A Zero-Trust Model for Intrusion Detection in Drone Networks

Said OUIAZZANE, Malika ADDOU, Fatimazahra BARRAMOU
ASYR RT, LaGeS Laboratory, Hassania School of Public Works, Morocco

***Abstract*—Today's worldwide introduction of drone fleets in a range of industrial applications has led to numerous network security issues, opening drones up to cyberthreats. In response to these challenges, an innovative approach has been proposed to protect drone fleet networks against potentially dangerous cyberattacks. Indeed, drones are considered as flying computers, and the proposed approach takes into account their complex network structure and communication protocols. The proposed system is designed around a multi-agent architecture, with a hybrid zero-trust detection mechanism against known and emerging cyberthreats. The CICIDS2017 dataset was exploited after performing some essential pre-processing tasks including data cleaning, balancing, binarization and dimension reduction. The proposed approach guaranteed high levels of accuracy and scalability, enabling an effective response to potentially dangerous cyber threat scenarios threatening drone fleets. To evaluate the effectiveness of the proposed system, a test portion of CICIDS2017 was used. The accuracy in recognizing benign network traffic reached 99.99% with a very low false alarm rate, ensuring the system's effectiveness against known and unknown cyber threats. Extensive experimental testing has been carried out on never-before-seen data, highlighting the system's remarkable ability to rapidly recognize cyber threats in real time, thereby enhancing the overall security of drone networks. The contribution of the proposed approach is significant for drone network security, as it introduces a comprehensive model designed to meet the specific security requirements of drone fleets. Finally, the proposed approach offers practical prospects for improving the security of drone applications.**

***Keywords*—Fleet of drones; security; zero trust; intrusions; cybersecurity; zero day; Multi-Agent**

I. Introduction

Over the past few years, the use of drones has grown dramatically, as the Federal Aviation Administration estimates that there are between 2 and 7 million consumer drones in use by 2020 [1]. This major growth in drone usage is reflected in the projected trajectory of the drone industry, which is expected to reach an impressive $30 billion by 2036 [2]. The widespread accessibility and affordability of consumer drones has facilitated their adoption by hobbyists and enthusiasts alike. Consequently, airspace has seen a surge in unmanned aerial vehicles, leading to radical changes in aviation practices and raising important questions about security, privacy and regulatory measures [3].

Drones and drone fleets have become priceless assets for critical and sensitive situation management in a variety of fields. For example, their role in border surveillance is vital for strengthening security and control through the detection of illicit activities [4]. And when it comes to critical infrastructure, drones carry out inspections to identify vulnerabilities in assets such as power lines, bridges and pipelines, thus promoting early detection and preventive maintenance [5]. As for police forces, they use drones to monitor crowds at public events and investigate crime scenes, keeping officers safe. In addition, drones can also contribute to damage assessment, survivor location and rescue operations by providing real-time aerial data during disasters. For environmental monitoring, drones investigate sensitive areas and nature reserves to combat illegal activities, while studying the behavior of flora and fauna [6]. Finally, drones can contribute to healthcare by rapidly delivering medical supplies to inaccessible regions [7].

So, as mentioned, drones are therefore used to deal with more critical situations, as any bypassing of the drone network security can result in material and human damage, and even endanger human lives [8]. Nevertheless, the scientific community focused on communication protocols, battery autonomy and drone network use cases, and ignored the security aspect of this widely used technology [9].

As a result, addressing the security aspect is a crucial point to be taken very seriously by the scientific community, in order to protect drone networks against emerging cyber-attacks. The present work aims to propose a new intrusion detection approach based on a multi-agent architecture with a hybrid detection mechanism respecting the zero-trust principle.

The remainder of this paper is structured as follows: In Section II, we describe the background to the study, defining some concepts relevant to our study. Section II examines the current state of the art regarding the security of UAV networks and several proposed approaches to deal with intrusion detection in UAV networks. Section III presents an in-depth exploration of the proposed IDS architecture, detailing its components and its operational principle. Section IV describes the simulation and testing methodology of the proposed system. Section V delves into results and discussion and Section VI concludes the paper.

A. Drone

A "drone" is any unmanned aircraft, commonly known as an unmanned aerial vehicle (UAV), unmanned aerial system (UAS) or unmanned combat aerial vehicle (UCAV) [10]. For the purposes of this discussion, we focus on consumer/recreational UAVs, as illustrated in Fig. 1.

Fig. 1. Examples of consumer drones.

B. Fleet of Drones

1) Definition of a fleet of drones: A drone fleet is a coordinated and synchronized set of drones that cooperate to accomplish a given mission [11]. UAVs can be configured with various functionalities to satisfy a wide range of user needs. Managing a fleet of drones can involve complex operations, such as centralized control, mission planning, inter-drone communication and real-time data collection [12]. There are a wide variety of possible deployments for drone fleets, from surveillance and inspection to precision farming, logistics and delivery, and disaster relief. [13]

2) Communication modes of a fleet of drones: A drone fleet can be designed based on four possible communication architectures: centralized communication architecture, cellular communication architecture, satellite communication architecture and adhoc communication architecture [14] [15].

a) Satellite communication architecture: This architecture involves establishing communications between drones via satellite connections as shown in Fig. 2. Such links offer global coverage and are therefore particularly well suited to applications requiring wide coverage in remote or extensive areas. [16]

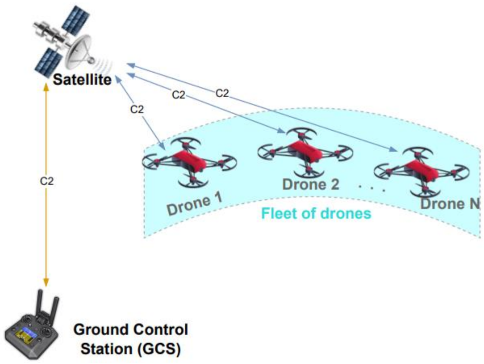


Fig. 2. Satellite communication architecture.

b) Ad hoc communication architecture: This type of communication is based on cooperation between drones in a dynamic, autonomous network, with no dependence on a static infrastructure, as shown in Fig. 3. Nearby drones interconnect autonomously, contributing to decentralized communications and collaborative data exchange, especially in scenarios where there is no conventional infrastructure. [17]

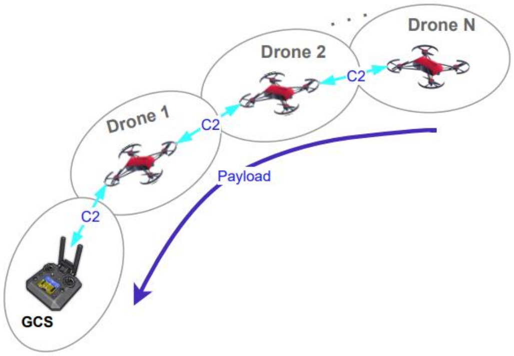


Fig. 3. Ad-hoc communication architecture.

c) Cellular communication architecture: A cellular architecture exploits existing cellular networks to ensure interaction between drones and ground stations as demonstrated by Fig. 4. Acting as roving nodes in the cellular network, drones ensure stable communication over extended distances, especially in urban environments or densely populated areas with established cellular coverage [18].

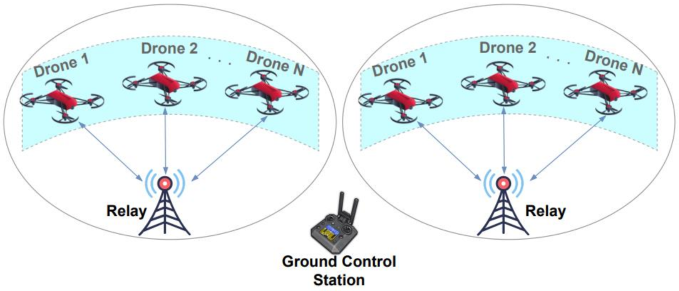


Fig. 4. Cellular communication architecture.

d) Centralized communications architecture: As shown in Fig. 5, this type of architecture relies primarily on a ground control station to manage communications. This configuration enables simplified coordination, efficient data processing and immediate decision-making, for scenarios requiring centralized control and supervision of drone fleets. [19]

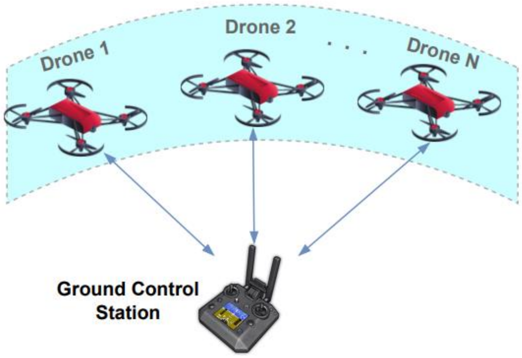


Fig. 5. Centralized communication architecture.

