Sybil Attack Detection in Internet of Flying Things-IoFT: A Machine Learning Approach

Donpiti Chulerttiyawong and Abbas Jamalipour, *Fellow, IEEE*

***Abstract*—Sybil attack refers to the situation when a malicious node falsely claims to have numerous identities and is known to be one of the security threats to the Internet of Things (IoT). Due to recent increase usage of unmanned aerial vehicles (UAVs) in various applications, Sybil attack has been identified as a threat to the flying ad hoc network (FANET) paradigm and its integration with the IoT to form the Internet of Flying Things (IoFT). In this article, we propose an intelligent Sybil attack detection approach for FANETs-based IoFT using physical layer characteristics of the radio signals emitted from the UAVs as detected by two ground nodes. A supervised machine learning approach is employed and experimented with several different classifiers available in the Weka workbench platform. The experiment was carried out based on two features of the radio signals, namely, the received signal strength difference (RSSD) and the Time Difference of Arrival (TDoA). Simulation results revealed that the proposed scheme can achieve a high correct classification accuracy of above 91% on average, even for smart malicious nodes with power control capability operating at power levels not directly trained. In addition to its high performance, the proposed scheme is also less susceptible to various attacks commonly carried out on the upper layers, such as data spoofing, due to the use of only intrinsically generated physical layer data. Furthermore, no additional communications overheads of the UAV nodes are required for the functionality of this scheme.**

***Index Terms*—Flying ad hoc network (FANET), Internet of Flying Things (IoFT), Internet of Things (IoT), machine learning (ML), received signal strength difference (RSSD), Sybil attack, Time Difference of Arrival (TDoA), unmanned aerial vehicle (UAV).**

I. INTRODUCTION

UNMANNED aerial vehicles (UAVs), also known as drones, refer to pilotless aerial vehicles that are either autonomously controlled by a computer or remotely controlled by a pilot on the ground. UAVs deployment in the military domain dates back several decades, with the primary applications being strike, reconnaissance, and border surveillance. However, more recently, UAVs have also gained increasing usage in civilian applications, including search and rescue operations, environmental sensing and monitoring, and delivery of food and other products. In this context, the flying ad hoc network (FANET) paradigm, which is a subclass of mobile ad hoc network (MANET) where the nodes possess aviation characteristics, is strongly tied to the operation of UAVs due to the needs for UAV nodes to communicate with each other or with other node types, such as ground control station and satellite. Consequently, the FANET paradigm and its integration with the Internet of Things (IoT) to form the Internet of Flying Things (IoFT) have been gaining increased attention in the research community [1], [2], [3].

The arrangement of UAVs to form a swarm has been increasingly highlighted as an operating model of great potential for various applications. For example, Gao et al. [4] and Zhang et al. [5] discussed the use of UAV swarms for search and rescue operations, while Liu et al. [6] discussed the use of UAV swarms for air quality index monitoring. Although the deployment of UAV swarms can bring about immense advantages from the aspects of resource allocation, control, and cooperation, such a deployment model can also concurrently introduce additional security risks associated with malicious use [7]. For instance, there could be a greater potential for attacks involving identity falsification, one of which is the Sybil attack.

Sybil attack is well known to be one of the security threats to the IoT. It refers to the situation when a malicious node falsely claims to have numerous identities [8], [9]. There are several incentives for a node to act in such a way; in the context of FANETs, examples are such as to allow it to illegitimately acquire more weight in a voting system and to create an illusion of traffic congestion in a particular area [10], [11]. Countermeasures for Sybil attack include prevention, detection, and mitigation. Prevention refers to the inhibition of the attack from occurring at all. Detection refers to the identification of security breach, the identification of attack type, as well as the initiation of relevant mitigation solutions. Finally, mitigation refers to the alleviation of resulting outcomes of the attack [12].

More recently, the use of machine learning (ML) has increasingly been leveraged to address various challenges, including IoT security. ML does this by intelligently choosing the actions to be taken in response to a given situation based on knowledge that the system has learned. Well-known examples of applications are, such as computer vision, bio-informatics, fraud/malware detection, authentication, and speech recognition [13].

As will be discussed further in Section II, there exist numerous studies in the literature that discuss Sybil attack detection methods for wireless ad hoc networks, wireless sensor networks, and vehicular ad hoc networks (VANETs). However, this is not the case for FANETs, which would have

Manuscript received 25 August 2022; revised 17 January 2023; accepted 28 February 2023. Date of publication 16 March 2023; date of current version 7 July 2023. *(Corresponding author: Donpiti Chulerttiyawong.)*

The authors are with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: mick.chulerttiyawong@sydney.edu.au; a.jamalipour@ieee.org).

Digital Object Identifier 10.1109/JIOT.2023.3257848

had relatively fewer Sybil attack threats due to the lower expectation of having high node density presented in an area; but the more recent increase in UAV usage is changing all that. Adapting one of the numerous existing non-FANET Sybil attack detection methods is also deemed to require significant effort, as those schemes were not designed to suit nodes with complex 3-D mobility. These facts motivated us to develop a novel approach for Sybil attack detection to fill this gap, which should be lightweight, highly secure, and able to detect smart malicious nodes with power control capability. ML has been identified as a tool with high potential to aid in the delivery of such identified features.

In this article, we propose a new intelligent Sybil attack detection approach for FANETs-based IoFT. The proposed approach employs range-based location verification using physical layer characteristics of the radio signals emitted from the UAVs as detected by two ground nodes. This is done by utilizing a supervised ML approach and experimenting with several different classifiers available in the Weka [14] workbench platform. The learning is carried out on two features of the radio signals, namely, the received signal strength difference (RSSD) and the Time Difference of Arrival (TDoA). To the best of our knowledge, no other similar schemes have been proposed so far.

The technical contributions of this article are summarized as follows.

1) To fill a knowledge gap in the literature relating to Sybil attack detection in FANETs-based IoFT which is still quite deficient in general.
2) To achieve Sybil attack detection in FANETs-based IoFT using intrinsically generated physical layer data of radio signals emitted from the UAVs. Advantages associated with this are such as less susceptibility to attacks involving information spoofing and not requiring additional communications overheads.
3) To achieve Sybil attack detection in FANETs-based IoFT, where both classic malicious nodes with fixed power and smart malicious nodes with power control capability may be presented.
4) To investigate and demonstrate the use of ML in carrying out Sybil attack classification determination based on two attributes, namely, the RSSD and TDoA ratios of two different radio signals, obtained using only two monitoring nodes.

The remainder of this article is organized as follows. In Section II, we review existing related works and deduce the contributions of this study. We then outline the details of the proposed scheme in Section III. In Section IV, we describe the simulation environment, including all the key simulation parameters. The results and evaluation of the simulation are then discussed in Section V. Finally, we conclude in Section VI.

II. Related Works

A. Position Localization Using Physical Layer Data

Many existing positioning systems are known to function using measurements of physical layer features of the radio signal. Very commonly used features include received signal strength (RSS), Angle of Arrival (AoA), Time of Arrival (ToA), and TDoA. Classical usage of these measurements involves a two-step process. These steps are briefly described below; however, interested readers can also refer to more comprehensive publications, such as Dardari et al. [15] and Munoz et al. [16] for more details, including mathematical descriptions.

In the first step, the position-related signal parameters of interest are measured. Out of the four features, RSS is known to be the most easily obtainable because it is simply a measurement of the received power, which can easily be done in any system without the need for time synchronization. On the other hand, ToA and TDoA require some sort of time synchronization. In essence, ToA is a measurement of signal propagation delay; therefore, time synchronization between the receivers and the transmitter would be required. Similarly, TDoA is a measurement of signals propagation delay difference between the receivers; therefore, time synchronization between the receivers of interest would be required. On the other hand, AoA is known to perhaps be the least favorable feature, as it requires characterization of the direction of signal propagation; consequently, the use of AoA may dictate the need for costly specialized hardware, such as the use of antenna arrays. Additionally, AoA position estimation performance also degrades as the distance between transmitter and receiver increases [15], [16].

The second step is the application of position estimation techniques based on the parameters obtained in the first step. This can be achieved by using techniques, such as lateration and angulation. The use of multiple types of position-related parameters can also be combined to form hybrid methods [15]. One constraint of this step is that more than two receiver nodes are generally required for accurate positioning. For example, as outlined by Li et al. [17], according to the principles of trilateration, if ToA or TDoA are used, three receiver nodes would be required for 2-D position estimation. More relevant to FANETs is the fact that for 3-D position estimation, four receivers would be required.

