KISS: Knowledge Integration System Service for ML End Attacks Detection and Classification

Sergei Chuprov, Leon Reznik, and Raman Zatsarenko

***Abstract*—Machine Learning (ML) systems are integrated with other parts of modern cyberinfrastructure (CI), which forms the novel ML with Integrated Network (MLIN) structure. Various security tools are already employed in CI to detect malicious attacks. However, their operation is limited to a certain CI part or MLIN component without considering their integration and the ML end performance as the major indicator. We design and implement Knowledge Integration System Service (KISS) that introduces a novel approach to detect and classify attacks into adversarial attacks against ML end system (A-Attacks), and against base MLIN infrastructure (B-Attacks). Unlike traditional security mechanisms, KISS examines how attacks on individual MLIN components impact the overall ML end system performance, and extracts and integrates knowledge from each MLIN CI component to better distinguish between the attack types. As our major contributions, we (1) investigate the effects of A- and B-Attacks on ML systems, actively used in industry. We (2) develop the KISS prototype and verify it in practice. We demonstrate how KISS can be integrated into already established MLIN CI and combined with the traditional security tools. Our experiments verify that KISS improves attack detection and their initial classification between A- and B-Attacks in MLIN operational setups.**

***Index Terms*—Machine learning, Services integration framework, Adversarial attacks, Data quality**

I. Introduction

OVER the last decades, Machine Learning (ML) trained systems and services have become a critical part of the modern cyberinfrastructure (CI), building up a novel paradigm, which we introduced and formalized in [14] as ML with Integrated Network (MLIN) systems. This technological advancement was almost immediately followed up by generating new attacks, which we call adversarial or A-Attacks, against ML systems, that significantly expanded an old arsenal of attacks against data generation and communication systems and networks, which we call in this paper base infrastructure attacks or B-Attacks. Both types typically result in Data Quality (DQ) degradation of data fed into ML systems and consequently in their performance decrease. While multiple methods and tools for A-Attacks detection have been developed [11], [17], [41] and added to the old tools arsenal, they are not properly integrated together, even at the structural level. These methods typically concentrate on a particular factor affecting the performance (e.g., a specific aspect of DQ or ML model) and consider only a separate CI component (e.g., either sensors vs. communication channels and networks vs. ML-trained classifiers as the end systems). Such tools fail in many cases to detect and reliably classify malicious attacks against MLIN components and MLIN CI resulting in ML end system performance deterioration [30].

In contrast to the fore mentioned approaches, we consider attack detection in MLIN from the system's integration perspective based on the knowledge utilization. We propose to extract, accumulate, and employ the knowledge on the MLIN components and attacks against them to design the Knowledge Integration System Service (KISS) that facilitates detection and classification of attacks against MLIN systems that eventually result in the ML end performance degradation. We propose a generic design approach that collects knowledge from all available intrusion detection, system and network monitoring, and security evaluation systems and tools. The integration idea means that the proposed KISS is not intended to replace the existing attack detection mechanisms and tools but to integrate them. In our implementation, we develop a KISS prototype that executes the attack classification at the high level, differentiating between A- and B-Attacks. We demonstrate how KISS can be merged with the existing tools, such as SNORT. We illustrate the practical ways of the knowledge extraction, accumulation and application not only for attack classification but also for producing recommendations on the defense against the detected attacks.

Being a knowledge integration system, KISS (see sec. IV for its generic design description) requires knowledge on how various attacks against MLIN components affect ML end performance for a specific use case. To derive this knowledge, we conduct an empirical study (see sec. II) that investigated how various conditions, particularly malicious A- and B-Attacks, affect DQ on different stages of the data life-cycle and various MLIN components. Critically, we analyze (see sec. III) the combined effects of these attacks on the overall performance of the ML end system. Another part of the paper describes the KISS prototype design, implementation and application in case studies that integrated ML system performance evaluations and intrusion detection tools. In our use cases, we integrate KISS with the existing industrial security solutions, such as SNORT, develop and verify a prototype – see Figure 1. In sec. V, we demonstrate a practical case of KISS integration into a real MLIN system operation with the case analysis in sec. VI. The second use case described in sec. VII demonstrates KISS knowledge integration with industrial knowledge and databases.

Sergei Chuprov is with the Department of Computer Science, The University of Texas Rio Grande Valley, Edinburg, TX 78539 USA (e-mail: sergei.chuprov@utrgv.edu). Leon Reznik, and Raman Zatsarenko are with the B. Thomas Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: leon.reznik@rit.edu; rz4983@rit.edu).

Manuscript received XX, XXXX; revised XX, XXXX.

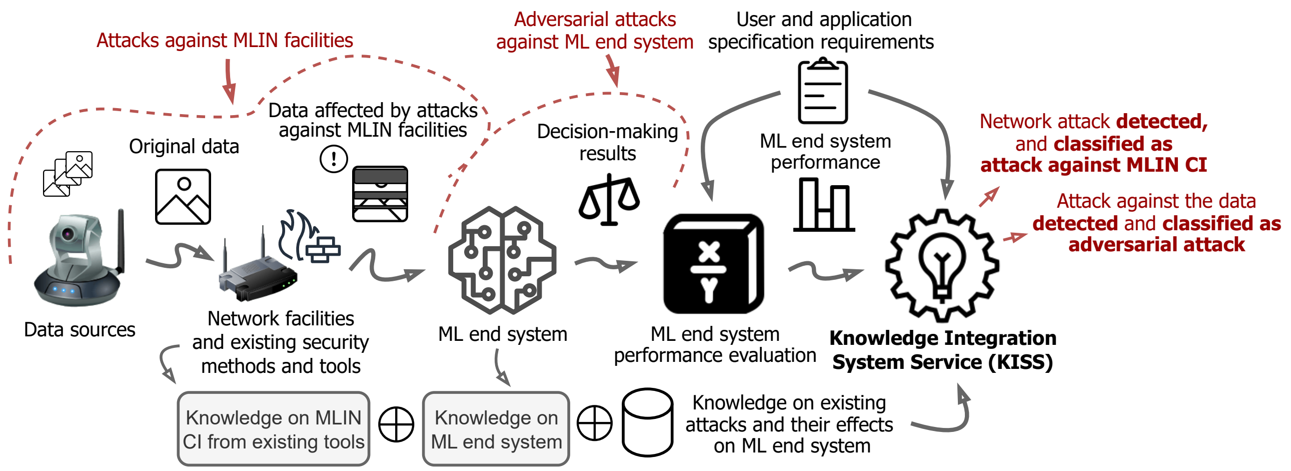
Attacks against MLIN facilities
Adversarial attacks against ML end system
User and application specification requirements
Data affected by attacks against MLIN facilities
Original data
Decision-making results
ML end system performance
Network attack detected, and classified as attack against MLIN CI
Attack against the data detected and classified as adversarial attack
Data sources
Network facilities and existing security methods and tools
ML end system
ML end system performance evaluation
Knowledge Integration System Service (KISS)
Knowledge on MLIN CI from existing tools
Knowledge on ML end system
Knowledge on existing attacks and their effects on ML end system

Fig. 1. MLIN composition and operation with, and interrelations between its components. The figure represents the data life-cycle in MLIN; the types of knowledge derived and integrated by KISS; and the types of malicious attacks detected and mitigated by KISS based on the knowledge integration, and MLIN components vulnerable to these attacks

II. Investigating Attacks Impact on ML End Systems

We examine two major attack types: A(dversarial)-Attacks, which directly manipulate data to disrupt ML systems, and B(ase CI)-Attacks, which target base CI infrastructure, impacting ML performance. We analyze these two attack types in video and image object detection using cloud ML-as-a-service and open-source models. Table I summarizes our investigation, detailed in subsequent subsections.

A. Employed ML Models

In our study, as an ML end system we consider image recognition systems, both open-source (white box) and commercial (black box) that provide As-A-Service (AAS) access to their features, such as AWS Rekognition[1] and Google Vision AI[2], with no information about their models shared. We employ Faster-RCNN [29] and SSD [22] architectures to configure the parameters for the applied image distortion methods. The effects of A-Attacks are evaluated on AWS Rekognition and Google Vision AI black box systems. We also evaluate the effect of image examples corrupted by B-Attacks on YOLO [28], being the example of one-stage image detection architecture.