II. State of the Art

A. An Overview of Dangerous Tactics Targeting Drone Networks

Drone networks are vulnerable to a wide range of cyber-attacks. These attacks are all based on well-thought-out strategies, enabling hackers to achieve their malicious objectives, sometimes with life-threatening consequences. These attacks include:

- Communications jamming: The aim of communications jamming attacks is to disrupt or interfere with communications between UAVs and their control stations. Attackers can thus disrupt communication signals and cause loss of control, with serious consequences for ongoing operations. [20]
- DoS and DDoS attacks: This kind of attacks floods the UAV network with a large number of malicious requests, making it unavailable for legitimate communications. These attacks may lead to the interruption of critical business operations or even paralyze the network. [21]
- Data interception: Interception attacks capture sensitive data exchanged between drones and their associated control stations. This can lead to the exposure of confidential data and compromise mission confidentiality. [22]
- Usurpation of control: Spoofing is a serious threat that allows hackers to take control of a drone remotely, bypassing the legitimate control system. This can lead to malicious use of the drone for illegal or dangerous purposes. [23]

B. Attack Surface and Compromising Risk of a Drone Network

Drone networks are designed to cover large geographical areas and rely mainly on WiFi and radio waves as their communications medium [24]. Indeed, various types of communication can be involved in a network of drone fleets (see Fig. 6), including ad-hoc exchanges between drones, interactions with the control station and satellite links for GPS. Consequently, the attack surface of a drone network is considerable, exposing these networks to potential cybersecurity risks [25]. Given that anyone in the vicinity with a WiFi antenna/packet sniffer can potentially attempt to compromise the security principles of drone networks.

Drone manufacturers and researchers have mainly focused on developing communication protocols and improving battery life, while often ignoring the security aspect of drone networks. If the security of a drone network is compromised, numerous security risks can arise [9]. For example, a hacker could take control of the drone and gain access to its payload and the sensitive information it carries [21]. The hacker could also bring down the drone, resulting in property damage and the loss of the aircraft. In addition, the hacker could take control of the drone for malicious purposes, or integrate his own drone into the fleet, thus disrupting delivery and causing a denial-of-service issue [26].

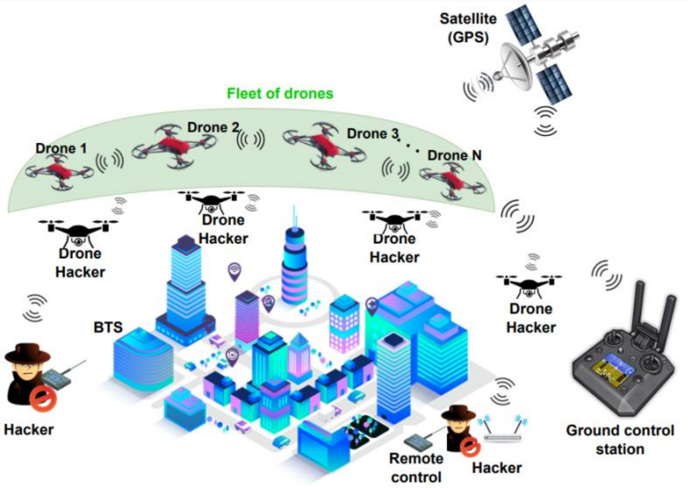


Fig. 6. Overview of a drone network and its attack surface.

C. Related Work

Recent research efforts in the field of intrusion detection systems for UAV fleets have seen the emergence of new, innovative approaches, each addressing distinct security issues and employing diversified methodologies. Fotohi, Abdan and Ghasemi (2022) [27] presented SID-UAV, an innovative system designed to counter the security risks associated with malicious drones, particularly in the context of drone-to-drone communication. In addition, Shrestha et al (2021) [28] proposed a UAV- and satellite-based 5G-network security model that can harness machine learning to effectively detect vulnerabilities and cyberattacks within a drone network. The proposed approach was suitable for both drone and satellite connections. Ouiazzane et al. (2022) [9] proposed a multi-agent intrusion detection system primarily designed according to a multi-agent architecture to counter denial-of-service (DoS) attacks targeting drone networks. The authors demonstrated remarkable accuracy in detecting DoS attacks while minimizing false alarms, underlining the system's effectiveness in protecting drone networks against a spectrum of known and unknown threats. Ihekoronye et al. (2022) [29] introduced a hierarchical intrusion detection system based on anomaly detection, suggesting an effective strategy for securing military UAV networks. They employed random cross-validation and finely tuned hyperparameters, guaranteeing resilience to network delays. The system considers payload constraints, battery limitations and the high mobility that characterizes Internet of Drones (IoD) networks. In a deep learning context, Abu Al-Haija and Al Badawi (2022) [30] proposed an autonomous intrusion detection approach adapted to drone networks. Such an approach makes use of deep convolutional neural networks to discern malicious activities within drone networks, efficiently processing encrypted Wi-Fi data records from commonly used drone brands such as Parrot, DBPower and DJI Spark drones.

Finally, Ouiazzane et al. took up the challenge of intrusion detection in UAV fleet networks in their 2020 study [14], focusing on ad hoc communication architectures. Their multi-agent system, enriched with detection mechanisms based on machine learning, underlines the importance of adaptable, intelligent security measures in the constantly evolving context of drone fleet networks. These studies have made a significant contribution to the advancement of intrusion detection systems designed to meet the specific security challenges inherent in drone networks.

D. Discussion of Related Work Limitations

The state-of-the-art study carried out on drone security aims to understand the literature surrounding the topic. The security of a drone fleet is rarely addressed, even though these fleets represent the future trend in the use of drones for civilian missions. Most of the work cited in the state of the art has focused on routing protocols, autonomy optimization, and communication architectures while ignoring the security aspect, to which particular attention must be paid given the disastrous damage likely to occur whenever fleet security principles are successfully circumvented.

Little research has been carried out on the problem of intrusion detection in drone fleet networks. In fact, the approaches proposed by the community only address an aspect of the architecture or intrusion detection mechanism, and no complete work tackles all the aspects in question. Furthermore, there are no effective datasets for dealing with the problem of intrusion detection in drone fleets. In addition, even work based on machine learning sometimes uses polluted or obsolete datasets that fail to represent the real network traffic of a drone fleet. These factors lead to more attractive research avenues to further strengthen the security of drone fleet networks.

A drone, like a computer, is made up of a set of fundamental components. Among these, the central processing unit (CPU) and random-access memory (RAM) provide the computing power needed to execute the drone's tasks and missions. In addition, for communication with ground controllers and other aircraft, drones use diverse transmission methods, including WIFI and radio waves. Drones are also equipped with cameras to capture images and record video in their environment. Data storage is essential for drones to retain essential operating systems and files. Furthermore, drones are equipped with built-in sensors, such as GPS, that provide essential information on position and orientation. In the same way that computers depend on power sources, drones rely on batteries for their energy needs. Finally, drones integrate specialized aeronautical hardware enabling them to navigate in the air, while computers employ control peripherals for hands-on operations. To sum up, the analogy between drones and computers underlines the similarity of their components, asserting that the drone is a veritable flying computer.

Cyber security is becoming a major concern for drones and drone fleets [31]. Given their vast attack surface and their reliance on wifi networks and radio waves, these devices are vulnerable to a whole range of cyberattacks.

Like computer networks, drone networks are exposed to potential cybersecurity risks that can jeopardize the security of this widely used technology, essential for carrying out vital missions. In our approach, we see drones as flying computers, and therefore we address their security issues in a similar way to traditional computer network security practices.

III. Proposed Approach

A. Architecture of the Proposed ZT-NIDS System

In this research project, ZT-NIDS (Zero Trust-based Network Intrusion Detection System), a new network intrusion detection system, is introduced. The proposed system is designed with a multi-agent architecture consisting of a set of independent and cooperative agents working together to efficiently detect intrusions in drone. Fig. 7 illustrates the overall architecture of the proposed ZT-NIDS system.