B. Sybil Attack Detection in IoFT

There are quite a number of published articles that outline different Sybil attack detection methods that are applicable to slightly different IoT domains; several recent survey papers summarize these into their associated categories. Recent surveys on Sybil attack detection in wireless ad hoc networks and wireless sensor networks can be found in Arshad et al. [12], Vasudeva and Sood [18], and Singh [19]. There are also several recent survey papers on Sybil attack detection in VANETs, including Shobana and Arockia [20], Zhang et al. [21], Velayudhan and Anitha [22], and Hammi et al. [23].

Existing Sybil attack detection approaches found in the literature include the use of location verification, network behavior monitoring, resource testing, trust systems, and cryptography. As mentioned in Section I, the scheme proposed in this article focuses on the range-based location verification approach. To elaborate further, the location verification approach is classified into range-free and range-based methods. In the range-free methods, high-accuracy location is calculated based on data supplied through external means, such as global positioning system (GPS), radar, or other localization schemes. The range-based methods, however, generally can work simply by using data obtainable from the physical layer characteristics of the radio signals being sent and received [19]. There are several reasons why methods that use intrinsically generated physical layer data to detect Sybil attack might be more preferable than others. For instance, the use of intrinsically generated physical layer data also brings about a security advantage over methods that use extrinsic data, in that such use would be less susceptible to spoofing attacks. Furthermore, unlike many other methods in other detection approaches, authentication would not be required; consequently, misidentification due to potentially stolen credentials would be less of a risk. Cryptography, which is a widely used technique for authentication, also consumes a lot of energy [11]. Accordingly, since UAVs operate on limited energy, for some applications, it may be desirable to cut down on their cryptographic usage. Nevertheless, there may be other advantages associated with the other Sybil attack detection methods; therefore, in some situations, it may be desirable to combine the advantages associated with different schemes by using two or more detection mechanisms on a complementary basis.

Schemes that use physical layer characteristics of the radio signals do exist in the literature. These schemes use features, such as RSS, AoA, ToA, and TDoA for their location verification determination. However, to the best of our knowledge, none of these schemes are designed for mobile nodes that possess aviation characteristics like UAVs in FANETs. In fact, apart from those designed for VANETs, most schemes only cater for static nodes. Additionally, most schemes also do not cater for the situation in which malicious nodes can adjust their transmit power to fool detectors while carrying out Sybil attacks. Furthermore, the ways in which some of these schemes operate impose various other undesirable constraints. For example, schemes, such as Kabbur and Kumar [24] and Yuan et al. [25] used RSS indication values obtained through triangulation, requiring at least three monitoring nodes to be used [12], [23]. Other examples include schemes like Lv et al. [26], Abbas et al. [27], and Angappan et al. [28], which require the use of additional localization information such as those obtainable through neighbors of the suspicious nodes [12]; consequently, unlike schemes that purely and directly use intrinsically generated physical layer data, these schemes may be more susceptible to attacks involving information spoofing.

When looking more specifically at Sybil attack detection for FANETs, to the best of our knowledge, there are currently no survey papers that discuss this topic. Nevertheless, we did find a limited number of existing research works in this area, including de Melo et al. [29], Sun et al. [10], and Walia et al. [30], details of which are summarized in the following paragraphs. Note that none of these schemes operate on pure use of physical layer characteristics of the radio signals.

In de Melo et al. [29], an identity and location validation scheme called UAVouch is proposed to detect malicious UAVs that do not follow expected trajectories, including the potential scenario where a Sybil attack is being carried out. The idea is for this scheme to supplement the authentication mechanism by requesting position validation from neighboring nodes inside a cell and by using a position plausibility/classifier model to detect movement inconsistencies. The scheme is reported to have an average position falsification attack detection accuracy of above 85%.

In Sun et al. [10], a Bayesian Nash equilibrium game theory-based intrusion detection scheme is proposed, which can detect Sybil attacks among other attack types. The game is between the intrusion detection nodes and the attacking nodes, with each side strategizing to maximize their profits. The scheme works by studying the past behavior of UAV nodes and determining the deployment of intrusion detection nodes to achieve optimization by minimizing the overhead while achieving a high detection rate. Specific details on the Sybil attack detection mechanism and the associated detection accuracy rate are not provided due to not being the focus of this article.

In Walia et al. [30], a mutual authentication technique to detect Sybil attack in FANETs is proposed. The scheme works by having each node checking its neighboring nodes for identification. If nodes with the same identification but with different neighbors are found, they are marked as intruder nodes. Each intruder node is then monitored more closely and if found to change its identity then it would get identified as malicious. In terms of performance, this article reports high throughput, low overhead, and low packet loss; however, it does not mention the overall Sybil attack detection accuracy rate.

C. Machine Learning for Sybil Attack Detection in IoFT

A typical ML system has three layers: 1) input; 2) feature extraction and processing; and 3) output. The input layer takes in preprocessed data, which is then passed onto the feature extraction and processing layer where the data patterns get extracted; basically, this is where the training of an ML system takes place. Several classifiers exist in this layer, each of which defines a different methodology for data pattern extraction; well-known ones are, such as support vector machines (SVMs), principal component analysis (PCA), and hidden Markov model (HMM). Finally, the output layer produces the prediction results of the task, such as classification for discrete outputs (class labels) and regression for continuous numeric outputs [13], [31].

ML methods can commonly be grouped into *supervised*, *unsupervised*, *semi-supervised*, or *reinforcement learning* approaches. Interested readers can refer to survey papers, such as Jamalipour and Murali [32], Hussain et al. [13], Al-Garadi et al. [33], and Wang et al. [34] for more information on these ML approaches and on the use of ML in IoT security in general. Of most relevant to this article is the supervised learning approach, where a class label is assigned to identify each data entry in the training set. Learning then takes place based on this known identification and the other input features parameters. Subsequently, the learned system can be deployed on other data sets to make predictions regarding the correct class label associated with each entry.

D. Motivations and Contributions

As can be seen from previous discussions, there is currently a gap for a Sybil attack detection mechanism that can achieve highly accurate detection of mobile Sybil nodes in FANETs-based IoFT. This is especially true if the scheme can detect Sybil nodes with power control capability. In addition, the use of physical layer features was identified as potentially being very useful for Sybil attack detection applications in FANETs. The pure use of intrinsically generated physical layer data to carry out detection also minimizes potential problems such as the risk of data spoofing. A potential approach might be to try and adapt existing methods developed for wireless ad hoc networks, wireless sensor networks, or VANETs to cater for FANETs; however, significant extensions would be required and there is no guarantee that such solutions will work well. As an alternative, it is worth investigating a new innovative scheme, as proposed in this article.

From the perspective of selecting the most appropriate physical layer features to use, the use of RSS and/or TDoA features makes the most sense. The use of ToA is undesirable because it requires synchronization with the transmitter, which would be impractical to implement. Similarly, the use of AoA feature would also be impractical unless antenna arrays are already required for other reasons. From the amount of monitoring nodes perspective, it would also be desirable to minimize these while still maintaining a highly accurate detection functionality.

It can be seen that ML can potentially be used to aid the construction of the scheme. More specifically, it is known that RSS and TDoA features capture some location information. An ML system can be developed to learn certain characteristics associated with RSS and TDoA values confirmed as belonging to Sybil attack events, in preparation for it to identify similar malicious instances in the future. More importantly, the learning can be performed without the system requiring to know the exact underlying mechanisms, such as mathematical operations. Because ML can easily learn from both features concurrently, a hypothesis can be formed that a minimal number of two monitoring nodes may already be sufficient for accurate Sybil attack detection functionality. It is important to note that the exact formats of attributes to be fed into the ML system need to be refined to suit the intended application, which is Sybil attack detection in this case. This process is a bit of an artwork, and for our study, it resulted in two attributes, namely, the RSSD and TDoA ratios of two different radio signals, more details of which can be found in Section III.

As will be demonstrated in later sections, the scheme proposed in this article, incorporating an artificial intelligence mechanism, has been designed with the intention of filling the gap for Sybil attack detection in the FANETs environment. The proposed scheme addresses all of the above-mentioned design criteria and does not require any additional communications overheads. With the use of only two monitoring nodes at fixed locations while still able to achieve a high detection accuracy of above 91% on average, it supports our hypothesis that such a minimal number of nodes may already be sufficient when assisted by an ML mechanism. To provide further illustration, Table I summarizes the contribution of our proposed scheme compared with the existing Sybil attack detection approaches described by Singh [19].

III. Proposed Scheme

In this section, we discuss the architecture of the proposed scheme. As depicted in Fig. 1, we look at a situation where a number of UAVs fly within a given area to carry out certain operations. While doing so, the UAVs communicate with each other and/or with ground stations. Some members of the nodes have malicious purposes and would attempt to carry out Sybil attacks by falsely identifying themselves as other entities. Two monitoring nodes are placed on the ground at fixed locations within the operational area in an attempt to detect Sybil UAV nodes.

