IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 22, 2021, accepted February 19, 2021, date of publication March 1, 2021, date of current version March 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063002

Active Machine Learning Adversarial Attack Detection in the User Feedback Process

VICTOR R. KEBANDE[1,2,3], SADI ALAWADI[4], FERAS M. AWAYSHEH[5], AND JAN A. PERSSON[1,2]

[1]Internet of Things and People (IOTAP) Center, Malmö University, 211 19 Malmö, Sweden
[2]Department of Computer Science, Malmö University, 211 19 Malmö, Sweden
[3]Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden
[4]Division of Scientific Computing, Department of Information Technology, Uppsala University, 752 36 Uppsala, Sweden
[5]Delta Research Center, Data Systems Group, University of Tartu, 51009 Tartu, Estonia

Corresponding author: Victor R. Kebande (victor.kebande@mau.se)

This work was supported in part by The Swedish Knowledge Foundation through the Internet of Things and People (IOTAP), Malmö University, Malmö, Sweden, under Grant 20140035.

ABSTRACT Modern Information and Communication Technology (ICT)-based applications utilize current technological advancements for purposes of streaming data, as a way of adapting to the ever-changing technological landscape. Such efforts require providing accurate, meaningful, and trustworthy output from the streaming sensors particularly during dynamic virtual sensing. However, to ensure that the sensing ecosystem is devoid of any sensor threats or active attacks, it is paramount to implement secure real-time strategies. Fundamentally, real-time detection of adversarial attacks/instances during the User Feedback Process (UFP) is the key to forecasting potential attacks in active learning. Also, according to existing literature, there lacks a comprehensive study that has a focus on adversarial detection from an active machine learning perspective at the time of writing this paper. Therefore, the authors posit the importance of detecting adversarial attacks in active learning strategy. Attack in the context of this paper through a *UFP-Threat driven model* has been presented as any action that exerts an alteration to the learning system or data. To achieve this, the study employed ambient data collected from a smart environment human activity recognition from (Continuous Ambient Sensors Dataset, CASA) with fully labeled connections, where we intentionally subject the Dataset to wrong labels as a targeted/manipulative attack (by a malevolent labeler) in the UFP, with an assumption that the user-labels were connected to unique identities. While the dataset's focus is to classify tasks and predict activities, our study gives a focus on active adversarial strategies from an information security point of view. Furthermore, the strategies for modeling threats have been presented using the Meta Attack Language (MAL) compiler for purposes adversarial detection. The findings from the experiments conducted have shown that real-time adversarial identification and profiling during the UFP could significantly increase the accuracy during the learning process with a high degree of certainty and paves the way towards an automated adversarial detection and profiling approaches on the Internet of Cognitive Things (ICoT).

INDEX TERMS Adversarial detection, user-feedback-process, active machine learning, monitoring industrial feedback.

I. INTRODUCTION

While many Internet-of-Things (IoT) technologies are applying Machine Learning (ML) in implementing security solutions, it has become apparent that most sophisticated attacks are propagated against machine learning-based systems [1]. Furthermore, most of the IoT infrastructure-based attacks succeed as a result of varying adversary intentions and expectations. Targeted or manipulative attacks where a ML model may be deliberately tuned to take in altered training sets, inputs, and to provide false output are particular examples how these adversarial attacks are propagated. While targeted attacks are assumed to be deliberate or intentional in nature [2], it is imperative to note that the success of targeted

The associate editor coordinating the review of this manuscript and approving it for publication was Mervat Adib Bamiah.

attacks is mainly dependent on the threat and vulnerability surface of the machine learning model, IoT infrastructure, or the nature of the attack.

Even though it is important to ensure that a machine learning model's accuracy is maintained or achieved during classification, there is also a possibility of having malicious content in form of adversaries (targeted, unintentional etc), that can influence the outcome of machine learning systems during active learning. This, is owing to the fact that, there are instances an oracle/human agent may be needed to provide expert labels. Hence, it becomes important to prioritize the probabilities of an oracle/human agent exacerbating malicious content [3], or having reluctant fallible users [4] based on existing vulnerabilities. As a result, vulnerabilities may easily be used by an adversary to intentionally obstruct the learning process, which may result to an interference with the output's accuracy. Obscurity, among other targeted attacks in the author's perspective, could occur during the User-Feedback Process (UFP) through an active learning strategy, for example, where a Dynamic Intelligent Virtual Sensor (DIVS) is deployed. In this context, DIVS is presented as a virtual sensor within a heterogeneous environment that consists of an abstraction layer that overlays the physical infrastructure [5].

During the UFP (See Fig. 1), the user/oracle/human agent's behavior being queried during continued learning may differ based on their perceived intentions or motives. In some circumstances these intentions could either be deliberate or unintentional. Deliberate intentions sometimes may be assessed in situations where an oracle/ human agent may be a malicious labeler or a normal uncoordinated attack by an adversary. The prevalence of adversarial attacks in ML, in this context, presents a significant challenge that is worth exploring. Also, based on the ever-rising complexity of attacks and integration to real world applications, there is need, from an information security standpoint to realise desirable approaches that can defend the learning system against the varying intentions of a potential attacker. An attacker, in the context of this study has been used to portray any human profile or agent that interacts with the system — with the sole aim of altering the learning system or data particularly during UFP. We argue that characterizing the identities and actions of potential attackers or IoT devices during the process of continued learning is a viable step towards the creation of a suitable threat model that can be used to develop automated adversarial detection and profiling approaches.

As a step towards identifying baseline attacks that can succeed in this context, we have identified a general knowledge-base adversary tactics based on MITRE ATT&CK, which have also been used as a foundation for detecting adversarial attack points. Notably, from a preliminary perspective, we coin the term Generic Induced Attacks (GIA) that has relevance to potential attacks that emanates from unique identities, which also forms the generic or fundamental attacks from the documented novel CAPEC/MITRE ATT&CK matrices. CAPEC/ MITRE ATT&CK matrices are presented as standard adversarial attacks that can prevail in any vulnerable environment. From these generic attacks, we map the GIA's behavior to the UFP threat model [6] in order to identify different assumptions that are modeled as a step towards the detection of potential security goals violations in the perspective of secure online learning. These assumptions could easily be exploited by an adversary, which plays a vital role in threat and attack detection.

Therefore, this paper sets a precedence in exploring hurdles that exist due to the presence of adversarial active attacks on the ML model (with a focus on interactive and online learning), specifically during the UFP. We have countered this by employing (Human activity recognition from continuous ambient sensors Dataset, CASA) with fully labeled connections by intentionally falsifying the labels as a targeted/manipulative attack.

Furthermore, we have employed nine ML algorithms in our experiments to aid in detecting potential attacks, this has been evaluated before the attack, after the attack and also during the attack with interactive learning, UFP. It is important to note that nine ML algorithms have been used to check the influence that an attack can have during active learning strategy and to also show the performance of various classifiers. In the long run, we aim to investigate whether a ML model can be improved in a fashion that it allows secure learning. Notably, we assess situations that can allow compromise-where a ML model can be subjected to exacerbate potential vulnerabilities based on the existing general adversarial tactics (MITRE ATT&CK) in active learning. By identifying this, it would ultimately guarantee secure learning for a ML algorithm during UFP.

The remainder of this paper is organized as follows: Section II briefly presents the Background and Related Literature followed by Adversarial Detection Approaches in Section III. Section IV presents Modeling Attacks in UFP Process alongside the threat modeling. Experimental evaluations are presented in Section V while comparative analysis is presented in Section VII. We conclude and make mention of the future research work in Section VIII.

II. BACKGROUND AND RELATED LITERATURE

A. UFP THREAT MODEL

The UFP relates to a querying strategy in uncertain sampling [8], where in active learning, an oracle or a human agent is considered to be in the loop and the learning model is able to query this oracle or a human agent to give labels. Generally, active learning assumes that the DIVS is able to expect some feedback when requested (precisely, when labels are requested durng active learning), while there is continuous interaction with other users [7], [9]. The important aspect of the DIVS is that, it is able to adjust based on the changing nature of IoT environment. For example, during online learning, there is need for the DIVS to query the user/oracle during data labeling in a UFP as is shown in Figure 1, in the DIVS processing pipeline [7], [9]. We make assumptions

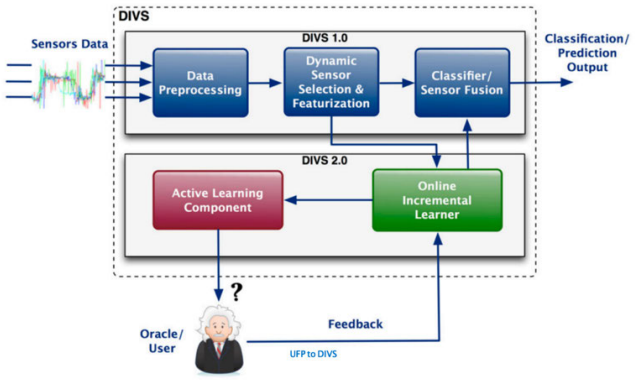


FIGURE 1. DIVS processing pipeline [7].

that for a case of multiple oracles, they may posses unique identities which gives labels to the learning model. Furthermore, the UFP-TM has been modeled to assume that an adversary's attempt are basically aimed to capitalize on the non-robustness of machine learning algorithms through targeted attacks (Logical and physical), thus leading to the assumption that, there always may exist a vulnerability that can be exploited. Based on this premise, we argue from this context that, it is possible for a trained classifier, C to be able to correctly classify an instance $x \in X$, where the actual goal of an adversary is to influence the classifier to classify an instance $x' \in X$ wrongly based on a vulnerability, as a targeted attack as $(x' \in X)' \text{–> } oracle$. From this perspective, deliberate false labels injection to the DIVS could, for example, be an aspect that interferes with online learning. Consequently, while existing researches mainly have a focus on how machine learning models can be fooled, the UFP-TM, from an information security stand point assumes based on classification (*object*, *activity*) in a setting, and restrictions can prevent the human oracle/agent that poses as an adversary from manipulations.

