Towards Context-aware Intrusion Detection in Individual-oriented Information Systems: An Empirical Study on Android Malware

Van-Tien Nguyen[1,2], Renzo E. Navas[2], Guillaume Doyen[2]
[1]*INSA Toulouse, LAAS-CNRS, UPR 8001, Toulouse, France* [2]*IMT Atlantique, IRISA, UMR CNRS 6074, Rennes, France*
vtnguyen@laas.fr; {van-tien.nguyen, renzo.navas, guillaume.doyen}@imt-atlantique.fr

***Abstract*—In recent years, the range and volume of Internet services utilized by an individual have significantly expanded. This growing relationship between an individual user and diverse digital services has led to the emergence of Individual-oriented Information System (IIS) that encompasses the user, their physical devices, and the information systems they interact with. Current security approaches within an IIS suffer from three main limitations: (1) they are restricted to specific services, (2) they require intrusive instrumentation of each user single device, or (3) they rely on specific integration between client and server components. As a result, they fail to globally protect against attackers who possess enough information to bypass standalone security schemes. To overcome these constraints, we propose to (1) consider a network-oriented approach to detect intrusions which may occur within an IIS and (2) to make sensors taking part of the IIS contribute to the intrusion detection by providing user-related contextual data. In the absence of any suitable dataset that mixes network and physical contextual data, we construct a new integrated dataset comprising benign data captured through an in-situ experiment and intrusion traces extracted from CIC-AndMal2017—a widely referenced dataset in the literature. Our evaluation confirms that considering both user physical context and network features improves the performance of intrusion detection, thereby making IIS more resilient to attackers.**

***Index Terms*—Individual-oriented Information System, Intrusion Detection System, NIDS, Context-aware, Android, Malware, Dataset, Explainable AI (XAI)**

I. Introduction

In the digital era, people increasingly rely on a wide range of Information Systems (IS) for everyday activities, including social networking, emailing, online banking, e-commerce, working, and educating. These services involve the processing, storage, and transmission of users' personal and sensitive data. For instance, online banking systems require log-in credentials, e-commerce platforms request credit card information, social networks store personal images and private messages. The close relevance of an individual's data with such ISs forms what we define as an **Individual-oriented Information System (IIS)** [1]. We conceptualize the IIS as a composite system comprising three components: (1) an individual user, (2) digital elements (e.g., information systems), and (3) physical elements (e.g., a smartphone, smart watch) that enable the user's interaction with these digital components.

The wide variety of information systems within an IIS results in a diversity of security solutions. These solutions are typically limited in scope: they are either designed to secure communication between multiple users and a single system or tailored to protect a specific device that requires intrusive control and management. Such isolated mechanisms fail against attacks where the adversary has sufficient knowledge to bypass them. For instance, an attacker with stolen credentials can authenticate on the victim's device. In this case, the actions appear legitimate from a system-centric or device-centric viewpoint, yet they represent clear breaches of personal security. This highlights a critical gap in conventional approaches: the lack of user global context in security models.

Adopting a user-centric security perspective within an IIS, we leverage the user's global context–where digital activities originate from digital elements and physical activities from physical elements–to detect inconsistencies that may signal security threats. Since these inconsistencies can occur across multiple devices and services, the network layer provides a comprehensive view point to observe and correlate such behaviors. For instance, if a device initiates network connections while the user is known to be running outdoors without the device in use, such a mismatch between physical and digital behavior could serve as a strong anomaly signal. We hypothesize that incorporating user physical contextual information improves the performance of Network-based Intrusion Detection Systems (NIDS).

In summary, our work makes the following contributions:

1) **A new dataset** (ibIDS) combining network traffic and physical sensor data collected from a real user's activities filling the gap in datasets that integrate user context with network behavior.
2) **A framework** leveraging user physical context data in network intrusion detection systems while ensuring user transparency and encrypted data confidentiality.
3) **An experimental validation** of the hypothesis that physical context improves NIDS performance using an integrated dataset combining ibIDS with malicious traffic from CIC-AndMal2017 [2].

The remainder of this paper is organized as follows. Section II reviews the relevant literature. Section III presents our proposed framework. Section IV describes the construction of a novel integrated dataset. Finally, Section V evaluates the research hypothesis using this dataset.

978-3-903176-75-1 ©2025 IFIP

II. State of the art

In this section, we explore three major research directions related to IIS security in the literature, with a particular focus on the use of user contextual and biometric data.