The focus of the architecture is on the use of ML system to detect Sybil attack instances. The supervised ML approach was identified as the most suitable approach in this study due to the nature of the problem being addressed. This is because there are simply two known distinct outcome classes, which are whether or not a Sybil attack event is taking place. Furthermore, the use of supervised learning is also favorable from the performance assessment perspective, as training and test data sets with correctly labeled class events can be generated in a straightforward manner through the simulation of UAV networks.

As outlined in Section II, a typical ML system has three layers: 1) input; 2) feature extraction and processing; and 3) output. In this architecture, the focus is mostly on the input layer, more specifically, the derivation of data attributes to be fed into the ML system. Feature extraction and processing activities, which result in the determination of classification output, are mostly performed by the ML system based on specific algorithms. There exist numerous well-researched supervised ML algorithms which can potentially be used with the proposed architecture, as long as they support two numerical attributes (i.e., RSSD and TDoA ratios) and a class attribute (i.e., Sybil attack instance or not). Some of these algorithms have been selected for the simulations carried out in this study, the details of which can be found in Section IV.

Before proceeding further, it is important to note that the proposed scheme has been designed with the intention of being flexible for use with a range of UAV mobility patterns, density levels, transmit power levels, and signal emission rates; however, the exact limitations are outside the scope of this study. Another point to note is that this study was conducted based on the assumption that the free space path-loss propagation model holds true. Furthermore, it is also assumed that signals from other UAVs and other systems in the surrounding area are coordinated in such a way that results in negligible interference effects on the functionality of the system, such as through the use of orthogonal frequency-division multiplexing.

Regarding how the ML attributes were designed, as discussed in Section II, our literature review suggests that RSS and TDoA physical layer features contain location information most suitable for the application scenario in this study.

TABLE I
COMPARISON OF THE PROPOSED SCHEME TO EXISTING SYBIL ATTACK DETECTION APPROACHES

Sybil Attack Detection Approach	**Approach Description**	**Typical Advantages**	**Typical Disadvantages**
Range-based location verification	Use data obtainable from the physical layer characteristics of radio signals being sent and received.	• Low in cost since device already has physical layer characteristics of communicating radio signals by default.	• Accuracy may be reduced by rapid changes in node position. • Difficulties in detecting nodes that can manipulate signal strength. • Accuracy may be reduced by interference, multipath fading, shadowing, etc.
Range-free location verification	Location calculated from external data (e.g., GPS, radar, etc).	• Can provide high accuracy distance calculation.	• External data means more susceptibility to data spoofing attacks (compared with range-based location verification).
Network behavior monitoring	Based on nodes features and behavior in the network.	• Allow features and behavior in the network to be used for accurate detection of malicious nodes.	• Malicious nodes with specific knowledge can escape detection. • Specialized tools required for data collection and analysis.
Resource testing	Node challenged to provide knowledge about specific resources (usually physical fingerprinting or energy).	• Allow uniqueness in resources of each node to be used for verification.	• Extensive power consumption. • Genuine nodes with resource problems due to other reasons may be falsely classified as malicious.
Trust systems	Trust value obtainable from trusted devices or trusted neighbors must be maintained by each node to remain in the network.	• Allow periodic evaluation which can be done in centralized or decentralized manner.	• Inability to detect malicious node already dominating trust determination process.
Cryptography	Authenticate nodes and communicate securely using public/private keys. Use watermarking to guarantee valid data.	• Can also offer protection against various other attack types.	• High memory, computing, and communications overhead for resource constraint devices. • High costs associated with key management.
Proposed scheme	Range-based location verification using physical layer characteristics of the radio signals, namely, RSSD and TDoA, paired with supervised machine learning.	• Low in cost since device already has physical layer characteristics and only require two monitoring nodes. • Less susceptible to attacks on upper layers, such as data spoofing, stolen credential, etc. • Designed to work with mobile nodes in FANETs-based IoFT. • High detection performance even with malicious nodes that can manipulate signal strength. • Potential to extend to detect other attack types and/or utilize unsupervised machine learning approach.	• Prior to deployment, some further performance studies may still be required, for example on: 1. effects of interference and structural blockages; and 2. networks with high node density.

However, this is not the end of the process as the exact formats of ML input attributes that would allow for a high classification success rate still need to be derived. Based on the assumptions given in the previous paragraph, the received power level is assumed to follow the free space path-loss model developed by Friis [35] as

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2 \tag{1}$$

where P_t is the transmitted signal power, G_t is the transmit antenna gain, G_r is the receive antenna gain, λ is the wavelength of the signal, and d is the distance between transmitter and receiver. As for the signal propagation time taken between the transmitter and the receiver, such delay can be represented by the equation

$$\tau = \frac{d}{c} \tag{2}$$

where the constant $c \approx 3 \times 10^8$ m/s can be used for the speed of light [15].

The proposition given in this Sybil attack detection problem is that there are to be only two monitoring nodes, and the system is to detect if two signals identified as transmitted from different UAVs in near real-time are actually likely coming from the same location (i.e., the same UAV). Therefore, it is necessary to ensure that the attributes are designed to capture maximal information to enable the ML classifier to recognize such underlying pattern differences. As the use of ML is an experimental science, the process of determining the precise formats of the attributes used in this study requires some creativity and preliminary experiments to verify their effectiveness. Because of the proposition to use two monitoring nodes, the use of TDoA measured at these different monitoring nodes already makes sense. The next step is to represent this characteristic as a numeric value that captures

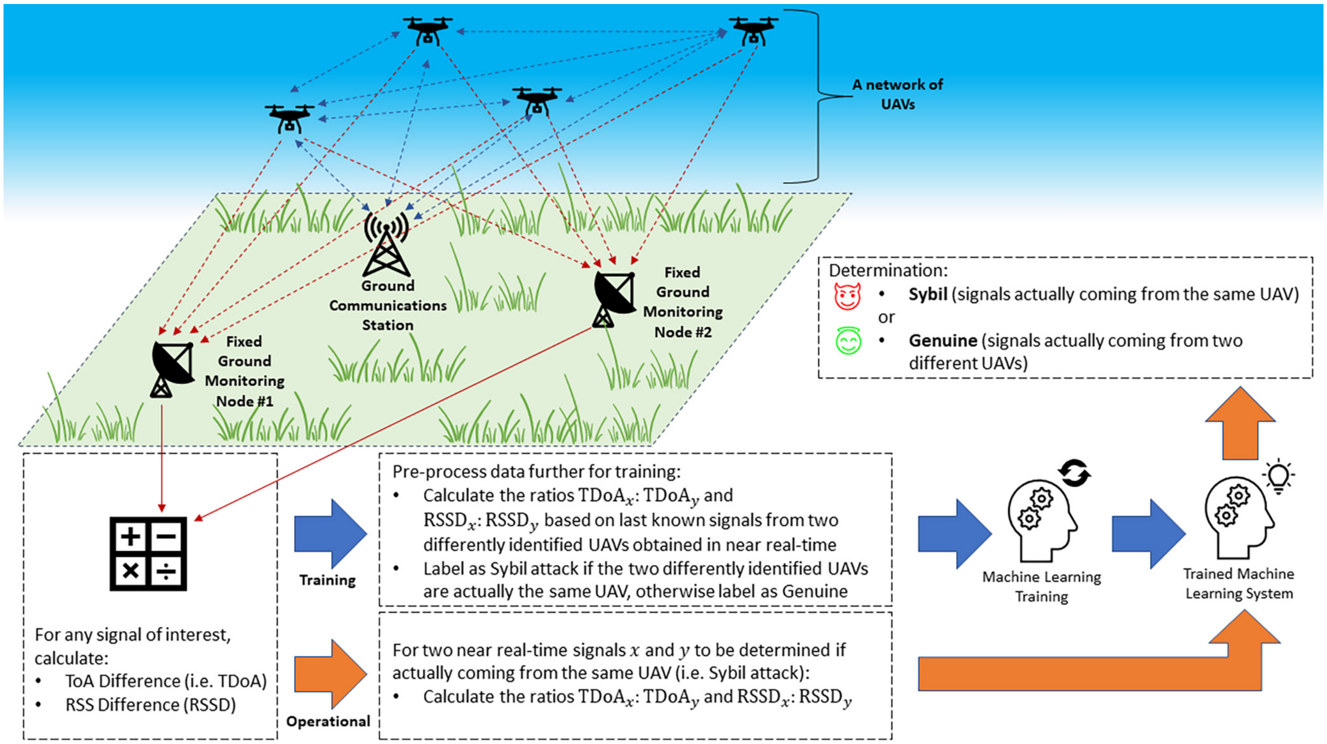
A network of UAVs
Ground Communications Station
Fixed Ground Monitoring Node #1
Fixed Ground Monitoring Node #2
Determination:
Sybil (signals actually coming from the same UAV)
or
Genuine (signals actually coming from two different UAVs)
For any signal of interest, calculate:
ToA Difference (i.e. TDoA)
RSS Difference (RSSD)
Training
Operational
Pre-process data further for training:
Calculate the ratios $TDoA_x : TDoA_y$ and $RSSD_x : RSSD_y$ based on last known signals from two differently identified UAVs obtained in near real-time
Label as Sybil attack if the two differently identified UAVs are actually the same UAV, otherwise label as Genuine
For two near real-time signals x and y to be determined if actually coming from the same UAV (i.e. Sybil attack):
Calculate the ratios $TDoA_x : TDoA_y$ and $RSSD_x : RSSD_y$
Machine Learning Training
Trained Machine Learning System

