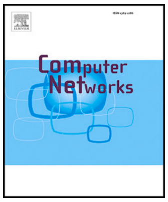
Computer Networks 250 (2024) 110576

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Machine learning-enabled hybrid intrusion detection system with host data transformation and an advanced two-stage classifier

Zhiyan Chen [a], Murat Simsek [a], Burak Kantarci [a,*], Mehran Bagheri [b], Petar Djukic [c]

[a] *University of Ottawa, 800 King Edward Avenue, Ottawa, ON, Canada, K1N 6N5*
[b] *Ciena, 5050 Innovation Drive Ottawa, ON, K2K 0J2, Canada*
[c] *Nokia Bell Labs, 600 March Rd, Kanata, ON, Canada K2K 2T6*

ARTICLE INFO

Keywords:
Machine learning
Network security
Network intrusion detection
Hybrid intrusion detection
Feature flattening
Collaborative classifier

ABSTRACT

Network Intrusion Detection Systems (NIDS) have been extensively investigated by monitoring real network traffic and analyzing suspicious activities. However, there are limitations in detecting specific types of attacks with NIDS, such as Advanced Persistent Threats (APT). Additionally, NIDS is restricted in observing complete traffic information due to encrypted traffic or a lack of authority. To address these limitations, a Host-based Intrusion Detection system (HIDS) evaluates resources in the host, including logs, files, and folders, to identify APT attacks that routinely inject malicious files into victimized nodes. In this study, a hybrid network intrusion detection system that combines NIDS and HIDS is proposed to improve intrusion detection performance. The host data undergoes a Language Processing (NLP)-based Bidirectional Encoder Representations from Transformers (BERT) model from textual representation to a numerical one in order to process host data in a similar way to the network flow data through machine learning models. The feature flattening technique is applied to flatten two-dimensional host-based features that is provided by BERT into one-dimensional vectors so that host-based and network flow-based features can be processed by advanced Machine Learning (ML) models. In order to enhance HIDS effectiveness, a two-stage collaborative classifier is utilized, which applies two tiers of machine learning algorithms, binary and multi-class classifiers, to detect network intrusions. Once a binary classifier is used to detect benign samples to reduce the complexity of the original problem, the attack data are classified by a multi-class supervised learner to identify attack types. Hence, the overall performance of the two-stage collaborative model outperforms the baseline classifier, XGBoost. The proposed method is shown to generalize across two well-known datasets, CICIDS 2018 and NDSec-1. The performance of XGBoost, which represents conventional ML, is evaluated. Combining host and network features enhances attack detection performance (macro average F1 score) by 8.1% under the CICIDS 2018 dataset and 3.7% under the NDSec-1 dataset. Meanwhile, the two-stage collaborative classifier improves detection performance for most single classes, especially for DoS-LOIC-UDP and DoS-SlowHTTPTest, with improvements of 30.7% and 84.3%, respectively, when compared with the traditional ML models.

1. Introduction

The importance of security solutions for networked systems has increased with the advances in information and communication technologies [1,2]. In the area of safeguarding network systems, considerable research and implementation efforts have been dedicated to the exploration of Network Intrusion Detection Systems (NIDSs [3–5]. ML-based NIDSs have been proven to detect prevalent and zero-day attacks, which have also been studied in the context of defensive and proactive/adversarial ML [6].

NIDS identifies suspicious activities by analyzing data from a single packet in captured global network traffic [7]. However, monitoring entire network traffic can be challenging in scenarios such as encrypted traffic or without authentication monitoring. NIDS cannot inspect encrypted network traffic, limiting its ability to detect only external attacks [8]. Moreover, NIDS requires improved solutions to identify specific malicious activities, particularly APT, which often utilize malicious files attached to various applications [9]. Host-based intrusion detection evaluates resources in a host, including logs, files, and folders, to detect attacks on hosts such as servers. HIDS provides a fine-grained solution to detect anomalous patterns internally, making it a valuable tool in detecting APT [10]. HIDS also has the advantage of detecting anomalies without analyzing or monitoring network traffic. As an

[*] Corresponding author.
E-mail address: burak.kantarci@uottawa.ca (B. Kantarci).

https://doi.org/10.1016/j.comnet.2024.110576
Received 14 November 2023; Received in revised form 7 April 2024; Accepted 5 June 2024
Available online 7 June 2024

example, [11] presents a HIDS use case for securing Android mobile equipment. HIDSs are software components installed on observed systems, and they scan the entire system to prevent intrusions. HIDSs offer rich context, enabling excellent knowledge for data processing and analysis [12]. Although HIDSs increase costs due to their connectivity to the server, setup of distributed clients, and collection and management of massive and sensitive data from host devices, they have recently gained attention from researchers. Industrial companies often implement both NIDS and HIDS to achieve promising detection performance and secure their systems [13].

In this work, we investigate intrusion detection system combining NIDS and HIDS via utilizing network and host features together. Various public datasets, such as NSL KDD, KDD 99, Bot-IoT, and MQTTTest, are utilized in network intrusion detection research [1]. However, there are only a limited number of public datasets that include host information. The CICIDS 2018 dataset [14,15] and NDSec-1 dataset [16], which incorporate host-based information, including logs detailing events and messages, have been employed in various intrusion detection schemes. Therefore, we selected the two datasets to demonstrate the method integrating network and host features and evaluate the proposed method. Despite the potential benefits of host-based information in intrusion detection, very few studies have applied hybrid host features and flow features together for intrusion detection via machine learning or deep learning models. The authors in [17] introduce a deep learning model named CIDS-Net to detect intrusions using hybrid host and flow features. In this work, we propose machine learning-enabled intrusion detection system and an advanced two-stage classifier for network intrusion detection with hybrid host and flow data. Furthermore, the literature demonstrates dataset pre-processing in details for hybrid features, including host feature selection and host feature flattening. Host-based content is stored as a string and transformed into a numeric array by BERT [18,19]. Event data and message data stored in the host are transformed separately to obtain host-based features. Moreover, the transformed host features are applied data pre-processing techniques to reduce the size of host features and finally are flattened from two-dimensional into a vector. The following step is to combine network/flow features[1] and host features and send to detection model, including traditional ML and the two-stage collaborative classifier.

One of the motivations behind this work is the large number of samples in intrusion datasets. For example, the CICIDS 2018 dataset contains approximately 1 million samples. Typically, the majority of the dataset consists of benign traffic, and attack samples are a minority since only a limited number of attack points can be appropriately hidden (except for denial-of-service related attacks). As a result, the large number of benign samples increases the complexity of the intrusion detection system due to the increased training overhead of the ML models. To address this, this work leverages both network-based and host-based intrusion detection frameworks simultaneously to take advantage of both systems. A two-stage collaborative classifier is proposed to improve intrusion detection accuracy and minimize false alarm probability. Fig. 1 illustrates the proposed intrusion detection system, which bridges HIDS and NIDS to consider both real network traffic and records saved in host devices. To reduce detection system complexity while maintaining detection performance, this article introduces a two-stage collaborative classifier that comprises a binary classifier and a multi-classifier. The binary classifier discriminates between benign and attack instances, with all attacks sharing the same label. The multi-class classifier characterizes the attack instances after the binary classifier has filtered out the benign instances. The proposed framework initially eliminates benign traffic and then focuses on recognizing individual attack categories.

The main contributions of this work can be listed as below:

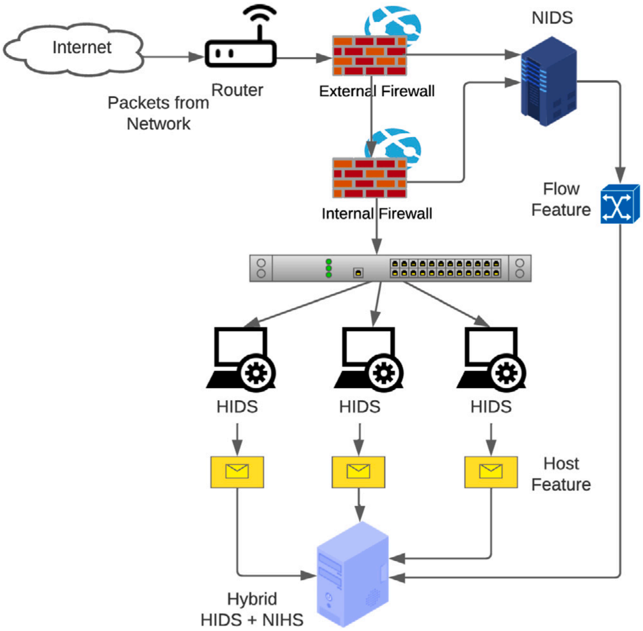


Fig. 1. An example of hybrid HIDS and NIDS system.

- The process involves converting host data from textual to a numerical structure, wherein host-based features are further categorized into event features and message features. The efficacy of the proposed hybrid intrusion detection system has been validated through comparative analysis against state-of-the-art machine learning-based detection methods.
- Data flattening is a technique employed to convert multidimensional features into a vector, facilitating their seamless integration into machine learning or deep learning networks for both training and testing purposes. This straightforward method simplifies the transformation of host features into a vector, making them readily applicable in machine learning models.
- The design entails a two-stage collaborative classifier. The proposed novel classifier not only reduces computational complexity but also improves the detection performance when compared to the baseline machine learning-based detection model.

The remainder of the article is structured as follows. Section 2 presents the state-of-the-art in network security, emphasizing NIDS and HIDS. Section 3 introduces two public datasets, CICIDS 2018 and NDSec-1, and provides a sample distribution for each. In Section 4, a machine learning-based approach using hybrid and network-based features is presented for detecting network intrusions, demonstrating the benefits of using host features alongside flow features. Section 5 introduces a two-stage machine learning framework, which filters out benign samples before characterizing the attack instances. Finally, the article concludes in Section 6 by discussing future directions.

2. Related work

Network security encompasses more than just communication networks; it also encompasses the security of the entire cyberspace environment, including the information infrastructure, application systems, and data resources [20]. To achieve comprehensive network system security, several approaches can be employed, such as firewalls, antivirus software, access control, and anti-malware [1,21–24]. For instance, the study in [25] introduces a hardware/software-based firewall that plays the role of monitoring and filtering traffic packets seeking to enter or exit a safeguarded private network. The authors in [26] present novel

[1] We use network and flow feature to represent the same feature in this work that are extracted from network traffic data and different with host features.

methodologies, including emulation techniques and heuristic scanning to detect viruses. Current antivirus software products are designed with including but not limited to static and dynamic heuristics, machine learning mechanisms, neural networks, hidden Markov models, and rootkit heuristics. Meanwhile, the study in [27] shows an innovative and scalable approach to implement immutable, verifiable, adaptive, and automated access control policies for Internet of Things (IoT) equipment. Additionally, the authors provide a successful proof of concept for scalability. Furthermore, configuration and path analysis can help to safeguard critical infrastructure, such as a smart-grid network, and can be viewed as a firewall use case [28]. The study in [29] presents a refined iteration of Pigeon-Inspired Optimization (PIO), that integrates PIO with the inclusion of a local search algorithm. Furthermore, an ensemble learning approach, employing multiple one-class classifiers, is implemented to enhance the performance of the proposed NIDS. The authors in [30] demonstrate a NIDS method built in Pseudo-Siamese Stacked Autoencoder (PSSAE) to address limited resource issue in IoT devices. The presented approach is strategically implemented in the fog computing layer, involving unsupervised training of stacked autoencoders (SAEs) to extract features from traffic. The following step is to utilize, supervised learning to identify abnormal traffic. Meanwhile, in [31], a novel ensemble-based machine-learning technique for intrusion detection is introduced, aiming to address these challenges and enhance detection capabilities across diverse attack scenarios. The study in [32] demonstrates a novel system to detect intrusions in an IoT network in the context of smart agriculture with restricted computing resources in the IoT system. The study in [33] illustrates how selecting a higher frequency of features can enhance intrusion detection accuracy through the utilization of ML algorithms. The authors in [34] proposes a neural network-based classification system that uses firewall logs to ensure system security. NIDS have also become widely used for intrusion detection, particularly when combined with machine learning algorithms, which have shown promising detection performance. Another study [35] presents a neural network-based framework that identifies abnormal actions in the system to protect against attacks. In [36], the authors introduce UNICORN, which uses time-efficient figures to extract provenance images, providing in-depth information to identify APT attacks. The study in [37] introduces a novel hybrid model that integrates ML and DL approaches to enhance detection rates while ensuring reliability. The proposed approach achieves efficient pre-processing by employing SMOTE and XGBoost for oversampling and feature selection, respectively. Finally, [38] presents a deep learning-based attack detection system for industrial internet of things, called the Sparse Evolutionary Training Approach. The proposed method demonstrates promising detection performance. One study [39] proposes a deep learning-integrated recurrent approach to detect network intrusions by extracting features from recurrent models. The approach also employs a feature selection technique to identify optimal features from the extracted and original ones, and an ensemble model for classification. Another study [40] presents a NIDS framework that can detect specific attacks by integrating supervised (such as XGBoost) and unsupervised (such as expectation–maximization) machine learning algorithms. The study in [41] presents an intrusion detection system to prevent routing attacks such as sinkhole and selective forwarding attacks in IoT networks. The studies in [42] demonstrate ensemble learning-based NIDS that take advantage of various individual ML models to estimate wisely. It is worth to note that there have been studies that leverage federated learning-based intrusion detection [43] however, we scope this work to centralized ML-based intrusion detection solutions.