That notwithstanding, it is essential to anticipate potential attacks on DIVS and to identify respective countermeasures on the DIVS. As a result, we explore the UFP Threat Model (UFP-TM) [6], which mainly is focused on addressing major security assumptions in continued learning of the DIVS which is shown in Figure 1. Basically, the DIVS utilizes online strategies that allows a ML model to undergo training by way of labelled instances in order to give the desired output. Apart from that, the context of this study concentrates on considering active learning [7], [9] strategies with the involvement of a user/oracle that allows the DIVS to be able to provide the user feedback, which together in this study has been used to model the threat model [6]. The UFP-TM address some security assumptions, which, may hamper proper learning of the DIVS or the output when the oracle/user that is queried by DIVS can, for instance, decide to falsify or tamper with the labels based on streaming sensor data or if the learning model itself is attacked. It is worth noting that tampering can also be directed to the input data when the model is learning or after the learning model has been trained. Consequently, it is always important that there is continued learning while the user is queried during the UFP. Based on the UFP-TM [6], the author highlights the threat assumptions in Table 1 that have been identified in the buildup of the UFP threat model.

B. RELATED WORK

Machine Learning has made great strides in recent years, with an impressive performance on many applications such as real-time feedback analysis. Although ML systems can be useful when deployed in an iterative supervised learning realm, they are not perfect. Especially that most exciting advances in ML require large-scale volumes of data, making data labeling the new bottleneck. Hence, a new adaptive and incremental learning algorithm and strategies that combine concepts from the field of ML is required to improve the quality of the classification model and decrease the complexity of training instances. This learning strategy proactively selects the subset of available examples to be labeled next from a pool of yet unlabeled instances in what is called active ML [10], [11]. This approach is well-motivated in many modern ML problems, in particular, when labels are complicated, time-consuming, or expensive to collect [12]. Also studies have highlighted continued research on activity recognition techniques based on interactive machine learning, for example in dynamic sensor environment where streaming data is able to be used to measure accuracy [9], [13], [14].

Aiming to optimize the active learning with semi-supervised feature extraction, [15] proposed to tackle the high-dimensional features' problems by selecting the most representative samples in the low-dimensional space. Their study conduct sample selection and feature extraction recursively at each iteration of the process to learn more accurate models. Another effort to tackle the difficulties of data collection in activity recognition pipelines is [16] that suggests an activity recognition model using active learning. On the other hand, the work in [17] assumes that the attacker can access a subset of sensors in a white box set and maliciously manipulate the controller's commands to the actuators. In this technical note, the effect of false data injection actuator attacks was reported [18] in the face of the adversarial sensor and actuator attacks that are time-varying and partial asymptotic stability when the sensor and actuator attacks are time-invariant. However, all previous studies did not consider a typical UFP deployment architecture with adequate in-depth analysis, as this study proposes for the streamed sensor. Also, it is imperative to highlight that most literature assumes that attacker's knowledge precedes the attacks [19], however, our assumptions for the adversary model based on the UFP-TM [6] had Dolev-Yao model as a baseline, which in numerous situations assumes the presence of adversarial defences. Other research on adversarial detection and security mechanism have been realised by [20] where blockchain has been utilised to create computationally infeasible blocks to prevent contaminating data during incremental training.

TABLE 1. UFP threat model assumptions [6].

Security Violation	Assumptions
Targeted attack or unintentional	False labels to the DIVS could interfere with the online learning process by giving outputs other than the originally intended.
Authentication	UFP between on the DIVS service may be an insecure transmission mechanism through which an adversary is able to have full or partial control which may make him able to modify or tamper.
Denial of Service (DoS)	An adversary can deny service to the DIVS communication channel which may interfere with the UFP.
Targeted and masquerade attacks	An adversary can attack and capture it which eventually may break the entire communication channel of the UFP.
Privacy and confidentiality	Obtain sensitive data in a malicious way that could violate data privacy.
Targeted attacks	An adversary could use the sensor instances of the DIVS as instruments of launching sensor-based or other malicious attacks.

III. ADVERSARIAL DETECTION APPROACHES

This section gives approaches towards realizing adversarial detection approaches. It mainly encompasses various fundamental techniques that can be used to detect adversarial attack techniques during active ML process.

A. PROBLEM STATEMENT

Security during activity recognition in smart environments is still a concern and this concern is incredibly genuine due to the activities conducted by users, data, nature of devices, their interoperability, and administration. The core problem that is explored in the context of this paper is inclined on realizing attacks exhibited during active ML, for example, in the case of the UFP, a subprocess of the DIVS [7], [9]. It is possible (during active learning) that the accuracy of the DIVS could deliberately be influenced or induced by way of providing false labels by an adversary during the UFP in order to tamper with the input, training sets, and output. A significant disadvantage or challenge that may lead to a fall in the accuracy or deterioration of the learning model, in this context, is if a powerful targeted attack against the ML model succeeds. Based on these shortcomings, it is imperative to identify, profile adversaries at the same time limit adversarial motives to ensure that in continued learning, the output generated by the DIVS is reasonably accurate.

During the UFP, attacks can be initiated by an adversary as targeted attacks or unintentionally giving a wrong set of labels. In an intentional or targeted attack, the attacker may use an existing service deliberately to target the learning system by abusing or subverting the ML model's expected output. An unintentional attack may be a wrong label given without knowledge, wrong input, bug, or failure that may be witnessed either in the software's sensors running the model or the physical hardware. Attackers can quickly launch an attack on the learning system using diverse approaches to compromise the learning system to give false outputs. Also, to detect adversarial patterns during active learning, it is paramount to understand the adversaries' techniques and motives during the adversarial attack. In this context, an adversarial attack could either be exploitative or manipulative attacks, and detecting this kind of attack generally requires one to understand the different stages of compromise, especially during the UFP, before the system can fully be considered to be compromised or before a potential attack can be detected. It is worth noting that this needs identification of behaviors of the learning system, how inputs and labels are given to ascertain if an adversary's activity at any given time had an impact on the target system. While it is essential to mention that adversarial attacks in the UFP are regarded as targeted or unintentional attacks, it is also vital to understand the stages that an adversary can use to attack the system.

The authors have detailed several techniques that predominantly are aligned to the UFP-threat model and the identified techniques mainly considers the following; the stages of UFP compromise, Generic Induced Attacks (GIA), mapping GIA with UFP threat model [6], constitutes of GIA and mapping GIA to the UFP-Threat model, and general modeling of attacks in the UFP during active ML. Each of these techniques is explained further on.

B. PROBLEM FORMULATION

Based on the problem statement that has been presented on the need for security techniques during activity recognition, we present a problem formulation that is centered on adversarial detection during an active learning strategy. We then show how adversarial attacks are modeled in a UFP approach, by basing the study on the UFP-TM (Table 1). It is important to note that, to the best of the author's knowledge the concept of adversarial attack detection in active learning strategy in the UFP holds entrancing novelty that is worth exploring and as a result, the formulation of this problem is based on the following generic preliminaries:

- We model adversarial attacks based on the GIA constitutes that emanates from the UFP-TM, as unusual attack propagation, At, Targeted Attacks, TGA, Targeted Behavior, TGB Learning System degradation, LSD, malicious intention, MAL_{int}, Malicious injection, MAL_{inj}, Adversarial Obstruction, Adv_{ob} and Integrity Attacks, $Intg_{At}$.
- Additionally, we define an adversary driven attack representation based on 4-tuples $< \beta, \delta, \phi, \varphi >$ to represent the security goals (CIA) based on the objective of this research. We also denote the UFP-TM as a 5 tuple

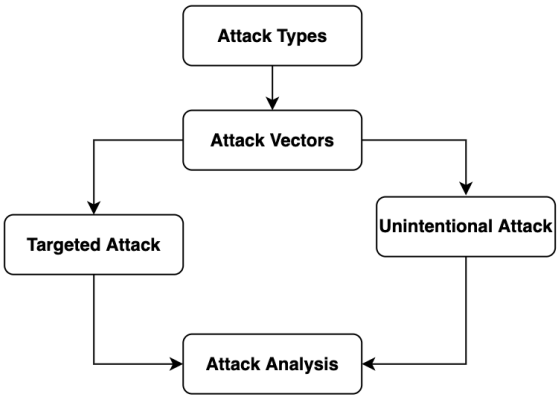


FIGURE 2. Taxonomy of UFP attacks.

TABLE 2. List of notations.

Notations	Descriptions
At_n	n^{th} sensor instance attack
LSD	Learning system degradation
flb	False labeling in streamed sensor data
GKC	Gate keeper capture
TGA	Targeted attacks (intentional)
TGB	Targeted behavior
MAL_{int}	Malicious intention
WO	Wrong output
MAL_{inj}	Malicious injection
Adv_{ob}	obstructing an adversary
DoS	Denial of Service
TMP	Tampering/Modification
$Intg_{At}$	Integrity attacks
$R(E)$	Learning environment representation
TM	Threat model
$TM.R(E)$	Threat model representation in $R(E)$
AS	Attack severity

$< TM, \beta, \delta, \phi, \varphi >$ entity that represents an existence of threat in a learning environment, E. This is based on the existence of *activity*, *object* and *output* respectively.

- Finally, based on the presence of an *activity*, in a learning environment, E, we then use assumptions of known threats to illustrate the effects that the UFP-TM may hold and how adversarial attacks are able to lead to vulnerabilities.

C. STAGES OF UFP COMPROMISE: ADVERSARIAL VIEW

The stages for UFP compromise illustrate attack types that an adversary launches, and then it sets a precedence that can be explored based on the novel attacks mentioned in novel CAPEC/MITRE ATT&CK matrices. The stages of potential UFP compromise are shown as a taxonomy in Figure 2 that highlights the intent that an adversary may have to target the learning system that is launched over the attack vectors.

Attack types represent the dimensions used by an adversary to achieve his adversarial goals or arrive at his destination. In this context, the attack vector represents approaches or vulnerabilities that an adversary uses to gain access to the learning system. Notably, our study identifies misconfigurations, gatekeeper takeover/control, or lack of sufficient authentication as a possible adversary vector of entry. We classify these attacks as targeted (intentional) or unintentional attacks. This classification is necessary to show how compromise may be reached by an adversary. This only happens when a targeted adversary deliberately compromises the security properties of the learning system. A possible targeted attack may be directed to the learning system, applications linked to the system, the network, sensors, physical tampering, ML model, training and test data, or the data being relayed, while unintentional attacks come in many dimensions.