Fig. 1. Architecture of the proposed scheme.

```
Algorithm 1: Calculate TDoA and RSSD for a Given Signal x
Input: ToA_{x#1}: ToA of radio signal x obtained by fixed ground monitoring station #1,
       RSS_{x#1}: RSS of radio signal x obtained by fixed ground monitoring station #1,
       ToA_{x#2}: ToA of radio signal x obtained by fixed ground monitoring station #2,
       RSS_{x#2}: RSS of radio signal x obtained by fixed ground monitoring station #2
begin
    Calculate $TDoA_x = ToA_{x\#1} - ToA_{x\#2}$
    Calculate $RSSD_x = RSS_{x\#1} - RSS_{x\#2}$
end
```

the relationship between the two signals. The numeric value needs to somehow encompass such a relationship pattern to make it distinguishable if the value is likely coming from two signals belonging to the same UAV. Consequently, the TDoA ratio between the two signals was selected as one of the attributes. Similarly, the process also resulted in the RSSD ratio of the two signals as the other attribute, where the RSSD of any signal of interest is the difference between the RSS values of the signal measured at the two different monitoring nodes. The precise details of these two attributes are captured in Algorithms 1 and 3. Since the differences in RSS and ToA values received at the two monitoring nodes play important roles in the characterization of the two attributes, it is worth noting that the two locations should be sufficiently far apart to enhance effectiveness.

There are two different phases in this scheme: 1) the *training* and 2) *operational* phases. Detailed descriptions are elaborated in the following sections.

A. Training Phase

In the training phase, UAVs would be deployed and carry out radio communications in a controlled manner. A given number of these UAVs would be programmed to act maliciously and execute Sybil attacks by falsely using multiple identities. Signals from each UAV would be sampled by the monitoring nodes every certain interval for a given number of times until the end of the training period. The two monitoring nodes would detect and collect the RSS and ToA of each signal being sampled. Subsequently and in accordance with Algorithm 1, the corresponding RSSD and TDoA of each signal sampled would be calculated based on the variations in RSS and ToA received at the two different nodes. This is followed by the execution of Algorithm 2, which also calls Algorithm 3, to calculate the RSSD and TDoA ratios between the signal being sampled and all other latest signals sampled from every other UAV with a differently declared identity. The data generated by Algorithm 2 for all collected signals would then be collectively fed to the ML classifier as training data.

As previously discussed, the Sybil attack characterization being performed is carried out through the discovering of patterns within the RSSD and TDoA ratios from two different signals that would have been emitted from somewhat nearby physical positions during the UAV's movement in the air. Consequently, it is also important to note that any signals that were emitted in the past beyond a certain near real-time threshold need to be excluded, as the positions of the

Algorithm 2: Execute Algorithm 3 on a Given Signal x and All the Latest Signals Collected From Each of the Other Differently Identified UAVs Within a Given Near Real-Time Limit t_{thres}

```
Input: Current time $t$,
Near real-time threshold $t_{\text{thres}}$,
UAV$_{\text{ID}}$: the identity of UAV claimed to have emitted signal $x$
begin
    for each collected signal claimed to have emitted from a UAV other than UAV$_{\text{ID}}$
        if the signal $y$ being checked is the latest emitted from such claimed UAV identity at current time $t$ and the signal $y$ was not sampled prior to $t - t_{\text{thres}}$ do
            Execute Algorithm 3 with signal $x$ and signal $y$ as the two signal inputs
            if the real identity of emitters of signal $x$ and signal $y$ are actually the same UAV do
                Mark corresponding entry as "Sybil" class
            else
                Mark corresponding entry as "Genuine" class
            end if
        end if
    end for
end
```

UAVs emitting those would potentially have already changed quite significantly. This threshold is represented by the near real-time limit t_{thres} in Algorithm 2.

B. Operational Phase

In the operational phase, UAVs would be deployed and carry out communications using either genuine or fake identities. In this phase, the two monitoring nodes would collect RSS and ToA data of radio signals and use Algorithm 1 to calculate RSSD and TDoA similarly to the training phase; however, the difference is that the true identity of each signal's emitter is not known. To predict whether any two near real-time signals collected and identified as coming from different UAVs are actually coming from the same UAV (i.e., a Sybil attack event), Algorithm 3 is executed and the resulting values of RSSD and TDoA ratios are passed on to the trained ML classifier for determination.

There are various ways in which the operational phase detection mechanism can be deployed. As an example, detection can be performed on all pairs of signals detected by the monitoring nodes and the ML classification results are passed on to the upper layers for appropriate risk-based decisions subjecting to other relevant information available. Alternatively, perhaps more efficiently, an individual request can be made by a mechanism in one of the upper layers to perform a check on any particular signals suspicious of being from a Sybil node.

IV. Simulation Environment

A simulation of the proposed scheme was set up on a desktop computer with an Intel i7 2.90-GHz processor, 32 GB of random access memory (RAM), and Windows 10 Enterprise operating system. The simulation can be divided into three stages: 1) simulation of a network of flying and communicating UAVs; 2) data preprocessing prior to ML classification; and 3) ML classification. We used the network simulator OMNeT++ [36] (Version 5.7) in conjunction with the INET framework [37] (Version 4.2.9) for the first stage, the output of which is a log file containing all communication records. Subsequently, for the second stage, a Python script was written and applied to the log file. This was performed to extract all relevant data, execute relevant algorithms described in Section III, and arrange the collated data to a format readable by the ML classifier used in the next stage. Finally, in the third stage, ML classification was carried out using the previously prepared training and test data. The tool used for the third stage was the Weka workbench platform (Version 3.8.5). Details of the three stages and further information on the Weka workbench platform are described in the following sections.

Algorithm 3: Calculate Ratios TDoA$_x$:TDoA$_y$ and RSSD$_x$:RSSD$_y$ for Two Given Signals x and y

```
Input: TDoA$_x$: Latest known near real-time TDoA of signal $x$ obtained from Algorithm 1,
       RSSD$_x$: Latest known near real-time RSSD of signal $x$ obtained from Algorithm 1,
       TDoA$_y$: Latest known near real-time TDoA of signal $y$ obtained from Algorithm 1,
       RSSD$_y$: Latest known near real-time RSSD of signal $y$ obtained from Algorithm 1
begin
    Calculate $\text{TDoA}_x : \text{TDoA}_y = \text{TDoA}_x \div \text{TDoA}_y$
    Calculate $\text{RSSD}_x : \text{RSSD}_y = \text{RSSD}_x \div \text{RSSD}_y$
end
```

A. Stage 1: Simulation of UAVs

In this stage, a network of flying and communicating UAVs was simulated in OMNeT++ using INET's "MassMobility" model. The UAVs movement model was based on INET's "3D Mobility" showcase [38], in which each UAV node moves in a 3-D space. To summarize, the UAV nodes moved at a speed randomly selected from a uniform distribution range between 10 and 20 m/s. Each node also turns at a random uniform distribution angle range between $-10°$ and $10°$ around a random elevation angle of the same uniformly distributed angle range. The positioning of the UAVs was configured to update every 1 s. In terms of the UAVs flying space, this was defined as a square of dimensions 1000×1000 m. As for the elevation, we restricted the range to be between 5–50 m to better reflect a more realistic permitted flying height for UAVs. On the ground, we added three fixed nodes: 1) the ground communications station at coordinates (250, 400); 2) the first monitoring node at coordinates (250, 250); and 3) the second monitoring node at coordinates (750, 750).

On the communications side, we used INET's "AckingWirelessInterface" wireless network interface module together with "ApskScalarRadio" hypothetical radios and the “ApskScalarRadioMedium” radio model which uses free space path loss by default [39]. A transmission frequency of 2 GHz was specified for use with this radio model. We did not define any antenna gains, which means that an isotropic antenna with a gain of 1 (0 dB) was used for each radio [40].