B. Employed Image and Video Datasets

1) A-Attack Scenario: To study ML object detector performance under adversarial attacks, we utilize the "Traffic Sign" and "Stop Sign" image subsets from the Open Images V6 dataset [9], dividing them into a 20-image set for preliminary black-box attack tuning and a 101-image set for precise evaluation of effective adversarial techniques, generating adversarial images as described in sec. II-C. Additionally, to assess video-based attacks, we employ AWS Rekognition on 25 handpicked 8-10 second videos from the Berkeley Deep Drive (BDD110K) Dataset, focusing on "Traffic Lights" and "Road Sign" labels, which initially yielded confidence scores above 90% in selected original videos.

[1]https://aws.amazon.com/rekognition/
[2]https://cloud.google.com/vision

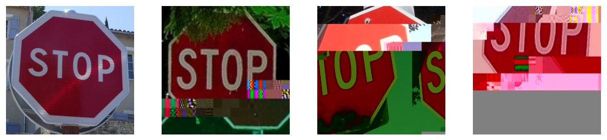
(a) (b) (c) (d)

Fig. 2. Examples of images distorted by different packet loss and buffer overflow during their network communication: (a) – original stop sign image; (b) – stop sign image transmitted with buffer 128B and 5% packet loss; (c) – stop sign image transmitted with buffer 256B and 5% packet loss; (d) – stop sign image transmitted with buffer 512B and 20% packet loss

2) B-Attack Scenario: In this case, we also utilize the same images from the employed Open Images V6 dataset. We utilize the original images from the employed subset to train our models, and a separate set of testing images with various corruptions caused by network packet loss due to B-Attacks. For the images with various corruptions, we employ the data produced in [14], which includes "Traffic Sign" and "Stop Sign" images corrupted by network packet loss while transmitting them over a real wireless network employed using POWDER testbed [10]. Some examples of the employed images can be seen in Figure 2. As B-Attacks can lead to pixel distortions and color manipulations in the images reconstructed on the receiver [14], we also produced images with color adjustments: grayscaled images and images with increased contrast.

C. Investigated Attacks

1) A-Attacks: A-Attacks introduce subtle, imperceptible perturbations to fool ML object detectors [2], using techniques

TABLE I
SUMMARY OF INVESTIGATIONS OF A- & B-ATTACK IMPACT ON ML END SYSTEMS

Study	Attack Type	Attacks	Data Modality	Dataset	Models	Evaluation Metrics
Sec. III	A-Attacks	LowKey [13], TI [16], Masked Adversarial Images [27], Admix [37], Momentum [15]	Images	Open Images V6 (121 images) [9]	AWS Rekognition, Google Vision AI	Percentage of images from the whole set in which the target pattern was detected; Average confidence score
		Random Noise	Videos	Berkeley Deep Drive 110K (25 videos)	AWS Rekognition	Confidence score; Calculated percentage of correct decisions
	B-Attacks	Flood attacks [19], black hole attacks [6], gray hole attacks [21], sink hole attacks [25], jamming attacks [7], man-in-the-middle (MITM) attacks [4]	Images	Open Images V6 (3200 images)	YOLO	Average Precision (AP), mean average precision (mAP)
Sec. V and VI	A, B-Attacks	Random Noise, UDP flooding	Videos	Berkeley Deep Drive 110K (34 videos)	YOLO	Confidence score ratio; Detection ratio; Weighted confidence score
Sec. VII	A, B-Attacks	Random Noise, UDP flooding	Images	Open Images V6 [9], DAmageNet [12] (663 images)	YOLO	Confidence Score

like Box-Constrained L-BFGS [36], FGSM [18], Carlini & Wagner Attacks [11], Deepfool [23], and JSMA [39]. We investigate the impact of the LowKey ensemble attack [13], targeting AWS Rekognition and Google Vision AI, with parameters tuned against YOLO, Faster-RCNN, and SSD [22], enhanced with Gaussian blur [40], Translation Invariance (TI) [16], Masked Adversarial Images [27], Admix [37], Momentum [15], and random noise injection. For random noise addition, noise is injected into input data using predetermined levels. Images are altered with intensities of 100 and 200 using the Pillow library, while videos are attacked by randomly altering 2–15% of pixel colors to white. We limit the maximum noise intensity introduced into the videos to 15% as with higher percentages AWS Rekognition yields confidence levels under 50%, rendering it ineffective for object recognition tasks. Defense techniques employed to mitigate A-Attacks often include anomaly detection [8], [26], gradient-based detection [32], and ensemble detection [1]. However, these often introduce computational overhead [33], [34], lack generalization [20], and produce false positives [35]. Retrofitting these techniques can disrupt MLIN operation and require ongoing maintenance. In contrast, our work proposes a complementary KISS system for integration with existing security, network, and ML tools.

2) B-Attacks: Malicious attacks against MLIN base CI may affect the quality of the data communicated in MLIN. Many Denial of Service (DoS) attacks, e.g. flood attacks [19]; black hole attacks [6]; grey hole attacks [21]; sink hole attacks [25]; jamming attacks [7]; and man-in-the-middle (MITM) attacks [4] may result in packet loss if UDP protocol is employed [31]. As we investigate video and image communication and streaming, where UDP is frequently employed, packet loss impact on the performance of end ML systems, such as object detectors or image classifiers need to get examined. Aqqa *et al.* [3] studied such renowned open-source ML architectures as YOLOv3, Faster-RCNN, SSD512, and RetinaNet, and showed that a decrease in performance can be expected when dealing with packet loss if those object detectors were trained on high-quality data only. Baidya and Levorato [5] showed that network interference during data communication negatively influences the performance of intelligent object detectors. In our experiments, we investigate the case with images and videos affected by packet loss, that might be caused by DoS attacks, during their transmission over a network (see sec. II-B)

D. Evaluation Metrics and ML Models Preparation

For A-Attacks against images, we evaluated both AWS Rekognition and Google Vision AI, using similar metrics but including bounding box counts for Rekognition, which also measured classification performance across Stop Sign (SS), Traffic Sign (TS), and Sign labels. Video attacks were assessed using AWS Rekognition's confidence scores and correct decision percentages. For B-Attacks, we used Average Precision (AP) and mean Average Precision (mAP) to evaluate pre-trained ML models fine-tuned on the undistorted Open Images V6 subset, comparing baseline performance on original data to that on attacked data. The evaluation dataset consisted of 3200 images, with 10% randomly selected as a test set, which was then augmented with various distortions before being fed to the model.

III. ANALYSIS OF ATTACKS' EFFECTS ON ML END SYSTEMS

A. A-Attacks Scenario

1) Attacks against Images: Using 20 preliminary images, we evaluated all the adversarial techniques (excluding Ensemble [13]) against AWS Rekognition (Figure 3, Table II). Rekognition struggled with Stop Sign (SS) detection, identifying only 9/20 original images, but surprisingly, most attacks

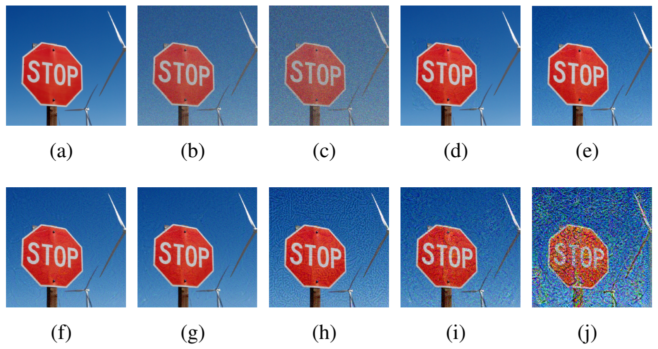

Fig. 3. Samples of the images used in the A-Attacks scenario: (a) – original image; (b) – noise (100); (c) – noise (200); (d) – masked TI; (e) – admix; (f) – TI (30 iterations); (g) – TI (60 iterations); (h) – ensemble (6 iterations) and high LR; (i) – ensemble (50 iterations) and momentum; (j) – ensemble (100 iterations)

TABLE II
EVALUATION OF VARIOUS A-ATTACKS AGAINST AWS REKOGNITION (PRELIMINARY EVALUATION)