Related work discusses the use of host-based intrusion detection (HIDS) to detect specific attacks that are challenging for a NIDS. The study in [49] positions HIDS as a crucial last mile of defense against cybersecurity attacks, especially following upon the failure or bypassing of perimeter defenses such as NIDS and firewalls. The study in [44] proposes a host-based system to detect network intrusion via monitoring and analyzing containerized environments (e.g., system calls), that is helpful to make accurate prediction for attacks. In [45], the authors investigate observing information from system logs, which are host-based features, and show that such systems could classify suspicious activities with high accuracy. Furthermore, the authors in [11] utilize host-based information to develop a detection system by organizing statistical data and ML models. The study [47] demonstrates the use of HIDS in Industrial Automation Systems to identify intrusions targeting specific entities of embedded industrial equipment by analyzing information on host devices. In [46], a HIDS framework examines processes that consist of sequences of DLL (Dynamic Link Library) instruction calls made by diverse application and system tasks to the Windows operating system kernel to detect anomalous tasks. In [50], a HIDS system integrated with Convolutional Neural Networks (CNN) is described to ensure security in IoT, with general characteristics that make it compatible with all IoT products. Another study [48] introduces HIDS to enhance cloud system security, integrating several machine learning algorithms (such as KNN, Logistic Regression, and Naive Bayes). As demonstrated in [17], the CICIDS 2018 dataset was processed to yield a numerical representation of host-based features. A fully connected layer-based machine learning model was employed to aggregate all features. In order to improve the performance, the proposed methodology introduces more in-depth knowledge and new ML-based strategies to the data transformation strategy in [17], which is later utilized to generalize the performance under the two well-known datasets. Thus, this study quantitatively demonstrates the unique contribution of the numerical representation of the host-based features (i.e., event and message features) to the network flow features. Moreover, an advanced two-stage classifier provides more accurate performance beyond the HIDS performance. The proposed HIDS detection system is efficient in handling large volume data in the cloud, in comparison to traditional NIDS systems. Table 1 presents current works to identify network intrusion. A thorough qualitative comparison in the table unveils the need for demonstrating a hybrid HIDS and NIDS.

While NIDS and HIDS are both effective in detecting various types of network intrusions, most current research tends to rely solely on one of them. Because there are limited ML-based frameworks that integrate both HIDS and NIDS to identify attack samples, a hybrid framework that leverages the strengths of both approaches would be valuable for improving detection performance. Additionally, previous studies have shown that machine learning algorithms are effective in building network intrusion detection systems. With this in mind, the goal of this work is to propose a machine learning-based hybrid approach that combines HIDS and NIDS to identify intrusions.

3. Dataset introduction

This work utilizes two public datasets that contain network and host information. Host-related data is text-based, extracted from host resources such as logs, files, and folders. The Transformer technique (Bert) is applied to convert text information into numerical data. Fig. 2 illustrates the structure of network-based and host-based features in the CICIDS 2018 and NDSec-1 datasets. The message text and event text information are transformed into two-dimensional numerical matrices separately by Bert. Bert converts event text into an $m * n$ matrix and message text into a $p * q$ matrix. Similarly, the flow feature/network feature dimension is represented by a $1 * f$ matrix, where f is determined by the number of network features in the dataset. This transformation enables machine learning algorithms to utilize flow features and host features to train themselves, which benefits from both network intrusion detection systems and host-based intrusion detection systems [17].

3.1. CICIDS 2018 dataset

The CICIDS 2018 dataset is a public dataset that contains seven types of attacks, including Brute-force, Heartbleed, Botnet, DoS, DDoS,

Table 1
List of existing works for network intrusion detection.

Ref.	Methodology	Dataset	Benefit	Further improvement	Features
[29]	NIDS; enhanced PIO feature selection	BoT-IoT, UNSW-NB15, NLS-KDD and KDD99	Boosted performance	Overfitting	Flow
[34]	Identify intrusions in firewall via SNN	IFW-2019	High (98.50%) accuracy	Dataset uniformity	Flow
[35]	DDoS identification using DWT and auto-encoder	Private	High detection rate 100%	Testing in a real SDN network	Flow
[30]	NIDS based on pseudo-siamese stacked autoencoder	KDDTest	Suitable for IoT device with low computing resource	Reduce false-positive	Flow
[38]	SET integrated attack detection model in IIoT	CICIDS 2017, DS2OS	Accelerated inference time	Diversity of test equipment	Flow
[39]	Via a end-to-end approach to identify intrusions	SDN-IoT, KDD 99, etc.	Generalization	Testing in a real-time system	Flow
[31]	Ensemble based NIDS	UNSW-NB15, NSL-KDD, etc	Generalization to multiple datasets	False positive improvements	Flow
[32]	CNN-based NIDS suitable for agriculture	NSL KDD	High accuracy	Application of other DNNs	Flow
[42]	Ensemble learning-based NIDS	NSL-KDD	Wise prediction is made	Integration of other ML models	Flow
[44]	Intrusion detection via HIDS using two datasets	CB-DS, LID-DS	Reasonable run-time overhead	Identifying Heartbleed and a few other real-time intrusions.	Host
[45]	Observe logs in hosts and identify illegitimate logs via OSSEC tool	N/A	Identify malicious logs	Further test result analysis	Host
[46]	NLP-based HIDS via analysis dll to identify anomalous	ADFA-WD	Identifying a diverse array of attacks	Extension with a transformer-based HIDS	Host
[47]	HIDS to secure embedded industrial equipment	N/A	Deploy HIDS firstly in a PLC, that has a real-time operation system	Extension of features and capability of the presented system	Host
[48]	Two IDS merged to protect cloud	N/A	Identify intrusions and create alerts	Combine different ML models	Host
[17]	Introduce CIDS-Net for intrusion detection	CICIDS 2018, SCVIC-CIDS-2021	99.89% macro F1 score	Improving performance for some classes	Host (Event and Message), Flow
Our	Hybrid NIDS and HIDS; Two-stage collaborative classifier	CICIDS 2018, NDSec-1	Use public dataset; boost performance	Apply deep learning model	Host (Event and Message), Flow

Table 2
CICIDS 2018 dataset sample distribution.

Class	Training	Test	Total
Benign	308 375	152 172	**460 547**
Bot	60 767	29 693	**90 460**
DDOS-HOIC	137 147	67 449	**204 596**
DDOS-LOIC-HTTP	39 019	19 166	**58 185**
DDOS-LOIC-UDP	760	342	**1102**
DoS-GoldenEye	2271	1163	**3434**
DoS-Hulk	13 388	6553	**19 941**
DoS-SlowHTTPTest	10 579	5351	**15 930**
DoS-Slowloris	1394	702	**2096**
FTP-BruteForce	32 222	15 918	**48 140**
SSH-Bruteforce	11 143	5418	**16561**

Web attacks, and infiltration of the network from inside[2]. The attacking infrastructure consists of 50 devices, and the victim system has 5 branches, comprising 420 devices and 30 servers. The CICIDS 2018 dataset comprises captured network traffic and recorded system logs of each device, along with 80 features obtained from the captured traffic [51]. The seven attack scenarios are categorized into 14 types, including Bot, DDOS-HOIC, DDOS-LOIC-HTTP, DDOS-LOIC-UDP, DoS-GoldenEye, DoS-Hulk, DoS-SlowHTTPTest, DoS-Slowingloris, FTP-BruteForce, SSH-Bruteforce, Brute Force-Web, Brute Force-XSS, Infiltration, and SQL Injection. Since Brute Force-Web, Brute Force-XSS, Infiltration, and SQL Injection consist of very few samples (less than 50), it is challenging to train machine learning and deep learning algorithms effectively. Compared to the significant number of other attack scenarios, the limited samples can be considered noise in the training procedure and result in lower detection performance. Special techniques are required for effective handling of rare samples. For instance, in [52], the authors merge five classes with limited data into one class for the training of the ML algorithm. Those five classes included Infiltration, Brute Force-Web, Brute Force-XSS, SQL Injection, and Heartbleed. Similarly, the study in [53] excluded rare classes such as Infiltration, FTP-Brute Force, and DoS-SlowHTTPTest in the performance evaluation under the CICIDS 2018 dataset. Further, in [54], the authors excluded certain flood classes such as Brute Force web attack due to their limited numbers in the CICIDS 2018 dataset. In the same study, it is observed that ML models may recognize alternative patterns within the data, pointing out the need for further investigation. This work adopts a similar approach by excluding four rare classes

[2] The dataset is introduced in detail at: https://www.unb.ca/cic/datasets/ids-2018.html