We simulated ML training data based on a network of 100 UAV nodes, 80 of which were genuine in that they only used their true identities to identify themselves in communications. Each genuine node transmitted one UDP packet to the ground communications station every 1-s period. The initial transmission time was different for each node, but ranged between simulation times $t = 1$ and $t = 2$ s. The other 20 UAV nodes were Sybil nodes, each of which used two different identities, namely, “A” and “B,” to identify itself. Each identity transmitted one UDP packet to the ground communications station every 1-s period. Similar to genuine nodes, the initial transmission time for Sybil nodes was different for each identity, but ranged between simulation times $t = 1$ and $t = 2$ s. Note that relating back to the near real-time limit t_{thres} described in Section III, the limit used here can be considered as not exceeding 1 s. The transmission period of 1 s can also be viewed either literally as each UAV identity communicated once a second, or perhaps more realistically, that each UAV identity communicated numerous times a second but only one of those got sampled.

In terms of the transmission power, we assumed that all UAV nodes are supposed to be operating at a power level that is not too high, in order to preserve their limited onboard battery energy. At the same time, the transmit power needs to be high enough to achieve reliable radio transmission in various environments and distances. Therefore, we defined all genuine UAV nodes to transmit at a power level of 100 mW, which is also assumed to be the maximum transmit power level. Conversely, Sybil nodes had the ability to adjust their transmission power down to a smaller level in an attempt to fool more traditional Sybil attack detectors.

We generated the training data for two scenarios: 1) where each Sybil node operates at a fixed transmit power level of 100 mW and 2) where each Sybil node operates at a fixed transmit power level of 100 mW for Identity A but at a range of power levels from 100 mW down to as low as 0.001 mW for Identity B. More specifically, power levels assigned to different Identity B UAV nodes are 100, 75, 50, 25, 10, 0.1, and 0.001 mW. The training data simulation for each scenario was carried out for a duration of 50 simulated seconds using seed-set value of “0.” Such a timing duration was chosen to achieve a balance of having sufficient training data samples while minimizing actual simulation execution time.

For the generation of test cases, we also used two main different transmit power scenarios similar to what we did for the training data and also used an execution duration of 50 simulated seconds; however, we generated more diverse cases. For instance, the tests include some power levels presented in the training data as well as some power levels not presented in the training data but still within the 0.001 to 100 mW range. For each test case, we did the evaluation on more diverse seed-set values, being from “1” through to “5.” Furthermore, we generated supplementary test cases for a new UAV network composition consisting of 98 genuine nodes and two Sybil nodes, also using various power levels within the same range and seed-set values of “1” through to “5.” Note that we will use the designation “Gx80Sx20” to refer to the network composition comprising 80 genuine nodes and 20 Sybil nodes. Similarly, we will use the designation “Gx98Sx2” to refer to the network composition comprising 98 genuine nodes and two Sybil nodes.

B. Stage 2: Data Preprocessing Prior to Machine Learning Classification

In this stage, for both the training and test data, we wrote a Python script to extract all relevant data from the output log file generated by OMNeT++ and arrange the data into an “ARFF” data set format readable by Weka. Each data set had three attributes: 1) TDoA_x:TDoA_y ratio; 2) RSSD_x:RSSD_y ratio; and 3) class label of either Sybil or Genuine. The generation of these attributes using Algorithm 2 is described in detail in Section III.

Because the simulated UAV networks consisted of a substantially higher number of genuine nodes than Sybil nodes, the generated data sets contained substantially more entries of the Genuine class. This means that the ML classifier would learn more characteristics of Genuine class data than Sybil class data, and thus would be more susceptible to overfitting the data to the characteristics of the Genuine class nodes. To mitigate this issue, we decided to also create a trimmed down version of the training data which randomly skips some entries of the Genuine class so that there are roughly equal entries for the Genuine and Sybil classes overall. We performed some quick experiments and confirmed that using the untrimmed version for training resulted in the classifier having a much poorer performance in detecting Sybil class entries. As an example, Table II illustrates the OneR classification results when using the trimmed and untrimmed training data sets for Scenario 2 described in the previous section evaluated against the trimmed and untrimmed versions of one of the Gx80Sx20 test data sets. Note that the details of how this table was populated can be referred to in the next section. Unsurprisingly, the use of untrimmed training data led to very high true positive detection rates of Genuine class entries but very low true positive detection rates of Sybil class entries. Although such use led to a very high average overall accuracy percentage when evaluated with the untrimmed test data set, this was only so because there were significantly more instances of Genuine class data. As can be seen, when using such untrimmed training data evaluated with the trimmed test data set, the average overall accuracy percentage was very low. Similar results were also obtained with the use of different test data sets and classifiers. Consequently, we decided to use the trimmed version of the data for training. For testing, although it may be more realistic to use the untrimmed data, we decided to also experiment with the trimmed data for the Gx80Sx20 composition in order to observe the ML classification performance more thoroughly.

TABLE II
ONER CLASSIFICATION RESULTS—TRAINING DATA SET EVALUATED WITH A GX80SX20 TEST DATA SET—TRIMMED VERSUS UNTRIMMED

Training Dataset	Overall Correct Classification Percentage	True Positive Sybil Instances	False Negative Sybil Instances	True Positive Genuine Instances	False Negative Genuine Instances	True Positive Sybil Percentage	True Positive Genuine Percentage
	Evaluated with untrimmed version of test dataset:						
Trimmed	91.07%	1837	83	622412	61108	95.68%	91.06%
Untrimmed	99.71%	66	1854	683380	140	3.44%	99.98%
	Evaluated with trimmed version of test dataset:						
Trimmed	94.16%	1837	83	1775	141	95.68%	92.64%
Untrimmed	51.67%	66	1854	1916	0	3.44%	100%

C. Stage 3: Machine Learning Classification

In this stage, we evaluated the training and test data in Weka. The Weka platform comes included with a collection of classifiers of different algorithm types. Furthermore, additional classifiers are also available as optional downloadable packages. We carried out preliminary experiments with most, if not all, of the classifiers that support the problem scenario in this study and shortlisted a few high-performing ones. This process then narrowed down to the four chosen algorithms, namely, J48, Classification via Regression, OneR, and JRip. Note that these well-researched algorithms are of three different types, more details of which can be found in the next section for interested readers. These diverse algorithms were then used in carrying out the full experiments to test the robustness of the scheme.

We captured the following output results for evaluation: 1) the accuracy of correctly classified instances overall; 2) the number of Sybil class entries correctly identified as Sybil class (i.e., "true positive Sybil" or equivalently "true negative Genuine"); 3) the number of Sybil class entries incorrectly identified as Genuine class (i.e., "false negative Sybil" or equivalently "false positive Genuine"); 4) the number of Genuine class entries correctly identified as Genuine class (i.e., "true positive Genuine" or equivalently "true negative Sybil"); and 5) the number of Genuine class entries incorrectly identified as Sybil class (i.e., "false negative Genuine" or equivalently "false positive Sybil").

Note that the second and third outputs can be used to calculate the percentage of true positive Sybil entries detection. Similarly, the fourth and fifth outputs can be used to calculate the percentage of true positive Genuine entries detection. Another point to note is that, we can add the second and fourth outputs together to obtain the overall correct classification instances. Likewise, we can add the third and fifth outputs together to obtain the overall incorrect classification instances.

D. Weka Workbench Platform

The Weka workbench platform is a popular open-source software for ML [41]. Weka comes with a collection of classifiers, where we focus on the following four: 1) J48; 2) Classification via Regression; 3) OneR; and 4) JRip. These four classifiers are based on three different algorithm types: 1) decision tree; 2) metalearning; and 3) rules. We briefly describe these different classifiers below.

The J48 classifier is a decision tree-type algorithm. Decision trees define the sequences of decisions to be made together with the resulting recommendation. Each node in a decision tree evaluates a specific attribute until a leaf node is reached, which is where the classification decision is made. The J48 classifier is a derivation of a straightforward divide-and-conquer algorithm called "C4.5" [42] which needed to be extended in order to cater for real-world problems [43].

The Classification via Regression classifier is a metalearning type algorithm. Metalearning algorithms take classifiers and make them into more powerful learners or change them for other applications [43]. In the case of the Classification via Regression classifier, it performs classification on discrete classes using regression methods which would otherwise only be suitable for continuous classes. Note that the M5P decision tree classifier [44], which is the default option, was used in our experiments.

The OneR and JRip classifiers are rules-type algorithms. Rules-based classifiers are popular alternatives to decision trees. Rules can be much more consolidated than decision trees, especially when it is possible to have a default rule covering cases not defined by other rules. Another reason for rules popularity is that new rules can be added to existing ones without disrupting the other rules already in place [43].