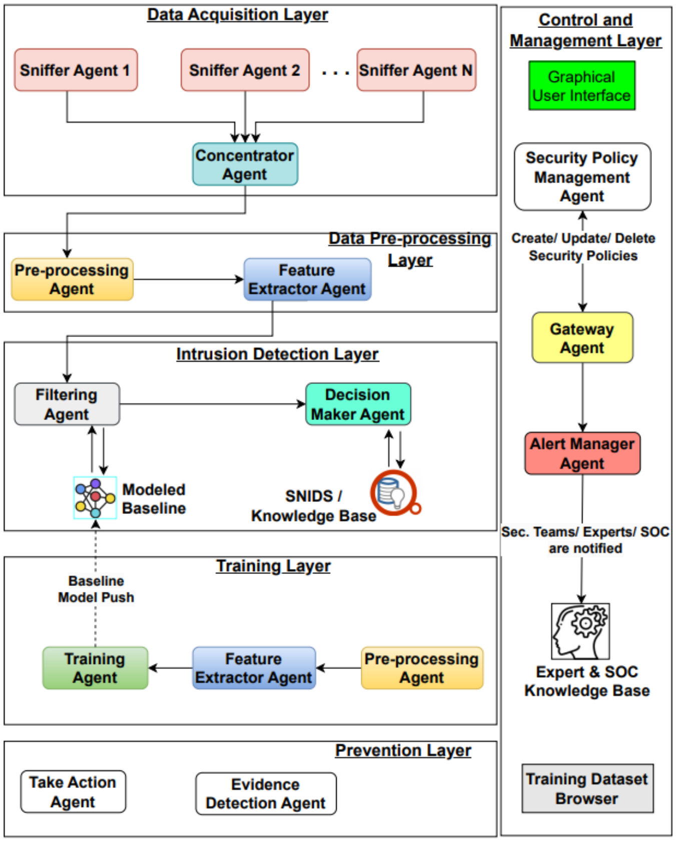


Fig. 7. The architecture of the proposed HNID&PS system.

B. Components of the Proposed Model

The ZT-NIDS system is designed according to a multi-agent architecture and consists of six layers, each with a set of agents.

1) Data Acquisition Layer – DAL: This layer is the input interface to the system, capturing and aggregating traffic from UAV fleet networks. It is made up of two types of agents:

- Sniffer agent: This agent captures drone network traffic in real time. Each sniffer can be positioned to cover a segment of the drone network.
- Concentrator agent: it aggregates all network events originating from all sniffers and timestamps all network events.

2) Data Pre-processing Layer – DPL: The DPL layer pre-processes captured network traffic to clean and process it. In this layer, we consider two types of agents:

- Pre-processing Agent: It cleans, normalizes, and correlates network traffic originating from the concentrator agent to eliminate missing and infinite values.

- Feature Extraction Agent: Its aim is to extract relevant features and reduce the dimensionality of network traffic, while retaining a maximum amount of information for recognizing network packets. To do this, it removes correlated, constant and quasi-constant attributes and passes the extracted features on to the adjacent layer.

3) Intrusion Detection Layer – IDL: The IDL layer receives the extracted attributes and compares them with the baseline and the signature database. It includes a set of two agents:

- Filtering Agent: It receives network packet attributes and checks the match values first against the baseline to recognize normal network traffic, then against the signature database to identify known attacks.

- Decision Making Agent: This agent is responsible for making decisions about the detected intrusions and their severity; it intervenes in accordance with the security policy governing the detection of abnormal activities on drone networks; it generates alerts and sends them to the Alert Manager Agent; it sends orders to the prevention layer, for example, to quarantine network equipment/systems contaminated by the intrusion.

4) Training Layer – TL: The TL layer is responsible for regularly training our system's detection mechanism. It includes three types of agents, the first two already presented: the Pre-processing Agent and the Feature Extraction Agent, plus a third, the Training Agent, described as follows:

- Training Agent: This agent is responsible for receiving pre-processed data sets as input; modeling the network baseline to recognize normal network behavior; generating and updating the modeled baseline module of the intrusion detection layer.

5) Prevention Layer – PL: The PL layer acts in the case of an intrusive attack through a set of two agents:

- Action agent: It is responsible for isolating and eliminating detected intrusions, e.g. by quarantining infected equipment; taking security measures, e.g. revoking access, blocking ports, suspending accounts, etc.

- Evidence Detection Agent: This agent is tasked with identifying the root cause of the security incident based on detailed information about the intrusion; it keeps a history of security incidents and all related documentation.

6) Control and Management Layer – CML: The CML layer enables IT security managers of UAV networks to define security policies and undertake configuration actions using three types of agents:

- Security Policy Management Agent: Managing and controlling the intrusion security policies; Updating machine learning models; Optimizing whitelists and blacklists with information on hacking sources; Managing threshold levels

- Gateway Agent: This agent is responsible for promoting communication and coordination between the various agents; managing data exchanges and communication protocols; coordinating response actions between agents in case of network intrusion; synchronizing agent activities for consistent system functioning.

- Alert Manager Agent: This agent is tasked with correlating alerts generated by the system, so that only relevant alarms are triggered; reducing the false positive rate; notifying security administrators in real time, so that they can anticipate and intervene in case of network intrusion.

C. ZT-NIDS Detection Mechanism / Zero-trust Principle

Since a drone is a sort of flying computer, the network configurations of drone fleets are comparable to those of modern computer networks. Consequently, the proposed intrusion detection system is well suited to overcoming the challenges inherent in cyber threats that could compromise the security of flying aircraft.

The diagram in Fig. 8 illustrates the deployment mode of the proposed system in a drone fleet network. The idea behind it is to monitor the flow of data between the various actors in a fleet of drones, then analyze it to identify known and unknown attacks. The system consists of a group of autonomous entities that work together to successfully accomplish the tasks involved in identifying intrusions into a drone network.

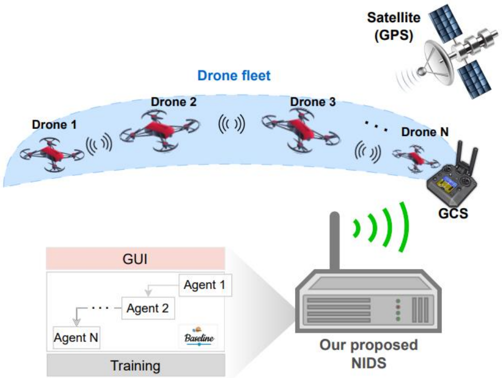


Fig. 8. Deployment mode of the proposed system.

A key element of the proposed system is its adherence to the "zero-trust" principle, which means that the detection mechanism works by considering a minimal level of trust in the processing of network traffic originating from drone fleets networks. To be more precise, the adopted detection mechanism is based on a network baseline as a reference, enabling us to identify any deviation from normal behavior. The diagram in Fig. 9 provides an overview of the system's detection mechanism. Its aim is to detect any attack attempts against drone networks, no matter whether they are already registered or not.

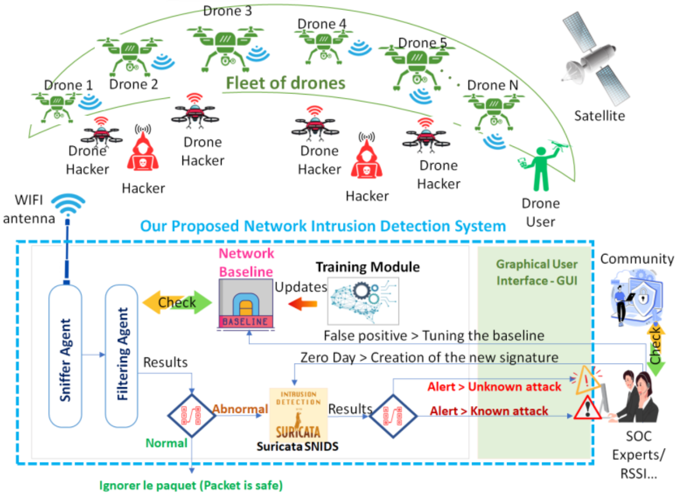


Fig. 9. Zero trust based intrusion detection mechanism.

For this purpose, the system is equipped with a baseline of normal behavior patterns, enabling it to distinguish legitimate transactions on the network. Any deviation from this basic pattern is interpreted as suspicious activity, triggering a notification to security managers. At the same time, the system has a standard signature database containing listed attack patterns. This signature database is continually updated as new attacks are identified through the process of comparing them with the network's reference baseline.

IV. Experimentation

A. Technical Architecture of the Lab used for Simulation

1) Lab environment: The entire laboratory is hosted by VMware on a workstation, whose characteristics are illustrated in Fig. 10.