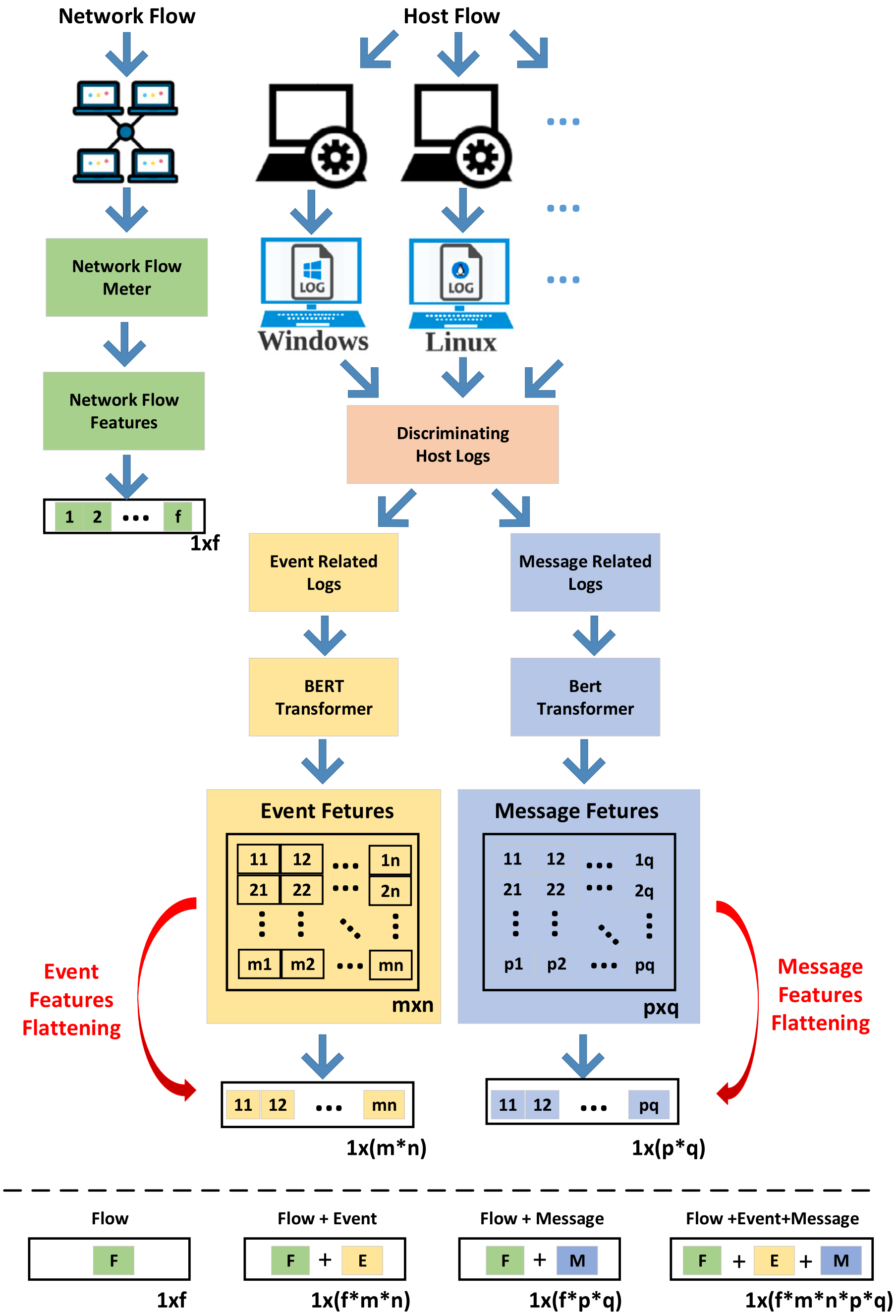


Fig. 2. Flow features and host-based event and message features after transforming through BERT which is one of the well known NLP model. This transform is previously applied one of our dataset, CICIDS 2018 as shown in [17].

in the CICIDS 2018 dataset. Therefore, the four classes are eliminated from the CICIDS 2018 dataset for later evaluation section. The sample distribution of the remaining attack scenarios in the CICIDS 2018 dataset is presented in Table 2. Indeed, excluding minority samples introduces a potential side effect, as complete elimination from a real system is inevitable. It is crucial to identify all attack samples, including minority instances, for comprehensive analysis. We have identified the inclusion of all attack types in the dataset as a prospective avenue for future research and intend to employ effective data augmentation techniques on minority classes.

The CICIDS 2018 dataset contains network traffic with 132 features that represent numerical arrays, which can be used directly by machine learning algorithms. This means that $f = 132$ in Fig. 2. Most current research utilizes network-based data to build NIDS when using the CICIDS 2018 dataset. However, the CICIDS 2018 dataset also contains logs in the host with event data, message data, and network traffic. The logged data in host equipment is saved as text representation to ensure readability for clients and within the organization. To integrate network- and host-based data, text data needs to be transformed into numerical arrays. In this paper, we adopt the strategy proposed in [17] and use Bert for word embedding to extract host features, a transformer architecture trained to obtain language representations. We derive event data and message data separately. Table 4 demonstrates that the CICIDS 2018 dataset consists of 132 network features, 224 event features, and 76,800 message features.

3.2. NDSec-1 dataset

The NDSec-1 dataset was initially introduced in [55] and was generated for research in attack composition in network security schemes.

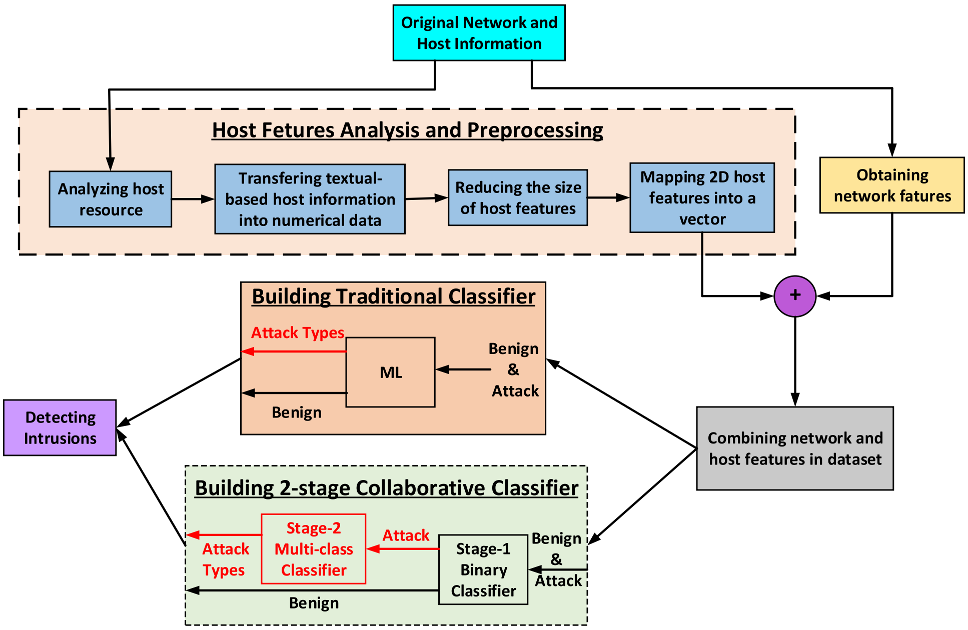

Original Network and Host Information
Host Fetures Analysis and Preprocessing
Analyzing host resource
Transfering textual-based host information into numerical data
Reducing the size of host features
Mapping 2D host features into a vector
Obtaining network fatures
+
Building Traditional Classifier
Attack Types
ML
Benign & Attack
Benign
Detecting Intrusions
Combining network and host features in dataset
Building 2-stage Collaborative Classifier
Stage-2 Multi-class Classifier
Attack
Attack Types
Stage-1 Binary Classifier
Benign & Attack
Benign

Fig. 3. General framework for the proposed methodology. Algorithmic details of two-stage Collaborative Classifier are given in Fig. 8.

Table 3
NDSec-1 dataset sample distribution.

Class	Training	Testing	Total
BOTNET	52	40	**92**
BRUTEFORCE	2150	1006	**3156**
DOS	12 677	6389	**19 066**
EXPLOIT	4	2	**6**
MALWARE	23	12	**35**
MISC	31	18	**49**
NORMAL	5701	2284	**7985**
PROBE	777	307	**1084**
WEBATTACK	18	12	**30**

Table 4
Original dimension of CICIDS 2018 and NDSec-1 for flow, event, and message features.

	Flow (1*f)	Event (m*n)	Message (p*q)
CICIDS	1*132	28*8	100*768
NDSec-1	1*63	2182*8	512*768

The NDSec-1 dataset is a public dataset used in network intrusion detection [16,56]. The dataset consists of 8 types of attacks, including Botnet, Bruteforce, DoS, Exploit, Malware, Misc, Probe, Webattack, Spoofing, and benign samples. However, there is only one sample for the Spoofing class, so this class is excluded from the evaluation section. Table 3 presents the sample distribution across the training and test sets. Flow features are extracted from the NDSec-1 dataset and included with 63 network features. Meanwhile, log records in hosts are provided, so host features are extracted using Bert for text-to-numeric array transforming. The CICIDS 2018 dataset's host features extraction procedures are adopted. Finally, the NDSec-1 dataset contains two parts of host features, including two-dimensional event and message features, as shown in Fig. 2. Table 4 demonstrates that the NDSec-1 dataset consists of 63 network features, 17,456 event features, and 393,216 message features.

4. Hybrid network features and host features for intrusion detection

This section explains the framework to combine network and host features for machine learning-integrated intrusion detection. Fig. 3 demonstrates general framework of the hybrid network and host features that are utilized via ML-based detection approaches.

4.1. Multiple dimension features combination for network and host features

As introduced before, host-based features are extracted from event data and message information, which both have two dimensions as presented in Fig. 2 (as presented in [17]). It is a problem to efficiently utilize the two-dimensional host features, like regular one-dimensional flow features. Feature flattening is a direct, easy, and low-cost technique to map a multi-dimension matrix into a vector. Feature flattening is used in CNN due to the lack of support for multidimensional data in the densely connected layers of a CNN [57,58]. Feature flattening is a feasible approach to adjusting multidimensional data to vectorized features. It is mainly used by ML algorithms.

A sample has f flow features, which is represented as a vector (1). Thus, a_{11} to a_{1f} stand for individual flow feature.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1f} \end{bmatrix} \tag{1}$$

Two-dimensional event features are denoted in (2). Matrix B is extracted from contextual event data in hosts via transformer technique, that includes numerical value in it. Element in B, such as b_{ij}, denotes single event feature.

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{n2} & \dots & b_{mn} \end{bmatrix} \tag{2}$$

Finally, two-dimensional message features are represented in (3). Matrix C is extracted from contextual message data in hosts via transformer technique, that consists of numerical value in it. Element in C, such as c_{ij}, denotes single message feature.

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1q} \\ c_{21} & c_{22} & \dots & c_{2q} \\ \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pq} \end{bmatrix} \tag{3}$$

Flow features, also called network-based features, are the major features in NIDS, especially in ML-based detection systems [58,59]. In order to utilize host-based features, two-dimensional feature data is flattened into a one-dimensional vector. As a result, after flattening, a vector B' of event-based features is obtained.

$$\mathbf{B'} = \begin{bmatrix} b_{11} & \dots & b_{1n} & b_{21} & \dots & b_{2n} & \dots & b_{m1} & \dots & b_{mn} \end{bmatrix} \tag{4}$$

Message-based host features are adjusted into a vector, denoted by C'.

$$\mathbf{C'} = \begin{bmatrix} c_{11} & \dots & c_{1q} & c_{21} & \dots c_{2q} & \dots & c_{p1} & \dots & c_{pq} \end{bmatrix} \tag{5}$$

As a result, multi-dimensional host features are mapped to a vector, so it is easy to obtain hybrid features, denoted by H_1 combining flow features and event features, H_2 combining flow features and message features, and H_3 combining flow features, event features and message features, respectively.

$$\mathbf{H_1} = \begin{bmatrix} A & B' \end{bmatrix} \tag{6}$$

$$\mathbf{H_2} = \begin{bmatrix} A & C' \end{bmatrix} \tag{7}$$

$$\mathbf{H_3} = \begin{bmatrix} A & B' & C' \end{bmatrix} \tag{8}$$

Therefore, the number of hybrid features depends on the number of flow features, event features, and message features. It is straightforward to calculate the number of elements in vectors B' and C', which stand for event features (t_{event}) and message features (t_{mes}), respectively.

$$t_{event} = m \times n \tag{9}$$

$$t_{mes} = p \times q \tag{10}$$

Thus, H_3 contains the maximum number of features, represented in (11).

$$t_{total} = f + t_{event} + t_{mes} \tag{11}$$

4.2. Host features reduction

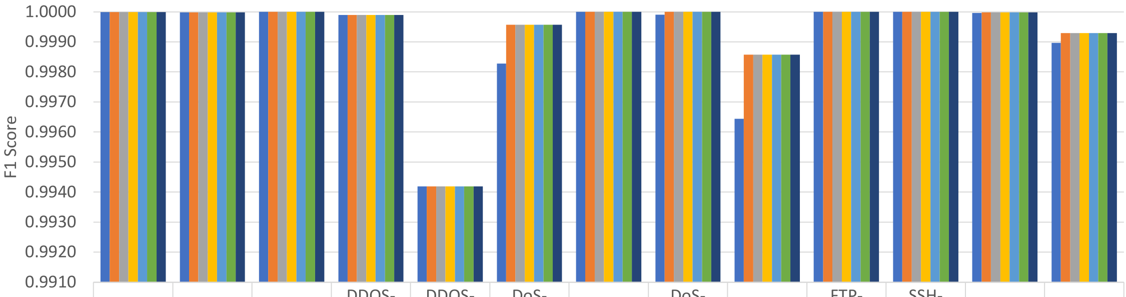
As introduced in Section 4.1, this work flattens host features into a vector and combines vector-based host features and flow features together for the training and testing of ML algorithms. However, feature flattening results in the phenomenon of dealing with too many features, making the dataset complicated, challenging to use, and potential overfitting for ML models. Specifically, in the CICIDS 2018 dataset, as introduced in Section 3, there are 132 flow features, so f is 132 in A. Moreover, event features are extracted into a 288 matrix, so m equals 28, and n is 8 in B. After flattening the event and message features, B' contains t_{event}, which equals 224 elements, indicating that there are 224 event features in vector B'. Meanwhile, the message feature dimension is 100 × 768, represented by C, so p is 100, and q is 768 in C. After flattening the message feature, we obtain a vector C' consisting of 76800 message features. The maximum number of features in the CICIDS 2018 dataset t_{total} is 77,156 features, according to Eq. (11), including flow and host features. The magnitude of host features is notably large, necessitating feature reduction before employing them for training ML models. Moreover, the CICIDS 2018 training dataset includes about 1 million samples, as described in Table 2. Therefore, it is challenging to apply a total of 77,156 features under our current test environment. Meanwhile, in the NDSec-1 dataset, as described in Section 3.2, the number of flow features is 63 (f). Event features are extracted into a 17,456 matrix, and the message feature dimension is 393,216. Thus, a sample in the NDSec-1 dataset consists of hybrid features up to 410,735 when mixing flattened host features and network features, according to Eq. (11). With an overwhelming number of features to be processed by an ML model, the NDSec-1 dataset faces the same dimensionality challenge as the CICIDS 2018 dataset.