To defeat the role of detection, an adversary needs to defeat the capability of the learning system. Doing so allows the system to do a self-learning to detect or learn the various malicious actions and anomalies. Using the novel CAPEC/MITRE ATT&CK matrices it is important to map and identify key or relevant attacks that affect the learning system that is labeled as Generic Induced Attacks (GIA) as baselines, which are discussed next.

D. GENERIC INDUCED ATTACKS

We explore the common attacks that can be propagated within a connected environment based on novel CAPEC/MITRE ATT&CK matrices and the behavior of those attacks. Also, we take a step further to explore attacks that quickly prevail during continued ML approaches based on these attacks' presence during continued learning. This has been presented as Generic Induced Attacks (GIA), a term coined to depict different attack behaviors exhibited by an attacker while the system is learning. Contrary to how most attacks are propagated on the threat and vulnerability landscapes, we take the attacker's skills into account as a major contributing factor. Therefore, we can generate the GIA based on the classifications that have been highlighted by the novel CAPEC/MITRE ATT&CK matrices.

In the context of the GIA, the authors are motivated to use novel CAPEC/MITRE ATT&CK matrices, as these approaches explicitly give a hierarchy of attack features that are used when a vulnerable point is being exploited. By trying to separate the adversarial activities to identify GIA for easy assessment purposes, we have identified the GIA shown in Table 3 based on the general recommendations by novel CAPEC/MITRE ATT&CK matrices on basic attacks.

Consequently, we explore the GIA from the context of the collected raw sensor data that may contain unique identities, which, if falsified may make the labels implausible-ultimately this may lead to a security violation of tampering or manipulation. In this context, an adversary can directly mislead the learning model through incorrect input/output information during interactive machine learning.

E. CONSTITUTES OF GIA AND UFP THREAT MODEL

The GIA representation in a typical attack pattern is given in this section based on the mapping shown in Table 3. While it is important to note that some of the attacks are advertently actualized as a targeted attack, we also note that some may

TABLE 3. GIA descriptions.

GIA	GIA Behavior	Description
1	Unusual attack propagation techniques	For an attack to be successful, the attacker may launch multiple attacks using different techniques and based on this the system is bound to exhibit different characteristics emanating from the attack.
2	Degradation in system performance	The performance of the system will be degraded based on the number of attacks and the intention of the attackers.
3	Repetitive Attack	The aspect of intentional attack denotes targeted attacks, however, in this instance the attacker may deliberately launch attacks continuously until it is successful.
4	System Manipulation	The system may maliciously be tailored to give wrong output and this interferes with the accuracy or the prediction of the ML model.
5	Malicious logic insertion	Mostly, an adversary may advertently add malicious logic that can deliberately be hidden from users in order for the system to achieve negative impacts aimed at obscurity.

be unintentional. We rely on the fact that the sole aim of an adversary is to challenge the system's security, and as a result, we provide insights that show how the vulnerable threat landscape is [6]. Based on this shortcoming, an attacker may easily invalidate most or all attack paths of the learning system. Ultimately, while the severity of this adversarial action is a general point of concern, care is given in this context on identifying and estimating the likelihood of orchestrated targeted attacks and their impact during active learning. The descriptions of the notations that have been used in the paper are provided as a summary based on Table 2. The discussion on the constitutes as highlighted in Table 3 is given based on the bullets that follow:

- Unlimited unusual attack propagation techniques can come in form of sets, that may compromise the learning system which shows that based on the assumptions from the UFP-TM, it may lead to, for example, a number of sensor instance attacks, At, which is represented as follows:

$$At = \{At_1, At_2 \ldots At_n\}; \tag{1}$$

- A learning system may suffer degradation based on the number of active attacks that are channeled either intentionally or unintentionally through Learning System Degradation (LSD). While LSD could target both hardware and software, the focus of our study mainly targets the learning model. In this context the target is channeled to the act falsifying labeling, *flb* or capturing the gatekeeper node, *GKC* which is represented as is shown in Equation 2:

$$LSD \rightarrow \{At_{no} \rightarrow flb | GKC\} \tag{2}$$

- A Targeted Attack (TGA) can be detected based on the behavior exhibited when the attack is directed to the learning system. We present this as a Targeted Attack Behavior (TAB) that is aimed at giving false labels, *flb*, and this stems from repetitive behavior, which in this context regarded as intentional (targeted) attack. These are represented as follows:

$$TGA \rightarrow \{TAB(At_{no}) \rightarrow \{flb\}\} \tag{3}$$

- Constant learning system manipulation is based on the presence or influence of an adversary which eventually tailors the system in a way that, it is able to give wrong output, or attack training sets through service disruption or alteration of the learning system. This is based on the malicious intention, MAL_{int}, by an adversary, that deliberately manipulates the learning system to give Wrong Output, *WO*, which is denoted as follows:

$$MAL_{int} \rightarrow \{flb \rightarrow WO\} \tag{4}$$

- Deliberate malware injection, MAL_{inj} during the UFP is attributed to the malicious intention, MAL_{int}, of an adversary which is advertently propagated in order to create a negative impact to the system and this is represented as follows:

$$INJ \rightarrow \{Adv \rightarrow D_{inj}\} \tag{5}$$

- Adversarial obstruction, Adv_{ob}, allows limiting of services offered by the learning system and this is achieved by disrupting the learning system, though, for example, Denial of Service (DoS), in this context, the system's resources that queries the human agent/oracle/user for the feedback, during the UFP may have its resources depleted in a manner that allows the system to give an adversary the desired output, which is represented as follows:

$$Adv_{ob} \rightarrow \{DoS\} \tag{6}$$

- Information modification and tampering, *TMP*, is channeled by integrity attacks and this is advertently propagated based on direct attack to the data, in form of manipulation which is represented as shown next

$$Intg_{At} \rightarrow \{TMP\} \tag{7}$$

By carefully analyzing the aforementioned GIA descriptions, we are able to come up with possible attack approaches, and in order to refine it further for purposes of detection and profiling approaches, the author maps the GIA, whose selection has been based on the prevalence of the learning system, i.e. DIVS, and the novel CAPEC/MITRE ATT&CK matrices on basic attacks coupled with the potential UFP threat Model [6].

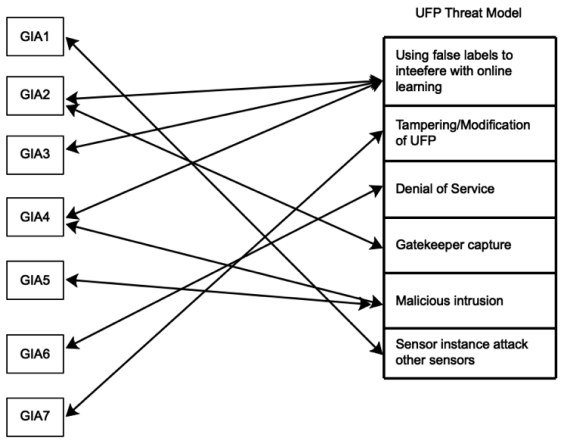


FIGURE 3. Mapping GIA to UFP threat model.

F. MAPPING GIA WITH UFP THREAT MODEL

The UFP threat model has been described as a culmination of possibilities experienced due to the execution of the DIVS service [5] to and from the oracle/user (presented in Figure 1). Additionally, we managed to put an argument that an adversary will be more interested in attacking the UFP, which in the long run, may lead to inaccurate predictions. In return, this calls the need for detecting potential attacks. Consequently, the UFP threat model has also created several assumptions that may be directed to DIVS. We map each of the GIA against assumptions from the threat model, as is shown in Figure 3.

From the mapping of (GIA and UFP assumptions), it is important also to note that the descriptions have been used to show adversarial motives and approaches that can be used to conduct profiling. GIA1, for example has been mapped to the sensor instance attack of the UFP threat model, while the use of false labels from the UFP threat model has been mapped to GIA2, GIA3, and GIA4, respectively. Next, tampering and modification have been mapped to GIA7, while DoS is mapped to GIA6. This process is followed by the malicious intrusion that is mapped with GIA4 and GIA5 respectively.

IV. MODELING ATTACKS IN UFP PROCESS

Herein, a scheme that is used to align the UFP with possible attacks is presented. Doing so is motivated by the need for identifying weaknesses that conceptually may result from the user's activity, which are regarded to be targeted or unintentional attacks in this context. We model this by providing the fundamental representation of the attacks based on the UFP threat model's assumptions.

A. ATTACK REPRESENTATION

Based on the UFP threat model, we suggest that an adversary is able to be detected based on the presence of the following aspects; *activity*, *object* and *output* that are generated from the learning environment. In this context, output has been represented as a potential output from an oracle that has

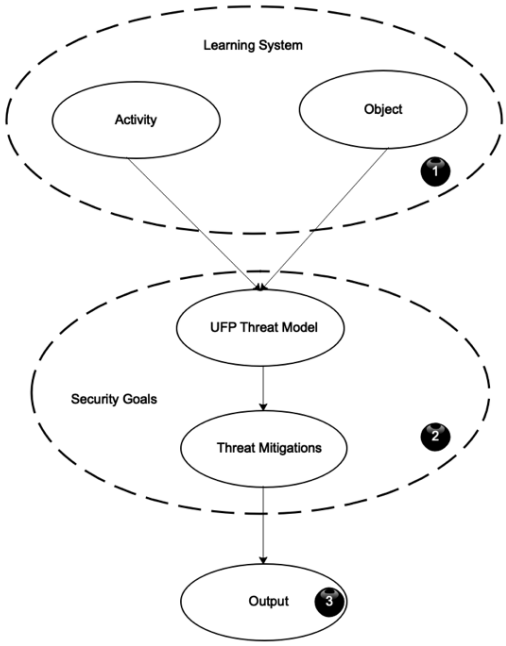


FIGURE 4. Threat-driven technique for UFP.

unique identities that could be subject to deliberate manipulation. Based on this premise, our choice of *activity*, *object* and *output* is necessitated by the fact that these three aspects give continuous interaction, as long as there is an *activity* and *object* which may lead to a given *output*. Furthermore, this is represented as a threat-driven technique that is shown in Figure 4. In normal cases, there would be no output without an input, however, it is worth noting that in this context, output is not categorically illustrated to have emanated from correct or wrong input.