Cat.*	**Met.†**	**A-Attack type**					
		Or.‡	**Mask TI**	**Ad-mix**	**TI (30 iter.)**	**TI (60 iter.)**	**M-tum§**
SS	Det.** (%)	45	45	60	60	55	65
	Conf.	78.5	79.7	76.5	74.9	77	74.3
	Av. conf. (all)	35.3	35.8	45.9	45	42.4	48.3
	BBs detected	0	0	0	0	0	0
TS	Det. (%)	90	80	80	85	80	80
	Conf.	95.7	91.3	92.2	87.8	90.3	90
	Av. conf. (all)	86.1	73.1	73.7	74.6	72.2	72
	BBs detected	24	12	10	8	7	6
Sign	Det. (%)	95	85	95	95	95	90
	Conf.	88.5	89.8	88.3	86.9	87	87.6
	Av. conf. (all)	84	76.3	83.8	82.5	82.7	78.8
	BBs detected	0	0	0	0	0	0

* Category; † Evaluation Metrics; ‡ Original Images; § Momentum; ** Detection

increased this rate. It performed better on Traffic Sign (TS), detecting 18/20 original images, but all attacks halved this. Google Vision AI (Table III) detected all 20 original SS images, with most attacks reducing detection, and the 100-iteration Ensemble attack failing to detect any. We then tested both platforms on the extended set of 101 SS images (Table IV). Rekognition showed higher robustness against attacks but lower baseline performance (67.3% detection) compared to Vision AI. Masked TI had the largest impact, reducing detection to 20.8% for Vision AI and 31.7% for Rekognition.

TABLE III
EVALUATION OF VARIOUS A-ATTACKS AGAINST GOOGLE VISION AI (PRELIMINARY EVALUATION)

Met.	**A-Attack type**								
	Or.	**M-TI***	**Ad-mix**	**TI (30 iter.)**	**TI (60 iter.)**	**M-tum**	**Ens.† (6 iter.)**	**Ens. (50 iter.)**	**Ens. (100 iter.)**
Det. (%)	100	85	95	100	100	80	90	40	0
Conf.	87.1	88.2	88.8	86.9	86.8	89.8	86.8	91.2	0
Av. conf. (all)	87.1	75	84.4	86.9	86.8	71.8	78.1	36.5	0

* Masked TI; † Ensemble attack

TABLE IV
EVALUATION OF VARIOUS A-ATTACKS AGAINST GOOGLE VISION AI AND AWS REKOGNITION (EXTENDED DATASET)

Class.*	**Met.**	**A-Attack type**				
		Or.	**TI (30 iter.), Mask, LR=.01**	**TI (30 iter.), LR= .015**	**TI (30 iter.), LR=.02**	**TI (5 iter.), LR=.04**
GV†	Det. (%)	99	20.8	31.7	58	57
	Conf.	87.8	85.9	87.2	89.1	85.2
	Av. conf. (all)	86.9	17.9	27.6	52	48.9
Rek.‡	Det. (%)	67.3	31.7	49.5	71.3	66.3
	Conf.	74.7	71.9	79.6	78.4	76
	Av. conf. (all)	50.3	22.8	39.4	55.9	50.4

* Classifier; † Google Vision AI; ‡ AWS Rekognition

Overall, Rekognition demonstrated greater resilience to adversarial attacks, despite its lower accuracy on original images.

2) A-Attacks against Videos: For A-Attacks against videos, we evaluate Rekognition's performance using confidence scores, presented in Figure 4(a), and the overall rate of correct decisions, presented in Figure 4(b). Since we employ BDD110K dataset for this case, we use object labels different from the cases with images: "Traffic Lights" and "Road Sign", which are provided with the data. Our findings reveal significant degradation in both evaluated metrics as noise levels increased across the videos. For "Traffic Lights" detection, confidence scores exhibit a steady decline with increasing noise, reaching a minimum of 30.1% at the noise intensity of 15%. Similarly, the rate of correct detections for "Traffic Lights" drops rapidly at the noise level of 2% compared to original videos, followed by a more gradual decline at higher noise intensities. "Road Sign" correct decisions rate exhibits in general lower performance than "Traffic Lights" across all noise levels. While confidence scores also decrease with noise for "Road Signs", they remain overall lower compared to "Traffic Lights". The correct detection rate for "Road Signs" also suffers a steeper decline with increasing noise, highlighting a greater sensitivity to noise compared to "Traffic Lights". For all the noise intensities, we also record how the video

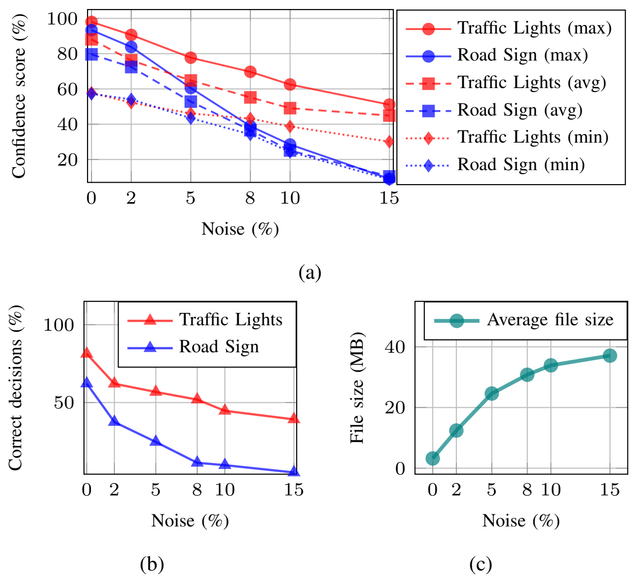

Fig. 4. Experimental results obtained by processing the videos, affected by the noise of various intensity, with AWS Rekognition: (a) – mean values of maximum, average, and minimal confidence scores provided by Rekognition across 25 videos affected by various noise intensities; (b) – the correct decisions ratio demonstrated by Rekognition across 25 videos affected by various noise intensities; (c) – the average video file size for 25 videos affected by various noise intensities

file size changes after the pixels were corrupted in the video. Figure 4(c) indicates that adding noise significantly increases the average size of video files processed by Rekognition.

B. B-Attacks Scenario

Table V demonstrates the impact of various image corruption factors on a pre-trained YOLO object detector. Surprisingly, “Noise (100)”, “Noise (200)”, and “Contrast increase” positively affected Stop Sign (SS) category performance, while conversely, Traffic Sign (TS) performance and overall mAP decreased across all corruptions. Notably, “Noise (200)”, “2% Packet loss”, and “5% Packet loss” resulted in extreme performance drops for the TS category. As expected, packet loss significantly degraded performance due to image corruption, rendering images unrecognizable. Interestingly, “Noise (100)”, “Noise (200)”, “Grayscale”, and “Contrast increase” appeared to enhance SS detection, potentially by filtering less essential features and preserving critical ones. However, TS images, with their greater feature diversity, suffered negative impacts from all distortions, increasing the diversity of feature distributions in the test images.

C. Discussion of the Attacks’ Effects Evaluation Results

Based on our evaluations, B-Attacks, which assume an attacker has limited knowledge of the ML system and its architecture, can cause substantial and unpredictable decreases in ML application performance. In contrast, A-Attacks require a high intensity of distortion to significantly degrade performance; the image distortions must be strong enough to alter the image substantially or completely obscure the patterns of interest for correct detection and classification. Therefore, MLIN CI security and protection are vital for ensuring ML application performance, as even attacks like network flooding can significantly affect it. However, conventional security mechanisms like IDS are unable to associate B-Attacks with potential ML performance degradation. This leaves MLIN users and operators challenged to investigate the root cause of performance degradation, as reasons can vary, hindering their ability to apply appropriate and effective measures to maintain performance at the required level.

TABLE V
IMAGE OBJECT DETECTION PERFORMANCE FOR B-ATTACKS (YOLOV3)

Category	Distortion Type	AP/mAP
SS	Original	84.46
	Noise (100)	99.32
	Noise (200)	92.57
	Grayscale	99.03
	Contrast increase	99.26
	2% Packet loss	41.38
	5% Packet loss	30.77
TS	Original	70.69
	Noise (100)	61.29
	Noise (200)	43.55
	Grayscale	63.25
	Contrast increase	68.04
	2% Packet loss	27.78
	5% Packet loss	10.14

IV. KISS Architecture and Operation

To address the gap between current mechanisms employed in MLIN, and to facilitate more informed attacks detection and classification in cutting-edge MLIN systems, we propose KISS: the service that integrates knowledge extracted from MLIN components. KISS allows to: (1) extract knowledge on interrelations between MLIN components, effects of known attacks, and changes in ML end performance; and (2) to accumulate and integrate this knowledge to better detect and classify malicious attacks into A-Attacks targeted against ML end system, and B-Attacks targeted against MLIN CI; (3) use the results of the attack classification to better inform recommendations on defense measures to prevent or mitigate the degradation in ML end performance. The aforementioned points facilitate meeting user and application requirements towards the ML end system performance. KISS integrates knowledge available in MLIN from three key elements:

MLIN components: this includes understanding the functionalities and vulnerabilities of each component and their influence on quality of the processed data and ML end application performance.