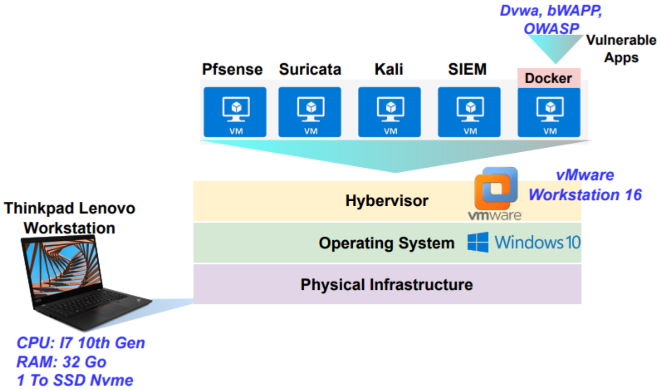


Fig. 10. Technical infrastructure hosting the ZT-NIDS experimental lab.

2) Applicative architecture of the Lab used for simulation: This laboratory aims to demonstrate the practical implementation of the ZT-NIDS system introduced in this research project. Fig. 11 gives a visual presentation of the laboratory's application architecture, highlighting the inputs and outputs of the platform designed to simulate the ZT-NIDS system.

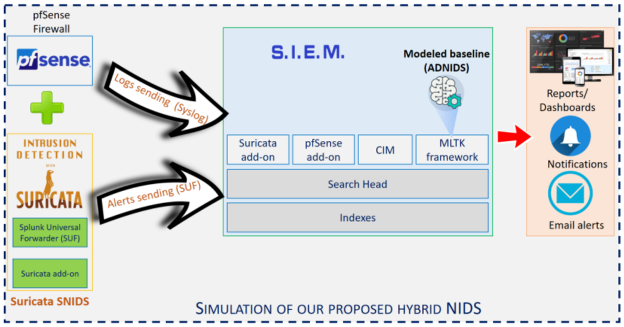


Fig. 11. Applicative architecture of the lab simulating the ZT-NIDS model.

The architecture diagram above shows two modules on the left-hand side: pfSense and Suricata. Both modules play a crucial role in intercepting, collecting and aggregating real-time events originating from drone networks. Suricata functions as a signature-based network intrusion detection system, and is either connected to a SPAN port on the switch, or to a network card operating in Promiscuous mode, enabling it to capture a copy of all ongoing network traffic in real time. The modules Splunk Universal Forwarder (SUF) and Suricata-Add-Ons have been deployed on the Suricata host to facilitate the extraction and recognition of the intercepted network events.

The network events collected are then directed to Splunk's SIEM (Security Information and Event Management) for processing, thanks to a number of additional modules. In particular, the Suricata-Addon, pfsense-Add-On and CIM modules integrated into the SIEM facilitate the normalization, indexing and storage of actionable network events, making it easier to extract attributes describing network packet behaviors.

In the SIEM, we have incorporated the MLTK framework to establish the baseline model for normal network operation. This enables the SIEM to recognize deviations from typical network behavior. With this configuration, Suricata excels at recognizing known attacks, while the Baseline module is specially designed to identify anomalies and previously unidentified zero-day attacks.

The SIEM is configurable to trigger alerts when real attacks are detected, and provides managers with comprehensive security indicators for in-depth analysis. In addition, it provides routine reports to administrators, giving them an overview of the security posture of the monitored networks.

B. Laboratory Components for Simulating the Proposed System

The practical simulation of our ZT-NIDS system relies on the set-up of a laboratory environment, as described in the previous section. This laboratory includes several components and software tools used to emulate the ZT-NIDS model proposed as part of our research. Table I below gives an overview of the components that make up our model and the simulation tools used.

C. Data Collection

Given that a drone is essentially a flying computer, and considering the similarity between traditional computer networks and those used by drones, the CICIDS2017 dataset was deliberately chosen to evaluate the effectiveness of the system. The main focus was on the benign traffic portion of the data, with the aim of creating a representative baseline of normal network operation. CICIDS2017 was chosen for its consistency and relevance to modern network behavior, which differentiates it from conventional data sets that are often more theoretical than practical.

TABLE I. THE PROPOSED MODEL COMPONENTS VS. SIMULATION TOOLS

Our ZT-NIDS model components	***Simulation tools (Lab)***
Sniffer Agent	Pfsense
Concentrator Agent	Pfsense
Preprocessing Agent	Splunk Universal Forwarder
	Splunk TA for Suricata
	Splunk Common Information Model
	TA-Pfsense
Feature Extraction Agent	Machine Learning ToolKit (MLTK)/ Python/ Splunk Add-ons
Filtering Agent	Splunk Search Head
Decision Maker Agent	Splunk SIEM
Training Agent	Python & Machine Learning ToolKit
Take Action Agent	Ansible
Security Policy Management Agent	Splunk Administration Interface
Gateway Agent	Splunk SIEM
Evidence Detection Agent	Splunk Search App
Alert Manager Agent	Splunk SIEM
Signature-based NIDS	Suricata NIDS
Graphical User Interface	Splunk Dashboard
Benign traffic/ Baseline	CICIDS2017/ benign traffic

1) CICIDS2017 dataset used for evaluating the proposed system: The Canadian Cybersecurity Institute has released the CICIDS2017 dataset to help researchers tackle the challenges of intrusion detection [33]. The CICIDS2017 dataset is available in two distinct formats: CSV files for learning purposes, and PCAP files to enable rigorous evaluation of detection mechanisms proposed by the community.

In this study, we exploited the CICIDS2017 dataset in CSV format to train the ZT-NIDS system in recognizing normal behaviors using machine learning algorithms. This process has created the Baseline module integrated into the architecture of the ZT-NIDS system.

To test the system under real-life conditions, the PCAP format was adopted. Several packets were replayed using tools such as tcpdump, Wireshark and Snort. These tools enable the system to intercept generated network traffic and automatically identify records corresponding to possible signs of attack.

The different data structures used in this case study are shown in Fig. 12. On the one hand, CSV format data sets are used to feed the learning module, enabling the network baseline to be established. On the other hand, PCAP files are used to simulate real network traffic, in order to test the system in a practical, authentic environment.

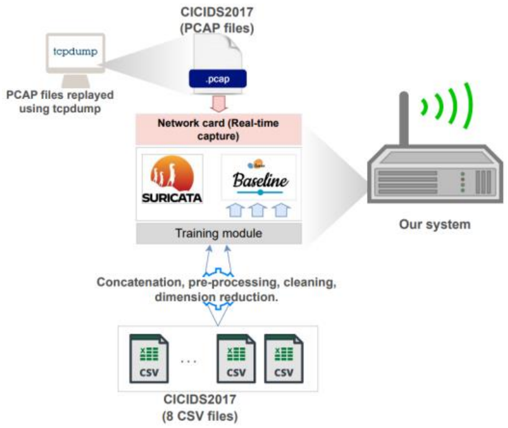

CICIDS2017
(PCAP files)
tcpdump
.pcap
PCAP files replayed using tcpdump
Network card (Real-time capture)
SURICATA
Baseline
Training module
Our system
Concatenation, pre-processing, cleaning, dimension reduction.
CSV
...
CSV
CSV
CICIDS2017
(8 CSV files)

Fig. 12. Dataset used to train and to test the proposed system.

2) CICIDS2017 dataset preprocessing

a) Composition of the intial CICIDS2017 dataset: The CICIDS2017 dataset has been chosen to model the network baseline, as it is reliable, up-to-date and can represent the modern real network traffic [33]. However, it poses certain cleaning, scaling and conversion problems for the use by machine learning algorithms [34]. Accordingly, pre-processing operations need to be undertaken before using the benign class to model the network baseline.

The CICIDS2017 dataset is multi-class in nature, including a "Benign" category reflecting regular network traffic. Additional categories are also included, representing distinct types of known attacks, as shown in Fig. 13.

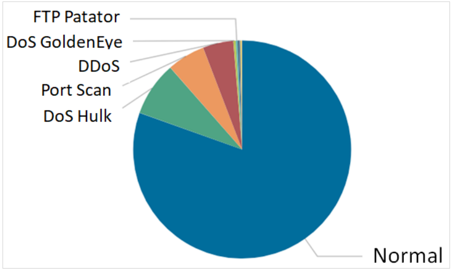


Fig. 13. Traffic classes contained in the intial CICIDS2017 dataset.

b) Cleaning up the CICIDS2017 dataset: The CICIDS2017 dataset suffers from problems of sanitisation, essentially due to the presence of infinite, null or sometimes missing records. Such problems can generally lead to falsified classification results during the machine learning process.

Some transformation processes can produce errors associated with undefined, infinite and oversized values. That's why we used certain Pandas methods like "drop()" to clean up the CICIDS2017 dataset of null, missing and infinite values.

c) Balancing of the CICIDS2017 dataset: The CICIDS2017 dataset is unbalanced regarding normal and abnormal records, as shown in Table II. Indeed, the class labeled "Normal" dominates over that labeled "Abnormal". The abnormal class refers to all the attack classes mentioned in Table I. As a result, over-fitting and under-fitting problems can be generated during the learning phase [35].