In order to reduce the number of features, this work diminishes the message feature matrix and event feature matrix prior to transforming them into a vector by flattening. In the case of the CICIDS 2018 dataset, there is a need to address the reduction of message features, given that they form the majority (e.g., 76,800) among all feature types. The process involves condensing the message feature matrix from 100 × 768 to a smaller matrix, such as 10 × 10, directly diminishing the message features. Additionally, it is crucial to reduce both the event feature matrix (i.e., 2182 × 8) and the message feature matrix (i.e., 512 × 768) for the NDSec-1 dataset before flattening them into a vector array. A method must be designed to select a small matrix from the original large matrix and maintain intrusion detection performance. This work applies the random selection method to determine rows and columns for a matrix prior to feature flattening. The random selection method aims to ensure the selected message matrix is representative of the large one and less likely to be subject to bias [60]. Specifically, message features in CICIDS 2018 are aimed to be reduced from 100 × 768 to 10 × 15; 10 is randomly selected among the range 1 to 100, and 15 is randomly selected from the range 1 to 768. It is worth noting that the average of several rounds of test results is measured to reduce the impact of fluctuations across samples and to take all features into account. We will demonstrate the determination of the appropriate dimension of a matrix considering computation performance and detection performance in Sections 4.3 and 4.4.

4.3. Performance evaluation under CICIDS 2018 dataset

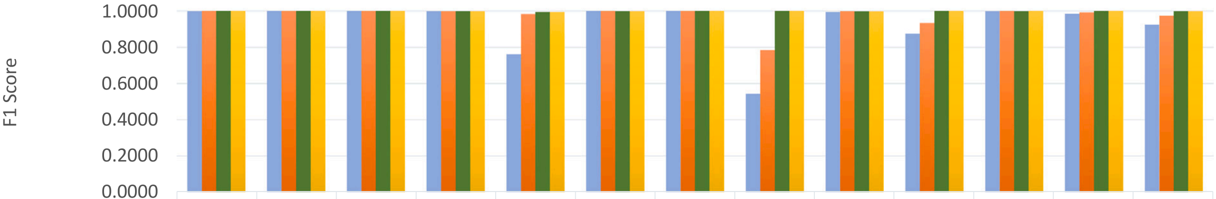
As stated in Section 4.2, the message features matrix should be reduced before flattening so as to address the issues regarding large number of host features that is mentioned earlier. Based on our pressure test, our current test environment can handle about 800 message features among CICIDS 2018 dataset. Thus, feature reduction goal is to condense the original message matrix of 100 × 768 into a smaller matrix C' with a $p \times q$ dimension. Furthermore, C' should have the number of message features $p \times q$ equal to 800. To determine the most appropriate dimension, we evaluated various combinations of integers p and q that satisfy $p \times q$ not being greater than 800. XGBoost is selected to evaluate the detection performance with various matrices (C') of message features, keeping the same flow features (132) and event features (226). The message matrix was reduced to 10 × 80, 15 × 50, and 20 × 40. The detection performance under different message matrices is shown in Fig. 4. It was observed that the performance was almost the same under different message matrices. Therefore, we selected the message feature matrix 15 × 50 (750 message features) as a representative in the subsequent tests. The subsequent evaluation employs various combinations of flow, event, and message features in the CICIDS 2018 dataset. While flow and event features remain unchanged in the original dataset, message features are reduced to 750 and arranged in a 15 × 50 matrix. Consequently, the reduced message features in the CICIDS 2018 dataset are integrated with flow and event features to create representations denoted as H_1, H_2, and H_3 as elaborated in Section 4.1. Calculation of the total number of features, as per Eq. (11), involves the addition of 132 flow features, 226 event features, and 750 message features.

Fig. 5 shows XGBoost (i.e., F1 score) results for four scenarios: (1) only flow feature A, (2) a combination of flow and event features H_1, (3) a combination of flow and message features H_2, and (4) a combination of flow, event, and message features H_3. XGBoost with only flow features is benchmark for comparison with the proposed hybrid flow and host-based performance. The results show that integrating message and event features improves the detection performance of all classes that cannot be fully detected. Particularly for the DDOS-LOIC-UDP attack, F1 score increases from 0.7609 (flow features only) to 0.9828 (with event features) and 0.9942 (by integrating message and event features). The most significant improvement is achieved for the detection of DoS-SlowHTTPTest attack, which increases from 0.5425 to 1.0000. Moreover, the overall performance is significantly improved. For instance, the macro F1 score raises up to 0.9993 with the event and message features, compared to 0.9246 with benchmark flow features



	Benign	Bot	DDOS-HOIC	DDOS-LOIC-HTTP	DDOS-LOIC-UDP	DoS-GoldenEye	DoS-Hulk	DoS-SlowHTTPTest	DoS-Slowloris	FTP-BruteForce	SSH-Bruteforce	WAvg	MAvg
🟦 10x80	1.0000	1.0000	1.0000	0.9999	0.9942	0.9983	1.0000	0.9999	0.9964	1.0000	1.0000	1.0000	0.9990
🟧 15x50	1.0000	1.0000	1.0000	0.9999	0.9942	0.9996	1.0000	1.0000	0.9986	1.0000	1.0000	1.0000	0.9993
⬜ 20x40	1.0000	1.0000	1.0000	0.9999	0.9942	0.9996	1.0000	1.0000	0.9986	1.0000	1.0000	1.0000	0.9993
🟨 30x25	1.0000	1.0000	1.0000	0.9999	0.9942	0.9996	1.0000	1.0000	0.9986	1.0000	1.0000	1.0000	0.9993
🟦 40x20	1.0000	1.0000	1.0000	0.9999	0.9942	0.9996	1.0000	1.0000	0.9986	1.0000	1.0000	1.0000	0.9993
🟩 50x15	1.0000	1.0000	1.0000	0.9999	0.9942	0.9996	1.0000	1.0000	0.9986	1.0000	1.0000	1.0000	0.9993
🟦 80x10	1.0000	1.0000	1.0000	0.9999	0.9942	0.9996	1.0000	1.0000	0.9986	1.0000	1.0000	1.0000	0.9993

Fig. 4. Performance comparison under various dimensions of message feature (host-based feature).



	Benign	Bot	DDOS-HOIC	DDOS-LOIC-HTTP	DDOS-LOIC-UDP	DoS-GoldenEye	DoS-Hulk	DoS-SlowHTTPTest	DoS-Slowloris	FTP-BruteForce	SSH-Bruteforce	WAvg	MAvg
🟦 Flow (Benchmark)	0.9995	1.0000	1.0000	0.9997	0.7609	1.0000	1.0000	0.5425	0.9936	0.8749	0.9999	0.9848	0.9246
🟧 Flow+Event	1.0000	1.0000	1.0000	0.9997	0.9828	1.0000	1.0000	0.7842	0.9993	0.9339	1.0000	0.9927	0.9727
🟩 Flow+Mes	1.0000	1.0000	1.0000	0.9999	0.9942	0.9996	1.0000	1.0000	0.9986	1.0000	0.9999	1.0000	0.9993
🟨 Flow+Event+Mes	1.0000	1.0000	1.0000	0.9999	0.9942	0.9996	1.0000	1.0000	0.9986	1.0000	1.0000	1.0000	0.9993

Fig. 5. NIDS (flow-based) and hybrid NIDS and HIDS (flow, event, and message-based) performance comparison under CICIDS 2018. ‘Mes’ abbreviates message. Flow-based NIDS system performance is the benchmark against which the proposed hybrid flow and host features approaches are compared.

only. The host features, whether event or message, clearly improve the attack detection accuracy. Moreover, message features show further advantages in boosting the intrusion detection performance compared to the event features. As an example, the combination of flow features and message features results in a macro F1 score of 0.9993, whereas integrating flow features and event features yields a macro F1 score of 0.9727.

4.4. Performance evaluation under ndsec-1 dataset

We followed a similar rule to combine flow and host features for the NDSec-1 dataset, starting by reducing the event and message feature matrices into smaller ones and flattening the two-dimensional event and message feature matrices into vectors. As introduced in Section 3.2, event features are stored in a 2182 × 8 matrix B, while message features are stored in a 512 × 768 matrix C. We used the random selection method to determine the smaller matrices for the NDSec-1 dataset, based on previous experience and methodology for the CICIDS 2018 evaluation. We selected the event feature matrix as B' and the message feature matrix as 500 × 8, and the message feature matrix as C' with dimensions 100 × 768 for the NDSec-1 dataset. We skip the detailed procedure in matrix size determination, which is almost the same as the CICIDS 2018 dataset, as illustrated in Section 4.3. Meanwhile, numerical results using the CICIDS 2018 dataset indicate that the dimensions of the message feature matrix marginally impact the intrusion detection performance, as shown in Fig. 4. Most of the event features are excluded if a small matrix, such as 10 × 10, is chosen instead of the original 100 × 768 event feature matrix. One might expect this to reduce the effectiveness of the detection performance. However, with a smaller matrix of event and message features, the number of host features is also significant, up to 80800 with t_{event} = 4000 and t_{mes} = 76800. Therefore, we employed Principle Component Analysis (PCA) [61] to reduce dimensions further for the mix of flow, event, and message features.

Fig. 6 presents a comparison of performance results (F1 score) using PCA to reduce features to different dimensions. The highest F1 score is achieved when PCA reduces the features to 10, with a macro F1 score of 0.91. Therefore, in the following tests, PCA is applied to reduce the features to 10. Fig. 7 presents the attack detection performance (F1 score) with flow and host features. The detection system using the composition of flow features, event features, and message features H_3 achieves the highest overall performance in macro average F1 score (MAvg) with 0.8913 and a weighted average F1 score (WAvg) of 0.9964. With H_3, the detection system demonstrates the best performance for DoS, Normal, and Probe. Specifically, MAvg is at 0.8913 when compared to the flow-only case (0.8595), combined flow and event features H_1 (0.8591), and the combination of flow and message features H_2 (0.8906). When flow and event feature H_1 is utilized, the best performance for Botnet and Exploit is obtained with F1 scores of 0.9217 and 0.7933, respectively. Moreover, the flow

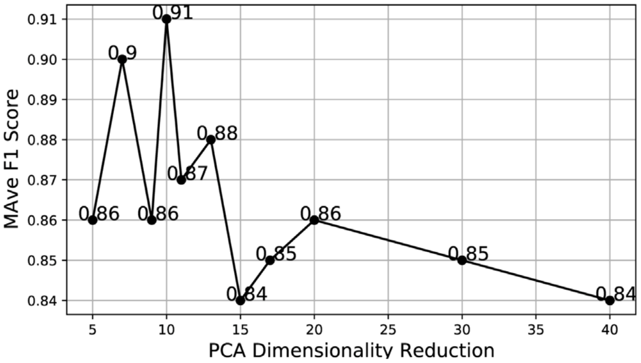


Fig. 6. Comparison of the impact of various dimensions (reduced by PCA) under the NDSec-1.