Existing security mechanisms: KISS complements existing security tools, incorporated into the current MLIN CI, by considering their reports and outputs.

ML end system performance: KISS employs the existing ML end performance evaluation methods and tools for the performance deterioration.

KISS, designed to complement existing security, enhances attack detection and classification without significant resource overhead or MLIN redesign. Figure 1 demonstrates the integration of knowledge by KISS to identify attacks and their impact on MLIN components, addressing data lifecycle vulnerabilities.

A. Knowledge Integration to Detect Malicious Attacks against MLIN

Our approach uses ML end system performance as a core monitoring indicator for KISS. Considering MLIN component interrelations and leveraging existing security mechanisms typically employed in practice, such as IDS, Intrusion Prevention System (IPS), and Security Orchestration, Automation, and Response (SOAR), KISS integrates operational knowledge to provide context on potential attack types. This contextual information aids in recommending actions to prevent or mitigate ML performance degradation.

1) Knowledge on MLIN CI from Existing Tools: Data transmitted to the ML system via network facilities can experience QoS degradation, potentially from B-Attacks, leading to DQ deterioration. By integrating network QoS data with security tool feedback (IDS/IPS), KISS enables informed reasoning about DQ degradation. For instance, simultaneous network QoS drops, network attack alerts, and ML performance degradation are classified as a B-Attack based on integrated MLIN component knowledge.

2) Knowledge on ML End Application Performance: After receiving and processing the data, ML system performance is evaluated against specifications. Performance drops, without network or security anomalies, indicate potential A-Attacks. KISS can integrate with dedicated A-Attack detection tools to scan input data and classify performance degradation due to these attacks. In practice, detailed component knowledge allows for more accurate mitigation. For instance, if KISS detects a network QoS drop during data transmission, it can analyze QoS metrics and network characteristics to distinguish between technological and adversarial origins. Based on this analysis and knowledge integration, KISS can recommend specific actions to prevent or mitigate ML system performance degradation, such as enforcing stricter security requirements, changing network transport protocols, or re-balancing traffic loads.

V. Use Case 1: KISS Integration with MLIN Components

A. Technical Setup

In this use case, KISS is employed in a corporative network segment used for transmitting media files from the client to the server. To setup the MLIN architecture, we employ three machines: client, server, and monitor, connected to a CISCO Catalyst 2960G (8TC-L) switch via Ethernet. We also have the intruder machine connected to the switch, which we use for modeling of B-Attacks, in particular network flooding attack mentioned in sec. II-C2. The client and monitor machines are equipped with Intel i5-6500, 16 GiB RAM and running Ubuntu 22.04 LTS; the server machine has Intel Xeon E5-1620 v3, 16 GiB RAM, and is also running Ubuntu 22.04 LTS; intruder machine has Intel i5-6500, 8 GiB RAM, and is running Kali Rolling (kernel version 6.10.9-amd64). In Figure 5, we visualize the operational setup for our use case.

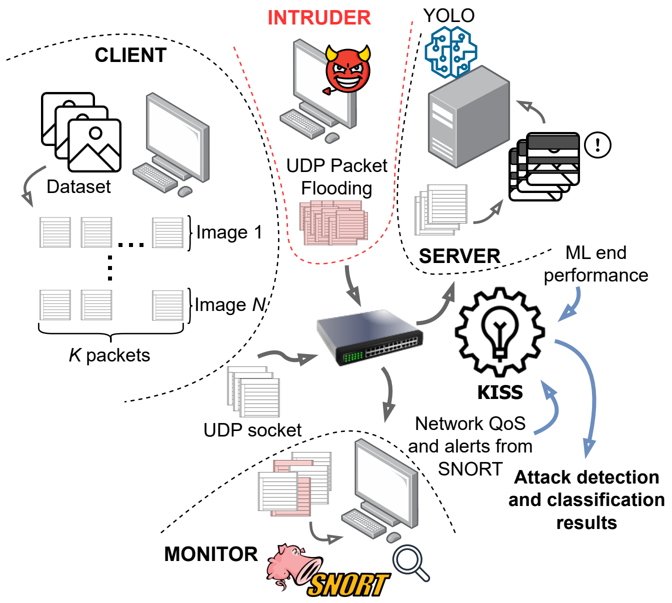

Fig. 5. Use case 1 operational setup: KISS integration with other MLIN components

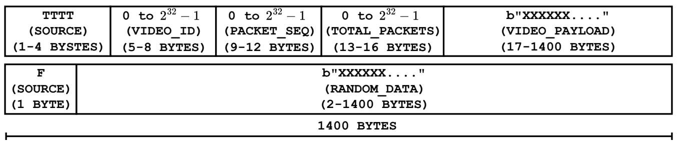
TTTT (SOURCE) (1-4 BYSTES)
0 to $2^{32}-1$ (VIDEO_ID) (5-8 BYTES)
0 to $2^{32}-1$ (PACKET_SEQ) (9-12 BYTES)
0 to $2^{32}-1$ (TOTAL_PACKETS) (13-16 BYTES)
b"XXXXXX...." (VIDEO_PAYLOAD) (17-1400 BYTES)
F (SOURCE) (1 BYTE)
b"XXXXXX...." (RANDOM_DATA) (2-1400 BYTES)
1400 BYTES

Fig. 6. Packet structure for client packets (top) and intruder packets (bottom)

The client machine stores and sends data to the server, while the monitor machine, connected to the switch, uses SNORT IDS and *tcpdump* to inspect all mirrored traffic from the client, intruder, and server. The server's network performance, tested with *iperf3*, handles approximately 1Gbps of UDP traffic. To quantify packet loss and ensure correct packet order, we developed a custom UDP protocol (Figure 6) with 1400-byte packets, below the default 1500-byte MTU, preventing fragmentation. This ensured accurate data transmission rate measurements, video reconstruction, and SNORT inspection, which avoided false positives caused by fragmented packets. The packet's SOURCE field is used to verify packet authenticity.

B. Employed Data and ML Model

We employed the BDD110K dataset used in our experiments in sec. III-A2. From the dataset, we manually selected 34 videos that contain various object, such as cars, traffic signs, and pedestrians. The videos are encoded in H.264 and have the length of approximately 10 seconds. All the 34 videos are first processed to create a single file containing the packets for all the videos. By doing so, we have better control over the packet transfer rate required for the experiment for recreating the data losses scenario due to packet flooding. This packet file is transmitted using UDP socket from client to the server.

The server receives the packets and saves them in a single packet file. This packet file is then processed on the server and all 34 videos are reconstructed ignoring missing packets if any.

The server is responsible for receiving the data and processing it with the YOLOv8n model pre-trained on COCO dataset, running in the inference mode. The frames are loaded from the input video file through the OpenCV functionality called VideoCapture. In this step, the dimension of the frame and the total count of the frames are retrieved while making sure every frame is returned sequentially. It skips if any frame fails to load. The YOLO model takes each and every frame independently. When the frame feed in, the model releases the results through performing inferences that detect objects and return bounding boxes, class labels, and confidence scores.

C. Attack Model

For A-Attacks targeting the ML system, we consider an adversary with limited capabilities who has gained access to client-stored data. This adversary aims to degrade ML performance while remaining undetected, lacking advanced knowledge of A-Attacks or the specific ML model used. They employ random noise addition, a universal attack method applicable across various data modalities and domains, requiring minimal expertise. Our evaluation, as shown in Table V, demonstrates that even this simple attack can effectively degrade ML performance, with the degree of degradation dependent on the intensity of the introduced noise. For B-Attacks, we assume an attacker has bypassed initial security and gained full access to an intruder machine on the corporate network, but possesses no knowledge of the ML system or data transmitted. The attacker's objective is to disrupt network functionality, affecting business processes, such as delaying data processing to harm reputation and cause financial losses. We simulate network flooding attacks of varying intensity using *hping3* to launch UDP packet floods from the intruder machine to the server. Due to *hping3*'s limitations, we specify packet intervals in microseconds, resulting in average attack rates of 945, 965, and 995 Mbps for video transmissions.