A. Datasets for Intrusion Detection System

An Intrusion Detection System (IDS) is a security mechanism that monitors network or system activities for malicious actions or policy violations and produces alerts to a management system [3]. It is categorized based on the data source into two main types: Network-based IDS (NIDS) and Host-based IDS (HIDS) [4]. NIDS monitor network traffic for signs of malicious activity, while HIDS focus on detecting anomalies within individual hosts.

Several recent surveys have reviewed the landscape of datasets used in the development and evaluation of NIDS [5, 6, 7], collectively identifying 32 datasets commonly cited in the field. Most of these datasets are conventional in nature, containing primarily network traffic traces (IP address, port number, bytes sent/received, etc.,), such as KDD Cup 99, Kyoto2006, NSL-KDD, CIC-IDS-2017, UNSW-NB15/18, and CICDoS2019. In contrast, HIDS datasets have been explored in many surveys [4, 8], focusing on data types such as system calls, log files, and other low-level indicators of host activity. These HIDS datasets offer fine-grained visibility into system behavior. Consequently, HIDS solutions require high privileges to monitor system-level activity.

Some datasets bridge the gap between traditional IT and cyber-physical systems, especially in Industrial Control Systems (ICS), such as SWaT, TON_IoT, and BETADAL, which include sensor data, actuator states, and control signals. TON_IoT also adds contextual system data like environmental sensors and system logs. However, the latter focus on system operations rather than user context which remains largely unexplored in mainstream IDS datasets.

B. Continuous Authentication

Biometrics and user contextual data play a pivotal role in the field of Authentication, especially in Continuous Authentication, as evidenced by several recent surveys [9, 10, 11]. These studies collectively examine how biometric modalities are integrated into authentication systems to enable ongoing user verification. In particular, Ryu et al. [9] identified two primary categories of biometric data used in such systems: behavioural data—including voice, keystroke dynamics, mouse movements, gait, touchscreen interactions, etc.,—and physiological data, such as iris patterns, fingerprints, palm prints and veins, heart or blood-related signals, etc.,. Focusing specifically on biometric authentication in smartphones, Syalevi et al. [10] conducted a more in-depth analysis of mobile use cases and reaffirmed the same two primary data modalities identified by Ryu et al. Notably, 87% of the reviewed studies centered on behavioural authentication, incorporating additional input sources such as mouth movements and user routine activities. Expanding the classification further, Ayeswarya et al. [11] introduced a third category —Context-aware biometrics— which encompasses contextual factors like IP address patterns, browsing history, user location, and device-specific attributes, thereby broadening the scope of data sources utilized in Continuous Authentication systems.

C. Wireless Body Area Network (WBAN)

Focusing on user context and biometrics in security, we investigate Wireless Body Area Networks (WBANs), where a system links to an individual's physiology. Since WBAN devices monitor physiological measurements via body attachment, many WBAN security solutions incorporate biometric data as input [12]. A comprehensive review by Gautam et al. [13] categorized WBAN security approaches into three key areas: key management, authentication, and trust management. Among these, biometric data plays a significant role. Indeed, its inherent randomness makes it valuable for key management, while its uniqueness to each user enhances authentication mechanisms. Notably, Electrocardiogram (ECG) signals have been used for cryptographic purposes, including random sequence generation [14] and biometric key management [15, 16]. Similarly, Han et al. [17] proposed a key generation scheme based on Photoplethysmography (PPG, a blood volume measurement metric), highlighting the potential of physiological data for cryptography in WBANs. Biometric data in WBANs is used to secure the body-area network itself, not to protect the user's broader digital activities.

In summary, this section reviews recent advances in leveraging user physical data for security. While user's physical data has been explored for authentication and key management, its integration into NIDS remains unexamined. This gap presents an opportunity to investigate how user physical context can complement network features to enhance NIDS in an IIS.

III. General Assessment Framework

This section introduces our framework used to examine the integration of network and physical contextual data and evaluate its impact within a NIDS-based detection module.

A. Overview

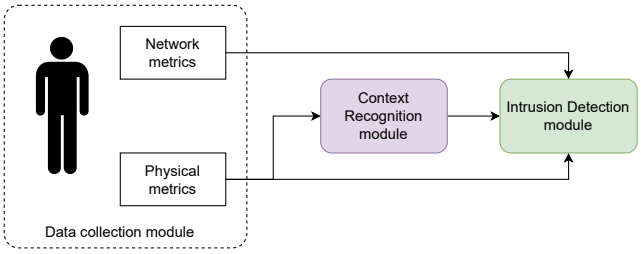

Fig. 1: Overview of our framework

The proposed framework is designed for intrusion detection by leveraging both network and physical metrics collected from an individual's context. As illustrated in Fig. 1, the architecture comprises three primary functional modules:

- Data Collection Module: This module collects raw data from the user's activities, including both Network metrics and Physical metrics.

