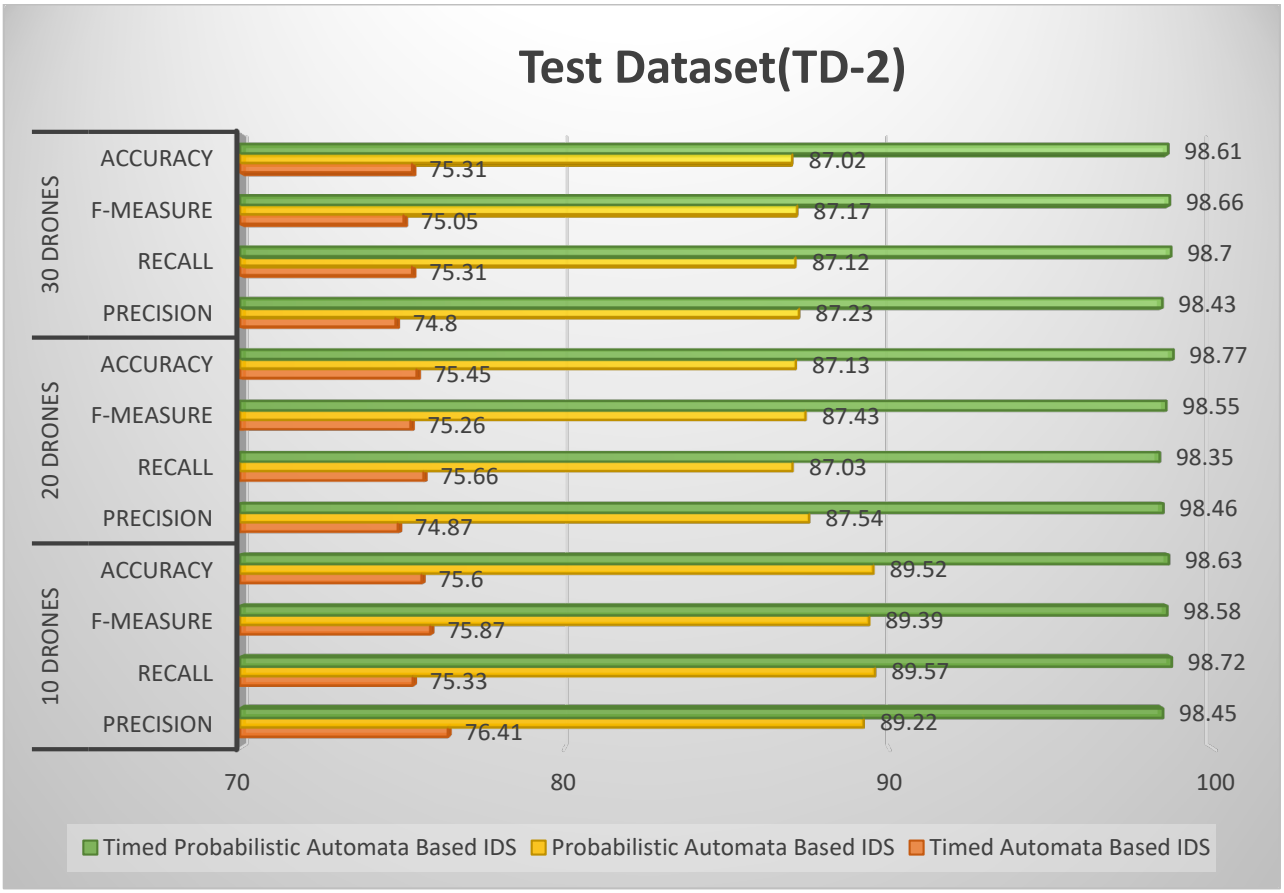

Figure 5. Performance metrics of our TPA-based IDS for test dataset-2.

Figure 6 shows the Timed Automata-based IDS: The system performs moderately well with an average F-measure of 85.31% and an accuracy of 85.43%. The performance of the system slightly varies with the number of drones, with the best performance at 10 drones. Probabilistic Automata-based IDS: The system performs relatively better than the Timed Automata-based IDS, achieving an F-measure of 94.37% and an accuracy of 94.32%. The performance of the system slightly varies with the number of drones, with the best performance at 20 drones. Timed Probabilistic Automata-based IDS: The system shows the best performance among all three IDSs with an F-measure of 98.28% and an accuracy of 98.24%. The performance of the system slightly varies with the number of drones, with the best performance at 10 drones.

