Li *et al. Cybersecurity* (2025) 8:80
https://doi.org/10.1186/s42400-025-00359-4

Cybersecurity

RESEARCH Open Access

IL-IDS: an incremental learning approach with confined data streams for intrusion detection

Jianming Li[1,2], Ye Wang[1*], Yan Jia[1,2], Liyi Zeng[2], Wenying Feng[2], Xiao Jing[1,2], Cui Luo[2], Zhaoquan Gu[1,2] and Binxing Fang[1]

Abstract

In cyberspace, intrusion and detection constitute dynamic and continuous game processes, where data streams are generated incrementally during intrusion. To mitigate intrusions and safeguard assets effectively, it is imperative to take prompt actions based on real-time detection and analysis of the currently available data streams. However, existing approaches that rely on complete and clean data struggle to keep pace with the continuous real-time flow of new network data. To address this issue, we introduce IL-IDS (Incremental Learning for Intrusion Detection Systems), a novel intrusion detection approach that utilizes incremental learning to enable accurate and timely detection of intrusions in real-world scenarios, where the need for real-time processing and learning from newly generated traffic data is paramount. IL-IDS performs in scenarios with limited data availability, where it initially transforms textual data streams into vectorized representations and leverages a variation autoencoder (VAE) to compress these vectors, efficiently extracting their latent features. Then a classifier is trained to distinguish attack and normal behaviors, and a three-way decision method is employed to establish a boundary for ambiguous data that pose challenges in direct classification. Concurrently, threat intelligence is integrated into this process to enhance the accuracy of decision-making. We validate the effectiveness and efficiency of IL-IDS with experiments on real-world deployments during an international activity, highlighting its robustness and reliability in intrusion detection applications, especially under conditions of confined data streams. Notably, IL-IDS has exhibited comparable accuracy and recall results, and attains exceptional 99.93% precision and 96.83% F1-score, which demonstrates a notable improvement of 5.27% and 2.59% respectively in comparison to intrusion detection models trained on complete and readily available data.

Keywords Intrusion detection, Incremental learning, Autoencoder, Three-way decision

Introduction

Existing deep-learning-based intrusion detection approaches often assume the availability of complete and clean data (Ramezany et al 2022; Farrukh et al 2022; Hassan et al 2021). However, this assumption does not hold in real-world scenarios, thereby limiting the applicability of these approaches. In the current era of big data, the expansion of cyberspace poses significant challenges to information security. Intrusion detection processes have to be contended with confined data streams at both macro and micro levels. At the macro level, the influx of gigabytes of data within minutes renders long-term

*Correspondence:
Ye Wang
wangye2020@hit.edu.cn
[1] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518052, Guangdong, China
[2] Pengcheng Laboratory, Shenzhen 518052, Guangdong, China

Springer Open

© The Author(s) 2025. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

storage of historical data infeasible. Additionally, in some important activities, information of protected assets is unknowable due to its sensitivity, and security engineers have only access to network or log data of their own but not of other participating security companies. At the micro level, network systems encounter traffic anomalies and node failures frequently, resulting in missing data fields in network flow or log data. Consequently, intrusion detection efforts have to perform with confined data streams.

A two-stage methodology (Duchi et al 2024) has been proposed to address the challenge of anomaly detection in asynchronous and incomplete data. This approach employs auto-regressive moving average to fill in missing data, and then conduct anomaly detection using the augmented dataset. However, this method exhibits certain limitations. It operates under a centralized framework, dividing the data solely into normal and abnormal categories. This necessitates the filling and storage of the entire dataset while lacking a boundary for data samples that pose difficulties in classification. The SCAN algorithm, introduced by Patcha and Park (2007), demonstrates the ability to swiftly and accurately detect intrusions even in the absence of complete and clean audit data, which is achieved by stochastic clustering algorithm based on expectation maximization and adaptive weighted sampling. Nevertheless, SCAN is specifically designed for audit data from hosts and does not readily adapt to open and dynamic environments where data arrives incrementally. The 1D-Dilated Causal Neural Network (1D-DCNN) was first applied to intrusion detection by Rizvi et al (2022). However, this approach primarily pertains to computing resources rather than confined data streams. Considering that the Pyramid and Elastic sketch cannot be adapted to various traffic sizes when conducting flow measurement in the data plane of high-speed switches, Kim et al (2023) adopted a minimum update strategy and present count-less sketch for robust data plane intrusion detection. It deals with different traffic sizes, but the confines of data remain unsolved. As a dynamic and continuous game process in the real world, intrusion detection approaches have to make a decision with only currently available data while accounting for the incremental arrival of additional data. Moreover, these approaches should exhibit necessary flexibility to handle samples that defy conclusive determination as intrusions or non-intrusions.