The threat-driven technique shown in Figure 4, categorically relies on the *activity*, *object* and *output* parameters as core aspects in the case of the UFP. Also, this technique shows the circumstances under which adversarial activities may prevail. This approach is divided into three parts namely learning system labelled (1), security goals, labelled (2) and output that is labelled (3). Based on the threat model that has been illustrated throughout this paper, the threat mitigation techniques should identify the measures that can wrongly influence the output (3), which if not identified and enforced may have a negative effect to the accuracy of the learning system (1). It is worth noting, that, the failure to achieve the security goals of Confidentiality, Integrity and Authenticity (CIA), leads to existence of vulnerabilities that are easily exploited by adversaries.

As a result, we give adversarial and threat centered definitions which helps the reader to comprehend the threat formation and path and the need of preserving the security system goals.

Definition 1 (Adversary-Driven Attack Model): is composed of 4-tuples; $<\beta, \delta, \phi, \varphi>$. These tuples translates and fits into the entire security goals that should be achieved

during the UFP, including threat mitigation strategies, where $<\beta, \delta, \phi>$ are also actions of the CIA security goals, and ϕ describes the threat mitigation approaches:

In this context, the concentration is on the security goals that the learning system can achieve based on the *activity*, *object* and *output* respectively. Basically, this definition is dependent on all the possible adversarial attacks, because, in essence the goals of the security model in such a context is to achieve the 4-tuple, $<\beta, \delta, \phi, \varphi>$ i.e. CIA together with the mitigation strategies respectively. Threat in this definition has been presented as any anomaly that is bound to have or exercise a negative impact to any of the CIA security goals, i.e., the actions that an adversary may use in order to exploit the system. As a consequence, based on the aspect of the learning system, we identify key areas of interest in the UFP. The key interest is on the targeted (intentional) attack that aims to deliberately manipulate the output of the system through the GIA constitutes or other anomalies and eventually this is regarded as a complex threat. As a result, we provide a definition for the UFP threat model next.

Definition 2 (UFP-Threat Model): is comprised of five tuples $<TM, \beta, \delta, \phi, \varphi>$, such that for any TM in $<\beta, \delta, \phi, \varphi>$ then a threat is said to exist in the security system:

Within the security system there exist objects and actions that rely on those objects. For example, in this case, we represent *activity* as actions by the *objects*, in a smart environment while we represent *output* as a result from a set of activities. This implies that, to generate actions that represent the threat model, there need to be at least a single action that may have some negative impacts on the 4-tuple, $<\beta, \delta, \phi, \varphi>$. The violation of these security goals is associated with attacks or anomalies.

Based on (*Def* 1 and *Def* 2), we provide simple conceptual formalisms that are centered on threats, vulnerabilities and mitigation strategies and the presence of an adversary in the learning environment. These formalisms still considers the *activity*, *object* and *output* from a learning environment and the notations that have been adjusted to fit the descriptions that was previously highlighted in Table 2. The *activity* denotes the actions by the *object* from a learning environment (could be an ambient or smart environment) that specifically represent the system. Also, *object* denotes the physical entity that performs some kind of activity from which the sensors are able to detect. While in most cases entities are attributed to be physical, we hold the same assumption because the aspect of having a user in between is associated to be human agent/user or an oracle based on the suggestions of the DIVS that is shown in Figure 1 of this article. We take importance in understanding the learning environment and continued interactions, which forms the basis of the actions that are generated from the objects. This is owing to the fact that it is the learning environment that ends up being susceptible to threats that may influence the output.

We take the learning environment as E, and to prevent an adversary actions the 4-tuple, $<\beta + \delta + \phi + \varphi>$ which is part of the security goals and mitigation strategies should be achieved. We also take the representation of the learning environment as $R(E)$, which consist of *activity* and *object* such that,

$$R(E) \equiv \{< activity >, < object >\} \tag{8}$$

We assume that the presence of an *activity* in E takes a 1 or a 0 otherwise assuming that all the factors are dependent of the availability of an *object*. We let $TM.R(E)$ to be a representation of the learning environment where a TM exist in $R(E)$ and the assumption is that a threat may exist that may lead to targeted attacks. In this context, we represent $TM.R(E)_{As}$ as a set of known threats or some form of assumptions based on the availability of *activities* in E, which is expressed as follows:

$$TM.R(E) \equiv \{TM.R(E)_{As} \| AS \in R(E)\} \tag{9}$$

This implies that the effect of TM in $R(E)$ gives room to an adversarial attack if the threats are able to lead to vulnerabilities. By satisfying the 4-tuple, $< \beta + \delta + \phi + \varphi >$ we argue that adversarial attacks may be prevented, an illustration is given below.

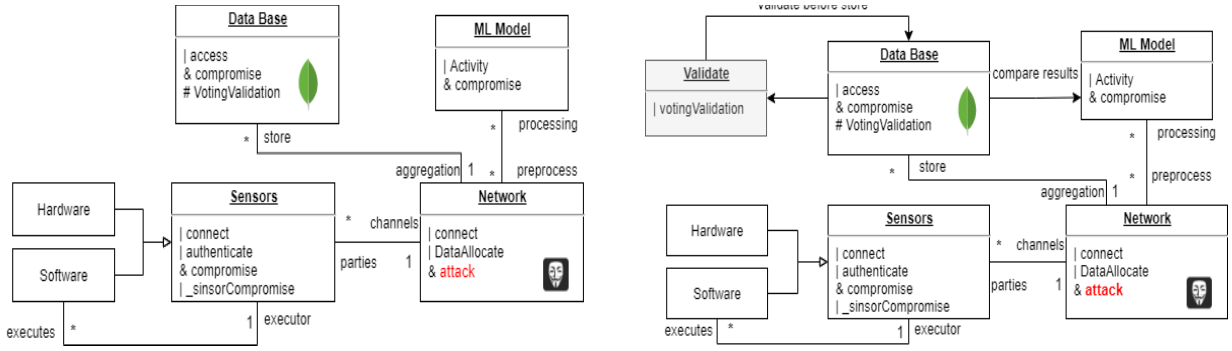
$$R(E) = \begin{Bmatrix} 1 \ TM.R(E)_{As} =< \beta + \delta + \phi + \varphi > \\ 0 \ TM.R(E)_{As} \neq< \beta + \delta + \phi + \varphi > \end{Bmatrix} \tag{10}$$

This conceptualization leads to the identification of an interaction among entities as is shown in Figure 4. In this conceptualization, we note the interaction between different entities that can influence the outcome of a learning system. This interaction is based on the sensors sensing activities that are generated by objects. Also, the threat model's assumptions help in identifying the threatening threats that can help to create a threat profile that can help in adversary identification and profiling.

B. THREAT MODELLING TECHNIQUE

In this section, we introduce a specific cyber-threat modeling as an attack simulation of the proposed adversary action components in a data streaming architecture using the Meta Attack Language (MAL) platform [21], [22]. MAL is a domain-specific language for probabilistic threat modeling to assess the cybersecurity of a system [23]. Figure 5 illustrates two scenarios using UML diagrams of the system during continued learning with and without the voting where in some instances the classified data rests in the database. Hence, we formalize the automated generation of attack graphs that can be utilized to improve the overall system security.

Below, we present a MAL specification related to the adversarial data streaming detection that composed of the following assets: (i) a set of sensors at the edge of the network, (ii) network, a representative entity of the data pipeline and data allocation, (iii) DataBase that represents a NoSQL MongoDB system, and the ML model.



(a) No validation

(b) With validation

FIGURE 5. A comparison UML diagram of the secure and non-secure UFP architecture with the validation extension.

```
abstract Asset Sensor {
  | connect
    -> compromise
  | authenticate
    -> compromise
  & compromise
    -> _sensorCompromise
  | _sensorCompromise
    -> executees.compromise,
      network.dataAllocate
}
Asset Hardware extends Machine {
}
Asset Software extends Machine {
  & compromise
    -> _sensorCompromise,
      executors.connect
}
Asset network {
  | connect
  | DataAllocate
  & attack
    -> aggregation.access}
Asset DataBase {
  | access
   -> compromiseUnencrypted,
  & compromiseUnencrypted
    -> compromise}
Asset ML_Model{
  | activity
   -> preProcess. dataAllocate}
associations {
Sensor [executor] 1 <-- Execution --> * [executees
    ] Software
Sensor [parties] * <-- Communication --> 1 [
    channels] Network
Network [store] * <-- Storage --> * [aggregation]
    DataBase
}
```

The collected data from the sensors at the edge of the network have two paths. The first is heading from the network to the ML model to be processed as an activity. Second, they are forwarded to the database (MongoDB in our deployment architecture) as storage capacity. Considering the attack steps, the attacker intercepts the aggregated data in the data allocation stage before reaching the ML model processing. This step is represented in the network class and highlighted in red. At the sensors class, compromise represents that the attacker has gained control over the network. To reach a compromise, both connect and authenticate must first be reached. Connect represents the attacker's establishment of contact with the system.

If a compromise is reached, then all Software that is executed by the compromised sensors (data stream) also becomes compromised. Furthermore, the data sets stored in the database become accessible. Finally, any connected network becomes accessible for data allocation and modification by the attacker. As a validation measurement, we propose the voting validation in Figure 5 (b) before proceeding to the database. This validation step can be implemented as a defense step of an attacker reading and manipulating the data. Therefore, even if an attacker has access to data, it cannot be fully compromised as it has to be validated first. This defense (represented by # in Figure 5) assume boolean values to indicate their status. If the voting validation is false, then, at the time of instantiation, the dataset is marked as compromised.

V. EXPERIMENTS

A. EXPERIMENTAL SETTINGS

In this section, we investigate the performance of ML classifiers in the wake of targeted/manipulate attacks in continued learning.

1) DATASET

To evaluate and validate the proposed concept, we have conducted our experiments using CASA dataset [1] that represent the actual daily activity of the volunteers living in these homes. Each dataset sample corresponds to specific activity which has been captured, that is composed of 36 features (as is shown in Table 4), which is linked to different sensors (E.g PIR, door, temperature, and light switch sensors) that are distributed over 30 different apartments. These sensors are installed in a location throughout the apartments to observe specific daily activities performed for example by the residents. In total, we have around 42 different activities from all volunteers such as reading, working, eating, sleeping, leaving the home,.etc. Moreover, CASA dataset

[1] https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+from+Continuous+Ambient+Sensor+Data

contains 13956534 samples, and have been collected from the selected apartments continuously in real-time while the residents undertake their day-to-day duties during two month time period [24]. In addition, CASA dataset has sensitive data that can be used to monitor the elderly people's health situation at their place. This makes this kind of data significant enough to want to test the adverse effect of manipulations, this explains its choice for this experiment. Moreover, the data patterns have diverse features that could likely be susceptible to attacks.