D. Evaluation Metrics

After processing each frame in the video, YOLO returns all the detected objects, object class, confidence score, and bounding box coordinates for each of the objects. As we have 34 videos that may contain various number of objects and object classes in the footage, we need to employ high-level metrics to evaluate the ML end performance across the overall data. We use three metrics: confidence score ratio (CSR), detection ratio (DR), and weighted confidence score (WCS). CSR quantifies the consistency of YOLO's detection confidence across frames by comparing the baseline performance with the cases when malicious attacks present. This metric is expressed in (1) and calculates the ratio of average confidence scores frame by frame, comparing the original video to the corrupted version.

$$\text{CSR} = \frac{\frac{1}{n}\sum_{i=1}^{n} C_i^{\text{attacked}}}{\frac{1}{n}\sum_{i=1}^{n} C_i^{\text{original}}}, \tag{1}$$

where C_i^{attacked} is the confidence score in the attacked frame i; C_i^{original} is the confidence score in the original frame i; and n is the number of confidence score outputs per each detection in the i'th frame.

DR demonstrates the difference in the number of detected objects between the original and attacked videos with respect to how YOLO consistently detects objects as packet loss changes. A drop in the DR when the attacks are present might refer to the case when YOLO misses the objects completely or shows lower detection performance in the corrupted data. This metric is used to determine if the number of consistent object detections is affected by the attack tested and flags frames where their count of detection differs from the baseline performance. Equation (2) is used to calculate DR.

$$\text{DR} = \frac{D_i^{\text{attacked}}}{D_i^{\text{original}}}, \tag{2}$$

where D_i^{attacked} is the number of detections in the attacked frame i; D_i^{original} is the number of detections in the original frame i.

The WCS metric incorporates detection frequency with the confidence score given by YOLO to provide clear indication regarding the reliability of detection in the case of variable attack intensity. This metric underscores the frames that have higher detection frequencies (more object detections per frame) to point out the ones on which the confidence level cannot be relied under an attack condition. The weighted confidence for each frame is calculated according to (3).

$$\text{WCS} = \frac{\frac{1}{n}\sum_{i=1}^{n} C_i^{\text{corrupted}}}{D_{\text{original}}} D_{\text{corrupted}}, \tag{3}$$

where $C_i^{\text{corrupted}}$ is the confidence score in the corrupted frame i; D_{original} is the average number of detections in the original frames; $D_{\text{corrupted}}$ is the average number of detections in the corrupted frames.

E. Threshold Selection

Threshold selection is a fundamental challenge in any security application, as it directly impacts detection performance. In our work, we determine these thresholds relevant for our use cases, but they can be fine-tuned based on specific data characteristics and application requirements. For CSR, DR, and WCS, we determine thresholds through analyzing experimental results on these metrics across all 34 videos. Specifically, for CSR and DR, ratios below 0.99 and above 1.01 consistently correspond to attack regions, as they indicate deviations caused by malicious interference in the transmitted data. These boundaries are established after analyzing the distribution of CSR and DR values in both normal and attack scenarios, ensuring they capture attack-induced variations while minimizing false positives. Similarly, for WCS, we set the threshold at $\mu_{wcs}^{i} \cdot 1.1$, where μ_{wcs}^{i} represents the mean WCS value for video i. This choice is guided by an empirical analysis of WCS fluctuations across frames, allowing us to effectively detect subtle attack-induced changes without introducing excessive false alarms.

F. Experimental Flow

For baseline evaluation, original videos were transmitted without malicious intervention to assess YOLO model performance under normal conditions. To simulate B-Attacks, UDP flooding attacks with varying intensities (945, 965, 995 Mbps) were launched against the server during video transmission; the reconstructed video streams were then processed by the YOLO model, and network packets were captured by *tcpdump* and analyzed by SNORT, with the client's rate maintained at 50 Mbps. To recreate A-Attacks, Gaussian noise (with $\sigma = 1$ and $\mu = 0$) was added to the videos, which were then transmitted without flooding; network packets were again captured and analyzed by *tcpdump* and SNORT, but the client's packet sending rate was increased to 300 Mbps to accommodate the larger video size resulting from the added noise.

It is important to consider the experimental approach of evaluating B- and A-Attacks detection in our study. In real-world operational systems, malicious attacks represent relatively infrequent occurrences, with the vast majority of system operation occurring under normal, non-attacked conditions [38], [24]. Furthermore, the simultaneous occurrence of distinct attack types, such as a network flood and adversarial content manipulation targeting the same data stream at precisely the same moment, constitutes an improbable event. Even in such an unlikely scenario, our framework is designed to detect anomalous behavior and classify it based on the dominant attack characteristics present. Further in-depth security investigation and forensic analysis can be triggered after the detection to ascertain the precise nature of the incident, potentially revealing the confluence of multiple attack vectors. Established security practices and incident response protocols are typically based on detecting and classifying anomalies, followed by a detailed investigation to fully characterize complex security events [30]. The design of our empirical study validates the core capability of our framework: the accurate differentiation between infrastructure-level anomalies (B-Attacks) and adversarial data manipulations (A-Attacks) when these events occur independently. This provides validation of KISS framework's feasibility and its capacity to effectively distinguish between these attack types in practice.

VI. Use Case 1: KISS Prototype Verification Results

To evaluate the impact of different attack types and intensities, our framework classified the considered attacks into two categories: B-Attacks and A-Attacks. This classification was determined based on the analysis of our established metrics and thresholds, and further integrating it with SNORT outputs. B-Attacks were implemented by introducing packet loss to induce network congestion and data corruption, while A-Attacks were implemented by adding noise to the transmitted video stream to assess its impact. For the evaluated B-Attack intensities of 945, 965, and 995 Mbps (94.5%, 96.5%, and 99.5% of the communication bandwidth respectively), the packet loss in the network resulted in 1.6%, 3.28%, and 16.86% respectively. Due to space limitations, we present detailed metric analysis for *video_2* only, with comprehensive results (data in *.csv*) and plots for all 34 videos available in the attached materials[3].

Based on the analysis of CSR metric, shown in Figures 7(a), 7(b), and 7(c) for *video_2* and consistently observed across all videos, a clear trend emerged where ML end performance deteriorates as B-Attack intensity increases. At 1.6% packet loss, CSR degradation becomes noticeable around frame 180, fluctuating between 0.8 and 1.0 in non-attacked regions but dropping significantly in attacked frames. With 3.28% packet loss, CSR degrades earlier and frequently falls below 0.5, even within the first 100 frames. Severe degradation is evident at 16.86% packet loss, where CSR remains consistently below 0.2 for almost the entire video. Similarly, DR metric analysis, depicted in Figures 7(d), 7(e), and 7(f) for *video_2* under these attacks. At 1.6% packet loss, DR remains stable around 1.0 until approximately frame 180 before degradation, while at higher packet loss intensities, DR shows significant drops and instability. WCS metric results, shown in Figures 7(g), 7(h), and 7(i) for *video_2* demonstrate drop in ML performance for the attack regions as well. For 1.6% packet loss, WCS begins around 0.6 and drops after frame 180. Higher variations and significant drops in WCS can be observed at 3.28% and 16.86% packet loss intensities.

Complementing the performance metric analysis, SNORT, employed as an IDS, logged a notable increase in alerts in the attack regions, as shown in Figure 7(l) (values for 94.5%, 96.5%, and 99.5% avg. attack rate). Specifically, SNORT flagged approximately 0.1% of packets as malicious in the absence of attacks, while this percentage increased to 0.3% in the attack regions. By combining the observed ML performance degradation with the simultaneous increase in SNORT alerts, our KISS framework effectively classified these attacks as B-Attacks, indicative of infrastructure-level attacks targeting the network.