The OneR classifier, which is also called "1R" or "1-rule," is Weka's implementation of Holte [45]. It works based on a set of rules applied to just one attribute by creating a different set of rules for each attribute and choosing the best one based on the resulting error rates. It is described as a simple and efficient method that can still produce effective rules that can often achieve surprisingly high accuracy. An explanation for such a phenomenon is that often the pattern underlying any real-world data is quite fundamental that even only just one attribute of the data is adequate for performing accurate predictions [43].

The JRip classifier is Weka's implementation of the repeated incremental pruning to produce error reduction (RIPPER) rule learner [46]. It is based on the idea of using incremental reduced-error pruning by Fürnkranz and Widmer [47] for quick and effective rule inference [43].

V. SIMULATION RESULTS AND EVALUATION

In this section, we look at the ML classification results obtained from the simulation exercises described in Section IV, where the results for all test data sets were

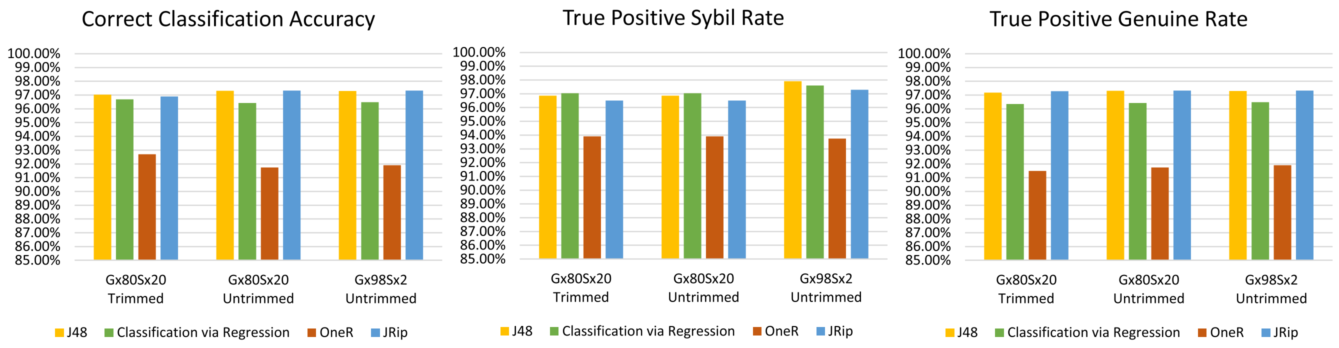

Fig. 2. ML classification results—Sybil nodes with fixed transmit power level.

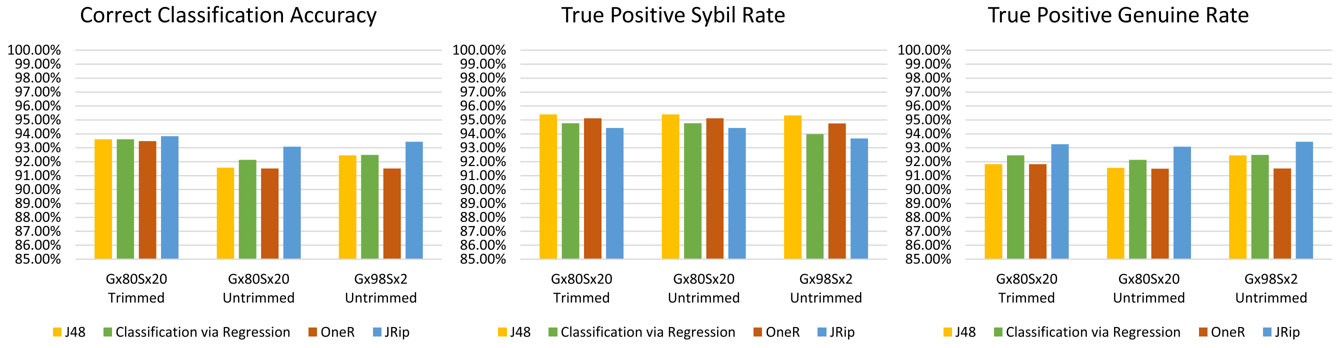

Fig. 3. ML classification results—average results for Sybil nodes with variable transmit power level.

obtained from five simulation runs using the five different seed-set values. To summarize, a high correct classification accuracy of above 91% on average was achieved across all four selected ML algorithms, even in scenarios with smart malicious nodes operating at power levels not directly trained. Such a high performance reflects the suitability of the design choices made for the proposed architecture, especially the selection of the two ML attributes, namely, the RSSD and TDoA ratios of two different signals. Additionally, the results also reflect the robustness of the proposed architecture in upholding high performance when different ML classifiers are used.

The following two sections discuss the results in more detail. Note that for the evaluation metrics, we use three criteria (refer Section IV): 1) the accuracy of correctly classified instances overall ("correct classification accuracy"); 2) the percentage of true positive Sybil entries detection ("true positive Sybil rate"); and 3) the percentage of true positive Genuine entries detection ("true positive Genuine rate").

A. Sybil Nodes With Fixed Transmit Power Level

In this section, we examine the performance of Scenario 1, the results of which are shown in Fig. 2. This is a simpler scenario in which Sybil nodes can only transmit at a fixed power level of 100 mW. It can be seen that the correct classification accuracies exceed 96% for all classifiers except for OneR which performs slightly worse in this scenario but still exceeds 91%. Similar results can also be observed when looking more specifically at true positive Sybil and true positive Genuine rates. Another observation about OneR is that it also performs worst in terms of its equitability in distinguishing Sybil and Genuine class entries, with the gaps between the true positive Sybil and true positive Genuine detection rates being the largest among the three classifiers.

B. Sybil Nodes With Variable Transmit Power Level

Here, we consider Scenario 2, which represents the more complex cases where Sybil nodes can vary their transmit power level. For these cases, a training data set containing Sybil nodes with seven different transmit power levels was used to train the classifiers. We conducted testing using diverse data sets with various transmit power levels, some of which were included in the training data set and some of which were not.

1) Average Results: To characterize the results more generally, we examine the average results obtained from the use of all test data sets for each of the two different node compositions, as shown in Fig. 3. When compared with the fixed power results shown in Fig. 2, it can be seen that the correct classification accuracies of the four classifiers decrease by a few percent, but all still exceed 91%. Likewise, the true positive Sybil and true positive Genuine rates also decrease slightly, with the results for true positive Sybil appearing to be slightly higher than that of true positive Genuine for all classifiers; however, the gaps are smallest for the JRip classifier, indicating that it is the most equitable one in distinguishing Sybil and Genuine class entries. Interestingly, unlike the results for the fixed power scenario, the performance of the OneR classifier is now more similar to that of the other three classifiers. This is perhaps not too surprising because as outlined in Section IV, OneR only uses one attribute to create

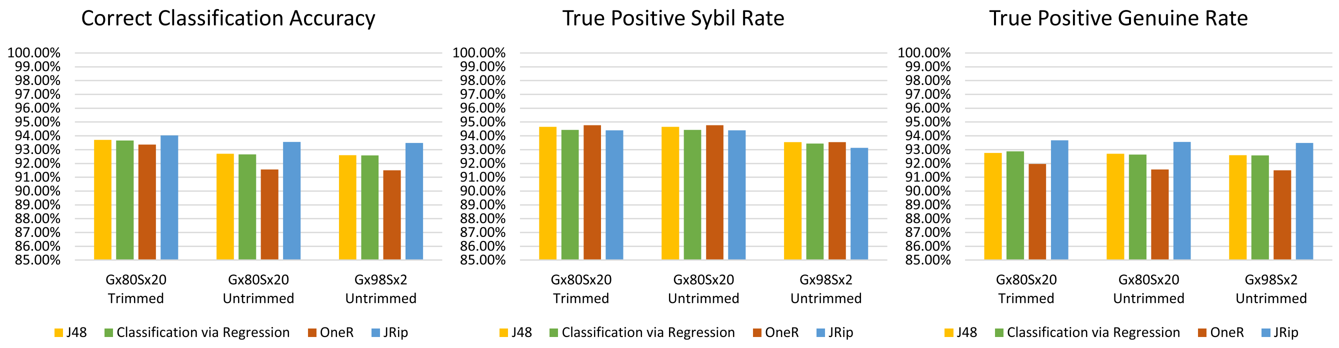

Fig. 4. ML classification results—Sybil nodes with trained identity B transmit power level (50 mW).