2) EXPERIMENTAL APPROACHES

We have made assumptions in our experiments that an adversarial attack can occur based on a number of threat levels. Based on that, the experiments have been conducted as a way of providing the proof of the concept given the following threat levels:

- Train a classifier without the attacker knowledge and without any defense technique and generate a baseline classifier
- Attack the input data by tampering with the labels (Targeted actions, malevolent labeler or uncoordinated attack)
- Apply interactive learning (UFP) and assess the performance of the classifiers
- Assess the Attack severity while leveraging interactive learning strategy, which based on the prevalence of unique identities may enable swift generation of defense and detection mechanism.

The CASA datset has been used to deploy our experiments that are focused in addressing adversarial attacks by training nine classifiers as follows: ExtraTrees, Random Forests, Bagging, Decision Trees, K-Nearest Neigbor (KNN), Light gradient boosting, Multi-layer perceptron, and Support Vector Machine (SVM) using the algorithm shown next.

B. RESULTS

Our approach has been mainly to keep track of the targeted attacks, specifically integrity attack on the classifiers and as a result we have utilised the Attack Severity Metric(ASM) to show instances where attacks prevail and the magnitude of these attacks using Eq 11.

$$AS = 1 - \frac{Recall \quad (after \quad the \quad attack)}{Recall \quad (before \quad the \quad attack)} \tag{11}$$

Tables 6 and 11 has shown the baseline performance of the classifiers when they are trained without any attack and when an attack is directed to the input data respectively. From these two tables, it is monitored that the accuracy of the classifiers deteriorates drastically once an attack is directed to the input data with a margin of 19 % (Precision). However, the precision, Recall, F1-score and Kappa indicators have been used to measure the accuracy of the classifiers before and after the attack on the labels. Given that this experiment is aimed at providing proof of the concept on the influence of targeted attacks to the learning model, the concentration has

Algorithm 1 Mapping GIA With UFP Threat Model

Input: $x_1, x_2, \ldots x_n, x_i \in X$
Output: $y_1, y_2, \ldots y_n, y_i \in Y$
dataAggregation(X, N); $X' =$ getData(N)
ML_Model=trainingML(Input X, Output Y)

Function `InteractiveLearningProcess` (*SelectedSamples X*) **:**
  $Y' =$ ML_Model.predict(X)
  if X' *in all N db are equals* **then**
    activeLearningProcess(X', Y')
  else
    dataAttacked(X', Y')
  end if
  New_X= Concat(X, X'); New_Y= Concat(Y, Y')
  New model: ML.retrain(New_X,New_Y)
  ML_Model=trainingML(New_X,New_Y)
  if *Acc(New model) > Acc(Old model)* **then**
    Old model = New model
  else
    Old model = Old model
  end if
Function `dataAggregation` (*dataSample X, replicationNumber N*) **:**
  Store x $\rightarrow DB_N, N \geq 3$
  return True
Function `trainingML` (*Input X, Output Y*) **:**
  TrainedModel $\leftarrow$ ML.train(X,Y)
  return TrainedModel
Function `getData` (*NumberOfSamples N*) **:**
  $X' \leftarrow$ Select X_N from all Dbs
  return X'
Function `activeLearningProcess` (*Input X, Prediction Y*) **:**
  foreach X **do**
    $y_i' \leftarrow (x_i \rightarrow Oracle)$
    if $y_i == y_i'$ **then**
      continue
    else
      New labeled data y_i': update $y_i \forall x_i$ for $\forall$ Dbs
    end if
  end foreach
Function `dataAttacked` (*Input X, Prediction Y*) **:**
  (Match, Unmatch) $\leftarrow$ CountMatchedSamples(X)
  if *Match > Unmatch* **then**
    Select matched x_i
    activeLearningProcess(x_i, y_i)
  else
    discard unmatched x_i samples
    Continue
  end if

mainly been on manipulative/integrity attack. The effect of attack on each of the aforementioned classifier is shown in Table 8, where the Attack Severity (AS) has been computed using Recall indicator. To test the effect of the attack with interactive learning, we employed UFP and as is projected in Table 8 and Table 9 respectively, there is an improvement of the accuracy on the classifiers. We have found out that when interactive learning is applied across the classifiers, there is significant improvement on the classifiers performance. Notably, the experimental verification that is shown in Table 9 shows that to initiate defense techniques, one need to harden incremental training techniques.

In analyzing the potentially existing vulnerabilities in the context of the experiment that has been conducted in this

TABLE 4. CASA dataset features characteristics.

Index	Features	Types	Index	Features	Types	Index	Features	Types
1	lastSensorEventHours	Discrete	14	areaTransitions	Discrete	27	sensorCount-Ignore	Continuous
2	lastSensorEventSeconds	Continuous	15	numDistinctSensors	Discrete	28	sensorCount-Kitchen	Continuous
3	lastSensorDayOfWeek	Discrete	16	sensorCount-Bathroom	Continuous	29	sensorCount-LivingRoom	Continuous
4	windowDuration	Continuous	17	sensorCount-Bedroom	Continuous	30	sensorCount-Office	Continuous
5	timeSinceLastSensorEvent	Continuous	18	sensorCount-Chair	Continuous	31	sensorCount-OutsideDoor	Continuous
6	prevDominantSensor1	Discrete	19	sensorCount-DiningRoom	Continuous	32	sensorCount-WorkArea	Continuous
7	prevDominantSensor2	Discrete	20	sensorElTime-Ignore	Continuous	33	sensorElTime-Bathroom	Continuous
8	lastSensorID	Discrete	21	sensorCount-Hall	Continuous	34	sensorElTime-Bedroom	Continuous
9	lastSensorLocation	Discrete	22	sensorElTime-Kitchen	Continuous	35	sensorElTime-Chair	Continuous
10	lastMotionLocation	Discrete	23	sensorElTime-LivingRoom	Continuous	36	sensorElTime-DiningRoom	Continuous
11	complexity	Continuous	24	sensorElTime-Office	Continuous	37	activity	Text (class label)
12	activityChange	Continuous	25	sensorElTime-OutsideDoor	Continuous			
13	sensorElTime-WorkArea	Continuous	26	sensorElTime-Hall	Continuous			

TABLE 5. Baseline performance of the classifiers without any attacks.

ML algorithms	Precision	Recall	F1-score	Kappa
ExtraTrees	0.9479	0.9478	0.9475	92.802%
Random forests	0.9405	0.9395	0.9389	91.595%
Bagging	0.9382	0.9376	0.9370	91.348%
Decision Tree	0.8971	0.8969	0.8969	85.922%
K-Nearest Neighbour	0.8658	0.8698	0.8656	82.074%
Light gradient boosting	0.8437	0.8378	0.8384	77.752%
Multi-Layer perceptron	0.7841	0.7920	0.7836	70.776%
Gradient boosting	0.7716	0.7641	0.7536	66.174%
SVM	0.7547	0.7464	0.7199	62.138%

TABLE 6. Classifiers performance after attacking the input data.

ML algorithms	Precision	Recall	F1-score	Kappa
ExtraTrees	0.7558	0.5953	0.5158	25.082%
Random forests	0.7548	0.5932	0.5119	24.522%
Bagging	0.7518	0.5929	0.5117	24.507%
Decision Tree	0.7151	0.5847	0.5030	23.282%
K-Nearest Neighbour	0.6963	0.5791	0.4950	22.170%
Light gradient boosting	0.8345	0.1780	0.2770	12.116%
Multi-Layer perceptron	0.7781	0.1635	0.2611	10.737%
Gradient boosting	0.6536	0.5583	0.4599	17.393%
SVM	0.6550	0.5546	0.4483	15.828%

TABLE 7. Effects of the evasion attack on each classifier.

ML algorithms	Recall (before the attack)	Recall (after the attack)	Attack Severity (AS)
ExtraTrees	0.9478	0.5953	0.3719
Random forests	0.9395	0.5932	0.3686
Bagging	0.9376	0.5929	0.3676
Decision Tree	0.8969	0.5847	0.3480
K-Nearest Neighbour	0.8698	0.5791	0.3342
Light gradient boosting	0.8378	0.1780	0.7875
Multi-Layer perceptron	0.7920	0.1635	0.7935
Gradient boosting	0.7641	0.5583	0.2693
SVM	0.7464	0.5546	0.2569

study, we have also explored and illustrated the potential attacks that have a possibility of violating major security goals (CIA), however, the main inclination of the study is towards targeted (integrity) attacks assuming the oracle that gives labels has unique identities. Generally, the objective of this attack is to deliberately falsify the contents of the dataset or actions that are being sensed in order to give wrong input/output to the learning model. Consequently, in the perspective of CASA dataset that has been utilised in this experiment, this has been achieved by the deliberate or unintentional falsification of dataset labels, or injection of malicious content. While monitoring the effect of this attack, a baseline performance of nine machine learning classifiers has been portrayed in this experiment in Table 11 based on Precision, Recall, F1-score and Kappa metrics respectively. Based on the outcome of these metrics, the effects of a targeted attack could, for example cause accuracy deterioration of the learning model, particularly during the UFP.

TABLE 8. Classifiers performance after attacking the input data and applying active learning.

ML algorithms	Precision	Recall	F1-score	Kappa
ExtraTrees	0.7946	0.6762	0.6379	44.02%
Random forests	0.7933	0.674	0.6345	43.505%
Bagging	0.7931	0.6761	0.6377	44.025%
Decision Tree	0.774	0.6673	0.6278	42.594%
K-Nearest Neighbour	0.7689	0.6638	0.623	41.899%
Light gradient boosting	0.9162	0.3417	0.4844	26.417%
Multi-Layer perceptron	0.9008	0.3298	0.4757	25.266%
Gradient boosting	0.7505	0.6462	0.5988	38.36%
SVM	0.753	0.643	0.5915	37.287%

TABLE 9. Evaluation of the countermeasure.