Figure 7(j) presents the average confidence score per frame for the original and A-Attacked video, using *video_2* as a representative example. While the original video's performance fluctuates between 0.5 and 0.8, the A-Attacked video exhibits consistently lower confidence scores, generally ranging from 0.4 to 0.7. CSR comparisons for A-Attacks, depicted in Figure 7(k), illustrate that for most frames, CSR remains within 0.8 to 1.0. This suggests that while noise generally impacts ML end performance, its effect, as captured by our metrics, is distinct from the severe disruptions characterized by B-Attacks. However, occasional significant performance drops are noticeable after frame 400 and towards the end of the video. Notably, SNORT logs for A-Attacks, as shown in Figure 7(l) (value for 0% avg. attacks rate), revealed significantly lower alert levels, around 0.04% of packets flagged as malicious. This alert rate is even lower than that observed for original, non-attacked videos, confirming that SNORT, operating with community rules focused on network traffic inspection, is not able to detect A-Attacks at all if used as a standalone security measure. This discrepancy in SNORT

[3]https://drive.google.com/drive/folders/1uJFGmwmjyof_KyQd_-UzW9tQ5_QYHo20?usp=drive_link

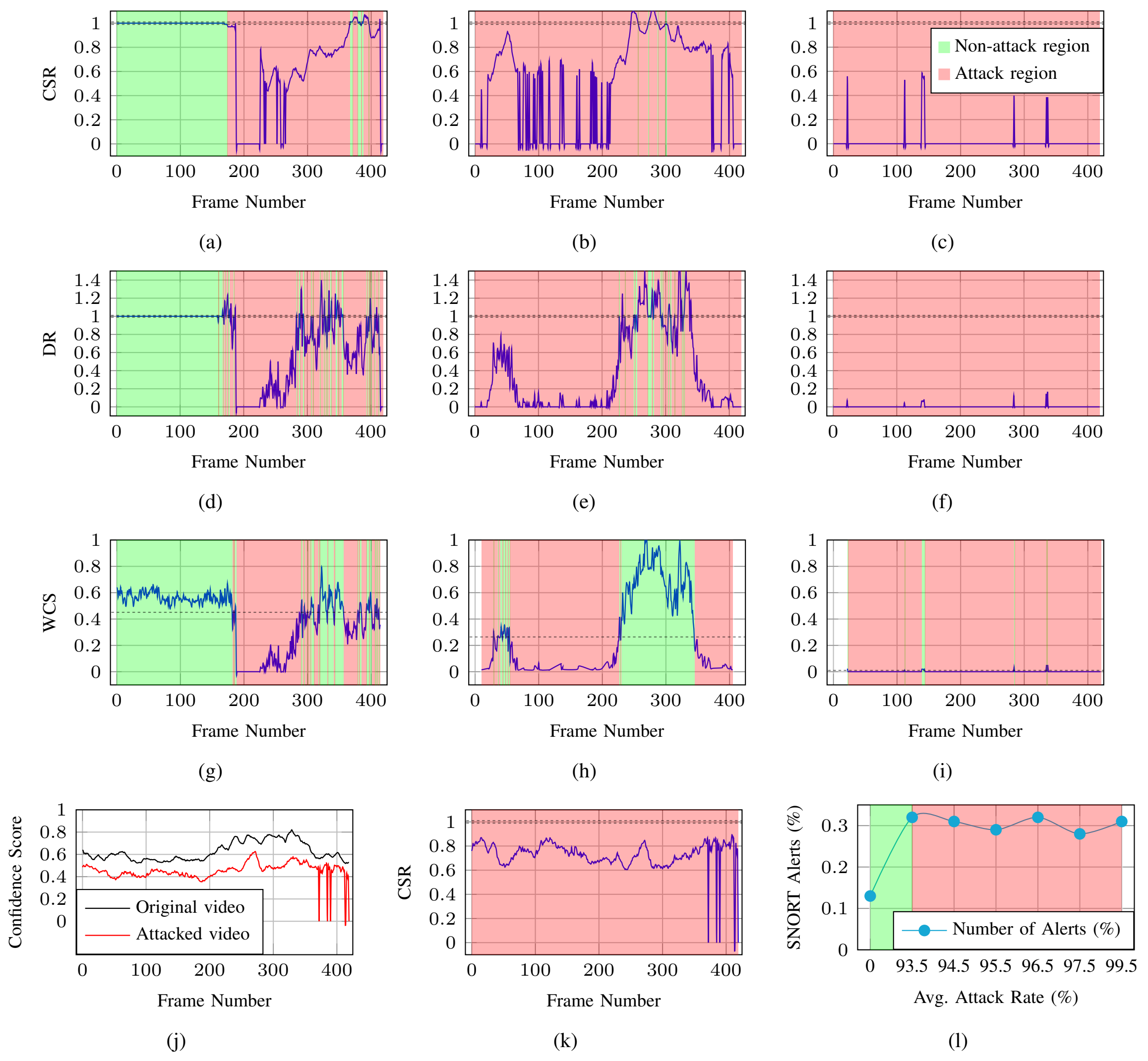

Fig. 7. Experimental results for one of the tested videos (video 2): (a) – CSR for 1.6% packet loss; (b) – CSR for 3.28% packet loss; (c) – CSR for 16.86% packet loss; (d) – DR for 1.6% packet loss; (e) – DR for 3.28% packet loss; (f) – DR for 16.86% packet loss; (g) – WCS for 1.6% packet loss; (h) – WCS for 3.28% packet loss; (i) – WCS for 16.86% packet loss; (j) – average confidence score per frame for the original video and video affected by A-Attack; (k) – CSR for the A-Attack; (l) – percentage of number of alerts issued by SNORT in relation to the average flooding attack rate. The dotted lines in (a) – (i), and (k) represent the threshold values, described in sec. V-D, for making a decision if the frame is affected by an attack

TABLE VI
SUMMARY OF EVALUATION METRICS ACROSS ALL TRANSMITTED VIDEOS

Attack	**Num. of Frames**		**Corrupted Frames**		**Num. of Detections**		**Confidence Score**		**WCS**		**DR**	
	Mean	**S.D.**	**Mean**	**S.D.**	**Mean**	**S.D.**	**Mean**	**S.D.**	**Mean**	**S.D.**	**Mean**	**S.D.**
Baseline (No Attack)	324.03	77.23	0	0	3218.84	1538.23	0.55	0.05	5.36	1.46	9.75	2.71
B-Attack (1.60%)	312.00	91.12	12.03	44.26	**2617.84**	**1562.13**	**0.52**	**0.05**	**4.34**	**1.65**	**8.26**	**3.00**
B-Attack (3.28%)	313.03	75.41	10.00	7.43	1280.73	980.08	0.48	0.05	2.04	1.51	4.04	2.74
B-Attack (16.86%)	**246.09**	**42.53**	**77.93**	**104.55**	**79.96**	**119.89**	**0.37**	**0.05**	**0.14**	**0.23**	**0.32**	**0.47**
A-Attack	**324.57**	**77.27**	**0**	**1.71**	750.00	854.43	0.50	0.06	1.07	0.86	2.10	1.72

alert levels, integrated with the observed ML performance degradation, allowed our framework to classify these attacks as A-Attacks targeting the ML end system rather than network infrastructure.

Across all 34 videos, both A- and B-Attacks degraded ML performance, with higher attack intensity worsening results. Table VII summarizes the **Mean** and standard deviation (**S.D.**) for key metrics, aggregated for all 34 videos, where values in green represent the best performance results and values in red the worst (excluding the baseline). B-Attacks caused highly dynamic, near-zero performance drops, reflecting the disruptive nature of packet loss. A-Attacks resulted in consistent, approximately 30% performance degradation, maintaining a similar trend to the original video. Integrating this metric analysis with SNORT alert patterns enabled accurate classification of all attack scenarios into B- and A-Attacks.