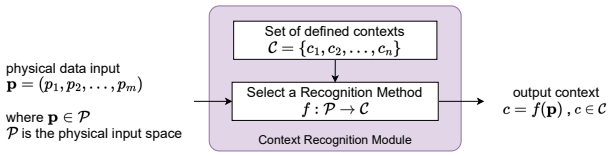

Fig. 2: Context Recognition Module

- Context Recognition Module: This module processes the physical inputs to determine the user's current context.
- Intrusion Detection Module: This module combines network features with other types of data from the previous modules to detect anomalies.

B. Data Collection Module

The data collection module serves as the initial stage of the proposed framework. This module is responsible for acquiring a diverse set of data from the activities of an individual and the surrounding environment. Specifically, it encompasses both **network data** and **physical data**. Network data are flow-based and include connection characteristics and communication protocols associated with the user's digital interactions, such as IP addresses, port numbers, or the number of bytes and packets transferred. Physical data are related to the user's physical state or the immediate environment, including sensor readings, biometric data, or location information. The integration of both network and physical data provides a contextual understanding for subsequent analysis by the context recognition and intrusion detection modules.

C. Context Recognition Module

Human Activity Recognition (HAR) has emerged as a prominent research domain across diverse applications. It aims to classify user activities from a predetermined activity set by analyzing multiple input data sources (e.g., vision-based and sensor-based) [18]. Within the context recognition module, we propose to adopt a HAR method, based on two key categories of information: (1) the physical contexts $\mathcal{C}$ to be detected and (2) the availability of physical input data $\mathbf{p}$. Fig. 2 represents our module.

First, the set of physical contexts must be carefully selected to ensure meaningful correlation with network usage behaviors. The rationale is that certain physical actions can provide strong indicators of legitimate or suspicious network activity. We define the set of possible contexts as a finite set $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$, where each c_i represents a specific context (e.g., walking, sitting, phone interacting).

Second, the framework must consider the set of available physical input data that can vary greatly depending on the environment and device capabilities. These data are represented as a vector $\mathbf{p} = (p_1, p_2, \ldots, p_m)$, where each p_i corresponds to a different physical metric.

Using both ($\mathcal{C}$ and $\mathbf{p}$), we select an activity recognition method f to provide the Intrusion Detection module with accurate, high-quality contextual insights.

D. Intrusion Detection Module

The Intrusion Detection Module receives three inputs: (1) Network data, (2) Physical data, and (3) User context $\boldsymbol{c}$. The network and physical data, collected from the Data Collection Module, are first pre-processed to create a combined dataset of network and physical features, denoted as D_{N+P}. Also, we retain a dataset containing only network features, D_N, that serves as a baseline to assess the contribution of physical features. The user context $\boldsymbol{c}$, provided by the Context Recognition Module, is utilized by the Model Type selection sub-module. Then, the Detection engine sub-module considers both a global detection model (M_g) and a context-aware multi-model approach (M_m). The global model (M_g) is trained on the entire dataset, while the multi-model scheme (M_m) selects a specific detection model for each recognized context. The detection engine, based on a machine-learning method, chooses a scheme between M_g and M_m, selects data sources with or without physical features, then analyzes the data and outputs alerts when traffic deviates from normal behavior. Fig. 3 illustrates the full pipeline of the proposed Intrusion Detection Module, from raw data ingestion and preprocessing, through context-driven model selection, to anomaly detecting and alerting.

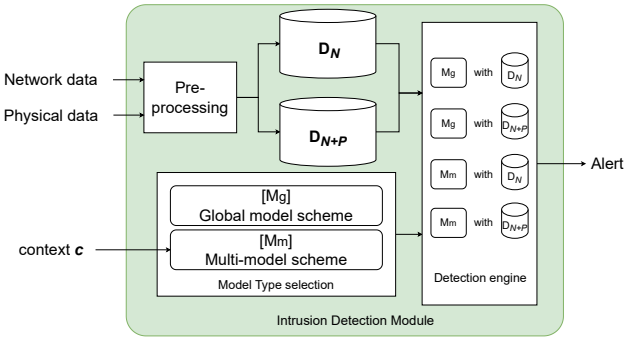

Fig. 3: Intrusion Detection Module

In the following sections, we leverage the proposed framework to conduct a comparative analysis of detection efficacy of a context-enriched model incorporating physical features (M_g with D_{N+P}) against the traditional model based solely on network features (M_g with D_N).