ML algorithms	AS before	AS after applying interactive learning	Improvement%
ExtraTrees	0.3719	0.2865	22.96%
Random forests	0.3686	0.2825	23,35%
Bagging	0.3676	0.2789	23.39%
Decision Tree	0.3480	0.2559	26.46%
K-Nearest Neighbour	0.3342	0.2368	29.14%
Light gradient boosting	0.7875	0.5921	24.81%
Multi-Layer perceptron	0.7935	0.5835	26.46%
Gradient Boosting	0.2693	0.1542	42.74%
SVM	0.2569	0.1385	46.08%

Tables (9-11) portrays various outputs that are based on baseline performance of classifiers, performance after attacks, effects of evasion attacks, attacks on input data with active learning and evaluation of the countermeasures respectively. From this experiment we compare the effects of the performance metrics as a result of attacks, for instance, we compare attacks to the input data (see Table 11) which represents the baseline performance of the classifiers and (see Table 6), that shows the classifier performance after the attack. Based on Table 6 and 11, there is a deteriorating performances for ExtraTrees (0.9479 to 0.7558), Recall (0.9478 to 0.5953), F1-Score (0.9474 to 0.5158) and Kappa (92.802 % to 25.082%) respectively. As a result, comparing the ExtraTree classifier, the Kappa metric portrays a deterioration of 67.72% after the initial attack to the input data. We observe that this deterioration has been consistent but with varying margins for Random Forest, Bagging, Decision Trees, K-Nearest Neighbor, Gradient Boosting and Multi-layer Perceptron respectively. Alternatively, Gradient Boosting and SVM classifiers have portrayed unique performances for the precision metric. Consequently, several observations are noted from Table 8 with regard to evasion attack on each classifier, which is based on the Attack Severity (AS) using the Recall metric, before and after the attack. This, has been used to portray instances and magnitudes of attack,

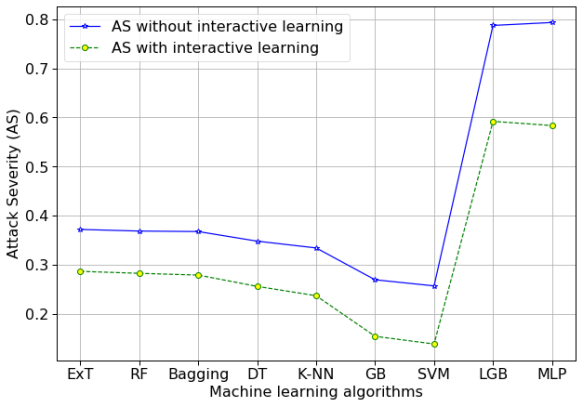


FIGURE 6. Depicting the attack severity (AS) for the machine learning classifiers with and without interactive learning.

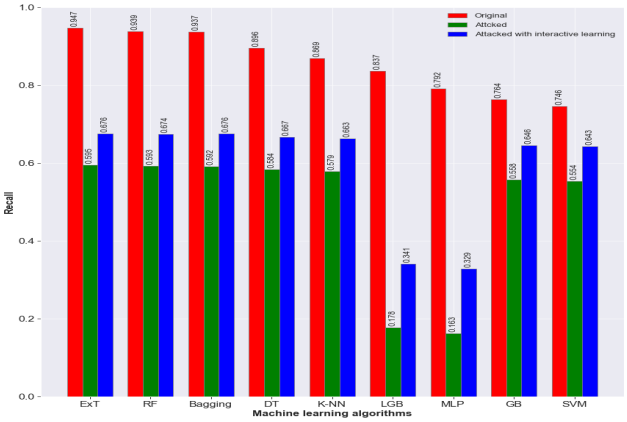


FIGURE 7. Recall value for all machine learning algorithms based on.

which in this context if launched towards the input data. Light gradient boosting and Multi-layer Perceptron portrays a higher attack severity (0.7875 and 0.7935) respectively when compared to other classifiers. Notably, the performance of the ML classifiers have also been assessed after attacking the input data, AS, while active learning strategy is being applied (see Table 9). In this case, Bagging classifier performs well as compared to other classifiers with a Kappa metric of 44.025% improvement, while Multi-layer Perceptron garners the lowest improvement with 25.266%. These evaluations have also been extrapolated as a countermeasure by showing the percentage improvement in Table 9. On the same note, Table 10 which show a replication of the findings from Tables (9-11), precisely shows performances of all the metrics against the classifiers with interactive learning while taking into account the baseline performance. Overall, ExtraTrees and Bagging emerges as the best based on Kappa classifier, while multi-layer perceptron emerges as the lowest performing classifier when active learning is incorporated.

C. EVALUATION MEASURES

We have used AS and active learning strategy to evaluate the behavior of the nine classifiers before the manipulative attack and after the attack. In this context, while the role of active learning is positioned to improve the classification, our approach also considers instances when an oracle is juxtaposed as a malevolent labeler $(x' \in X)'$–> *oracle* in a potential attack. We have evaluated the performance of the nine machine learning classifiers by measuring precision, Recall, F1-score and Kappa respectively with/without interactive learning at the same time with/without adversarial attack respectively. It is worth noting again that attack in the context of this study has been represented in the perspective of alteration/manipulation of labels which on the premise of this paper is portrayed as a targeted attack whose nature can violate integrity of data in a security perspective. Based on the manipulative attack, we are able to assess the classifier's improvements with SVM showing the most improvement as compared to the rest as is shown in Table 9.

Consequently, based on the experimental results, specifically from the baseline performances that are shown in Table 11, it is evident that one ML algorithm, ExtraTrees has outperformed the rest both in Precision, Recall, F1-score and Kappa, although RF and Bagging have a relatively close-matching accuracy. Although the accuracy looks higher prior to the manipulative/tampering attack, there seem to exist slight decrease in the accuracy after the attack, which is an indication that the influence of the correct labels affects the accurate prediction of ML algorithms. Furthermore, it also demonstrates the distinct role that a correct classification can play toward threat detection. Additionally, the modeling of attack steps that previously was highlighted using the Meta Attack Language is a significant step that could play a significant role towards real-time detection of other adversarial attacks in continued learning. In Table 8, we have observed the Attack Severity for the Recall as a result of the evasive attack on each of the classifier while after applying active learning strategy, there also seem to exist variations for precision, Recall, F1-score and Kappa as is shown in Table 9. Attack severity based on the experiments conducted when interactive learning is used and when it is not used has been shown in Figure 6 and the improvement has been factored in Figure 7, whereas a computation of baseline performance, classifier performance after attacking the input, evasion attacks on classifiers and evaluation before and after the attack is shown in Figure 5.

We can arrive at a number of conclusions based on the results exhibited by these experiments: Firstly, our approaches are more suitable for learning models leveraging virtual sensors, which makes it suitable for security violations detection mechanisms, which also could lead to development of defense mechanisms. Also, by utilising the CASA dataset, our evaluations shows that the outcome could still be applied in a real-time detection approaches when dealing with attacks that are confined with unique identities. Nevertheless, the accuracy portrayed by diverse ML algorithms indicates that adversarial attacks can emanate from the training sets, the learning model-in instances where classifiers can be fooled and the physical surrounding, however, this study is more

TABLE 10. Evaluation of the algorithms based on active learning and compared it with the baseline performance.

	Precision			Recall			F1-score			Kappa		
ML algorithms	Original	Attacked	Attacked with interactive learning	Original	Attacked	Attacked with interactive learning	Original	Attacked	Attacked with interactive learning	Original	Attacked	Attacked with interactive learning
ExtraTrees	0.9479	0.7558	0.7946	0.9478	0.5953	0.6762	0.9475	0.5158	0.6379	92.802%	25.082%	44.02%
Random forests	0.9405	0.7548	0.7933	0.9395	0.5932	0.674	0.9389	0.5119	0.6345	91.595%	24.522%	43.505%
Bagging	0.9382	0.7518	0.7931	0.9376	0.5929	0.6761	0.9370	0.5117	0.6377	91.348%	24.507%	44.025%
Decision Tree	0.8971	0.7151	0.774	0.8969	0.5847	0.6673	0.8969	0.5030	0.6278	85.922%	23.282%	42.594%
K-Nearest Neighbour	0.8658	0.6963	0.7689	0.8698	0.5791	0.6638	0.8656	0.4950	0.623	82.074%	22.170%	41.899%
Light gradient boosting	0.8437	0.8345	0.9162	0.8378	0.1780	0.3417	0.8384	0.2770	0.4844	77.752%	12.116%	26.417%
Multi-Layer perceptron	0.7841	0.7781	0.9008	0.7920	0.1635	0.3298	0.7836	0.2611	0.4757	70.776%	10.737%	25.266%
Gradient Boosting	0.7716	0.6536	0.7505	0.7641	0.5583	0.6462	0.7536	0.4599	0.5988	66.174%	17.393%	38.36%
SVM	0.7547	0.6550	0.753	0.7464	0.5546	0.643	0.7199	0.4483	0.5915	62.138%	15.828%	37.287%

focused on the influence/detection of the adversarial attacks have in continued learning.

From the perspective of validation, our approach provides a more effective technique in the context of active learning strategy in the UFP. More so, from the conducted experiments and the generated results, it is worth noting that the study can easily be generalized to fit in smart ecosystems. This, is owing to the fact that, the study employed nine algorithms that allowed the authors to conduct extensive experiments, which have also been used to validate the effects of adversaries and data manipulation/attacks in an ambient environment in continued active learning. These activities in a smart environment that utilizes CASA dataset were intentionally selected for this study so as in the long run its performance and outcome can be evaluated. As a result, the performance of our approach has duly been evaluated given that each of the nine algorithm-from a competition has diverse and varying output with best and worst performing. Furthermore, validation could also be deduced based on the variations on original data and attacked data, where this has been used to check the accuracy based on normal vs attacked data. Furthermore, a comparative study has also been drawn based on relevant studies to consolidate on the choice of the current proposition. Ultimately, the experiments that have been drawn from our study has portrayed that while leveraging active learning strategy, our approach is more effective.

Apart from that, it is important for a user of the machine learning model to be able to identify malicious activities due to how current attacks have been diversified. In order to illustrate the type of attacks that our approach realizes, our study has modeled the assumption that-the attacker or the malevolent labeler can be able to predict/estimate that in continued learning-instances of data are randomly or specifically generated, however, this could be a case when the attacker has prior know-how of the benign or the learning model, given that this knowledge may precede the attacks. As a result, this attacker has a probability of performing a sequence of malicious activities, potentially to the data or to the learning system. In that context, our study has been positioned to be able to detect active learning attacks, that is termed in this context as a 'malevolent labeler'. This has been shown in the variations before and after the attacks on the classifiers shown in Table 10.