TABLE II. COMPOSITION OF THE DATASET BEFORE BALANCING

Class	Records Count	%
NORMAL	1818477	80 %
DoS Hulk	184858	8 %
Port Scan	127144	5.6 %
DDoS	102421	5.02 %
DoS GoldenEye	8234	0.36 %
FTP-Patator	6350	0.28 %
SSH-Patator	4718	0.2 %
DoS slowloris	4637	0.2 %
DoS Slowhttptest	4399	0.19 %
Bot	1573	0.07 %
Web Attack Brute Force	1206	0.05 %
Web Attack XSS	522	0.028 %
Infiltration	29	0.001 %
Web Attack Sql Injection	17	0.0007 %
Heartbleed	9	0.0003 %

In order to overcome the problems of overfitting and underfitting, the dataset has been balanced to ensure a balanced presence of the different classes. We used the SMOTE Python technique to increase the percentage of minority classes. On the other hand, the randomUnderSampler Python technique was used to decrease the number of normal records in order to avoid the performance problems that the laboratory environment might envisage. Accordingly, Table III highlights the dataset composition after its balancing using the two aforementioned python techniques.

TABLE III. COMPOSITION OF THE DATASET AFTER BALANCING

Class	Records Count	%
NORMAL	618000	50.2 %
DoS Hulk	240000	19.49 %
Port Scan	160000	12.99 %
DDoS	150000	12.18 %
DoS GoldenEye	12000	0.97 %
FTP-Patator	7000	0.56 %
DoS Slowhttptest	7000	0.56 %
DoS slowloris	7000	0.56 %
SSH-Patator	5000	0.4 %
Bot	5000	0.4 %
Web Attack Brute Force	5000	0.4 %
Web Attack XSS	5000	0.4 %
Infiltration	5000	0.4 %
Web Attack Sql Injection	5000	0.4 %
Heartbleed	5000	0.4 %

d) CICIDS2017 dataset binarization: The modeling of the network baseline is based on the class tagged "Normal/Benign" in the CICIDS2017 dataset. Therefore, the balanced dataset was transformed into another binary dataset including two main classes of network traffic: Normal and Abnormal. Python libraries were used and the binary pre-processed dataset comprises 618000 normal traffic records and 618000 abnormal records. Consequently, both classes are present with an equal percentage of 50% each, as shown in Fig. 14.

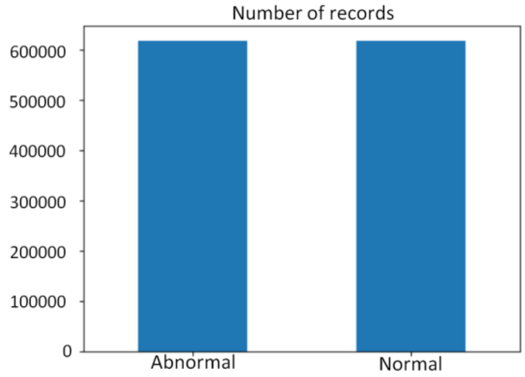


Fig. 14. Proportion distribution after balancing the binarized dataset.

e) Dimension reduction of the training dataset: The binarized balanced CICIDS2017 dataset contains a large number of attributes that are not necessarily relevant (79 attributes). This large number of attributes could cause enormous processing delays and could lead to falsified results when modeling the baseline. It makes more sense to eliminate unnecessary attributes and keep only the most relevant ones.

The StandardScaler imported from the sklearn.preprocessing library was used to downscale the features within the Pandas dataframe before applying the PCA transformation for dimension reduction. 30 principal components with zero cumulative variance were thus retained using the PCA technique, as shown in Fig. 15. This represents a considerable dimension reduction from 79 to 30 components that can describe 99.9% of the information within the standardized dataset.

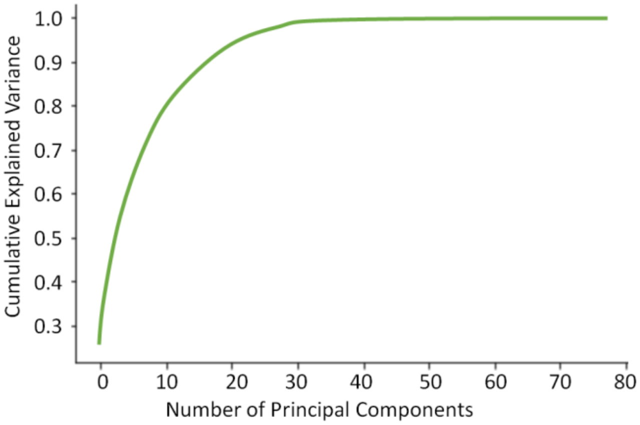


Fig. 15. PCA components used to represent the CICIDS2017 dataset.

D. Baseline Modelling and its Performance Evaluation

1) Network baseline modelling: After preprocessing the CICIDS2017 dataset, we used the “fit()” function in Splunk’s MLTK framework to train the system on normal network traffic and generate the baseline model as output.

The MLTK framework is based on the Scikit Learn python library to train machine learning models [36]. This framework is very powerful, fast and reliable, as it copies the events to be processed into memory before launching the pre-processing and training operations.

The sequence of actions undertaken by the "fit()" function is illustrated in Fig. 16 below, in order to generate a model capable of recognizing the network baseline.

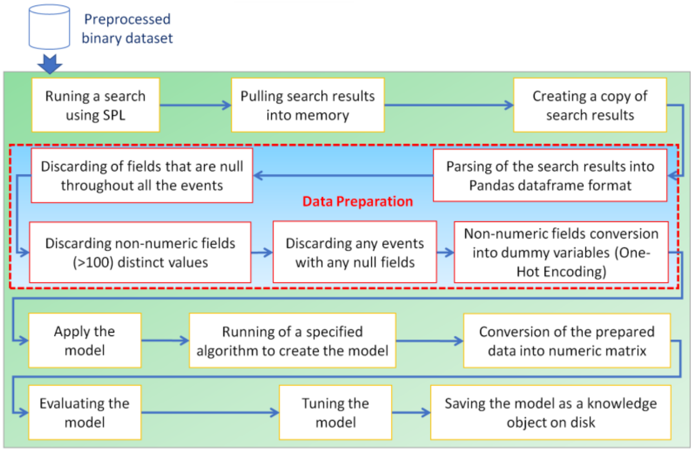

Preprocessed binary dataset
Runing a search using SPL
Pulling search results into memory
Creating a copy of search results
Discarding of fields that are null throughout all the events
Parsing of the search results into Pandas dataframe format
Data Preparation
Discarding non-numeric fields (>100) distinct values
Discarding any events with any null fields
Non-numeric fields conversion into dummy variables (One-Hot Encoding)
Apply the model
Running of a specified algorithm to create the model
Conversion of the prepared data into numeric matrix
Evaluating the model
Tuning the model
Saving the model as a knowledge object on disk

Fig. 16. Workflow of the training process using MLTK Framework applied on the optimized binary CICIDS2017 dataset.

2) *Machine learning algorithms used to train the baseline*

a) Machine learning algorithms and data sampling: After pre-processing and binarization of the CICIDS2017 dataset, a number of machine learning algorithms were used to build up a model capable of characterizing the network baseline. We used algorithms encompassing a variety of approaches, including Random Forest, Decision Tree, Naïve Bayes and Multi-Layer Perceptron. The key objective was to determine the most efficient and accurate machine learning algorithm that can differentiate between normal and abnormal drone's network traffic.

Having been pre-processed and binarized, the CICIDS2017 dataset was randomly separated into two sets: one for training (comprising 80% of the data) and the other for testing (comprising 20% of the data). In addition, PCAP files were used to evaluate system performance against real-time computer network data. These PCAP files were replayed using the tcpdump utility.

b) Performance evaluation metrics: The evaluation process incorporated key performance measures, such as Accuracy, Precision, Recall, F Score and Confusion Matrix. Importantly, cross-validation was applied with a k-fold value set at 10, guaranteeing a robust and reliable evaluation procedure.

Accuracy measures the number of correct predictions (both true positives and true negatives) that a model makes out of all predictions. It quantifies the overall accuracy of the model's predictions.

Precision measures the exactness of the positive predictions made by a model. It estimates the percentage of true positive predictions out of all positive predictions, and thus the model's ability to avoid false positives.

Recall, also known as sensitivity or true positive rate, measures the model's ability to identify all true positive instances. It refers to the proportion of true-positive predictions to all true-positive cases.