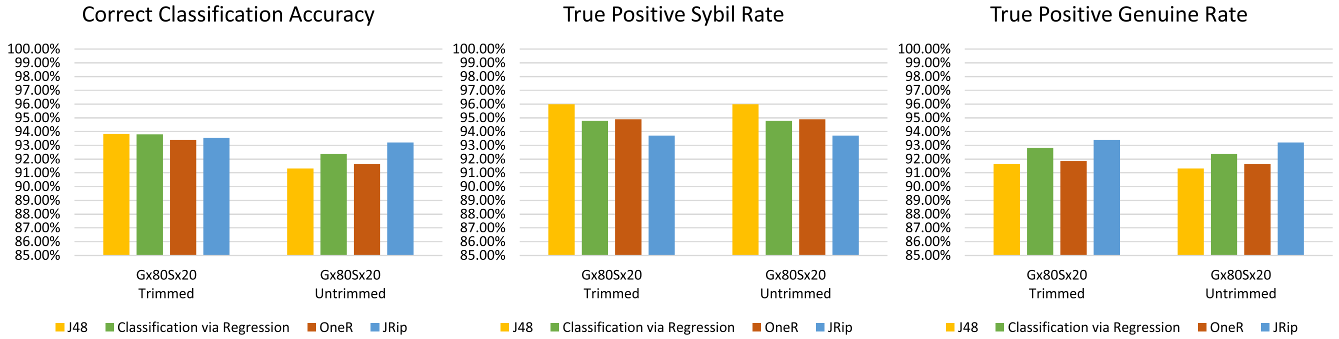

Fig. 5. ML classification results—Sybil nodes with untrained identity B transmit power levels (mixture of 40, 3, 0.6, 0.03, and 0.007 mW).

rules, and so the performance in some situations would be worse than the other classifiers that use all attributes available.

2) More Detailed Samples of Results: We can now look more closely at the performance differences between some sample test cases where the transmit power levels of Sybil nodes were included in the training data set versus those that were not.

First, let us look at an example situation where the Sybil nodes can only transmit at a fixed power level that is already included in the training data set. Fig. 4 illustrates the classification execution results of one such example situation, where each Sybil node transmits at a power level of 50 mW for its Identity B. When comparing this with the average results shown in Fig. 3, it can be seen that the results are fairly consistent with one another. The correct classification accuracies of all four classifiers exceed 91%. Similarly, the true positive Sybil rates are only slightly higher than the true positive Genuine rates for all classifiers.

Next, we consider a situation where the transmit power level of each Sybil node's Identity B has not been included in the training data set. An example situation is illustrated in Fig. 5, which captures the results of a diversified test case where there are five different transmit power levels of Identity B used among the Sybil nodes, namely, 40, 3, 0.6, 0.03, and 0.007 mW. Note that we only created such a situation for the Gx80Sx20 node composition. We did not create a similar situation for the Gx98Sx2 node composition because the small number of Sybil nodes in this case would not be effective in demonstrating the intended diversification. In terms of the classification results comparison, it can be seen that the results are also in line with the average results captured in Fig. 3, where the correct classification accuracies for all classifiers exceed 91%.

C. Future Works

The experimental results in this study were obtained from simulations carried out based on several assumptions which may not necessarily hold true in all situations. Therefore, more considerations are required prior to the actual deployment of the proposed scheme and may necessitate further experiments and adaptations, as appropriate. In addition, the proposed scheme may also be extendable to provide improvements and additional functionalities.

1) Additional Considerations Prior to Deployment: Examples of issues that may require additional consideration prior to deployment are as follows. First, this study was carried out based on the assumption that the free space path-loss propagation model holds true and that signals from other systems in the surrounding area are coordinated in such a way that results in negligible interference effects on the functionality of the system, such as through the use of orthogonal frequency-division multiplexing. Consequently, further assessments would need to be done on the effects of interference and structural blockages applicable at the physical location the system is planned to be deployed in.

This study also uses a specific UAV mobility model taken from an INET framework's showcase, which defines how different UAVs move around in a range of random speeds and directions. Simulations were also performed on specific flying space dimensions and node density levels. Furthermore, the simulations carried out used only one near real-time threshold value which is $t_{\text{thres}} = 1$ s. In practice, depending on the application, it is possible that nodes may be required to fly in a higher-density environment. They may also be required to use different mobility patterns or signal emission rates. Therefore, further studies are needed on these aspects prior to deployment, as appropriate.

The UAVs transmit power levels used in this study range from the maximum value of 100 mW down to the minimum value of 0.001 mW. Although these values led to great simulation results, further investigations would need to be done to confirm the performance based on the expected minimum and maximum UAVs transmit power levels applicable to the deployment scenario.

2) Use of Alternative Machine Learning Attributes: The RSSD and TDoA ratios of two different radio signals were selected as the attributes used for ML in this study following our hypothesis that they capture significant location information regarding any given UAV node at a particular point in time when used together. From the simulation results, the use of these two attributes was found to be quite effective in detecting Sybil attacks. Nevertheless, more studies can be carried out in the future to investigate whether the performance of the scheme can be improved even further if additional and/or different attributes are used, including those derived from other physical layer features, especially if such other features are easily obtainable in the intended deployment scenario.

3) Extension to Support Unsupervised Machine Learning and Other Attack Types: In reality, there may be situations in which data sets for ML training are not easily obtainable. In such situations, the use of supervised ML may not be ideal. As a potential solution, unsupervised ML, which uses input data sets without class labels to independently extract useful information and patterns [32], may need to be considered as an extension of the scheme. Likewise, considerations should be given to extending the scheme to cater for other attack types in FANETs, a good starting point of which might be those that also involve location verification.

4) Adaptation to Support Other Application Scenarios: Notwithstanding the fact that the proposed scheme was designed for and experimented in the FANETs environment, the approach may also function well in other application scenarios, either as is or with some modifications. As an example, in the case of VANETs, the mobility patterns where vehicles of certain height travel on known roads can be considered 2-D, which is more restrictive than the 3-D mobility in FANETs. However, there are similarities that may enable the mechanisms underlying the proposed scheme to also function well in such an environment. Additionally, research on VANETs is also more mature and thus trusted infrastructures exist, such as roadside units (RSUs), which may be advantageous for the adaptation of the proposed scheme (e.g., the RSUs can potentially be used as ground monitoring nodes).

VI. Conclusion

We proposed a supervised ML approach to intelligently detect Sybil attacks for FANETs-based IoFT. Simulation results revealed that the proposed scheme can achieve a high correct classification accuracy of above 91% on average, even for smart malicious nodes with power control capability operating at power levels not directly trained. Correspondingly, this means that the proposed scheme has a low false classification rate of less than 9% on average. Additionally, because of the use of only intrinsically generated physical layer data, the proposed scheme is also less susceptible to various attacks commonly carried out on the upper layers, such as data spoofing. Furthermore, no additional communications overheads of the UAV nodes are required for the functionality of this scheme. For future works, it may be possible to extend this scheme beyond Sybil attack detection applications, for example, to address other problems in FANETs that involve location verification. In addition, extensions and adaptations to support unsupervised ML and other application scenarios can also be investigated.

References

[1] D. F. Pigatto et al., "The Internet of Flying Things," in *Internet of Things A to Z: Technologies and Applications*. Hoboken, NJ, USA: Wiley, 2018, pp. 529–561.

[2] A. Chriki, H. Touati, H. Snoussi, and F. Kamoun, "FANET: Communication, mobility models and security issues," *Comput. Netw.*, vol. 163, Nov. 2019, Art. no. 106877.

[3] Q. Gu, T. Fan, F. Pan, and C. Zhang, "A vehicle-UAV operation scheme for instant delivery," *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106809.

[4] N. Gao, L. Liang, D. Cai, X. Li, and S. Jin, "Coverage control for UAV swarm communication networks: A distributed learning approach," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19854–19867, Oct. 2022.

[5] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT enabled UAV: Network architecture and routing algorithm," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3727–3742, Apr. 2019.

[6] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, "Federated learning in the sky: Aerial-ground air quality sensing framework with UAV swarms," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9827–9837, Jun. 2021.

[7] J. Zheng et al., "An efficient strategy for accurate detection and localization of UAV swarms," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15372–15381, Oct. 2021.

[8] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the Internet of Things," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1636–1675, 2nd Quart., 2019.

[9] S. Murali and A. Jamalipour, "A lightweight intrusion detection for Sybil attack under mobile RPL in the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 379–388, Jan. 2020.

[10] J. Sun et al., "An intrusion detection based on Bayesian game theory for UAV network," in *Proc. 11th EAI Int. Conf. Mobile Multimedia Commun.*, Qingdao, China, 2018, pp. 56–67.

[11] M. A. Lopez, M. Baddeley, W. T. Lunardi, A. Pandey, and J.-P. Giacalone, "Towards secure wireless mesh networks for UAV swarm connectivity: Current threats, research, and opportunities," in *Proc. 17th Int. Conf. Distrib. Comput. Sens. Syst. (DCOSS)*, 2021, pp. 319–326.