VI. COMPARATIVE ANALYSIS

In this section, we give selected pertinent examples that show adversarial threats/attacks, observations and defenses to machine learning while using active adversarial learning (proposed) as a baseline as is shown in Table 11. Firstly, researchers in [1], [25], [26] uses a supervised classification problem with the aim of identifying the need for secure machine learning based on adversarial decision making. These authors are able to identify a number of adversarial attacks; ranging from signature manipulations, allergy attacks, where active agents can easily be used for DoS attacks. Also this has been seen in instances where adversaries are able to build false labels in order to prevent the generation of accurate classifiers in order to propagate a technique through which an adversary can easily obstructs learning through delusive learning. Important to note is that, obstruction is a core attack that has been highlighted as a GIA under MITRE/CAPEC attack matrix in the context of this paper. Defense techniques for these adversarial attacks include keeping the corpus up to data and setting lower thresholds for false positives in new signatures in order to identify bogus traffic. Next, research by [27], [28] articulates SpamBayes learning method that shows how an adversary is able to exploit statistical machine learning vulnerabilities by identifying dictionary attacks where the SpamBayes is rendered useless. Also, an adversary can easily and intentionally prevent victims from receiving email messages, for example, for a competitive bid in order to have an advantage over others. Defense mechanisms adopted for this approach include the RONI defense that is able to test the performance difference with the victim's email. Another adversarial approach is the Valiant's model of machine learning in [27] through which adversarial learning is done from errors. Through this model, a classification error is identified over which an adversary is able to have control over a fraction of the sets that are being trained and learning is able to be achieved in the presence of noise or the errors that are created by an adversary. In this attack, there is need for a canonical algorithm that is able to cope with the presence of these malicious errors. An attack algorithm that is responsible for misclassification is identified as an adversarial attack in a three-dimensional classification based on learning styles in [28], where, these attacks are still considered to have no

TABLE 11. Comparison of different adversarial attacks and defenses.

Reference	Machine learning approaches	Objective	Adversarial attack Approaches	Defense
Proposed	Utilises 9 ML classifiers	Adversarial detection from an active machine learning perspective.	Employs ambient data collected from a smart environment human activity recognition from (Continuous Ambient Sensors Dataset, CASA) with fully labeled connections, where we intentionally subject the Dataset to wrong labels as a targeted/manipulative attack in the UFP if, for example, user-labels are connected to unique identities.	-Based on the UFP-Threat driven model, the actions of the oracle(human agaent)are subjected to real-time/secure monitoring. -Profiling and isolating potential attacks in continued learning
[1] [25] [26]	Supervised classification problem	Identify the need for sure machine learning based on real adversarial decision-making	Signature manipulation generation, allergy attacks where active agents are used for DoS attack. -Adversary builds false labels to prevent generation of accurate classifier. -Identifying delusive adversary that obstructs learning.	-Mitigations can only limit but not stop damages completely. Can be reduced by keeping corpus up to date, since up to date corpus gives way for effect of pool poisoning. -Setting lower threshold for false positives in new signatures, since successful pool poisoning need small amount of bogus traffic .
[27] [28]	SpamBayes learning method	To show how an adversary can exploit statistical machine learning vulnerabilities	-Dictionary attacks where an adversary is able to exploit the SpamBayes by rendering it to be useless. -Adversary focused attacks that prevents the victims from receiving email messages. -An adversary may prevent a victim from receiving messages that have competing bids.	RONI Defense mechanism that tests the performance difference with or without the email. -Dynamic threshold defense which work against the dictionary attacks.
[27]	Valiants model of machine learning	Adversarial learning from errors	Identifies the classification error through which an adversary has control over a fraction β of the sets being trained. -Learning in the presence of malicious errors or noise created by an adversary or emanating from a faulty equipment.	A canonical robust learning algorithm is employed that is able to cope with malicious errors. -Identified defense still theoretical. Generalized defense models against adversarial attacks exist in very early stages still and other defenses are more specific to a particular type of attack
[28]	3D classification based on learning attacks, learning styles and learning depth	-Identifying adversarial attacks on machine learning	Increased risk of adversarial vulnerabilities in machine learning models. -Most misclassification that is achieved by an adversary significantly is dependent with adversarial attack algorithm.	-Identified defense still theoretical. Generalized defense models against adversarial attacks exist in very early stages still and other defenses are more specific to a particular type of attack
[29]	Analytical model giving a lower bound on attackers work function and a taxonomy of attacks on machine learning system.	Identifying if machine learning can be a target of attack by a malicious adversary.	-Causative attacks channeled to alter the training techniques by influencing the training data. -Security violation through integrity attacks where intrusion is regarded as normal and availability attacks that renders the system to be unusable. -Adversarial attack on online learners by way of shaping a change based on how an online learner changes the prediction.	-Considers the statistical technique of information hiding (exploratory attacks) and regularization and randomization for (causative attacks
[30] [31]	Game theoretical model for adversarial learning	Model interactions between an adversary and data miner as an optimization problem using a game model.	-Contextualizes a game of two people for a spammer and a data miner based on an attack and a status quo. -Spammer attacks the classifier through modification of emails or maintain the status quo with four possible outcomes by tracking the adversary's movement.	-Based on the framework no exact defense because the spammer and data miner can reach an equilibrium when they seem to be playing at their best strategy at the same time
[32]	General adversarial attacks in machine learning and deep learning	Identify adversarial security and perturbation attacks in machine learning on training and testing set.	Causative attacks targeting training process, exploratory attack that targets after the sets have been trained, specify or targeted attacks, indiscriminate attacks that targets instances and security violations (integrity attacks, availability attacks and privacy violation attacks).	Literature opened for solutions
[33]	Anomaly-based intrusion system	Identify method that shows the weakness in anomaly-based intrusions.	Adversary escapes intrusion detection by crafting offensive techniques that blind anomaly -based intrusion detector while common attacks are on progress.	Possible to control the manifestation of this attack from the area of clarity to the area of detection blindness
[34]	Malware and spam based on network intrusion detection(Poisoning and Evasion attacks)	Utilises four classifiers (Random Forests, Multi-Layer perceptron and KNN)(Basically intergity violations).	-A set of malicious flows are crafted as benign. -Malicious samples are then injected into the training sets. -Classifiers are able to identify samples that were novel.	Attack severity before and after injecting the samples has shown a variation-as a way of crafting a countermeasure
[35]	Adversarial attack to exploit ML classifiers in Smart Healthcare Systems (SHS)	Deploying adversarial attacks in smart healthcare Systems (SHS).	-Adversary has partial knowledge of the data that is being distributed.ML algorithms are deployed to perform targeted and un-targeted attacks.Able to manipulate medical devices reading-with a focus of altering the status of the patient.Study has shown that this attack degraded the SHS and propagates erroneous approaches during treatment.Uses 5 ML ((HopSkipJump,Fast Gradient Method etc..) to perform malicious attacks.	No specific defenses identified

specific defenses. A particular focus on an analytical model that gives lower bound on attacker's work function by identifying if a machine learning approach can be a target of an attack by a malicious adversary has identified causative attacks that are channeled to do alterations of the training techniques by way of influencing the training data. Through this, a number of security attacks are identified like integrity attacks, intrusion and availability attacks, and attacks on online learners by shaping changes based on how prediction is done [29].

Defenses that have been identified in this context include the statistical techniques of information hiding to counter the causative attacks. Other pertinent research is the game theoretical model for adversarial learning in [30], [31] that considers a model of interaction between an adversary (spammer) and a data miner in an optimization problem. In this learning attack, the spammer is able to attack the classifier by way of modifying emails in order to maintain the status quo in order to have some desired outputs. No specific defenses are identified in this context given that the spammer and data miner

can reach an equilibrium when they seem to be playing at their best strategy at the same time. Other relevant attacks include the general adversarial attacks focused in machine and deep learning [32], whose main focus is to identify adversarial security and perturbation attacks in machine learning on the training and testing sets. In this context, the authors have identified causative attacks that are able to target the training process and exploratory attacks that is able to target the sets after training. The other focus includes security violations (integrity attacks, availability attacks and privacy violations). Lastly, it is an anomaly-based intrusion system in [33] that is able to show the weakness in anomaly-based intrusions. In this approach an adversary is able to escape intrusion detection systems by way of crafting various offensive techniques that can easily blind the anomaly-based intrusion detector while other common attacks are on progress. Defenses for this can of attack can be controlled by manifesting this attack from the area of clarity to the area of detecting anomaly blindness.

Based on the comparative analyses that has been shown in Table 11, it is important to note that, while the selected studies are relevant, there exist limitations on leveraging active learning strategy in the context of adversarial detection in the user-feedback process as is portrayed in this paper. The relevance in these analysis in Table 11 is that, these studies also explicitly outlines important strategies that could be integrated in futuristic adversarial attack detection models

VII. CONCLUSION AND FUTURE WORK

In this paper, we have elucidated diverse approaches that illustrate potential adversarial issues based on the initially suggested DIVS threat model. We have provided subsequent experiments by inclining the experiments on manipulative attacks to provide proof and observe the behavior of the trained classifiers before and after this attack. Results that have been portrayed in this paper have set a precedent for future work, where we will construct a real-time attack detection mechanism in continued learning as a step towards generating defense techniques from an information security standpoint.

The novelty of this work lies in the adversarial and threat detection approaches in continued learning while leveraging active learning. As a result, this work could still be extended in the following directions: Providing threat alerts that can enable cyber-response strategies in continued learning and also threat prediction strategies in continued learning. Since our suspicions are mainly on potential intrusions, we plan to utilize a Honeypot Dataset that has intrusion type attacks.

Future work aims to be able to develop specific attack techniques for active adversarial active learning. Also the authors aims to leverage and integrate honeypot based dataset with known attacks in order to model attack patterns from a real-time attack scenario. Also, we aim to address privacy preserving aspects by extending the study from active to federated learning techniques in the UFP, while suggesting now security mechanisms that can be used to strengthen active learning strategies.

ACKNOWLEDGMENT

The authors would like to thank Dynamic Intelligent Sensor Systems (DISS) project members, the Internet of things and People (IOTAP) Research Center, Malmö University, Sweden, for the support while coming up with this research. They would also like to acknowledges the opinions, findings, and conclusions expressed in this article are purely of the authors.