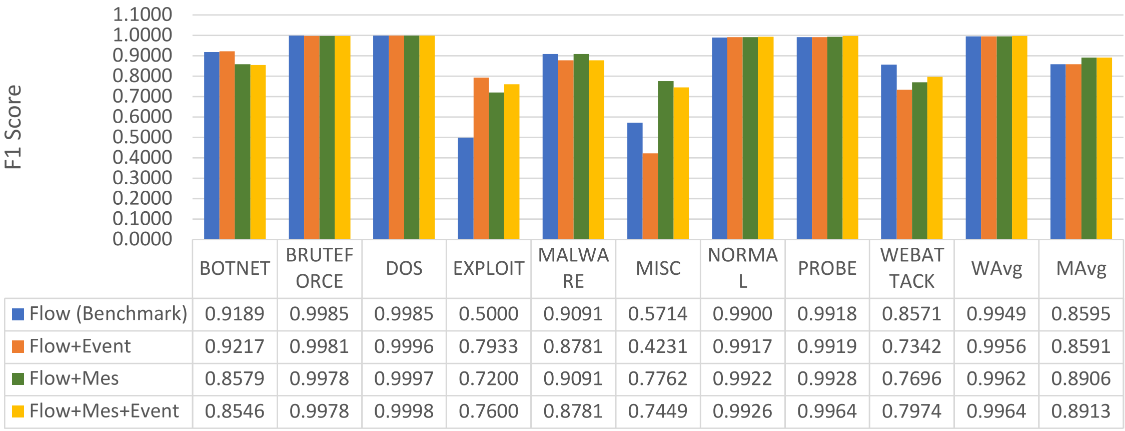
and message combination H_2 case performs the best under Malware (0.9091) and Misc (0.7762). Thus, host features are helpful in boosting attack detection performance in terms of overall performance and most individual classes when compared to the case where only network features are used. However, introducing host features leads to a reduction in performance in the detection of Bruteforce and Webattack. For instance, Bruteforce F1 score reduces slightly from 0.9985 (flow only) to 0.9981 (flow and event), and Webattack reduces from 0.8571 (flow only) to 0.7974 (flow, event, and message together).

5. Two-stage collaborative classifier

5.1. Method introduction

The CICIDS 2018 dataset contains a majority of benign samples, with 308,375 in the training dataset and 152,172 in the test dataset, contributing to approximately 50% in both datasets, as shown in Table 2. Eliminating the benign samples would reduce the computational complexity of the intrusion detection system. Moreover, filtering out half of the samples in the dataset would reduce the dataset's complexity and improve the intrusion detection performance per attack type. Meanwhile, the numerical results show that the ML algorithms perform extremely accurate, close to 100% F1 score, in normal/benign class among CICIDS 2018 dataset and NDSec-1 dataset as illustrated in Fig. 5 and Fig. 7, respectively. Hence, we propose a two-stage collaborative classifier, as shown in Fig. 8, that integrates a binary classifier $ML1$ and a multi-class classifier $ML2$. Within this architecture, the primary objective of $ML1$ is to classify samples into benign and attack categories. In the cases where $ML1$ classifies a sample as benign, encompassing both True Negatives (TN) and False Negatives (FN), it is excluded from subsequent phases. Only samples estimated as attack samples by $ML1$ are considered by $ML2$ regardless of those instances actually being False Positives (FP) or True Positives (TP). The rationale behind the two-stage collaborative classifier stems from the analysis of prior testing results on the CICID 2018 dataset, as detailed in Section 4. This analysis indicates that benign samples can be detected with an accuracy approaching 100%, rendering the detection performance of benign samples nearly unaffected by the presence of attack instances. Consequently, the architecture does not scrutinize $ML1$ predictions of TN and FN. The lack of analysis for FN may yield unintended consequences for the overall system performance as achieving 100% accuracy with $ML1$ is practically challenging. To measure the side effect of FN predictions by $ML1$, we evaluate an optimal scenario based on two assumptions: (a) $ML1$ does not make incorrect predictions for a specific intrusion sample; (b) $ML2$ accurately estimates this attack sample. This study contrasts the performance evaluation section's ideal scenario, where $ML1$ makes zero incorrect predictions (zero FN) for attack samples, with the actual scenario where $ML1$ predicts cases with FN. The quantified side effect that results from the variance between the optimal and real scenarios is shown in Section 5.2.

The proposed method conducts the training and testing procedures sequentially. In the first stage, the training dataset is used to train $ML1$. In the data preprocessing step, the original training dataset is labeled with two classes, including 0 and 1, representing benign and attack, respectively. All attack classes share one label as attack, while benign samples keep the same label as the original dataset in Table 2. $ML1$ is trained to discriminate between attack (label 1) and benign (label 0) classes. In the second stage for $ML2$, the original training dataset filters out benign samples, and the filtered dataset is used to train $ML2$. Thus, $ML2$ is trained to discriminate between multiple attack types without benign samples, forming a multi-classifier. $ML2$ reduces the complexity of the problem and saves computing time. The performance of the proposed two-stage collaborative classifier is verified by applying the test dataset. $ML1$ predicts whether a sample is benign or an attack. All samples estimated as intrusion samples are sent to $ML2$ for the second-stage classification. Performance evaluation of the proposed method is not straightforward since not all samples are predicted by one ML algorithm. In this work, the overall performance combines $ML1$ and $ML2$ prediction results. Specifically, we follow $ML1$ estimation for benign samples and $ML2$ for attack sample prediction. This approach enables the detection of different types of intrusions while maintaining high accuracy in identifying benign traffic. In the following evaluation section, $ML1$ and $ML2$ determination is based on classification performance. Specifically, $ML1$ should perform a promising performance for binary classification while $ML2$ performs better in multi-classification. It is worth noting that the base classifier ($ML1$ and $ML2$) can be any



	BOTNET	BRUTEFORCE	DOS	EXPLOIT	MALWARE	MISC	NORMAL	PROBE	WEBATTACK	WAvg	MAvg
🟦 Flow (Benchmark)	0.9189	0.9985	0.9985	0.5000	0.9091	0.5714	0.9900	0.9918	0.8571	0.9949	0.8595
🟧 Flow+Event	0.9217	0.9981	0.9996	0.7933	0.8781	0.4231	0.9917	0.9919	0.7342	0.9956	0.8591
🟩 Flow+Mes	0.8579	0.9978	0.9997	0.7200	0.9091	0.7762	0.9922	0.9928	0.7696	0.9962	0.8906
🟨 Flow+Mes+Event	0.8546	0.9978	0.9998	0.7600	0.8781	0.7449	0.9926	0.9964	0.7974	0.9964	0.8913

Fig. 7. Performance comparison of NIDS (flow-based) and hybrid NIDS and HIDS (flow, event and message-based) under NDSec-1. 'Mes' abbreviates message. Flow-based NIDS system performance is the benchmark against which the proposed hybrid flow and host features approaches are compared.

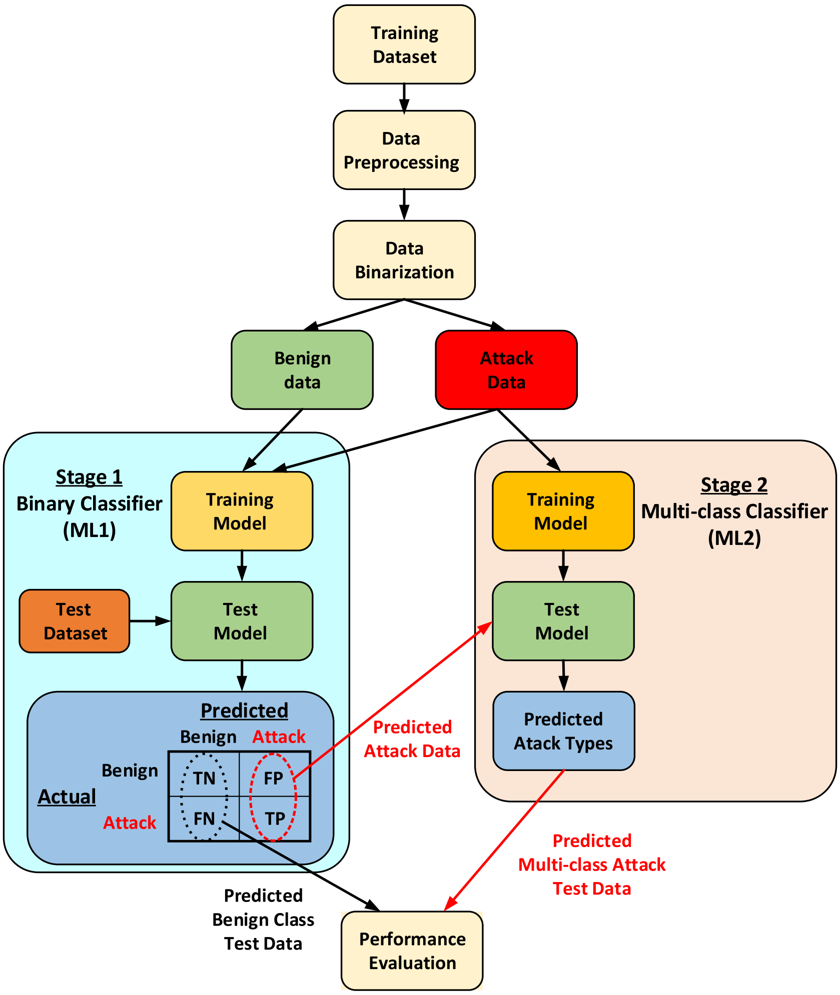


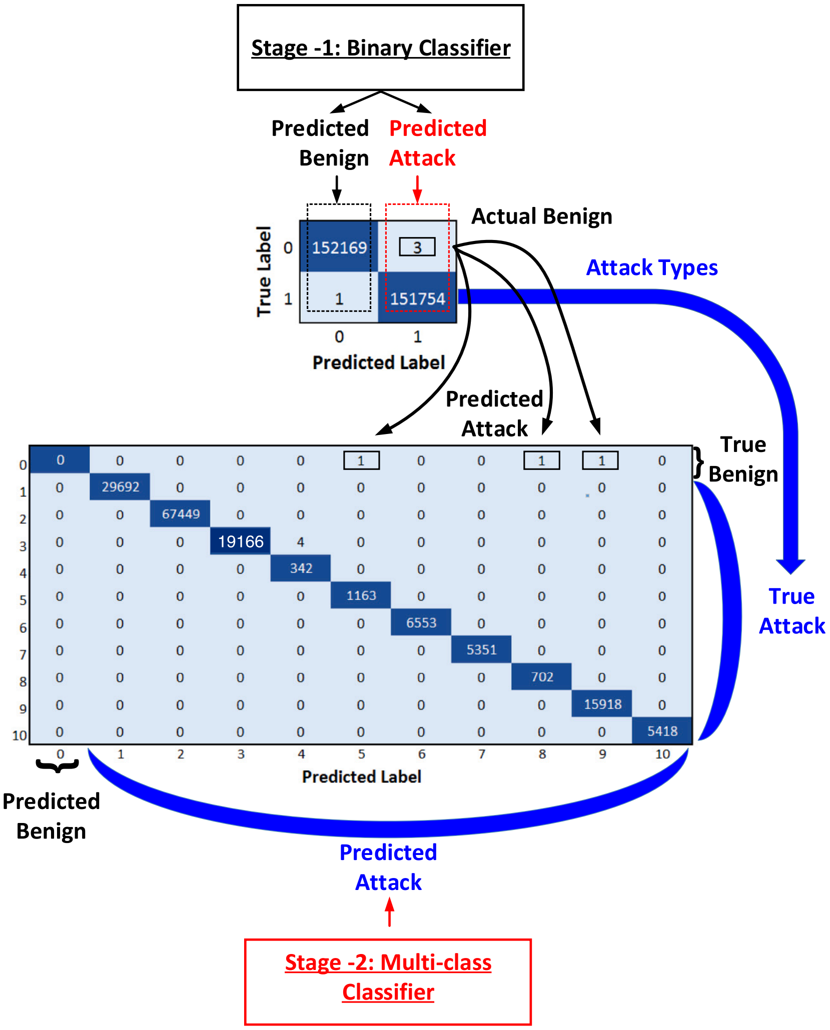
Fig. 8. Two-stage collaborative classifier architecture. First stage builds on a binary classifier (ML1), and the second stage builds on a multi-class classifier (ML2).