These problems pose great challenges for intrusion detection. When deploying in real network systems, intrusion detection systems (IDS) are required to process an immense volume of data, often in an order of gigabytes within a matter of minutes. As network flow and log data arrive incrementally, it becomes impractical to store the entire data for latter analysis of IDS. Consequently, decisions must be made by IDS based on the limited pool of currently available data. Furthermore, network data exhibits diverse formats, and certain data fields may be missing or incomplete when anomalies or failures occur within the network. The presence of such confined data forms complex scenarios for IDS, necessitating resilience in handling these instances. Rather than making immediate decisions, IDS must exhibit robustness in retaining such hard samples until the arrival of data with similar features.

To address these challenges, we propose **IL-IDS**, an **I**ncremental **L**earning approach for **I**ntrusion **D**etection **S**ystem specifically designed for real-world network environment, as opposed to relying solely on synthesized data (Moustafa and Slay 2015; Tavallaee et al 2009; Sharafaldin et al 2018; Lippmann et al 2000a, b). In the context of incrementally arriving data, IL-IDS operates under the limitation of confined data streams, which is in various forms. DeepAID (Han et al 2021) tackles this issue by segregating raw data into three distinct categories: tabular, time-series, and graph data. Anomaly detection is then performed using separate models for each data type. However, this approach is overly intricate for real-world deployment. On the other hand, the common data model (CDM) (Khoury et al 2020) effectively represents data from different operating systems such as Linux, Android, and Windows. Nevertheless, CDM is primarily tailored for process tracing rather than intrusion detection, and it is unable to represent network traffic data. In contrast, our proposed IL-IDS is capable of processing data of various forms simultaneously. Recognizing the significance of special characters in cyber attacks, such as the "%" symbol denoting annotation and rendering subsequent words ineffective in SQL injection attacks, our IL-IDS employs a character-level vectorization technique for input texts, instead of relying on word-level representations. Subsequently, the data items are padded to a fixed length with the use of a meaningless integer, specifically 0. A variation autoencoder (VAE) is trained to extract the feature matrix of the input data, enabling the learning of hidden features and the compression of a large volume of input data. To simplify the model, we employ a two-layer sparse autoencoder (SAE) trained with a data-replay incremental learning strategy (Van de Ven et al 2022) to determine whether the input data originates from malicious activities or not. Additionally, a three-way decision model is incorporated into the output of the decision model, establishing a boundary for challenging samples and refining the results with the guidance of threat intelligence. The main contributions of our work are as follows:

- We present a novel approach named IL-IDS for intrusion detection that leverages incremental learning to effectively handle confined data streams, aligning with the real-world scenario where data streams are generated incrementally.
- We design a text-to-vector algorithm to transform the input text data into vectors and padded them into a fix length. Then we use a VAE to learn and compress the underlying features of input data, which can effectively improve the computing and storage efficiency. Furthermore, we train and finetune a two-level autoencoder with incremental learning strategy to classify the input data. To deal with hard samples in the boundary, we integrate a three-way decision module that incorporates threat intelligence, improving the flexibility of detection models.
- We testify our proposed IL-IDS by deploying it into real-world network environment with the inclusion of honeypots. Extensive experiments show the effectiveness and efficiency of IL-IDS, which achieves comparable or even superior performance when compared with models trained on the whole complete data.

The remaining sections of this paper is organized as follows. The related works are analyzed in Related works" and the preliminaries of incremental learning and three-way decision are given in "Preliminaries". The detail description of our proposed IL-IDS is presented in "Methods", whereas the results of IL-IDS can be found in "Experiment and analysis". Finally, conclusion and future work is given in "Conclusion and future work".

Related works

IL-IDS is an incremental learning approach for intrusion detection, which also includes a boundary established by three-way decision method. In this section, we present a comprehensive review of the existing literature on incremental-based methods and three-way-decision-based methods for intrusion detection. By analyzing and evaluating their strengths and weaknesses, we aim to provide a thorough understanding of the current state-of-the-art in these areas. Ultimately, a discussion will serve as a foundation for the subsequent development and refinement of the IL-IDS framework.