TABLE VII
SUMMARY OF NETWORK METRICS AND ALERTS ISSUED BY SNORT

Attack	PL* (%)	P. Analyzed†	Alerts (#)	Alerts (%)‡
Baseline (No Attack)	0	157840	2	0.13
A-Attack	0	3151939	13	0.04
B-Attack (935 Mbps)	0.9	3826997	123	0.32
B-Attack (945 Mbps)	1.6	3817936	117	0.31
B-Attack (955 Mbps)	2.7	3826706	111	0.29
B-Attack (965 Mbps)	3.28	3827493	122	0.32
B-Attack (975 Mbps)	5.83	3827497	118	0.31
B-Attack (985 Mbps)	6.35	3826655	107	0.28
B-Attack (995 Mbps)	16.86	3826825	119	0.31

* Packet Loss; † Packets Analized by SNORT; ‡ Alerts % w.r.t. Packets Analized

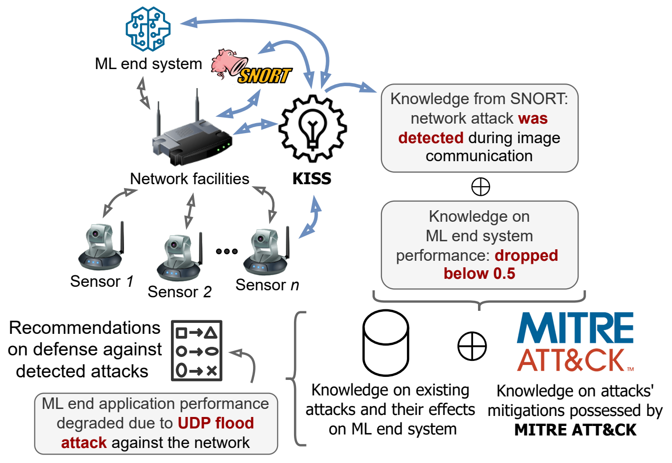
ML end system
SNORT
Network facilities
KISS
Sensor 1
Sensor 2
Sensor n
Knowledge from SNORT: network attack was detected during image communication
⊕
Knowledge on ML end system performance: dropped below 0.5
Recommendations on defense against detected attacks
ML end application performance degraded due to UDP flood attack against the network
Knowledge on existing attacks and their effects on ML end system
⊕
MITRE ATT&CK
Knowledge on attacks' mitigations possessed by MITRE ATT&CK

Fig. 8. Use case 2 operational setup: knowledge integration in KISS from existing security database

To assess whether the observed SNORT pattern can be generalized to a wider range of B-Attack scenarios, we performed additional runs with multiple loads for B-Attacks with attack intensities ranging from 935 to 995 Mbps. The results obtained demonstrate that the number (and the corresponding percentage) of alerts produced by SNORT is much higher than in baseline and A-Attack cases, which allows the framework to integrate this knowledge to correctly classify the attack. As summarized in Table VII, these additional runs consistently showed a significantly higher number of alerts for B-Attacks compared to the baseline and A-Attack.

VII. USE CASE 2: INTEGRATION OF KNOWLEDGE WITH EXISTING SECURITY DATABASES

In the previous section we demonstrated how KISS is leveraged to integrate the knowledge amongst CI components by combining the IDS logging information with the ML end performance. Here we demonstrate how we can utilize existing security databases to provide a more fine-grained classification of potential attacks and derive recommendations on the available defenses. Figure 8 captures our setup for this use case. We make the code and the description of our implementation and experiments available for public[4].

[4]https://drive.google.com/drive/folders/1g0wFk6eyCshik5n5ZF4wQMp7iRrmE45K?usp=drive_link

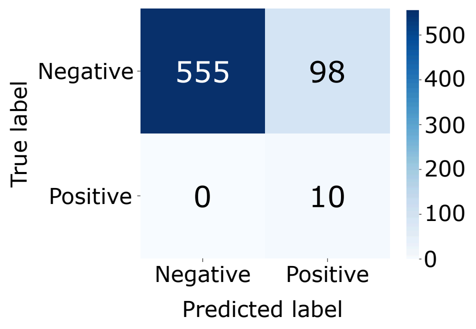

Fig. 9. Attack detection matrix for the case of KISS integration with MITRE ATT&CK. KISS was able to achieve accuracy of **85.2%** and recall of **100%** in detecting CI attacks

IT industry leaders host large security databases to protect their infrastructure. Our design targets the integration with existing security databases. In this case, we employ MITRE ATT&CK[5], which is an open-source knowledge base with API access that provides vast information on different attack signatures and can help KISS provide a more fine-grained classification of attacks, and recommendations on the defense actions against these attacks. To showcase how KISS can be combined with an existing security database, we employ a smaller traffic sign dataset containing 10 A-Attacked images, 15 B-Attacked images. Both attacks are performed in the same manner as described in sec. V, i.e., we use adversarial noise for A-Attacks and flooding for B-Attacks. The total size of the dataset is 663 images. The entire dataset is transmitted from the client to the server over the network and processed by YOLO, and the images affected by noise cause a drop in the confidence scores.

In the image communication process, 26 SNORT security alerts were extracted by KISS and matched in MITRE ATT&CK using API. Seven recommendations on mitigation actions (overlapping for all 26 records) were returned from MITRE ATT&CK to KISS based on the attacks detected and classified. Figure 9 presents a confusion matrix of attack classification. In a complex scenario where both A-Attacks and B-Attacks were present, KISS successfully detected 100% of A-Attacks and achieved an accuracy of 85.2%, recall of 100%, and a relatively low false positive rate of 14.8%. Listing 1 presents the output example on the recommendation actions generated by KISS based on the response received from MITRE ATT&CK after running the experiment. By integrating the existing security database with KISS, it was able to provide a more fine-grained classification of attacks, and to derive the actions aimed at defending or mitigating the attacks consequences. For instance, for the detected flooding attacks, KISS provided a fine-grained description of each attack based on the information from MITRE ATT&CK and matched attack records.

Listing 1. Snapshot of attack detection and recommended mitigations provided by KISS. 7 recommendations on mitigation actions (overlapping for 26 records) were returned from MITRE ATT&CK.

```
Match found: Original Type: Possible ICMP Echo Request Scan, Dictionary Key: ICMP
      ECHO REQUEST SCAN, Pattern ID: attack-pattern--67873dde-d728-45ae-83da-
      b12d5e73ca3b
```

[5]https://attack.mitre.org/

```
Match found: Original Type: Possible TCP SYN Scan, Dictionary Key: TCP SYN SCAN,
       Pattern ID: attack-pattern--e3a12395-188d-4851-9a16-ea8e1d4b78b8
Match found: Original Type: Possible UDP Scan, Dictionary Key: UDP SCAN, Pattern ID
       : attack-pattern--e3a12395-188d-4851-9a16-ea8e1d4b78b8

Mitigations mitigating attack-pattern--67873dde-d728-45ae-83da-b12d5e73ca3b (1):
* Pre-compromise (M1056)

Mitigations mitigating attack-pattern--e3a12395-188d-4851-9a16-ea8e1d4b78b8 (3):
* Disable or Remove Feature or Program (M1042)
* Network Intrusion Prevention (M1031)
* Network Segmentation (M1030)

Mitigations mitigating attack-pattern--e3a12395-188d-4851-9a16-ea8e1d4b78b8 (3):
* Disable or Remove Feature or Program (M1042)
* Network Intrusion Prevention (M1031)
* Network Segmentation (M1030)
```

VIII. KISS Limitations

The limitations of KISS can be categorized into architectural and implementation-based aspects. We highlight these aspects below.

(1) KISS relies on external security mechanisms (e.g., IDS/IPS, malware detection), networking components (e.g., network QoS monitors), and ML end systems (e.g., ML models and their performance monitoring tools), making its effectiveness dependent on the availability and accessibility of these tools, requiring proper interfaces for seamless data exchange. While KISS requires certain computational resources, it activates only when performance degradation is detected, minimizing its resource footprint.

(2) The framework currently focuses on image and video data for object detection, limiting its generalizability. While the methodology could be extended to other domains, adaptation and validation would be necessary.

(3) Practical implementation challenges include integration complexity, user training, and adoption costs. Integrating KISS into existing security and ML pipelines may require modifications to accommodate various system architectures and computational constraints. Additionally, user training is essential for security analysts and ML practitioners to correctly interpret KISS outputs, which could present adoption barriers. To address this limitation and make our framework more accessible, we are sharing it as an open-source solution and providing instructions on how to run and use it.

(4) The current design of the KISS framework is not explicitly engineered to detect and classify scenarios where both A- and B-Attacks are simultaneously active, targeting the same data stream at a specific point in time during communication. While KISS is capable of detecting anomalies under such conditions, its classification accuracy in distinguishing between or attributing performance degradation to concurrently occurring attack types may be limited.

(5) Depending on the use case and data, the appropriate threshold values need to be selected for the metrics integrated by KISS for the attack classification (e.g., CSR, DR, and WCS). While in our case thresholds were empirically determined through the analysis of experimental data across 34 videos, in general case the values are data-dependent and may require fine-tuning for different datasets, application contexts, and operational environments.