ML model. The objective of this study is not to propose a new ML algorithm but a new framework that can use any ML algorithm to boost the detection accuracy of multiple intrusive patterns. Based on previous test result, XGBoost demonstrates high performance, it will be applied in the proposed two-stage classifier framework.

5.2. Performance evaluation

The proposed two-stage classifier is evaluated using the CICIDS 2018 dataset. To compare with the previous work in Section 4, all 132 flow features, flattened event features (224), and the same matrix of message features (15 × 50) are used to evaluate the proposed approach. Based on previous test results, XGBoost achieves almost 100% accuracy for benign samples, making it the chosen base classifier in the two-stage collaborative classifier, as shown in Fig. 8. The presented method relies on an ML algorithm that achieves extremely high detection accuracy for benign samples. It uses a first-layer classifier, $ML1$, to identify and eliminate benign samples, and a second-layer classifier, $ML2$, to detect different types of intrusions. Using deep learning-based approaches is a prospective research direction to find a suitable $ML1$ instead of XGBoost.

When the test dataset is re-labeled for attack samples using one label instead of the original labels, the performance of XGBoost ($ML1$) can be seen in the confusion matrix illustrated in Fig. 9. The confusion matrix shows that most benign samples 152169 (TN) are predicted correctly and filtered by XGBoost (ML1). Only one (1) attack sample (FP) is predicted as benign, which is removed by $ML1$ prior to $ML2$. From $ML1$ binary classification results, we can observe that $ML1$ introduces extremely slim side effect without the analysis of FP. There are 3 benign (FN) samples predicted as attacks and sent to $ML2$ for classification. Fig. 9 demonstrates the classification results of the second-stage machine learning model ($ML2$) using XGBoost. Fig. 10 demonstrates confusion matrix for the two methods. We can see the proposed method improves attack detection rate especially for DoS-LOIC-UDP (class 4), DoS-SlowHTTPTest (class 7) and FTPBruteForce (class 9). Fig. 11 shows the performance comparison between the proposed two-stage collaborative classifier and XGBoost for individual classes and overall performance (macro average F1 score) by integrating $ML1$ and $ML2$ prediction results. The introduced framework demonstrates contributions by enhancing the performance of numerous individual attack types and overall system effectiveness when compared to traditional machine learning models. The proposed approach demonstrates a substantial enhancement in the MAvg F1 score, increasing from 0.9246 to 0.9994, which corresponds to an 8.1% improvement. Notably, it improves detection performance across various individual classes, particularly for DoS-LOIC-UDP and DoS-SlowHTTPTest, showing enhancements of 30.7% and 84.3% respectively. However, it should be noted that the introduced approach leads to a slight reduction in the performance of DDoS-GoldenEye, with a minimal decrease of 0.04% from 1 to 0.9996. This unintended consequence is a minor side effect of the proposed method. Furthermore, the two-stage collaborative classifier exhibits only a marginal improvement in performance compared to traditional machine learning methods presented in Section 4.3, progressing from 0.9993 to 0.9994 in terms of the macro



	0	1
0	152169	3
1	1	151754

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	1	0	0	1	1	0
1	0	29692	0	0	0	0	0	0	0	0	0
2	0	0	67449	0	0	0	0	0	0	0	0
3	0	0	0	19166	4	0	0	0	0	0	0
4	0	0	0	0	342	0	0	0	0	0	0
5	0	0	0	0	0	1163	0	0	0	0	0
6	0	0	0	0	0	0	6553	0	0	0	0
7	0	0	0	0	0	0	0	5351	0	0	0
8	0	0	0	0	0	0	0	0	702	0	0
9	0	0	0	0	0	0	0	0	0	15918	0
10	0	0	0	0	0	0	0	0	0	0	5418

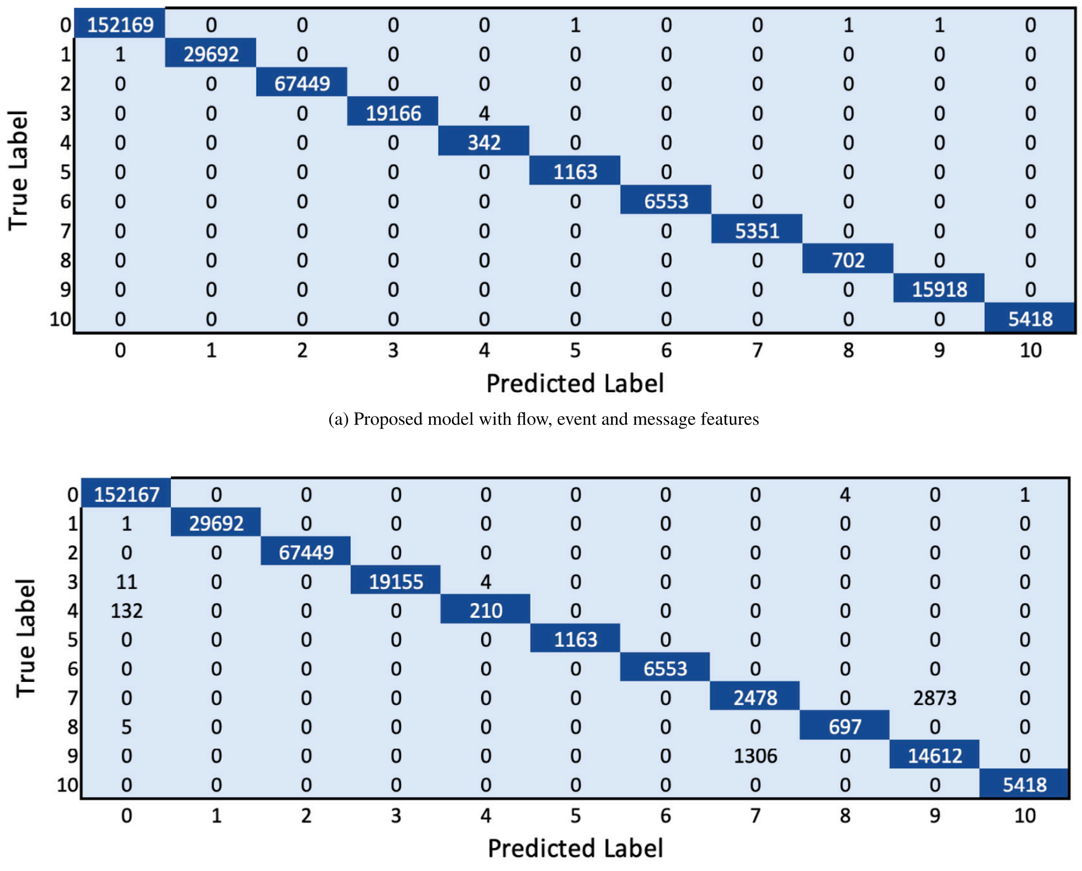
Fig. 9. The confusion matrix of stage-1 (Binary Classifier) and stage-2 (Multi-class Classifier) to demonstrate the prediction performance of the proposed collaborative two-stage method.

average F1 score. Given the already high performance nearing 100%, achieving significant improvements becomes challenging. In Fig. 9, $ML1$ identifies 1 attack sample as benign, which is later excluded from the estimation process of the next stage ML algorithm. When the confusion matrix of the proposed algorithm in Fig. 10(a) is analyzed, we observe that the FNs of $ML1$ are attributed to the Bot (class 1) instances. Ideally, $ML1$ would have zero false negatives, and $ML2$ would correctly predict 29,693 instances in class 1 with no affect on other classes. Based on these optimal predictions, the F1 score for the Benign class is calculated as 0.999990, and the F1 score for the Bot class is 1. However, in the actual scenario where $ML1$ contributes one FN for the Bot class, the F1 score of the Benign class is 0.999987, and 0.999983 for the Bot class. The negligible difference between these scores can be attributed to the slight side effect of $ML1$ due to incorrect classification of an intrusion as benign. Nevertheless, this difference is extremely small and does not affect other classes, resulting in a negligible impact on the overall evaluation.

The proposed method is advantageous when compared to the existing literature. The study in [62] presents the best performance of various machine learning models in terms of F1 score using the CICIDS 2018 dataset, including SVM with 0.9935, KNN with 0.9546, and TREE with 0.9994. In comparison, the proposed framework showcases a Weighted Average (WAvg) F1 score of 1.0000 and a Mean Average (MAvg) F1 score of 0.9994, surpassing the results reported in [62]. The authors in [63] demonstrate a CNN-integrated method to identify intrusions using CICIDS 2018 dataset. Evaluation results in that study, in terms of F1 score, show 0.735 for Benign, 0.43 for DoS-Hulk, 0.09 DoS-SlowHTTPTest, 0.95 for DoS-GoldenEye, 0.89 for DoS-Slowloris, 0.97 for DDoS-LOIC-HTTP, and 0.47 for DDoS-HOIC. Furthermore, our proposed framework demonstrates promising performance for individual attack types. For instance, the two-stage collaborative classifier results in 1.0000 for DoS-Hulk, significantly improves the 0.09 score in [63]. The studies in [51,64] present several ML models using CICIDS 2018 dataset and F1 score comparison is illustrated in Fig. 12. As seen in the figure, the proposed method improves the state-of-the-art by introducing the highest F1 Score. The suggested framework exhibits potential in augmenting the performance of intrusion detection systems by mitigating computational complexity and enhancing accuracy in the identification of multiple intrusive patterns.

6. Conclusion and future work

Network-based Intrusion Detection Systems (NIDS) have limitations in recognizing particular attack types, such as Advanced Persistent Threats, which require bridging with Host-based Network Intrusion Detection systems (HIDS). In this article, we proposed combining HIDS and NIDS to detect various types of intrusive patterns in a networked environment using Machine Learning (ML)-based approaches. We combined network-based and host-based features using a feature flattening approach and studied the impact of dimension reduction on message



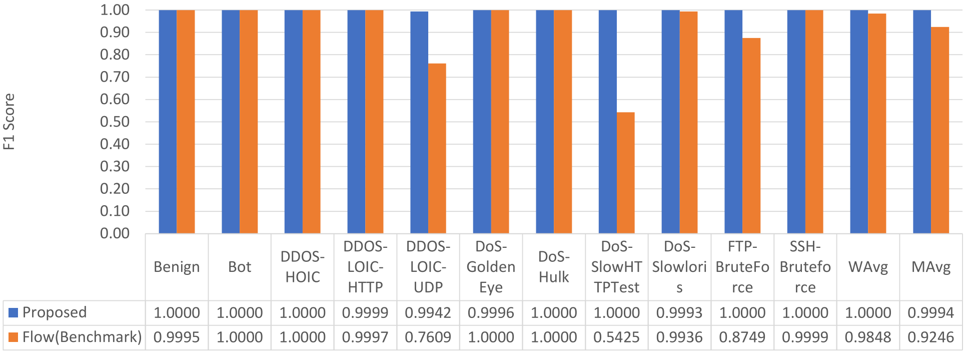
	0	1	2	3	4	5	6	7	8	9	10
0	152169	0	0	0	0	1	0	0	1	1	0
1	1	29692	0	0	0	0	0	0	0	0	0
2	0	0	67449	0	0	0	0	0	0	0	0
3	0	0	0	19166	4	0	0	0	0	0	0
4	0	0	0	0	342	0	0	0	0	0	0
5	0	0	0	0	0	1163	0	0	0	0	0
6	0	0	0	0	0	0	6553	0	0	0	0
7	0	0	0	0	0	0	0	5351	0	0	0
8	0	0	0	0	0	0	0	0	702	0	0
9	0	0	0	0	0	0	0	0	0	15918	0
10	0	0	0	0	0	0	0	0	0	0	5418

(a) Proposed model with flow, event and message features

	0	1	2	3	4	5	6	7	8	9	10
0	152167	0	0	0	0	0	0	0	4	0	1
1	1	29692	0	0	0	0	0	0	0	0	0
2	0	0	67449	0	0	0	0	0	0	0	0
3	11	0	0	19155	4	0	0	0	0	0	0
4	132	0	0	0	210	0	0	0	0	0	0
5	0	0	0	0	0	1163	0	0	0	0	0
6	0	0	0	0	0	0	6553	0	0	0	0
7	0	0	0	0	0	0	0	2478	0	2873	0
8	5	0	0	0	0	0	0	0	697	0	0
9	0	0	0	0	0	0	0	1306	0	14612	0
10	0	0	0	0	0	0	0	0	0	0	5418

(b) XGBoost with flow features

Fig. 10. Confusion matrix for XGBoost using flow feature and the proposed two-stage collaborative classifier with flow, event and message features.