Overall, each of these models has its own strengths and weaknesses when it comes to detecting malicious events in drone swarms. Timed automata are useful for systems with time-dependent behaviors, probabilistic automata are useful for systems with probabilistic behaviors, and time probabilistic automata are useful for systems with both temporal and probabilistic behaviors. Ultimately, the choice of model will depend on the specific characteristics of the system being analyzed and the types of malicious events that are being targeted. Abstraction techniques are used to reduce the state the exhaustive nature in our proposed TPA-based IDS in drone swarming. It involves grouping together similar states or behaviors in the TPA to create a smaller and more manageable model. This reduces the number of states in the TPA and improves the efficiency of the IDS.

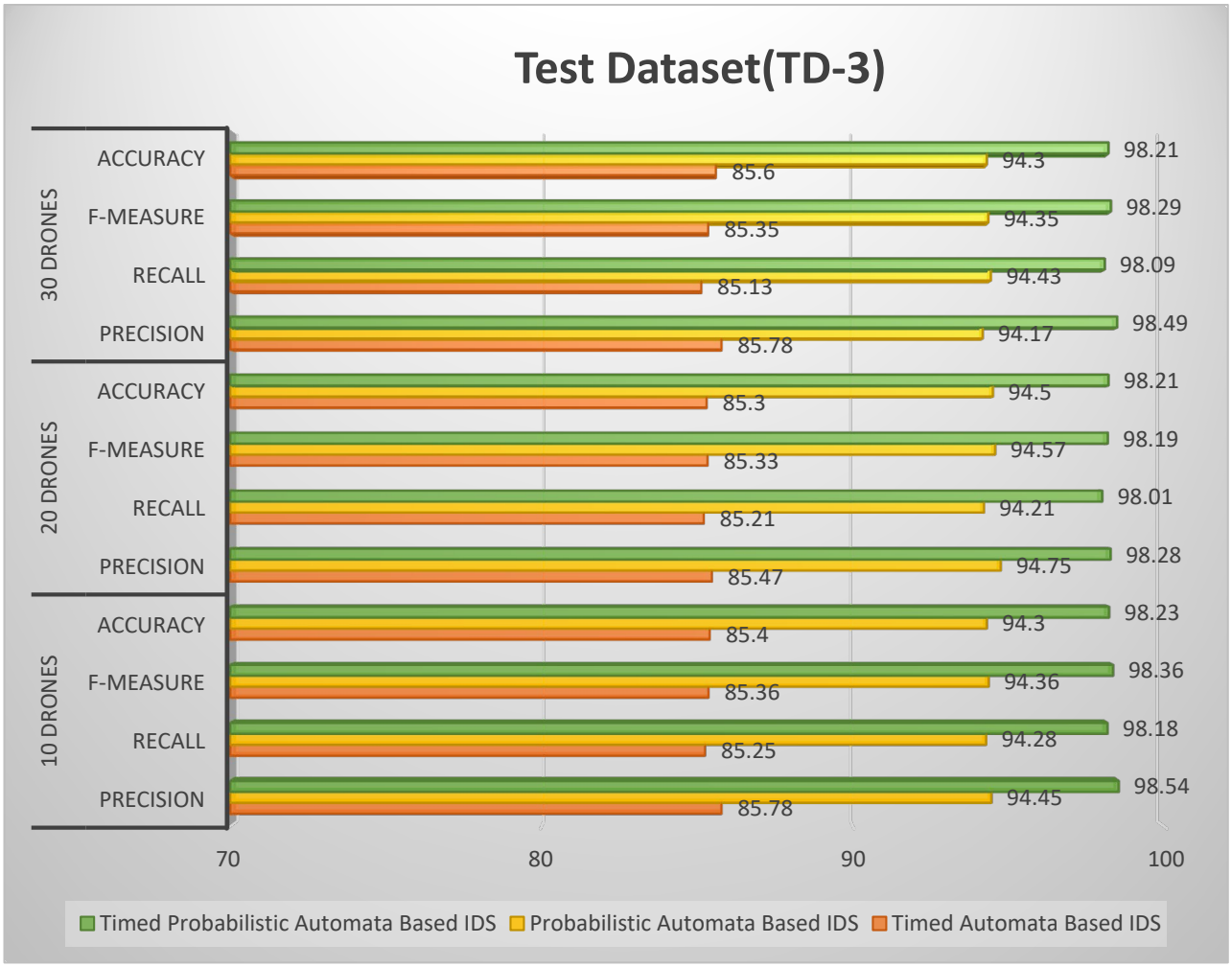

Figure 6. Performance metrics of our TPA-based IDS for test dataset-3.