F1 score is the harmonic mean of precision and recall. It is a balanced measure that combines both precision and recall. It is particularly useful in unbalanced data sets where one class clearly outnumbers another.

The confusion matrix is a powerful tool for evaluating the performance of machine learning algorithms. According to [32] and as shown in Fig. 17, the confusion matrix is generally composed of four basic elements:

- True positives (TP): These are cases for which the model was able to correctly predict the positive class.
- True negatives (TN): These are cases for which the model was able to correctly predict the negative class.
- False positives (FP): These are Type I errors, in which the model has incorrectly predicted the positive class when it should have been negative.
- False negatives (FN): Also known as Type II errors, these correspond to cases where the model has incorrectly predicted the negative class when it should have been positive.

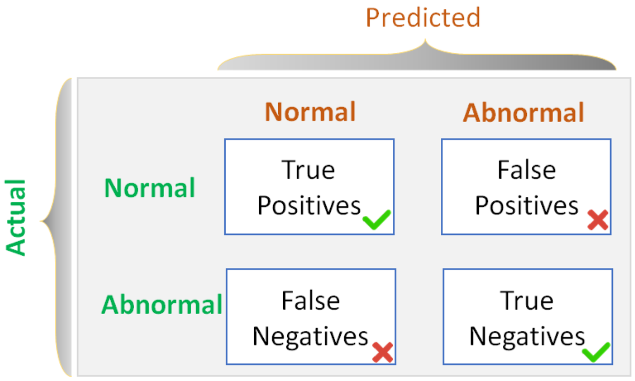


Fig. 17. Components of a confusion matrix (the normal class is our target).

3) Evaluation of the generated network baseline: To develop a highly efficient baseline network model, we tested a number of machine learning algorithms on the pre-processed and binarized CICIDS2017 dataset. In particular, we applied Random Forest, Decision Tree, Naïve Bayes and Multi-Layer Perceptron models.

a) Random Forest – RF: The Random Forest classifier was used to model the baseline of drone network traffic (regular operation of UAV networks). The test confusion matrix is shown in Table IV. The results obtained showed that the model generated was able to accurately recognize benign traffic with an accuracy, F1 score, precision and recall that all reached the exceptional level of 99.99%.

TABLE IV. RANDOM FOREST – TESTING CONFUSION MATRIX

	Predicted NORMAL	Predicted ABNORMAL
NORMAL	247194	6
ABNORMAL	10	247190

b) Decision Tree – DT: The decision tree was tested for modeling the network baseline (benign traffic modeling) and the confusion matrix for the tests carried out is presented in Table V. The generated model was capable of accurately recognizing benign traffic. The measures of accuracy, F1 score, precision and recall all reached the exceptional level of 99.99%.

TABLE V. DECISION TREE – TESTING CONFUSION MATRIX

	Predicted NORMAL	Predicted ABNORMAL
NORMAL	247192	8
ABNORMAL	12	247188

c) Naïve Bayes – NB: Table VI shows the confusion matrix results derived from the evaluation of the Naïve Bayes algorithm. This algorithm was employed to model the network baseline, with the pre-processed and binarized CICIDS2017 dataset. The performance measures obtained include 99.85% in accuracy, 99.84% in F1 score, 99.87% in precision and 99.83% in recall.

TABLE VI. NAÏVE BAYES – TESTING CONFUSION MATRIX

	Predicted NORMAL	Predicted ABNORMAL
NORMAL	246899	301
ABNORMAL	405	246795

d) Multi-Layer Perceptron – MLP: Table VII shows the testing confusion matrix following the use of the Multi-Layer Perceptron algorithm to model the network baseline, focusing on the classification of benign network traffic. The results reveal a precision rate of 99.93%, with F1 score and recall both reaching 99.92%. In addition, the precision measure achieved a high accuracy level of 99.94%.

TABLE VII. MULTI-LAYER PERCEPTRON – TESTING CONFUSION MATRIX

	Predicted NORMAL	Predicted ABNORMAL
NORMAL	247070	130
ABNORMAL	195	247005

4) Baseline modeling – Discussion: The classification results of benign traffic using a set of machine learning algorithms on the CICIDS2017 dataset are extremely promising. For the most part, the algorithms demonstrated an exceptional capacity to recognize normal network traffic with high performance and accuracy. As a result, the model generated can be integrated into the ZT-NIDS system to ensure the detection of deviations from regular network traffic. The positive results obtained are mainly attributable to the pre-processing, balancing, binarization and dimension reduction actions carried out on the CICIDS2017 dataset prior to its use.

In this laboratory, the Decision Tree algorithm was used to model the benign network traffic. This model was then integrated into the ZT-NIDS system, founded on the zero-trust principle, meaning that no network packet should be assumed to be trustworthy. The aim is to distinguish what is considered normal, and any deviation from this basis is automatically considered suspicious.

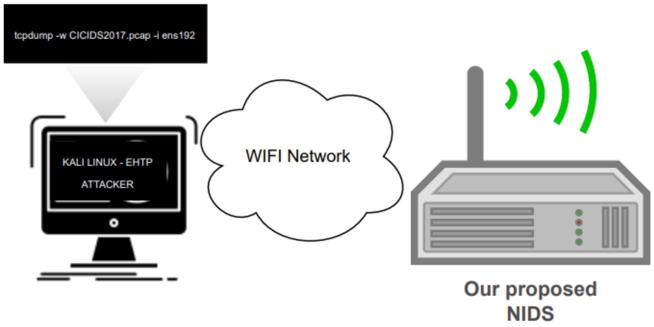
E. *Signature-based Module*

As mentioned earlier, the ZT-NIDS mechanism integrates both signature-based and anomaly-based modules enabling it to identify known and unknown cyber threats. Concerning the anomaly-based module, a baseline of benign network traffic has been created. The next focus is on the signature-based module, responsible for recognizing known intrusions.

We have already used Suricata as a signature-based module to detect known intrusions in one of our previous research works [34], and it has demonstrated excellent performance. We highlighted some of its advantages over Snort, such as its multithreading capability and its efficiency in handling known attacks. Consequently, for the ZT-NIDS mechanism, we used Suricata as a SNIDS (Signature-Based Network Intrusion Detection System) module to guarantee detection of known intrusions that deviate from the network baseline.

F. *Preparing Attack Scenarios – Going Into Production*

To evaluate the effectiveness of detection and simulate network attacks, we used tcpdump to replay pcap records from the CICIDS2017 dataset. Replaying a PCAP portion of CICIDS2017 that was never seen by the system allowed us to measure the performance and accuracy of the proposed approach. The illustration in Fig. 18 gives an overview of the lab architecture and attack scenarios.



```
tcpdump -w CICIDS2017.pcap -i ens192
```

Fig. 18. An overview of the architecture used to reproduce the attack scenarios.

G. *Testing the Baseline Model on New, Never-before-seen Data*

This section outlines the methodology adopted to evaluate the real-time functioning of the proposed ZT-NIDS system with particular emphasis on its components. Fig. 19 shows the sequential steps involved in testing the system with new network events. Notably, the step "Transform search results using data preparation" reproduces the procedures detailed in the training phase diagram (indicated by a red strikethrough outline in Fig. 16).

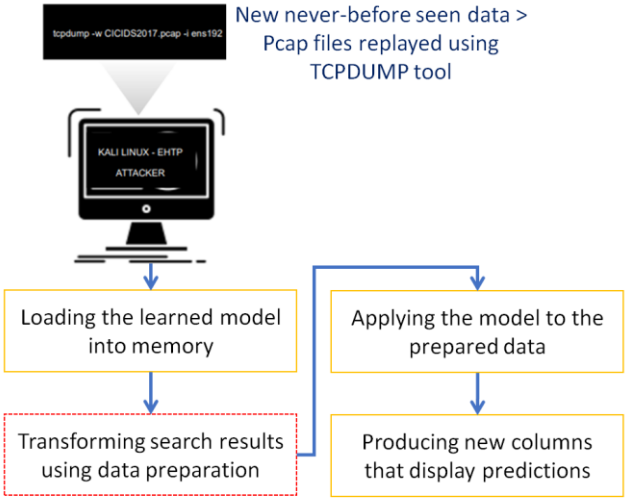
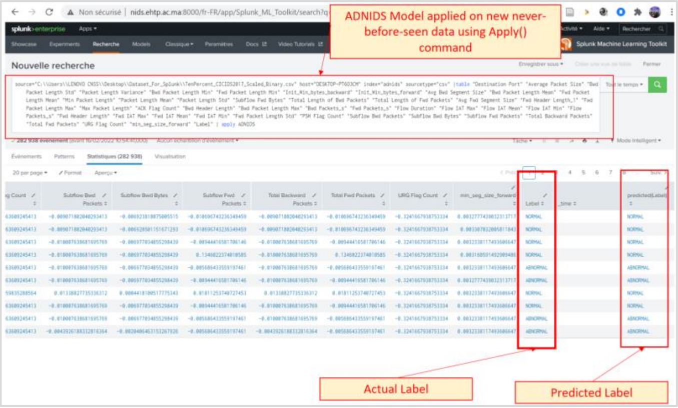

tcpdump -w CICIDS2017.pcap -i ens192
New never-before seen data > Pcap files replayed using TCPDUMP tool
KALI LINUX - EHTP ATTACKER
Loading the learned model into memory
Transforming search results using data preparation
Applying the model to the prepared data
Producing new columns that display predictions

Fig. 19. Worfkow of testing process of ZT-NIDS on never-before-seen network events.