IV. Dataset Construction

To evaluate the proposed framework under realistic conditions, we created a hybrid dataset that integrates both synthetic attack traces (from CIC-AndMal2017) and real-world benign behavior (ibIDS). This section provides a detailed description of the integrated dataset and its constituent components.

A. Attack Scenario

Android smartphones, with their widespread use and access to sensitive data, are prime targets for malware exploiting software flaws and permissions through stealthy, time-based activation strategies [19, 20]. From an IIS perspective, the malware compromises the physical element to generate digital

activity. To emulate this attack for evaluation, we define an attack scenario with the following parameters:

- Threat: Android-based malware
- Attack period: 8 hours
- Operational characteristics: Covert network communications occurring at hourly intervals

B. Benign data

To collect benign user data, we experimented on an Android smartphone (OS version 14) belonging to one of the authors, who engaged in routine activities such as commuting, working, walking outdoors, gaming, and sleeping. The data collection app ran unobtrusively in the background for two weekdays, continuously recording (1) network communications and (2) physical sensor readings.

We captured network traffic with PCAPDroid, an open-source Android tool validated in prior studies [20, 21], which stores data in *.pcap* format along with flow-based features. Since it only captures network packets, we extended it to log sensor data from the device's hardware (accelerometer, gyroscope, gravity sensor, magnetometer, proximity sensor, ambient light sensor, and air pressure sensor) as well as location data from the Global Positioning System (GPS). Sensor measurements were sampled at a 5 Hz frequency, while GPS updates were logged every second. Network traffic was collected continuously in real time.

As the study did not involve sensitive or personally identifiable information, formal ethical approval was deemed unnecessary. Following the data collection phase, all acquired information underwent processing for permanent archival and **our dataset, termed *ibIDS*, is available online at** [22] https://doi.org/10.5281/zenodo.15658730.

C. Malicious Data

Since our data collection was conducted on a personal smartphone which regularly connects to a university network, we were restricted from installing real malware and directly capturing its behavior. Instead, we constructed an integrated dataset by combining our **real benign smartphone traffic** (ibIDS) with **real malicious traffic samples** (CIC-AndMal2017). The CIC-AndMal2017 dataset [2] includes network captures of 5,065 benign apps and 429 malware samples, each capture tied to a single app. However, background and pre-installed applications introduce extra traffic, so the captures do not reflect isolated app behavior. To address this limitation, we implemented a filtering methodology [23] to isolate malicious traffic by removing benign IP addresses from the malware captures. Since each malware family was collected under distinct scenarios, and only the **Adware** family had same user interaction patterns with the **benign** apps, the filtering protocol was applied accordingly:

- From the captures of benign families in CIC-AndMal2017, we created a list of benign source IP addresses and benign destination IP addresses.
- Within the Adware family captures, we excluded network flows exhibiting both benign source and benign destination addresses. The remaining traffic was classified as isolated malware communication.

Following the filtering process, we obtained isolated network traffic from five distinct Adware families: *Ewind, Feiwo, Gooligan, Kemoge,* and *Youmi*. Each family is represented by 8 to 10 individual captures, each lasting 40 minutes, approximately. In accordance with the approach to generate the CIC-AndMal2017 dataset [2], we extracted a 15-minute segment (from minute 25 to minute 40) from each capture, corresponding to the period of confirmed malware activation.

V. Experiments and Results

In this section, we instantiate the proposed framework to validate the effectiveness of using physical contextual information in detecting network intrusion, using our ibIDS dataset.

A. Evaluation Pipeline

Given our attack scenario–where malware is assumed to activate once every hour–and the available malicious data (specifically, five malware families, each with eight 15-minute traffic captures), we conducted our experiments following the procedure described in Procedure 1.

Procedure 1 Malware Injection and Evaluation Pipeline

for $malware \in$ [Ewind, Feiwo, Gooligan, Kemoge, Youmi] **do**
  for $i = 0$ to 23 **do**
    Inject 8 samples of $malware$ starting at hour i
    Data pre-processing
    for $trial = 1$ to 10 **do**
      Randomly split the dataset into training/testing sets
      Create a machine learning model
      Train and evaluate the model on the current split
      Record the evaluation metric (i.e., PR AUC)
    end for
    Aggregate the metrics over 10 trials for the injection i
  end for
  Visualize and analyze the aggregated results for the current $malware$
end for
Visualize and analyze overall results across all malware samples

The procedure iterates over each selected malware family. For each malware, 24 injection scenarios are created. In the i^{th} scenario, 8 captures of the malware are injected from hour i to hour $i+7$ (mod 24), as illustrated in Figure 4. The network data is then transformed from a **flow-based format** into a **time-slice-based representation**, which is more suitable for modeling patterns.