6. Conclusions

Our proposed TPA-based intrusion detection system (IDS) is designed to safeguard drone swarming against various types of attacks, including Playback, DDoS, Zero-day, Mischievous series assaults, Hijacking, and Spoofing-Jamming assaults. The IDS operates in the ground control station of the drone swarm, which eliminates resource limitations and provides ample capacity to detect new and complex attack scenarios that may arise. The proposed algorithm for the intrusion detection system (IDS) is evaluated on three test datasets using three different models: timed automata-based IDS, probabilistic automata-based IDS, and timed probabilistic automata-based IDS. The performance of the algorithm is measured in terms of precision, recall, F-measure, and accuracy, and the results are presented for different numbers of drones (N), ranging from 10 to 30 drones. The results show that the timed probabilistic automata-based IDS outperforms the other models for all test datasets and all values of N. The algorithm achieved the high accuracy, precision, recall, and F-measure, indicating its effectiveness in detecting intrusions in drone swarming. These results suggest that the proposed algorithm is a promising approach for developing an efficient and reliable IDS for drone swarming.

Author Contributions: Conceptualization, methodology, project administration, supervision, formal analysis, investigation, V.S.; writing—review and editing, software, validation, visualization, M.A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Vellore Institute of Technology Chennai 600127, India.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Coppola, M.; McGuire, K.N.; De Wagter, C.; De Croon, G.C. A survey on swarming with micro air vehicles: Fundamental challenges and constraints. *Front. Robot. AI* **2020**, *7*, 18. [CrossRef] [PubMed]
2. Lowe, C.M.; Sarnell, J.A.; Stuehrmann, L.G.; Schwartzman, M.C.; Robbins, T.J. *A Heterogeneous Swarm Solution for Disaster Reconnaissance: A Feasibility Study*; Worcester Polytechnic Institute: Worcester, MA, USA, 2018; pp. 1–33.
3. Tahir, A.; Böling, J.; Haghbayan, M.H.; Toivonen, H.T.; Plosila, J. Swarms of unmanned aerial vehicles—A survey. *J. Ind. Inf. Integr.* **2019**, *16*, 100106. [CrossRef]
4. Al-Naji, A.; Perera, A.G.; Mohammed, S.L.; Chahl, J. Life Signs Detector Using a Drone in Disaster Zones. *Remote Sens.* **2019**, *11*, 2441. [CrossRef]
5. Gladence, L.M.; Anu, V.M.; Anderson, A.; Stanley, I.; Revathy, S. Swarm Intelligence in Disaster Recovery. In Proceedings of the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 6–8 May 2021; pp. 1–8.
6. Jeon, C.; Ha, J.; Ko, H.; Lee, B.; Ryu, B. Swarmsense: Effective and Resilient Drone Swarm and Search for Disaster Response and Management Application. Available online: https://www.wirelessinnovation.org/assets/Proceedings/2019/TS6.2%20Jeon%20presentation.pdf (accessed on 15 March 2023).
7. Subbarayalu, V.; Surendiran, B.; Kumar, P.A.R. Hybrid Network Intrusion Detection System for Smart Environments Based on Internet of Things. *Comput. J.* **2019**, *62*, 1822–1839. [CrossRef]
8. Alfeo, A.L.; Cimino, M.G.; Vaglini, G. Enhancing biologically inspired swarm behavior: Metaheuristics to foster the optimization of UAVs coordination in target search. *Comput. Oper. Res.* **2019**, *110*, 34–47. [CrossRef]
9. Asbach, J.; Chowdhury, S.; Lewis, K. Using an Intelligent UAV Swarm in Natural Disaster Environments. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec, QC, Canada, 26–29 August 2018; American Society of Mechanical Engineers: New York, NY, USA, 2018; Volume 51753, p. V02AT03A013.
10. Queralta, J.P.; Qingqing, L.; Gia, T.N.; Truong, H.L.; Westerlund, T. End-to-end design for self-reconfigurable heterogeneous robotic swarms. In Proceedings of the 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina del Rey, CA, USA, 25–27 May 2020; pp. 281–287.
11. Peng, J.; Zhang, Z.; Wu, Q.; Zhang, B. Anti-Jamming Communications in UAV Swarms: A Reinforcement Learning Approach. *IEEE Access* **2019**, *7*, 180532–180543. [CrossRef]
12. Fabra, F.; Wubben, J.; Calafate, C.T.; Cano, J.C.; Manzoni, P. Efficient and coordinated vertical takeoff of UAV swarms. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–5.
13. Chen, X.; Tang, J.; Lao, S. Review of unmanned aerial vehicle swarm communication architectures and routing protocols. *Appl. Sci.* **2020**, *10*, 3661. [CrossRef]
14. Hildmann, H.; Kovacs, E.; Saffre, F.; Isakovic, A.F. Nature-Inspired Drone Swarming for Real-Time Aerial Data-Collection Under Dynamic Operational Constraints. *Drones* **2019**, *3*, 71. [CrossRef]
15. Kusyk, J.; Uyar, M.U.; Ma, K.; Samoylov, E.; Valdez, R.; Plishka, J.; Hoque, S.E.; Bertoli, G.; Boksiner, J. Artificial intelligence and game theory controlled autonomous UAV swarms. *Evol. Intell.* **2020**, *14*, 1775–1792. [CrossRef]
16. Arnold, R.D.; Yamaguchi, H.; Tanaka, T. Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *J. Int. Humanit. Action* **2018**, *3*, 18. [CrossRef]
17. Olfati-Saber, R. Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. *IEEE Trans. Autom. Control* **2006**, *51*, 401–420. [CrossRef]
18. Lawton, J.; Beard, R.; Young, B. A decentralized approach to formation maneuvers. *IEEE Trans. Robot. Autom.* **2003**, *19*, 933–941. [CrossRef]
19. V, S.S.; Parasuraman, R.; Pidaparti, R. Impact of Heterogeneity in Multi-Robot Systems on Collective Behaviors Studied Using a Search and Rescue Problem. In Proceedings of the 2020 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Abu Dhabi, United Arab Emirates, 4–6 November 2020; pp. 290–297. [CrossRef]
20. Ramadan, R.A.; Emara, A.-H.; Al-Sarem, M.; Elhamahmy, M. Internet of Drones Intrusion Detection Using Deep Learning. *Electronics* **2021**, *10*, 2633. [CrossRef]
21. Jiang, R.; Zhou, Y.; Peng, Y. A review on intrusion drone target detection based on deep learning. In Proceedings of the 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 18–20 June 2021; Volume 4, pp. 1032–1039.
22. Xiao, W.; Li, M.; Alzahrani, B.; Alotaibi, R.; Barnawi, A.; Ai, Q. A Blockchain-Based Secure Crowd Monitoring System Using UAV Swarm. *IEEE Netw.* **2021**, *35*, 108–115. [CrossRef]