	Benign	Bot	DDOS-HOIC	DDOS-LOIC-HTTP	DDOS-LOIC-UDP	DoS-Golden Eye	DoS-Hulk	DoS-SlowHTTPTest	DoS-Slowloris	FTP-BruteForce	SSH-Bruteforce	WAvg	MAvg
🟦 Proposed	1.0000	1.0000	1.0000	0.9999	0.9942	0.9996	1.0000	1.0000	0.9993	1.0000	1.0000	1.0000	0.9994
🟧 Flow(Benchmark)	0.9995	1.0000	1.0000	0.9997	0.7609	1.0000	1.0000	0.5425	0.9936	0.8749	0.9999	0.9848	0.9246

Fig. 11. Performance comparison for the two-stage collaborative classifier under the CICIDS 2018 dataset. The benchmark is the same one as shown in Fig. 5, using flow-only features, while the proposed method uses flow, event, and host features.

features at the hosts. We used traditional ML algorithms (e.g., XGBoost) and a two-stage collaborative classifier to detect intrusions. The first estimation stage used a binary classifier to discriminate benign and attack traffic, and the second stage used a multi-class classifier to discriminate multiple attack types. We evaluated our approach using two public datasets, CICIDS 2018 and NDSec-1, which contain network information and host-based data (e.g., event data and message data in host resources).

Our numerical results demonstrated that the hybrid of network and host features significantly boosted performance. The overall performance (macro average F1 score) raised from 0.9246 to 0.9993 using CICIDS 2018, representing an 8.1% enhancement, and the detection performance for all individual attack types improved. For example, the F1 score of DDOS-LOIC-UDP increased from 0.7609 to 0.9942, and DoS-SlowHTTPTest improved from 0.5425 to approximately 1 under the CICIDS 2018 dataset. For the NDSec-1 dataset evaluation results, the hybrid of flow and host features improved macro average F1 score from 0.8595 to 0.8913. The two-stage collaborative classifier dramatically boosted performance, with the macro average F1 score increasing from 0.9246 to 0.9994 and 30.7% and 84.3% improvements in individual class performance for DoS-LOIC-UDP and DoS-SlowHTTPTest, respectively. The proposed two-stage collaborative classifier outperforms traditional ML models in literature as detailed in the numerical results. Our ongoing work includes evaluating graph-based datasets (e.g., StreamSpot, Long-hour dataset) introduced in [36] using the presented method in this work. We also plan to deploy

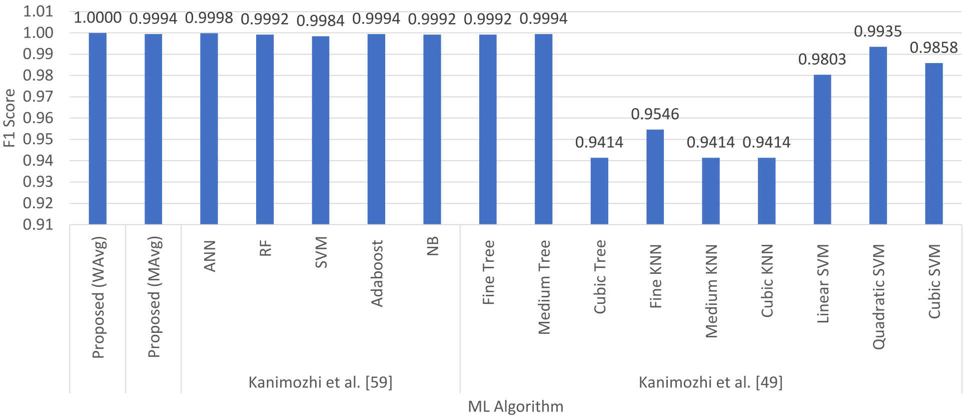


Fig. 12. Performance comparison for the proposed two-stage collaborative classifier and traditional ML algorithms, using the CICIDS 2018 dataset.

deep learning network-based approaches in the two-stage collaborative classifier instead of XGBoost to further boost detection accuracy.

In this work, four minority attack types are excluded from original dataset. It is essential to identify all instances of attacks, including those belonging to minority categories, to conduct a thorough analysis. Recognizing the importance of incorporating all attack types in the dataset, we consider it as a promising area for future investigation. Our research agenda includes the utilization of data augmentation techniques to enhance the representation of minority classes in the dataset. Furthermore, a thorough analysis on the two-stage collaborative classifier regarding the incorrect predictions of the first level classifier $ML1$ for benign samples is also included in our research agenda to further extend and optimize the proposed framework.

CRediT authorship contribution statement

Zhiyan Chen: Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Murat Simsek:** Writing – review & editing, Methodology, Investigation, Formal analysis, Conceptualization. **Burak Kantarci:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Conceptualization. **Mehran Bagheri:** Writing – review & editing, Validation, Formal analysis, Conceptualization. **Petar Djukic:** Writing – review & editing, Validation, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Zhiyan Chen has patent #US20220263842A1 pending to Ciena. Murat Simsek has patent #US20220263842A1 pending to Ciena. Burak Kantarci has patent #US20220263842A1 pending to Ciena. Mehran Bagheri has patent #US20220263842A1 pending to Ciena. Petar Djukic has patent #US20220263842A1 pending to Ciena. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that was used was publicly available and was cited in the paper.

Acknowledgment

This work is supported in part by the Ontario Centre for Innovation (OCI), Canada under ENCQOR 5G Project #31993. Petar Djukic and Mehran Bagheri were with Ciena (Kanata, ON) when this work was performed.

References

[1] Z. Chen, J. Liu, Y. Shen, M. Simsek, B. Kantarci, H.T. Mouftah, P. Djukic, Machine learning-enabled IoT security: Open issues and challenges under advanced persistent threats, ACM Comput. Surv. (2022) http://dx.doi.org/10.1145/3530812, Just Accepted.

[2] C. Zhang, I. Khan, V. Dagar, A. Saeed, M.W. Zafar, Environmental impact of information and communication technology: Unveiling the role of education in developing countries, Technol. Forecast. Soc. Change 178 (2022) 121570.

[3] Z. Yang, X. Liu, T. Li, D. Wu, J. Wang, Y. Zhao, H. Han, A systematic literature review of methods and datasets for anomaly-based network intrusion detection, Comput. Secur. (2022) 102675.

[4] A.B. de Neira, B. Kantarci, M. Nogueira, Distributed denial of service attack prediction: Challenges, open issues and opportunities, Comput. Netw. (2023) 109553.

[5] J.M. Kizza, System intrusion detection and prevention, in: Guide to Computer Network Security, Springer, 2024, pp. 295–323.

[6] J. Liu, M. Nogueira, J. Fernandes, B. Kantarci, Adversarial machine learning: A multilayer review of the state-of-the-art and challenges for wireless and mobile systems, IEEE Commun. Surv. Tutor. 24 (1) (2022) 123–159, http://dx.doi.org/10.1109/COMST.2021.3136132.

[7] K. He, D.D. Kim, M.R. Asghar, Adversarial machine learning for network intrusion detection systems: A comprehensive survey, IEEE Commun. Surv. Tutor. (2023).

[8] M. Zipperle, F. Gottwalt, E. Chang, T. Dillon, Provenance-based intrusion detection systems: A survey, ACM Comput. Surv. 55 (7) (2022) 1–36.

[9] D. Moon, S.B. Pan, I. Kim, Host-based intrusion detection system for secure human-centric computing, J. Supercomput. 72 (7) (2016) 2520–2536.

[10] P.K. Mvula, P. Branco, G.-V. Jourdan, H.L. Viktor, Evaluating word embedding feature extraction techniques for host-based intrusion detection systems, Discover Data 1 (1) (2023) 2.

[11] J. Ribeiro, F.B. Saghezchi, G. Mantas, J. Rodriguez, S.J. Shepherd, R.A. Abd-Alhameed, An autonomous host-based intrusion detection system for android mobile devices, Mob. Netw. Appl. 25 (1) (2020) 164–172.

[12] I. Martins, J.S. Resende, P.R. Sousa, S. Silva, L. Antunes, J. Gama, Host-based IDS: A review and open issues of an anomaly detection system in IoT, Future Gener. Comput. Syst. (2022).

[13] M. Rani, et al., A review of intrusion detection system in cloud computing, in: Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India, 2019.

[14] A.V. Turukmane, R. Devendiran, M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning, Comput. Secur. 137 (2024) 103587.

[15] H. Yang, J. Xu, Y. Xiao, L. Hu, SPE-ACGAN: A resampling approach for class imbalance problem in network intrusion detection systems, Electronics 12 (15) (2023) 3323.

[16] D. Nashat, F.A. Hussain, Multifractal detrended fluctuation analysis based detection for SYN flooding attack, Comput. Secur. 107 (2021) 102315.

[17] J. Liu, M. Simsek, B. Kantarci, M. Bagheri, P. Djukic, Collaborative feature maps of networks and hosts for AI-driven intrusion detection, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 2662–2667.

[18] N.J. Prottasha, A.A. Sami, M. Kowsher, S.A. Murad, A.K. Bairagi, M. Masud, M. Baz, Transfer learning for sentiment analysis using BERT based supervised fine-tuning, Sensors 22 (11) (2022) 4157.

[19] M. Kowsher, A.A. Sami, N.J. Prottasha, M.S. Arefin, P.K. Dhar, T. Koshiba, Bangla-BERT: Transformer-based efficient model for transfer learning and language understanding, IEEE Access 10 (2022) 91855–91870.

[20] J. Xu, F. Sun, Q. Chen, Network security, in: Introduction To the Smart Court System-of-Systems Engineering Project of China, Springer, 2022, pp. 343–384.

[21] S. Kim, S. Yoon, J. Narantuya, H. Lim, Secure collecting, optimizing, and deploying of firewall rules in software-defined networks, IEEE Access 8 (2020) 15166–15177, http://dx.doi.org/10.1109/ACCESS.2020.2967503.

[22] J.R. Vacca, Network and System Security, Elsevier, 2013.

[23] J. Liu, M. Simsek, B. Kantarci, M. Erol-Kantarci, A. Malton, A. Walenstein, Risk-aware fine-grained access control in cyber-physical contexts, 2021, arXiv preprint arXiv:2108.12739.

[24] A.N. Özalp, Z. Albayrak, M. Çakmak, E. ÖzdoĞan, Layer-based examination of cyber-attacks in IoT, in: 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications, HORA, IEEE, 2022, pp. 1–10.

[25] J.M. Kizza, Firewalls, in: Guide to Computer Network Security, Springer, 2024, pp. 265–294.

[26] U. Drakulić, E. Mujčić, A comparative performance analysis of various antivirus software, in: International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies, Springer, 2023, pp. 423–430.