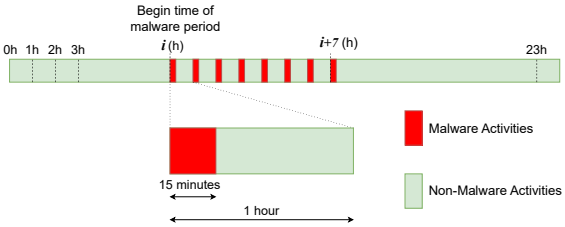

Fig. 4: Malware injection at time ***i*** (hour) (***i*** = 0,1,...23)

Each injection experiment is repeated ten times with different random train–test splits (11:1 benign-to-attack ratio), training and evaluating a supervised model in each run. Results are aggregated across runs, injection scenarios, and malware families to assess context-dependent performance and cross-family detection trends.

B. Data pre-processing

1) Network data - Flow-based to Time-slice-based: In this study, we adopt a time-slice framework, in which network and physical parameters are monitored and analyzed within fixed temporal windows. In our experiment, each time-slice is defined as a 60-second interval. We aggregate all network flows that terminate within the interval and compute corresponding time-slice-based features. These features are described in Table I.

TABLE I: Network Time-slice-based Features and Descriptions

#	Feature Name	Description
1	flow_count	Total number of network flows observed during the interval.
2	tcp_flow_count	Number of TCP flows observed.
3	udp_flow_count	Number of UDP flows observed.
4	unique_dst_ips	Count of unique destination IP addresses.
5	new_dst_ips	Number of destination IPs not seen in the previous time interval.
6	entropy_dst_ip	Entropy of destination IP addresses; measures randomness/distribution.
7	entropy_dst_port	Entropy of destination ports; captures the diversity in destination ports.
8	flow_per_dst_ip	Average number of flows per unique destination IP.
9	total_sent_pkts	Total number of packets sent.
10	total_rcvd_pkts	Total number of packets received.
11	total_pkts	Sum of sent and received packets.
12	pkts_per_flow	Average number of packets per flow.
13	total_sent_bytes	Total number of bytes sent.
14	total_rcvd_bytes	Total number of bytes received.
15	total_bytes	Sum of sent and received bytes.
16	bytes_per_flow	Average number of bytes per flow.
17	total_duration	Total duration of all flows in the observation window duration (in *millisecond*).

2) Physical data: In our experiment, we collect data from three physical sensors to capture user context: (1) Accelerometer, (2) Ambient Light, and (3) GPS. As these sources operate at different sampling rates, we perform preprocessing to align them for use in both the Context Recognition and IDS modules.

Accelerometer: The accelerometer provides 5 samples per second, each consisting of three-axis acceleration values: a_x, a_y, a_z (m/s^2). To summarize movement intensity, we compute the average magnitude of acceleration over one second:

$$\bar{m}_t = \frac{1}{5}\sum_{i=1}^{5}\sqrt{a_{x_i}^2 + a_{y_i}^2 + a_{z_i}^2}$$

where $a_{x_i}, a_{y_i}, a_{z_i}$ are the i-th sample values within second t. The resulting value $\bar{m}_t$ captures the movement intensity, irrespective of direction.

Ambient Light: Ambient light readings are sampled whenever there is a change in lighting conditions. It may occur within 1 second or more. To label the light value every 1 second, we take the latest value recorded. The light value at time t is computed as:

$$l_t = l_\tau$$

where l_τ is the most recent ambient light value recorded at timestamp $\tau \leq t$.

GPS: The GPS records geographic coordinates every second. To preserve privacy, raw locations are excluded; instead, we derive user speed, reflecting movement intensity without revealing exact positions. Speed at time t is computed as:

$$v_t = \frac{\Delta x}{\Delta t}$$

where Δx is the distance traveled between timestamps $t-1$ and t, calculated using the Haversine formula [24].

The raw GPS signal introduces noise, leading to fluctuations in the computed speed—even when the subject is stationary. To reduce this error, we apply a Butterworth low-pass filter to suppress rapid, noisy variations. Without any seek for optimiality, this filtering step efficiently reduces the GPS noise.

C. Intrusion Detection

1) Experimental Setup and Evaluation Metrics: In this section, we evaluate our detection scheme with a global model approach (M_g) and consequently, we do not utilize context $\boldsymbol{c}$ derived from the Context Recognition Module. Due to time and space constraints, the multi-model approach (M_m) is left for future work. We select the eXtreme Gradient Boosting (XGBoost) as a supervised classification algorithm. We aggregate the three physical signals ($\bar{m}_t$, l_t, and v_t) and produce three new time-sliced physical features at 60-second intervals to align with the defined time window: *count_light_1* (total number of seconds where light greater than 0), *count_acce_1* (total number of seconds indicating the smartphone's movement), and *avr_filtered_speed* (the average speed after applying a Butterworth low-pass filter).