IX. Conclusion

In this paper, we developed and presented KISS – service to be incorporated into CI of modern MLIN systems and combined with the existing security mechanisms to better detect and classify malicious attacks. KISS extracts knowledge from MLIN CI component tools, e.g. ML end system performance monitors and IDS, as well as from other specialized data and knowledge bases on particular tools operation in order to detect attacks leading to ML end performance degradation. Based on the integrated knowledge, these attacks are classified into adversarial A-Attacks against ML end systems, and B-Attacks against CI. KISS is designed to complement existing security and system monitoring tools in MLIN. Unlike other mechanisms KISS utilizes ML end system performance as the major metric for detecting attacks against MLIN and integrated knowledge to classify them. We verified our design and demonstrated KISS knowledge and service integration with existing tools in two use cases: in the first we showed how the knowledge can be extracted and integrated from MLIN CI components in-place; in the second we investigated how the existing security knowledge bases, such as MITRE ATT&CK, can be integrated with KISS for obtaining recommendations on the proper defense actions. While the general performance and correctness of KISS depends on the existing MLIN CI components in-place, in our study KISS was able to detect and classify correctly all the attacks tested.

Acknowledgment

We want to appreciate indispensable help with the experimental part of this paper provided to us by Abhinav Ajit Menon and Jheel Nikulkumar Patel, 2024 MS students in Computer Science at Rochester Institute of Technology, NY.

This research was supported in part by the National Science Foundation USA (awards #2321652 and #2415299).

References

[1] M. Abbasi, A. Rajabi, C. Gagné, and R. B. Bobba, "Toward adversarial robustness by diversity in an ensemble of specialized deep neural networks," in *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33*. Springer, 2020, pp. 1–14.

[2] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[3] M. Aqqa, P. Mantini, and S. K. Shah, "Understanding how video quality affects object detection algorithms," in *14th International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applicaitons (VISIGRAPP 2019)*. SCITEPRESS, 2019, pp. 96–104.

[4] M. Azees, P. Vijayakumar, and L. J. Deborah, "Comprehensive survey on security services in vehicular ad-hoc networks," *IET Intelligent Transport Systems*, vol. 10, no. 6, pp. 379–388, 2016.

[5] S. Baidya and M. Levorato, "Content-based cognitive interference control for city monitoring applications in the urban iot," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.

[6] A. Bala, M. Bansal, and J. Singh, "Performance analysis of manet under blackhole attack," in *2009 First International Conference on Networks & Communications*. IEEE, 2009, pp. 141–145.

[7] J. S. Berg, *Broadcasting on the Short Waves, 1945 to today*. McFarland, 2008.

[8] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–33, 2021.

[9] G. A. Blog. (2020) Overview of Open Images V6. (Date last accessed 30-April-2022). [Online]. Available: https://storage.googleapis.com/openimages/web/factsfigures.html

[10] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, A. Ghosh, M. Hibler, D. Johnson, S. K. Kasera, E. Lewis *et al.*, "Powder: Platform for open wireless data-driven experimental research," *Computer Networks*, p. 108281, 2021.

[11] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.
[12] S. Chen, X. Huang, Z. He, and C. Sun, "Damagenet: a universal adversarial dataset," *arXiv preprint arXiv:1912.07160*, 2019.
[13] V. Cherepanova, M. Goldblum, H. Foley, S. Duan, J. Dickerson, G. Taylor, and T. Goldstein, "Lowkey: leveraging adversarial attacks to protect social media users from facial recognition," *arXiv preprint arXiv:2101.07922*, 2021.
[14] S. Chuprov, L. Reznik, and G. Grigoryan, "Study on network importance for ml end application robustness," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 6627–6632.
[15] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.
[16] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4312–4321.
[17] M. Fan, Y. Liu, C. Chen, S. Yu, W. Guo, L. Wang, and X. Liu, "Towards evaluating the reliability of deep neural networks based iot devices," *IEEE Internet of Things Journal*, 2021.
[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
[19] L. Huraj, M. Simon, and T. Horák, "Iot measuring of udp-based distributed reflective dos attack," in *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 2018, pp. 000 209–000 214.
[20] I. Ilahi, M. Usama, J. Qadir, M. U. Janjua, A. Al-Fuqaha, D. T. Hoang, and D. Niyato, "Challenges and countermeasures for adversarial attacks on deep reinforcement learning," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 90–109, 2021.
[21] V. Krundyshev, M. Kalinin, and P. Zegzhda, "Artificial swarm algorithm for vanet protection against routing attacks," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2018, pp. 795–800.
[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
[23] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.
[24] A. Naha, A. Teixeira, A. Ahlén, and S. Dey, "Deception attack detection using reduced watermarking," in *2021 European control conference (ECC)*. IEEE, 2021, pp. 74–80.
[25] E. C. Ngai, J. Liu, and M. R. Lyu, "On the intruder detection for sinkhole attack in wireless sensor networks," in *2006 IEEE International Conference on Communications*, vol. 8. IEEE, 2006, pp. 3383–3389.
[26] M. Nooribakhsh and M. Mollamotalebi, "A review on statistical approaches for anomaly detection in ddos attacks," *Information Security Journal: A Global Perspective*, vol. 29, no. 3, pp. 118–133, 2020.
[27] S. Pandey, P. R. Singh, and J. Tian, "An image augmentation approach using two-stage generative adversarial network for nuclei image segmentation," *Biomedical Signal Processing and Control*, vol. 57, p. 101782, 2020.
[28] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.
[30] L. Reznik, *Intelligent Security Systems: How Artificial Intelligence, Machine Learning and Data Science Work for and Against Computer Security*. John Wiley & Sons - IEEE Press: Hoboken, NJ, USA, 2022.
[31] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in iot: a survey," *The Journal of Supercomputing*, pp. 1–44, 2019.
[32] J.-P. Schulze, P. Sperl, and K. Böttinger, "Da3g: Detecting adversarial attacks by analysing gradients," in *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26*. Springer, 2021, pp. 563–583.
[33] I. Shumailov, Y. Zhao, D. Bates, N. Papernot, R. Mullins, and R. Anderson, "Sponge examples: Energy-latency attacks on neural networks," in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2021, pp. 212–231.
[34] I. Shumailov, Y. Zhao, R. Mullins, and R. Anderson, "The taboo trap: Behavioural detection of adversarial samples," *arXiv preprint arXiv:1811.07375*, 2018.
[35] ——, "Towards certifiable adversarial sample detection," in *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*, 2020, pp. 13–24.
[36] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.
[37] X. Wang, X. He, J. Wang, and K. He, "Admix: Enhancing the transferability of adversarial attacks," *arXiv preprint arXiv:2102.00436*, 2021.
[38] X. Wang, "On the feasibility of real-time cyber attack attribution on the internet," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 289–294.
[39] R. Wiyatno and A. Xu, "Maximal jacobian-based saliency map attack," *arXiv preprint arXiv:1808.07945*, 2018.
[40] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.
[41] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 151–178, 2020.

X. Biography Section

Dr. Sergei Chuprov is an Assistant Professor in the Department of Computer Science at the University of Texas Rio Grande Valley, USA. His current research interests include enhancing security and robustness of intelligent systems, with a focus on machine learning and data quality, particularly in federated learning. He holds a PhD in Computing and Information Sciences from Rochester Institute of Technology (2024), an MS in Information Security (with honors, 2019), and a BS in Information Security (2017), both from ITMO University.

Dr. Leon Reznik is a Professor of Computer Science and the member of the ESL Global Cybersecurity Institute at the Rochester Institute of Technology, New York, USA. He conducts research and teaches classes in the areas of ML and AI with their applications in cybersecurity, computer vision, intelligent control systems, sensor systems and networks. His new textbook "Intelligent security systems: How artificial intelligence, machine learning and data science work for and against computer security" was recently released by IEEE Press - Wiley and Sons. Prof. Reznik authored another textbook and almost two hundred papers in these fields as well as edited a few research volumes.

Raman Zatsarenko is a PhD student in the Lab of Data Quality and Intelligent Security at Rochester Institute of Technology, New York, USA. His research interests include anomaly detection in federated learning with applications in machine learning security and science. He holds a BS in Computer Science (2023) from Rochester Institute of Technology.