[12] A. Arshad, Z. M. Hanapi, S. Subramaniam, and R. Latip, "A survey of Sybil attack countermeasures in IoT-based wireless sensor networks," *PeerJ Comput. Sci.*, vol. 7, p. e673, Sep. 2021.

[13] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1686–1721, 3rd Quart., 2020.

[14] E. Frank, M. A. Hall, and I. H. Witten, "The WEKA workbench," in *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Cambridge, MA, USA: Morgan Kaufmann, 2016.

[15] D. Dardari, E. Falletti, and M. Luise, *Satellite and Terrestrial Radio Positioning Techniques: A Signal Processing Perspective*. London, U.K.: Academic, 2012.

[16] D. Munoz, F. Bouchereau, C. Vargas, and R. Enriquez, *Position Location Techniques and Applications*. Amsterdam, The Netherlands: Academic, 2009.

[17] X. Li, Z. D. Deng, L. T. Rauchenstein, and T. J. Carlson, "Contributed review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements," *Rev. Sci. Instrum.*, vol. 87, no. 4, 2016, Art. no. 41502.

[18] A. Vasudeva and M. Sood, "Survey on Sybil attack defense mechanisms in wireless ad hoc networks," *J. Netw. Comput. Appl.*, vol. 120, pp. 78–118, Oct. 2018.

[19] A. K. Singh, "Innovative survey of defense machinery against Sybil attacks over wireless ad-hoc network on IoT," *J. Eng. Res.*, vol. 9, no. 2, pp. 92–105, 2021.

[20] G. Shobana and X. A. R. Arockia, "Detection mechanism on vehicular adhoc networks (VANETs) a comprehensive survey," *Int. J. Comput. Sci. Netw. Security*, vol. 21, no. 6, pp. 294–303, 2021.

[21] Y. Zhang, B. Das, and F. Qiao, "Sybil attack detection and prevention in VANETs: A survey," in *Proc. Future Technol. Conf. (FTC)*, Vancouver, BC, Canada, 2020, pp. 762–779.

[22] N. C. Velayudhan and A. Anitha, "Sybil attack in VANET operating in an urban environment: An overview," in *Advances in Communication Systems and Networks*. Singapore: Springer, 2020, pp. 433–442.

[23] B. Hammi, Y. M. Idir, S. Zeadally, R. Khatoun, and J. Nebhen, "Is it really easy to detect Sybil attacks in C-ITS environments: A position paper," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18273–18287, Oct. 2022.

[24] M. Kabbur and V. A. Kumar, "MAR_Sybil: Cooperative RSU based detection and prevention of Sybil attacks in routing process of VANET," in *Proc. J. Phys. Conf. Ser.*, 2020, Art. no. 12009.

[25] Y. Yuan, L. Huo, Z. Wang, and D. Hogrefe, "Secure APIT localization scheme against Sybil attacks in distributed wireless sensor networks," *IEEE Access*, vol. 6, pp. 27629–27636, 2018.

[26] S. Lv, X. Wang, X. Zhao, and X. Zhou, "Detecting the Sybil attack cooperatively in wireless sensor networks," in *Proc. Int. Conf. Comput. Intell. Security*, 2008, pp. 442–446.

[27] S. Abbas, M. Merabti, D. Llewellyn-Jones, and K. Kifayat, "Lightweight Sybil attack detection in MANETs," *IEEE Syst. J.*, vol. 7, no. 2, pp. 236–248, Jun. 2013.

[28] A. Angappan, T. P. Saravanabava, P. Sakthivel, and K. S. Vishvaksenan, "Novel Sybil attack detection using RSSI and neighbour information to ensure secure communication in WSN," *J. Ambient Intell. Humanized Comput.*, vol. 12, pp. 6567–6578, Jul. 2021.

[29] C. F. E. de Melo et al., "UAVouch: A secure identity and location validation scheme for UAV-networks," *IEEE Access*, vol. 9, pp. 82930–82946, 2021.

[30] E. Walia, V. Bhatia, and G. Kaur, "Detection of malicious nodes in flying ad-hoc networks (FANET)," *Int. J. Electron. Commun. Eng.*, vol. 5, no. 9, pp. 6–12, 2018.

[31] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[32] A. Jamalipour and S. Murali, "A taxonomy of machine-learning-based intrusion detection systems for the Internet of Things: A survey," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9444–9466, Jun. 2022.

[33] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020.

[34] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to Pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, 3rd Quart., 2020.

[35] H. T. Friis, "A note on a simple transmission formula," *Proc. IRE*, vol. 34, no. 5, pp. 254–256, May 1946.

[36] "OMNeT++ discrete event simulator." OpenSim Ltd. 2022. Accessed: Jun. 29, 2021. [Online]. Available: https://omnetpp.org

[37] "INET framework." OpenSim Ltd. 2022. Accessed: Jun. 29, 2022. [Online]. Available: https://inet.omnetpp.org

[38] "Showcases > mobility > 3D mobility." OpenSim Ltd. Accessed: Jun. 29, 2022. [Online]. Available: https://inet.omnetpp.org/docs/showcases/mobility/spatial/doc

[39] "Tutorials > wireless tutorial > step 13. Configuring a more accurate path loss model." OpenSim Ltd. Accessed: Jun. 29, 2022. [Online]. Available: https://inet.omnetpp.org/docs/tutorials/wireless/doc/step13.html

[40] "Tutorials > wireless tutorial > step 14. Introducing antenna gain." OpenSim Ltd. Accessed: Jun. 29, 2022. [Online]. Available: https://inet.omnetpp.org/docs/tutorials/wireless/doc/step14.html

[41] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9042–9053, Oct. 2019.

[42] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[43] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Cambridge, MA, USA: Morgan Kaufmann, 2017.

[44] Y. Wang and I. H. Witten, "Induction of model trees for predicting continuous classes," in *Proc. 9th Eur. Conf. Mach. Learn.*, Prague, Czechia, 1997, pp. 128–137.

[45] R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Mach. Learn.*, vol. 11, pp. 63–90, Apr. 1993.

[46] W. W. Cohen, "Fast effective rule induction," in *Proc. 12th Int. Conf. Mach. Learn.*, Tahoe City, CA, USA, 1995, pp. 115–123.

[47] J. Fürnkranz and G. Widmer, "Incremental reduced error pruning," in *Proc. 11th Int. Conf. Rutgers University*, New Brunswick, NJ, USA, 1994, pp. 70–77.

Donpiti (Mick) Chulerttiyawong received the B.Eng. (Hons.) and B.Inf.Tech. degrees from The Australian National University, Australia, and the M.Proj.Mgt. degree from The University of New South Wales, Australia. He is currently pursuing the Ph.D. degree with the School of Electrical and Information Engineering, the University of Sydney, Australia, where his research focus is on improving security for the Internet of Things.

He has worked as a Professional Engineer in different industries, including telecommunications, defense, and transport, in both public and private sectors. He is currently a Chartered Professional Member of Engineers Australia (MIEAust CPEng).

Abbas Jamalipour (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Nagoya University, Nagoya, Japan, in 1996.

He holds the position of Professor of Ubiquitous Mobile Networking with The University of Sydney, Camperdown, NSW, Australia. He has authored nine technical books, 11 book chapters, over 550 technical papers, and five patents, all in the area of wireless communications and networking.

Prof. Jamalipour is a recipient of the number of prestigious awards, such as the 2019 IEEE ComSoc Distinguished Technical Achievement Award in Green Communications, the 2016 IEEE ComSoc Distinguished Technical Achievement Award in Communications Switching and Routing, the 2010 IEEE ComSoc Harold Sobol Award, the 2006 IEEE ComSoc Best Tutorial Paper Award, as well as over 15 Best Paper Awards. He was the President of the IEEE Vehicular Technology Society from 2020 to 2021. Previously, he held the positions of the Executive Vice-President and the Editor-in-Chief of VTS Mobile World and has been an Elected Member of the Board of Governors of the IEEE Vehicular Technology Society since 2014. He was the Editor-in-Chief IEEE WIRELESS COMMUNICATIONS, the Vice President-Conferences, and a member of Board of Governors of the IEEE Communications Society. Since January 2022, he has been the Editor-in-Chief of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He sits on the Editorial Board of the IEEE ACCESS and several other journals and is a member of Advisory Board of IEEE INTERNET OF THINGS JOURNAL. He has been the General Chair or Technical Program Chair for several prestigious conferences, including IEEE ICC, GLOBECOM, WCNC, and PIMRC. He is a Fellow of the Institute of Electrical, Information, and Communication Engineers, and the Institution of Engineers Australia, an ACM Professional Member, and an IEEE Distinguished Speaker.