For each injection scenario, to ensure fair comparison, both D_N and D_{N+P} share **identical train/test instances**: an 80/20 split is applied to D_N, and the same row indices are used for D_{N+P}. This design ensures that performance differences arise solely from the added features. The training set is used for hyperparameter tuning and cross-validation of the XGBoost model, which is then evaluated on the test set. This process is repeated 10 times with different random seeds. Given the imbalanced attack-to-normal ratio (1:11), The Area Under the Precision-Recall Curve (PR AUC) is used as the evaluation metric, and results are aggregated across scenarios to assess the impact of context-aware features on detection performance.

2) Results: Fig. 5 compares PR AUC across 24 Ewind injection timestamps. For each timestamp, two bars show the mean and standard deviation over 10 runs using datasets with and without physical features (D_{N+P} vs. D_N). While overall PR AUC values remain moderate, both exceed the

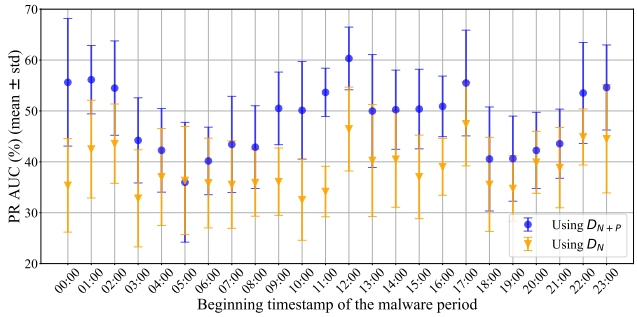

Fig. 5: PR AUC comparisons (by mean and standard deviation values) over 24 injection scenarios with *Ewind* in day 1

random baseline (1/12 ≈ 8.3%, where 1/12 is the ratio between positive and the whole samples). At certain timestamps, D_{N+P} achieves notably higher PR AUC than D_N.

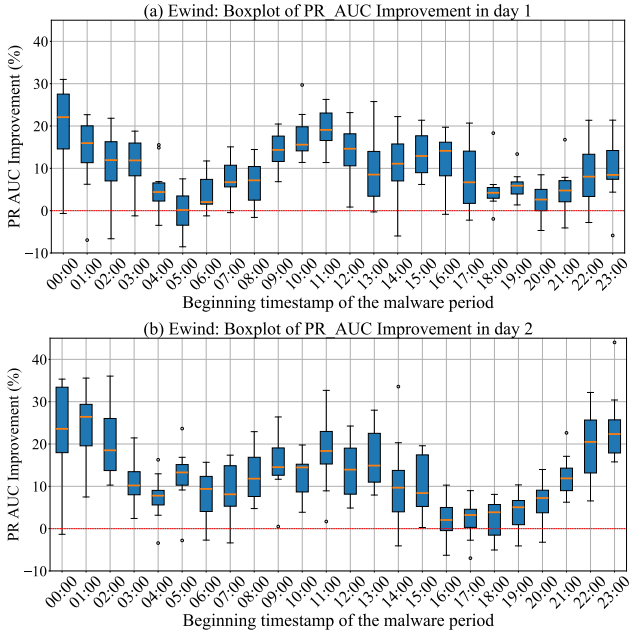

Fig. 6: Comparative enhancement of PR AUC when incorporating physical features alongside network-only features, demonstrated across two 24-hour periods: (a) day 1 and (b) day 2

Fig. 6 shows how D_{N+P} improve PR AUC compared with D_N on *Ewind*-injected data. Each boxplot displays the PR AUC improvement ($AUC_{\text{using } D_{N+P}} - AUC_{\text{using } D_N}$) across 10 runs for a given injection time. Most of the 24 timestamps show positive shifts in PR AUC, highlighting the value of user context in enhancing malware detection. Notably, beginning timestamps of malware duration, like 00:00, 01:00, and 11:00—periods when the user is likely inactive (e.g., sleeping or working)—show strong, consistent gains, while timestamps in the beginning of the morning or in the evening—when the user is more actively using the smartphone—yield minimal or negative improvements. This indicates that context-aware features generally help, though their effectiveness depends on the user's context during malware activity.