23. Restituyo, R.; Hayajneh, T. Vulnerabilities and Attacks Analysis for Military and Commercial IoT Drones. In Proceedings of the 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 8–10 November 2018; pp. 26–32. [CrossRef]
24. Condomines, J.-P.; Zhang, R.; Larrieu, N. Network intrusion detection system for UAV ad-hoc communication: From methodology design to real test validation. *Ad Hoc Netw.* **2019**, *90*, 101759. [CrossRef]
25. Guerber, C.; Royer, M.; Larrieu, N. Machine Learning and Software Defined Network to secure communications in a swarm of drones. *J. Inf. Secur. Appl.* **2021**, *61*, 102940. [CrossRef]
26. Miao, Y.; Hwang, K.; Wu, D.; Hao, Y.; Chen, M. Drone Swarm Path Planning for Mobile Edge Computing in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, 1–11. [CrossRef]
27. Asaamoning, G.; Mendes, P.; Rosário, D.; Cerqueira, E. Drone Swarms as Networked Control Systems by Integration of Networking and Computing. *Sensors* **2021**, *21*, 2642. [CrossRef]
28. Xue, R.; Zhao, M. Cognitive-Based High Robustness Frequency Hopping Strategy for UAV Swarms in Complex Electromagnetic Environment. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 4139345. [CrossRef]
29. Pardhasaradhi, B.; Cenkeramaddi, L.R. GPS Spoofing Detection and Mitigation for Drones Using Distributed Radar Tracking and Fusion. *IEEE Sens. J.* **2022**, *22*, 11122–11134. [CrossRef]
30. Feng, Z.; Ji, L.; Zhang, Q.; Li, W. Spectrum Management for MmWave Enabled UAV Swarm Networks: Challenges and Opportunities. *IEEE Commun. Mag.* **2018**, *57*, 146–153. [CrossRef]
31. Krichen, M. Improving Formal Verification and Testing Techniques for Internet of Things and Smart Cities. *Mob. Netw. Appl.* **2019**, 1–12. [CrossRef]
32. Krichen, M.; Lahami, M.; Cheikhrouhou, O.; Alroobaea, R.; Maâlej, A.J. Security testing of in-ternet of things for smart city applications: A formal approach. In *Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies*; Springer: Cham, Switzerland, 2020; pp. 629–653.