In order to evaluate the modeled baseline in a real network, we used an intact part of the CICIDS2017 dataset in PCAP format, which had not been used during the ZT-NIDS model training phase. The TCPDUMP tool, running under Kali Linux, was used to replay the CICIDS2017 pcap files. These network events were then transmitted to the SIEM system, where they were compared with the baseline and Suricata.

The SIEM then extracts the characteristics of the network events received and compares them to the baseline models, which had previously been stored in memory, and to Suricata's SNIDS. A new column is then generated to store the predicted values generated after application of the base model, as shown in Fig. 20. Further comparisons are made between the values predicted by the model and the actual values, enabling the calculation of performance metrics that serve as evidence of the model's effectiveness.



```
source="C:\\Users\\LENOVO CNSS\\Desktop\\Dataset_For_Splunk\\TenPercent_CICIDS2017_Scaled_Binary.csv" host="DESKTOP-PT603CM" index="adnids" sourcetype="csv" |table "Destination Port" "Average Packet Size" "Bwd Packet Length Std" "Packet Length Variance" "Bwd Packet Length Min" "Fwd Packet Length Min" "Init_Win_bytes_backward" "Init_Win_bytes_forward" "Avg Bwd Segment Size" "Bwd Packet Length Mean" "Fwd Packet Length Mean" "Min Packet Length" "Packet Length Mean" "Packet Length Std" "Subflow Fwd Bytes" "Total Length of Bwd Packets" "Total Length of Fwd Packets" "Avg Fwd Segment Size" "Fwd Header Length_1" "Fwd Packet Length Max" "Max Packet Length" "ACK Flag Count" "Bwd Header Length" "Bwd Packet Length Max" "Bwd Packets_s" "Fwd Packets_s" "Flow Duration" "Flow IAT Max" "Flow IAT Mean" "Flow IAT Min" "Flow Packets_s" "Fwd Header Length" "Fwd IAT Max" "Fwd IAT Mean" "Fwd IAT Min" "Fwd Packet Length Std" "PSH Flag Count" "Subflow Bwd Packets" "Subflow Bwd Bytes" "Subflow Fwd Packets" "Total Backward Packets" "Total Fwd Packets" "URG Flag Count" "min_seg_size_forward" "Label" | apply ADNIDS
```

g Count	Subflow Bwd Packets	Subflow Bwd Bytes	Subflow Fwd Packets	Total Backward Packets	Total Fwd Packets	URG Flag Count	min_seg_size_forward	Label	_time	predictedLabel
63609245413	-0.009071882040293413	-0.006923818875005515	-0.010696743236349459	-0.009071882040293413	-0.010696743236349459	-0.3241667938753334	0.003277743903231371	NORMAL		NORMAL
63609245413	-0.009071882040293413	-0.006928501151671293	-0.010696743236349459	-0.009071882040293413	-0.010696743236349459	-0.3241667938753334	0.003307832005811843	NORMAL		NORMAL
63609245413	-0.010007638681695769	-0.006977034855298439	-0.009444165817706146	-0.010007638681695769	-0.009444165817706146	-0.3241667938753334	0.003233811749360664	NORMAL		NORMAL
63609245413	-0.010007638681695769	-0.006977034855298439	0.1346022374010585	-0.010007638681695769	0.1346022374010585	-0.3241667938753334	0.003160591492909498	NORMAL		NORMAL
63609245413	-0.010007638681695769	-0.006977034855298439	-0.005686433559197461	-0.010007638681695769	-0.005686433559197461	-0.3241667938753334	0.003233811749360664	ABNORMAL		ABNORMAL
63609245413	-0.010007638681695769	-0.006977034855298439	-0.009444165817706146	-0.010007638681695769	-0.009444165817706146	-0.3241667938753334	0.003277743903231371	ABNORMAL		ABNORMAL
59835288564	0.01338827735336312	0.0004418100517775343	0.01811253749727453	0.01338827735336312	0.01811253749727453	-0.3241667938753334	0.003233811749360664	NORMAL		NORMAL
63609245413	-0.010007638681695769	-0.006977034855298439	-0.009444165817706146	-0.010007638681695769	-0.009444165817706146	-0.3241667938753334	0.003233811749360664	NORMAL		NORMAL
63609245413	-0.010007638681695769	-0.006977034855298439	-0.005686433559197461	-0.010007638681695769	-0.005686433559197461	-0.3241667938753334	0.003233811749360664	ABNORMAL		ABNORMAL
63609245413	-0.004392618833281636	-0.002840645315326792	-0.005686433559197461	-0.004392618833281636	-0.005686433559197461	-0.3241667938753334	0.003233811749360664	ABNORMAL		ABNORMAL

Fig. 20. The baseline model applied on never-before-seen network events.

V. Results and Discussion

A. Results

The following section summarizes all the results obtained after testing different machine learning algorithms on the pre-processed, balanced and binarized CICIDS2017 dataset. In particular, we focused on the recognition of benign traffic, i.e. the creation of the baseline. Consequently, Table VIII presents all the results obtained.

TABLE VIII. Summary of Results

Algorithms	Recall	Precision	Accuracy	F1
Random Forest	0,9999	0.9999	0.9999	0.9999
Decision Tree	0,9999	0.9999	0.9999	0.9999
Naïve Bayes	0,9983	0,9987	0,9985	0,9984
Multi-Layer Perceptron	0,9992	0,9994	0.9993	0,9992

B. Discussion of the Obtained Results

As we have previously noted, there has been limited research addressing the specific challenge of intrusion detection within drone fleet networks. Most existing studies have focused on aspects such as routing protocols, battery autonomy, and other functionalities, often overlooking security concerns. However, given the increasing importance of drone technologies and their modern applications, the security of these networks has become critically significant.

The analysis of drone networks has led us to the conclusion that drones can be regarded as flying computers, and their networks share similarities with modern computer networks. Building upon this insight, we have introduced a model named ZT-NDIS, designed with a multi-agent architecture to implement a detection mechanism in line with the zero-trust principle.

The multi-agent architecture has proven well-suited to the challenge of intrusion detection in drone networks due to its characteristics of distribution, autonomy, and cooperation among the various agents. The detection mechanism comprises two complementary modules: a signature-based detection module and an anomaly detection module.

To materialize the proposed model, we conducted a comprehensive simulation of all components of the ZT-NIDS architecture using a real-world laboratory setup. Initially, we preprocessed, balanced, and reduced the dimensionality of the CICIDS2017 dataset. We then assessed a range of machine learning algorithms to model the baseline of benign network traffic. Cross-validation techniques were employed to ensure classification performance. While all tested algorithms yielded promising results, we ultimately selected the "Decision Tree" model.

Subsequently, we constructed the baseline model using the MLTK framework based on Scikit Learn, implementing the Decision Tree algorithm. To complement the proposed detection mechanism, Suricata was integrated for recognizing attacks with known signatures.

Finally, a real-world testing was conducted by networking the system and replaying CICIDS2017 PCAP files. The results were highly satisfactory, and the system was able to effectively identify both normal and abnormal network.

This proposed system represents a significant contribution and marks the initial step toward enhancing the security of civilian drone networks, which continue to gain popularity in various applications.

VI. Conclusion and Perspectives

We introduced in this work a network intrusion detection system called ZT-NDIS, designed with a multi-agent architecture and a zero-trust detection mechanism adapted to drone networks. The multi-agent architecture is suitable to drone networks given their distributed, autonomous and cooperative characteristics. The zero-trust principle enabled us to detect various types of cyberattack, including zero-day threats, through continuous comparison of network traffic against the modelled baseline of benign traffic.

Thanks to the pre-processing of the CICIDS2017 dataset, we were able to create an effective network baseline model with almost 100% accuracy. The inclusion of Suricata has further enhanced the system's performance, particularly against known attacks.