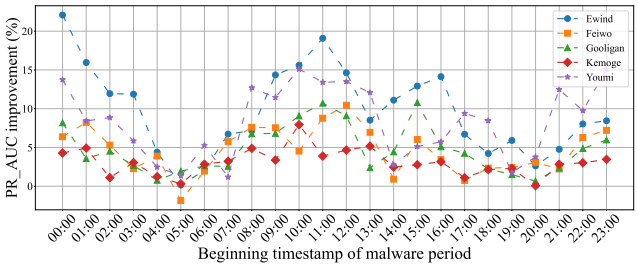

Fig. 7: Enhancement of PR AUC across distinct malware families.

Fig. 7 shows the median PR AUC improvement over 24 hours across the five malware families. While most families see consistent benefits–typically above 0%–the magnitude and timing of these improvements vary. *Ewind* and *Youmi* often exhibit the strongest gains, with several peaks exceeding 10%, suggesting that context data is particularly helpful for detecting these malware families. In contrast, *Kemoge* shows relatively minimal improvement throughout, indicating limited benefit from contextual features. These trends emphasize that the impact of user context is both malware-specific and context-dependent.

To highlight the effectiveness of user contextual features, we focus on the 00:00 *Ewind* case that exhibits one of the highest improvements in PR AUC across all periods. The result suggests that when user contextual data indicates a period of typical inactivity (e.g., sleeping), an observed increase in network activities is consequently perceived as more anomalous and thus more detectable. Fig. 8 represents the PR curves, showing significant improvements in malware detection when user context is included.

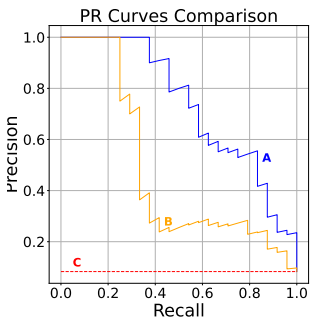

Fig. 8: PR curve comparing context-aware and baseline models detecting Ewind activities at 00:00, day 1, with same random data split. A: With D_{N+P} (PR AUC = 0.7393); B: With D_N (PR AUC = 0.4767); C: Random guesser (PR AUC = 0.0830)

D. Explanibility by SHAP

To identify which features contribute to the improved detection performance, we utilize SHapley Additive exPlanations (SHAP) [25]. SHAP is a game-theoretic approach that quantifies feature attribution by assigning importance values to individual variables based on their respective contributions to specific predictive outcomes.

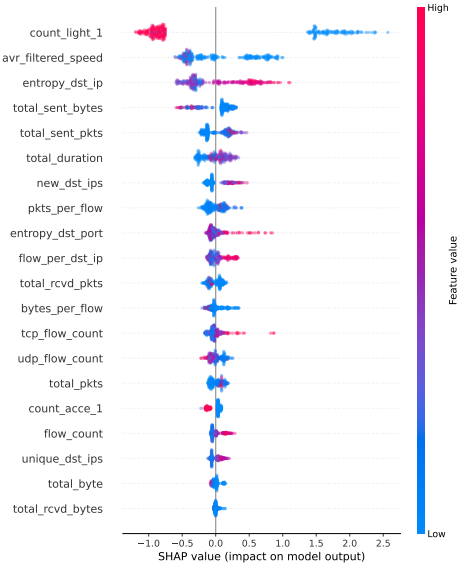

Fig. 9: Feature importance according to their relative influence as quantified on SHAP analysis with XGBoost detecting *Ewind* activities at 00h00, day 1

The SHAP summary plot in Fig. 9 shows that during nighttime, physical features such as *count_light_1*, *avr_filtered_speed*, and *entropi_dst_ip* play a key role in the model's predictions. Although their relationship to the output is not purely linear, patterns suggest that specific conditions–like low light or changes in motion–are critical signals the model uses to differentiate normal from anomalous behavior. The SHAP analysis demonstrates the relevance of physical data in the decision-making of the IDS algorithm and thus the global relevance of our approach.

VI. Conclusion

In this paper, we introduced a contextual security approach for Individual-oriented Information Systems. We proposed a novel framework that integrates the user's contextual information into NIDS. To evaluate this framework, we collected a unique dataset comprising both network traffic and physical sensor data from a personal smartphone over a 48-hour period during normal daily activities. The dataset is publicly available online. This benign dataset was then augmented with malware traffic from a public dataset to enable comprehensive evaluation. Our experimental results support the hypothesis that incorporating user contextual data can enhance the performance of a NIDS, highlighting the value of the contextual information of a user in an IIS for intrusion detection.