REFERENCES

[1] X. Liao, L. Ding, and Y. Wang, "Secure machine learning, a brief overview," in *Proc. 5th Int. Conf. Secure Softw. Integr. Rel. Improvement, Companion*, Jun. 2011, pp. 26–29.
[2] A. Lenin, J. Willemson, and D. P. Sari, "Attacker profiling in quantitative security assessment based on attack trees," in *Proc. 19th Nordic Conf. Secure IT (NordSec)*. Tromso, Norway: Springer, Oct. 2014, pp. 199–212.
[3] B. Miller, A. Kantchelian, S. Afroz, R. Bachwani, E. Dauber, L. Huang, M. C. Tschantz, A. D. Joseph, and J. D. Tygar, "Adversarial active learning," in *Proc. Workshop Artif. Intell. Secur. Workshop*, 2014, pp. 3–14.
[4] A. Tegen, P. Davidsson, and J. A. Persson, "The effects of reluctant and fallible users in interactive online machine learning," in *Proc. 4th Int. Workshop Interact. Adapt. Learn. (IAL)*, 2020, pp. 55–71.
[5] R.-C. Mihailescu, J. Persson, P. Davidsson, and U. Eklund, "Towards collaborative sensing using dynamic intelligent virtual sensors," in *Proc. Int. Symp. Intell. Distrib. Comput.* Paris, France: Springer, 2016, pp. 217–226.
[6] R. V. Kebande, J. Bugeja, and A. J. Persson, "Internet of threats introspection in dynamic intelligent virtual sensing," in *Proc. 1st Workshop Cyber-Phys. Social Syst., 9th Int. Conf. Internet Things (IoT), CEUR Workshop*, vol. 2530, A. Longo, M. Fazio, R. Ranjan, and M. Zappatore, eds, Bilbao, Spain, Oct. 2019, pp. 22–29.
[7] A. Tegen, P. Davidsson, R.-C. Mihailescu, and J. Persson, "Collaborative sensing with interactive learning using dynamic intelligent virtual sensors," *Sensors*, vol. 19, no. 3, p. 477, Jan. 2019.
[8] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Machine Learning Proceedings 1994*. Amsterdam, The Netherlands: Elsevier, 1994, pp. 148–156.
[9] A. Tegen, P. Davidsson, and J. A. Persson, "Interactive machine learning for the Internet of Things: A case study on activity detection," in *Proc. 9th Int. Conf. Internet Things*, Oct. 2019, pp. 1–8.
[10] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1648, 2009.
[11] D. Pereira-Santos, R. B. C. Prudêncio, and A. C. P. L. F. de Carvalho, "Empirical investigation of active learning strategies," *Neurocomputing*, vol. 326, pp. 15–27, Jan. 2019.
[12] H. Yu, X. Yang, S. Zheng, and C. Sun, "Active learning from imbalanced data: A solution of online weighted extreme learning machine," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1088–1103, Apr. 2019.
[13] A. Tegen, P. Davidsson, and J. A. Persson, "Activity recognition through interactive machine learning in a dynamic sensor setting," *Pers. Ubiquitous Comput.*, vols. 1–2, pp. 1–14, Jun. 2020.
[14] A. Tegen, "Approaches to interactive online machine learning," M.S. thesis, Dept. Comput. Sci. Media Technol., Malmö Universitet, Malmö, Sweden, 2020.
[15] S. Gu, Y. Jiao, H. Tao, and C. Hou, "Recursive maximum margin active learning," *IEEE Access*, vol. 7, pp. 59933–59943, 2019.
[16] H. M. S. Hossain, M. A. A. H. Khan, and N. Roy, "Active learning enabled activity recognition," *Pervas. Mobile Comput.*, vol. 38, pp. 312–330, Jul. 2017.
[17] M. Yadegar, N. Meskin, and W. M. Haddad, "An output-feedback adaptive control architecture for mitigating actuator attacks in cyber-physical systems," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 6, pp. 943–955, Jun. 2019.
[18] X. Jin, W. M. Haddad, and T. Yucelen, "An adaptive control architecture for mitigating sensor and actuator attacks in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 6058–6064, Nov. 2017.
[19] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," 2017, *arXiv:1702.02284*. [Online]. Available: http://arxiv.org/abs/1702.02284
[20] V. R. Kebande, S. Alawadi, J. Bugeja, J. A. Persson, and C. M. Olsson, "Leveraging federated learning & blockchain to counter adversarial attacks in incremental learning," in *Proc. 10th Int. Conf. Internet Things Companion*, Oct. 2020, pp. 1–5.

[21] *Meta Attack Language: A Domain Specific Language for Probabilistic Threat Modeling and Attack Simulations*, KTH Roy. Inst. Technol., Stockholm, Sweden, 2020. Accessed: Aug. 1, 2020.
[22] *Meta Attack Language Compiler*, KTH Roy. Inst. Technol., Stockholm, Sweden, 2020. Accessed: Aug. 1, 2020.
[23] P. Johnson, R. Lagerström, and M. Ekstedt, "A meta language for threat modeling and attack simulations," in *Proc. 13th Int. Conf. Availability, Rel. Secur.*, Aug. 2018, pp. 1–8.
[24] D. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE Intell. Syst.*, vol. 27, no. 1, pp. 32–38, Jan. 2012.
[25] S. P. Chung and A. K. Mok, "Allergy attack against automatic signature generation," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Hamburg, Germany: Springer, 2006, pp. 61–80.
[26] J. Newsome, B. Karp, and D. Song, "Paragraph: Thwarting signature learning by training maliciously," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Hamburg, Germany: Springer, 2006, pp. 81–105.
[27] M. Kearns and M. Li, "Learning in the presence of malicious errors," *SIAM J. Comput.*, vol. 22, no. 4, pp. 807–837, 1993.
[28] O. Ibitoye, R. Abou-Khamis, A. Matrawy, and M. O. Shafiq, "The threat of adversarial attacks on machine learning in network security—A survey," 2019, *arXiv:1911.02621*. [Online]. Available: http://arxiv.org/abs/1911.02621
[29] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf., Comput. Commun. Secur.*, 2006, pp. 16–25.
[30] W. Liu and S. Chawla, "Mining adversarial patterns via regularized loss minimization," *Mach. Learn.*, vol. 81, no. 1, pp. 69–83, Oct. 2010.
[31] W. Liu and S. Chawla, "A game theoretical model for adversarial learning," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2009, pp. 25–30.
[32] A. Siddiqi, "Adversarial security attacks and perturbations on machine learning and deep learning methods," 2019, *arXiv:1907.07291*. [Online]. Available: http://arxiv.org/abs/1907.07291
[33] K. M. Tan, K. S. Killourhy, and R. A. Maxion, "Undermining an anomaly-based intrusion detection system using common exploits," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Zürich, Switzerland: Springer, 2002, pp. 54–73.
[34] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, "Addressing adversarial attacks against security systems based on machine learning," in *Proc. 11th Int. Conf. Cyber Conflict (CyCon)*, May 2019, pp. 1–18.
[35] A. I. Newaz, N. I. Haque, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "Adversarial attacks to machine learning-based smart healthcare systems," 2020, *arXiv:2010.03671*. [Online]. Available: http://arxiv.org/abs/2010.03671

VICTOR R. KEBANDE received the Ph.D. degree in computer science (information and computer security architectures and digital forensics) from the University of Pretoria, South Africa. He was with the Information and Computer Security Architectures and Digital Forensics (ICSA) and DIgiFORS Research Group, University of Pretoria, and an Active Member of the Internet of Things and People (IOTAP) Center, Department of Computer Science, Malmö University, Malmö, Sweden. He is currently a Postdoctoral Researcher in cyber and information security with the Information System (IS) Research Subject, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden. His main research interests include cyber, information security and digital forensics in the area of the IoT, (the IoT security), digital forensics-incident response, cyber-physical system protection, critical infrastructure protection, cloud computing security, computer systems, distributed system security, threat hunting and modeling, cyber-security risk assessment, blockchain technologies, and privacy preserving techniques. He also serves as an Editorial Board Member for *Forensic Science International (Reports Journal)*.

SADI ALAWADI received the master's degree in soft computing and intelligence system from Granada University, Spain, in 2012, and the Ph.D. degree (Hons.) in computer science/AI from the Research Center of Intelligent Technologies (CiTIUS), University of Santiago de Compostela, Spain, in 2018. During the last two years he worked as a Postdoctoral Researcher with the IOTAP Research Center, Malmö University, Sweden. He is currently working with the Division of Scientific Computing, Department of Information Technology, Uppsala University. His main research interests include IOT systems, IoT middleware, end-user development in the IOT, machine learning, deep learning, federated learning, transfer learning, smart cities and their related systems, big data, real-time analysis, dimensionality reduction, and data visualization context awareness, and blockchain.

FERAS M. AWAYSHEH received the B.Sc. degree in software engineering from Al Balqa' Applied University, Jordan, in 2008, the M.Sc. degree (Hons.) in information, computer, and network security from the New York Institute of Technology (NYIT), New York, NY, USA, in 2010, and the Ph.D. degree in big data and cloud computing from the University of Santiago de Compostela, Santiago de Compostela, Spain, in 2020. He was a Researcher with the Research Center in Intelligent Technologies (CiTIUS), University of Santiago de Compostela. He is also a Visiting Fellow with the University of Edinburgh, Edinburgh, U.K, and Charles Darwin University, Australia. He is currently an Assistant Professor of Big Data with the Data System Group, University of Tartu, Estonia. His main research interests include large-scale distributed systems and big data analytics, particularly developing and running software reliably in production for resource allocation (on-premises and cloud-based clusters) and middlewares for load balancing and security solutions HPC, cloud, the IoT, and big data deployment architectures.

JAN A. PERSSON received the Ph.D. degree in optimization from Linköping University, Sweden, in 2002. He is currently an Associate Professor of Computer Science with Malmö University, Sweden. He is also a Researcher with the Internet of Things and People (IOTAP) Research Center and the K2–The Swedish Knowledge Centre for Public Transport. His research interests include artificial intelligence, optimization and simulation methods for decision support, and the application areas include the Internet of Things (IoT), sensor systems, and transport systems.