To evaluate the system's effectiveness, we carried out real laboratory tests. This enabled us to assess its ability to detect emerging cyber threats that could target drone networks.

The proposed system represents an important step towards improving the security of drone networks and gaining a better understanding of their security posture. Future work will focus on developing the various agents, generating real network traffic from drone networks and testing the system in a production environment.

References

[1] Hashimy, S. Q., & Benjamin, M. S. (2023). The Deployment of US Drones in Afghanistan: Deadly Sky and Unmanned Injustice.

[2] Shaikh, E., Mohammad, N., & Muhammad, S. (2021, March). Model checking based unmanned aerial vehicle (UAV) security analysis. In 2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA) (pp. 1-6). IEEE.

[3] Mekdad, Y., Aris, A., Babun, L., El Fergougui, A., Conti, M., Lazzeretti, R., & Uluagac, A. S. (2023). A survey on security and privacy issues of UAVs. Computer Networks, 224, 109626.

[4] Suresh Kumar, K., Prabakaran, D., Senthil Kumaran, R., & Yamuna, I. (2022). Privacy and security of smart systems. Intelligent Green Technologies for Sustainable Smart Cities, 291-315.

[5] Wojciechowski, P., & Wojtowicz, K. (2023, June). Detection of Critical Infrastructure Elements Damage with Drones. In 2023 IEEE 10th International Workshop on Metrology for AeroSpace (MetroAeroSpace) (pp. 341-345). IEEE.

[6] Keskin, B. B., Griffin, E. C., Prell, J. O., Dilkina, B., Ferber, A., MacDonald, J., ... & Gore, M. L. (2022). Quantitative investigation of wildlife trafficking supply chains: A review. Omega, 102780.

[7] Al-Wathinani, A. M., Alhallaf, M. A., Borowska-Stefańska, M., Wiśniewski, S., Sultan, M. A. S., Samman, O. Y., ... & Goniewicz, K. (2023, May). Elevating Healthcare: Rapid Literature Review on Drone Applications for Streamlining Disaster Management and Prehospital Care in Saudi Arabia. In Healthcare (Vol. 11, No. 11, p. 1575). MDPI.

[8] Botta, A., Rotbei, S., Zinno, S., & Ventre, G. (2023). Cyber Security of Robots: a Comprehensive Survey. Intelligent Systems with Applications, 200237.

[9] Ouiazzane, S., Addou, M., & Barramou, F. (2022). A multiagent and machine learning based denial of service intrusion detection system for drone networks. Geospatial Intelligence: Applications and Future Trends, 51-65.

[10] Mohsan, S. A. H., Khan, M. A., Noor, F., Ullah, I., & Alsharif, M. H. (2022). Towards the unmanned aerial vehicles (UAVs): A comprehensive review. Drones, 6(6), 147.

[11] Silva, M., Reis, A., & Sargento, S. (2023). A Mission Planning Framework for Fleets of Connected UAVs. Journal of Intelligent & Robotic Systems, 108(1), 2

[12] Saffre, F., Hildmann, H., Karvonen, H., & Lind, T. (2022). Self-swarming for multi-robot systems deployed for situational awareness. In New Developments and Environmental Applications of Drones: Proceedings of FinDrones 2020 (pp. 51-72). Springer International Publishing.

[13] Jean-Aimé Maxa. Architecture de communication sécurisée d'une flotte de drones. Réseaux et télécommunications [cs.NI]. Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), 2017. Français.

[14] Said OUIAZZANE, Fatimazahra BARRAMOU and Malika ADDOU, "Towards a Multi-Agent based Network Intrusion Detection System for a Fleet of Drones" International Journal of Advanced Computer Science and Applications(IJACSA), 11(10), 2020

[15] Eric W Frew and Timothy X Brown. Networking issues for small unmanned aircraft systems. Journal of Intelligent and Robotic Systems, 54(1-3) :21–37, 2009.

[16] Van, Y., Chen, X., Li, R., & Jiang, Y. (2022, August). The Intelligent Pelagic Communication System Architecture of the Fleet based on UAV Swarm Relay. In 2022 9th International Conference on Dependable Systems and Their Applications (DSA) (pp. 960-964). IEEE.

[17] Lakhwani, K., Singh, T., & Aruna, O. (2022). Multi-Layer UAV Ad Hoc Network Architecture, Protocol and Simulation. Artificial Intelligent Techniques for Wireless Communication and Networking, 193-209.

[18] Souli, N., Kolios, P., & Ellinas, G. (2023). Multi-Agent System for Rogue Drone Interception. IEEE Robotics and Automation Letters, 8(4), 2221-2228.

[19] Mahajan, N., Kaushal, S., & Kumar, H. (2023, March). IMS enabled Centralized UAV control system with seamless connectivity over mobile network. In 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA) (pp. 636-641). IEEE.

[20] Mademlis, I., Nousi, P., Lavaux, D., Aubourg, T., Le Barz, C., & Pitas, I. (2023). Secure Communications for Autonomous Multiple-UAV Media Production. In Unmanned Aerial Vehicles Applications: Challenges and Trends (pp. 323-347). Cham: Springer International Publishing.

[21] Omolara, A. E., Alawida, M., & Abiodun, O. I. (2023). Drone cybersecurity issues, solutions, trend insights and future perspectives: a survey. Neural Computing and Applications, 1-39.

[22] Lin, N., Yiyang, Y., Yingjie, Z., & Zihui, L. (2023, May). Network Threat Analysis and Protection Technology of UAV System. In 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI) (pp. 236-240). IEEE.

[23] Krichen, M. (2022). Défis de sécurité pour les communications par drones: menaces, attaques et contre-mesures possibles.

[24] Bajracharya, R., Shrestha, R., Kim, S., & Jung, H. (2022). 6G NR-U based wireless infrastructure UAV: Standardization, opportunities, challenges and future scopes. IEEE Access, 10, 30536-30555.

[25] Gosain, M. S., Aggarwal, N., & Kumar, R. (2023, April). A Study of 5G and Edge Computing Integration with IoT-A Review. In 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES) (pp. 705-710). IEEE.

[26] Karmakar, G., Petty, M., Ahmed, H., Das, R., & Kamruzzaman, J. (2022, December). Security of Internet of Things Devices: Ethical Hacking a Drone and its Mitigation Strategies. In 2022 IEEE Asia-

Pacific Conference on Computer Science and Data Engineering (CSDE) (pp. 1-5). IEEE.

[27] R., Abdan, M., & Ghasemi, S. (2022). A self-adaptive intrusion detection system for securing UAV-to-UAV communications based on the human immune system in UAV networks. Journal of Grid Computing, 20(3), 22.

[28] Shrestha, R., Omidkar, A., Roudi, S. A., Abbas, R., & Kim, S. (2021). Machine-learning-enabled intrusion detection system for cellular connected UAV networks. Electronics, 10(13), 1549.

[29] Ihekoronye, V. U., Ajakwe, S. O., Kim, D. S., & Lee, J. M. (2022, November). Hierarchical intrusion detection system for secured military drone network: A perspicacious approach. In MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM) (pp. 336-341). IEEE.

[30] Al-Haija, Q., & Al Badawi, A. (2022). High-performance intrusion detection system for networked UAVs via deep learning. Neural Computing and Applications, 34(13), 10885-10900.

[31] Kharchenko, V., & Torianyk, V. (2018, May). Cybersecurity of the internet of drones: Vulnerabilities analysis and imeca based assessment. In 2018 IEEE 9th international conference on dependable systems, services and technologies (DESSERT) (pp. 364-369). IEEE.

[32] Susmaga, R. (2004). Confusion matrix visualization. In Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM ‘04 Conference held in Zakopane, Poland, May 17–20, 2004 (pp. 107-116). Berlin, Heidelberg: Springer Berlin Heidelberg.

[33] Stiawan, D., Idris, M. Y. B., Bamhdi, A. M., & Budiarto, R. (2020). CICIDS-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access, 8, 132911-132921.

[34] Ouiazzane, S., Addou, M., & Barramou, F. (2022). A Suricata and Machine Learning Based Hybrid Network Intrusion Detection System. In Advances in Information, Communication and Cybersecurity: Proceedings of ICI2C’21 (pp. 474-485). Springer International Publishing.

[35] Ouiazzane, S., Addou, M., & Barramou, F. (2023). Cyberthreat Real-time Detection Based on an Intelligent Hybrid Network Intrusion Detection System. In Big Data Analytics and Intelligent Systems for Cyber Threat Intelligence (pp. 175-194). River Publishers.

[36] Mendoza, M. A., & Amistadi, H. R. (2018). Machine learning for anomaly detection on VM and host performance metrics.