Future research can focus on: (1) Exploring unsupervised learning methods–where models learn normal data patterns to detect anomalies–to further validate the effectiveness of context-aware intrusion detection in practical, deployable settings; (2) Evaluating multi-model approach to more thoroughly confirm the benefits of incorporating user context in enhancing IDS performance.

Acknowledgement

This work is supported by The Institut Cybersécurité Occitanie (ICO), funded by the Région Occitanie, France; and IMT Atlantique, Bretagne-Pays de la Loire, France. We kindly thank Eric Alata and Daniela Dragomirescu from LAAS-CNRS for their support in this work.

References

[1] V.-T. Nguyen *et al.*, "Toward context-aware security for individual information systems," in *Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information (RESSI)*, 2024.

[2] A. H. Lashkari *et al.*, "Toward developing a systematic approach to generate benchmark android malware datasets and classification," in *2018 International Carnahan conference on security technology (ICCST)*. ieee, 2018, pp. 1–7.

[3] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Department of Computer Engineering, Chalmers University of Technology, Tech. Rep., 2000.

[4] H. Satilmiş *et al.*, "A systematic literature review on host-based intrusion detection systems," *Ieee Access*, vol. 12, pp. 27 237–27 266, 2024.

[5] A. J. A. Immastephy and K. Punitha, "A systematic review on network intrusion detection system based on machine learning and deep learning approach," in *E3S Web of Conferences*, vol. 540. EDP Sciences, 2024.

[6] R. Holdbrook *et al.*, "Network-based intrusion detection for industrial and robotics systems: A comprehensive survey," *Electronics*, vol. 13, no. 22, p. 4440, 2024.

[7] S. A. Abdulkareem *et al.*, "Network intrusion detection: An iot and non iot-related survey," *IEEE Access*, 2024.

[8] Z. T. Sworna *et al.*, "Nlp methods in host-based intrusion detection systems: A systematic review and future directions," *Journal of Network and Computer Applications*, vol. 220, p. 103761, 2023.

[9] R. Ryu *et al.*, "The design and evaluation of adaptive biometric authentication systems: Current status, challenges and future direction," *ICT Express*, vol. 9, no. 6, pp. 1183–1197, 2023.

[10] R. Syalevi *et al.*, "Study on the implementation of multimodal continuous authentication in smartphones: A systematic review." *International Journal of Advanced Computer Science & Applications*, vol. 15, 2024.

[11] S. Ayeswarya and K. J. Singh, "A comprehensive review on secure biometric-based continuous authentication and user profiling," *IEEE Access*, 2024.

[12] M. Yaghoubi *et al.*, "Wireless body area network (wban): A survey on architecture, technologies, energy consumption, and security challenges," *Journal of Sensor and Actuator Networks*, vol. 11, no. 4, p. 67, 2022.

[13] A. K. Gautam and R. Kumar, "A comprehensive study on key management, authentication and trust management techniques in wireless sensor networks," *SN Applied Sciences*, vol. 3, no. 1, p. 50, 2021.

[14] C. Zenieh *et al.*, "A high-throughput random binary sequence generator based on ecg," *IEEE Access*, vol. 10, pp. 67 117–67 127, 2022.

[15] H. B. Hwang *et al.*, "Preliminary study of novel bio-crypto key generation using clustering-based binarization of ecg features," *Sensors*, 2024.

[16] N. Karimian *et al.*, "Never lose your ecg: A novel key generation and authentication scheme for implantable medical devices," *IEEE Access*, vol. 11, pp. 81 815–81 827, 2023.

[17] W. Han *et al.*, "An ppg signal and body channel based encryption method for wbans," *Future Generation Computer Systems*, vol. 141, 2023.

[18] M. H. Arshad *et al.*, "Human activity recognition: Review, taxonomy and open challenges," *Sensors*, vol. 22, no. 17, p. 6463, 2022.

[19] M. Conti *et al.*, "Asaint: a spy app identification system based on network traffic," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020.

[20] M. K. Qabalin *et al.*, "Android spyware detection using machine learning: a novel dataset," *sensors*, vol. 22, no. 15, p. 5765, 2022.

[21] M. Bayat *et al.*, "Itc-net-blend-60: a comprehensive dataset for robust network traffic classification in diverse environments," *BMC Research Notes*, vol. 17, no. 1, p. 165, 2024.

[22] V.-T. Nguyen, "ibids - a dataset of real-world network communication and user physical contextual data," Jun. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15658730

[23] D. Cao *et al.*, "Droidcollector: A high performance framework for high quality android traffic collection," in *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, 2016, pp. 1753–1758.

[24] R. W. Sinnott, "Virtues of the haversine," *Sky and Telescope*, 1984.

[25] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, 2017.