[27] M. Khalid, S. Hameed, A. Qadir, S.A. Shah, D. Draheim, Towards SDN-based smart contract solution for IoT access control, Comput. Commun. 198 (2023) 1–31.

[28] J. Tyav, S. Tufail, S. Roy, I. Parvez, A. Debnath, A. Sarwat, A comprehensive review on smart grid data security, in: SoutheastCon 2022, IEEE, 2022, pp. 8–15.

[29] O.A. Alghanam, W. Almobaideen, M. Saadeh, O. Adwan, An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning, Expert Syst. Appl. 213 (2023) 118745.

[30] S. Tu, M. Waqas, A. Badshah, M. Yin, G. Abbas, Network intrusion detection system (NIDS) based on pseudo-siamese stacked autoencoders in fog computing, IEEE Trans. Serv. Comput. (2023).

[31] M.A. Hossain, M.S. Islam, Ensuring network security with a robust intrusion detection system using ensemble-based machine learning, Array 19 (2023) 100306.

[32] A. El-Ghamry, A. Darwish, A.E. Hassanien, An optimized CNN-based intrusion detection system for reducing risks in smart farming, Internet Things 22 (2023) 100709.

[33] A.N. Özalp, Z. Albayrak, Detecting cyber attacks with high-frequency features using machine learning algorithms, Acta Polytech. Hungarica 19 (7) (2022) 213–233.

[34] Q.A. Al-Haija, A. Ishtaiwi, Multiclass classification of firewall log files using shallow neural network for network security applications, in: Soft Computing for Security Applications, Springer, 2022, pp. 27–41.

[35] R.F. Fouladi, O. Ermiş, E. Anarim, A ddos attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN, Comput. Netw. 214 (2022) 109140.

[36] X. Han, T. Pasquier, A. Bates, J. Mickens, M. Seltzer, Unicorn: Runtime provenance-based detector for advanced persistent threats, 2020, arXiv preprint arXiv:2001.01525.

[37] M.A. Talukder, K.F. Hasan, M.M. Islam, M.A. Uddin, A. Akhter, M.A. Yousuf, F. Alharbi, M.A. Moni, A dependable hybrid machine learning model for network intrusion detection, J. Inform. Secur. Appl. 72 (2023) 103405.

[38] R.V. Mendonça, J.C. Silva, R.L. Rosa, M. Saadi, D.Z. Rodriguez, A. Farouk, A lightweight intelligent intrusion detection system for industrial Internet of Things using deep learning algorithms, Expert Syst. 39 (5) (2022) e12917.

[39] V. Ravi, R. Chaganti, M. Alazab, Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system, Comput. Electr. Eng. 102 (2022) 108156.

[40] J. Liu, B. Kantarci, C. Adams, Machine learning-driven intrusion detection for contiki-NG-based IoT networks exposed to NSL-KDD dataset, in: Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, 2020, pp. 25–30.

[41] A.L. Santos, C.A. Cervantes, M. Nogueira, B. Kantarci, Clustering and reliability-driven mitigation of routing attacks in massive IoT systems, J. Internet Serv. Appl. 10 (1) (2019) 1–17.

[42] Z. Chen, M. Simsek, B. Kantarci, P. Djukic, All predict wisest decides: A novel ensemble method to detect intrusive traffic in IoT networks, in: 2021 IEEE Global Communications Conference, GLOBECOM, IEEE, 2021, pp. 01–06.

[43] O. Friha, M.A. Ferrag, M. Benbouzid, T. Berghout, B. Kantarci, K.-K.R. Choo, 2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT, Comput. Secur. (2023) 103097.

[44] A. El Khairi, M. Caselli, C. Knierim, A. Peter, A. Continella, Contextualizing system calls in containers for anomaly-based intrusion detection, in: Proceedings of the 2022 on Cloud Computing Security Workshop, 2022, pp. 9–21.

[45] C.G. Harshitha, M.K. Rao, P.N. Kumar, A novel mechanism for host-based intrusion detection system, in: First International Conference on Sustainable Technologies for Computational Intelligence, Springer, 2020, pp. 527–536.

[46] Y. Kumar, B. Subba, Stacking ensemble-based HIDS framework for detecting anomalous system processes in windows based operating systems using multiple word embedding, Comput. Secur. 125 (2023) 102961.

[47] C.V. Martinez, B. Vogel-Heuser, A host intrusion detection system architecture for embedded industrial devices, J. Franklin Inst. 358 (1) (2021) 210–236.

[48] V.K. Prasad, A. Raval Abhishek, M. Bhavsar, HIDSC2: Host-based intrusion detection system in cloud computing, in: Inventive Communication and Computational Technologies: Proceedings of ICICCT 2022, Springer, 2022, pp. 71–85.

[49] Z.T. Sworna, Z. Mousavi, M.A. Babar, NLP methods in host-based intrusion detection systems: A systematic review and future directions, J. Netw. Comput. Appl. (2023) 103761.

[50] S. Vinoth, H.L. Vemula, B. Haralayya, P. Mamgain, M.F. Hasan, M. Naved, Application of cloud computing in banking and e-commerce and related security threats, Mater. Today: Proc. 51 (2022) 2172–2175.

[51] V. Kanimozhi, T.P. Jacob, Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-cic-IDS2018 using cloud computing, in: 2019 International Conference on Communication and Signal Processing, ICCSP, IEEE, 2019, pp. 0033–0036.

[52] S. Roy, J. Li, B.-J. Choi, Y. Bai, A lightweight supervised intrusion detection mechanism for IoT networks, Future Gener. Comput. Syst. 127 (2022) 276–285.

[53] T. Kim, W. Pak, Robust network intrusion detection system based on machine-learning with early classification, IEEE Access 10 (2022) 10754–10767.

[54] R. Zuech, J. Hancock, T.M. Khoshgoftaar, A new feature popularity framework for detecting cyberattacks using popular features, J. Big Data 9 (1) (2022) 119.

[55] F. Beer, T. Hofer, D. Karimi, U. Bühler, A new attack composition for network security, in: P. Müller, B. Neumair, H. Raiser, G. Dreo Rodosek (Eds.), 10. DFN-Forum Kommunikationstechnologien, Gesellschaft für Informatik e.V., Bonn, 2017, pp. 11–20.

[56] J. Liang, M. Ma, Co-maintained database based on blockchain for IDSs: A lifetime learning framework, IEEE Trans. Netw. Serv. Manag. (2021).

[57] N. Abiwinanda, M. Hanif, S.T. Hesaputra, A. Handayani, T.R. Mengko, Brain tumor classification using convolutional neural network, in: World Congress on Medical Physics and Biomedical Engineering 2018, Springer, 2019, pp. 183–189.

[58] S. Albawi, T.A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural network, in: 2017 International Conference on Engineering and Technology, ICET, Ieee, 2017, pp. 1–6.

[59] K. Jiang, W. Wang, A. Wang, H. Wu, Network intrusion detection combined hybrid sampling with deep hierarchical network, IEEE Access 8 (2020) 32464–32476.

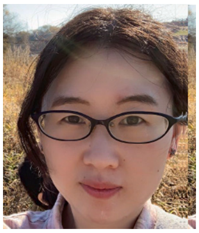
[60] L. Carson, P.F.L. Carson, B. Martin, Random Selection in Politics, Greenwood Publishing Group, 1999.

[61] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdisc. Rev.: Comput. Stat. 2 (4) (2010) 433–459.

[62] I.F. Kilincer, F. Ertam, A. Sengur, Machine learning methods for cyber security intrusion detection: Datasets and comparative study, Comput. Netw. 188 (2021) 107840.

[63] J. Kim, J. Kim, H. Kim, M. Shim, E. Choi, CNN-based network intrusion detection against denial-of-service attacks, Electronics 9 (6) (2020) 916.

[64] V. Kanimozhi, T.P. Jacob, Calibration of various optimized machine learning classifiers in network intrusion detection system on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing, Int. J. Eng. Appl. Sci. Technol. 4 (6) (2019) 209–213.



Zhiyan Chen has completed her PhD candidate in the Smart Connected Vehicles Innovation Centre at the University of Ottawa in 2023. Her research directions include two streams: (1) Secure Mobile CrowdSensing systems via machine learning-based methods; (2) Network intrusions detection based on AI methods. She was the founding Chair of the ACM Women in Computing Student Chapter at the University of Ottawa (Jan.2021–Jan.2022), and a former recipient of the N2Women Fellowship in IEEE ICC 2021. She is a student member of the IEEE Communications Society, and an active member of IEEE Communications Systems Integration and Modelling Technical Committee.

Murat Simsek is currently a member of the Smart Connected Vehicles Innovation Centre as a Senior Research Associate with the School of Electrical Engineering and Computer Science at the University of Ottawa. He received M.Sc. and Ph.D. degrees in Department of Electronics and Communication Engineering from Istanbul Technical University (ITU), Turkey in 2003 and 2012, respectively. During his Ph.D. study, he studied as a Visiting Scholar with Prof. Q.J. Zhang in Department of Electronics Engineering, Carleton University, Ottawa, Canada between Aug'09–Apr'10. He has served as the Technical Program Committee member of several IEEE conferences. His research interests include surrogate based modeling and optimization, artificial intelligence, machine learning, artificial neural networks, knowledge based modeling, cybersecurity, tabular data recognition and extraction, smart and connected autonomous vehicles.

Burak Kantarci is a Full Professor and the Founding Director of the Smart Connected Vehicles Innovation Centre (SCVIC) and Next Generation Communications and Computing Networks (NEXTCON) Research Lab at uOttawa. He holds a Ph.D. degree in computer engineering and is the author/co-author of 250+ publications in established journals and conferences, and 15 book chapters. Continuously listed among the top-cited scientists in telecommunications and networking based on the data reported by Stanford University since 2020, and since 2021, based on data collected from Microsoft Academic Graph, research.com has listed Dr. Kantarci among Canada's top computer scientists. Dr. Kantarci holds an Exemplary Editor Award from IEEE Communications Surveys and Tutorials (2021), and multiple best paper awards from various conferences, most recently from IEEE Globecom2021, Wireless World Research Forum 2022, and IEEE ICC2023, IEEE VCC2023. He is a recipient of the Minister's Award of Excellence from Ontario Ministry of Colleges and Universities (2021). He is the recipient of 2023 Technical Achievement Award of IEEE ComSoc Communications Software Technical Committee. He was a Distinguished Speaker of the Association of Computing Machinery (ACM) in 2019–2021. Currently he is a Distinguished Lecturer of the IEEE Communications Society and IEEE Systems Council. He has been a keynote/invited speaker or panelist in 40 events. In 2019–2020, Dr. Kantarci chaired the Communications Systems Integration and Modeling Technical Committee of the Institute of Electrical and Electronics Engineers (IEEE). He has been a general chair, program chair or track chair in 30+ international conferences. He is an Editor of the IEEE Communications Surveys and Tutorials, IEEE Internet of Things Journal, an Associate Editor for IEEE Networking Letters, and an Associate Editor for Elsevier Vehicular Communications.

Mehran Bagheri holds a Ph.D degree in computational biophysics from the University of Ottawa where he was a postdoctoral research fellow from 2017 to 2019. From 2022 to 2023, he was with Ciena where he was a Data Scientist, and working closely with other teams to develop AI based tools and build data stories. He is involved in a wide range of AI & analytics tasks, from defecting detection for manufacturing teams to forecasting and root cause analysis software/hardware performance for software and design teams.

Petar Djukic is currently with Nokia Bell Labs. When this work was performed, he was responsible for Ciena's AI & Analytics team at the office of the CTO. Petar has completed his B.A.Sc, M.A.Sc and Ph.D. at the University of Toronto. During his Ph.D. studies, Petar has used mathematics and engineering to solve networking problems. During his Ph.D. and later as a postdoctoral fellow, Petar has published over 30 papers with combined 2000 citations. Continuing in industry, Petar now has 47 patents with another 20 pending patents. At Ciena, Petar and his team were on a mission to apply AI to networking and supply chain use cases, to make both Ciena and its more efficient.