Incremental-based methods

In real-world context, training samples often arrive in incremental streams or are only available for a limited period of time due to privacy or storage constraints. To address the challenges posed by confined data streams, incremental learning (IL), also referred to as continual learning (Wang et al 2024), sequential learning (Clegg et al 1998), or lifelong learning (Zhou et al 2023), has emerged as a promising approach in designing machine learning models with continuous learning capacity. Van de Ven et al (2022) firstly divided incremental learning into three kinds: task-incremental learning (TIL), domain-incremental learning (DIL) and class-incremental learning (CIL). Later, Wang et al (2023) conducted a comprehensive survey about the theories, methods and applications of incremental learning. MetaSP (Sun et al 2022) was proposed to investigate the influence of examples in incremental learning, leveraging an influence function to accurately estimate the stability and plasticity impact of different data samples. Additionally, Huang and Ma (2023) stored suspicious data in a blockchain and manually labeled and classified them with the assistance of security agencies, which is proved to be time-consuming and expensive. Wu et al (2021) designed a dynamic ensemble incremental learning model for intrusion detection due to the dynamic changes of network data over time. This model addresses the challenge of evolving data by updating the ensemble model with new parameter. Sharma and Arora (2021) conducted a review of incremental learning methods specifically for network anomaly detection, focusing on challenges such as concept drift, noisy data, stability-plasticity dilemma, and model complexity. These obstacles represent the primary challenges encountered in intrusion detection based on incremental learning.

While incremental learning has found extensive use in computer vision (Van de Ven et al 2022; Wang et al 2023; Sun et al 2022; Rebuffi et al 2017), its application in intrusion detection remains limited. Considering the incremental nature and evolving distribution of data streams in real-world scenarios, we propose an intrusion detection approach based on domain-incremental learning in this paper.

Three-way-decision-based methods

Yao (2009, 2010) proposed the three-way decision model based on decision-theoretic rough sets. From then on, three-way decision has been used in many areas, including intrusion detection. Recognizing the limitations of binary decision models, such as their lack of flexibility and inability to handle misjudgments at the model level, Nauman et al (2016) proposed a three-way decision-based approach for malware analysis. This method enables the model to defer decisions instead of solely accepting or rejecting. With the assumption that boundaries between normal patterns and abnormal ones are not always distinct, Ye et al (2010) extracted a set of normal rules for intrusion detections based on rough set theory. Similarly, we add a decision boundary for those data samples that are difficult for classifiers to provide definitive

decisions due to insufficient information. Zhang et al (2020) constructed a feature space using denoising autoencoders and employed three-way decision model to immediately classify network behaviors as intrusive or normal. However, making decisions immediately without sufficient information carries inherent risks. Therefore, we advocate reaching conclusions after considering data with similar features that arrives moments later. Furthermore, it has been demonstrated that autoencoders outperform traditional shallow feature learning methods, and classification methods based on three-way decisions outperform those based on two-way decisions (Zhang et al 2020). Geng et al (2021) designed a sequential three-way decision approach based on multi-granularity feature sets extracted from different training epochs of encoders. Nevertheless, these methods require storing the entire data, which is impractical for continuous processes in cyberspace.

Discussion

In addition to incremental-based and three-way-decision-based methods, deep learning approaches have gained popularity in intrusion detection (Rizvi et al 2022; Han et al 2021; Du et al 2017; Bansal and Mahapatra 2017; Yang et al 2021; Latif et al 2024; Moustakidis and Karlsson 2020). These methods offer high accuracy, however, they rely on the availability of large amounts of clean and complete data, thereby limiting their practical deployment in real-world scenarios. While tagging and tracking methods (Ji et al 2018; Devecsery et al 2014) are effective for auditing logs in operating systems, they are impractical for tagging and tracking network flows in cyberspace. To address these challenges, Han et al (2023) proposed an explainable anomaly detection method in an open-world setting, encompassing four modules: output calibration, shift detection, shift explanation, and shift adaptation. In this paper, we propose a domain-incremental-learning-based approach for intrusion detection in real world with the shift of data distributions taking into consideration. Initially, we transform the input data into vectors and extract expressive features by training a VAE model. With the help of VAE model, our proposed IL-IDS can process data in various forms. We incrementally train and finetune a classification model using a two-layer autoencoder and introduce a decision boundary for samples with incomplete and inadequate information. Besides, threat intelligence can also be easily incorporated into the training process through the boundary of three-way decision.

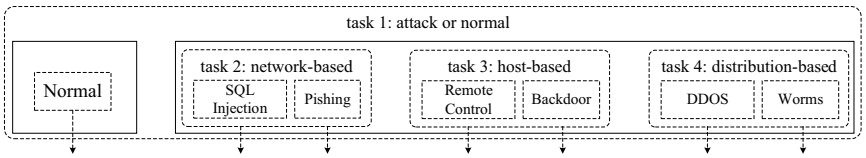
Preliminaries

This section primarily describes the concepts of incremental learning, including the domain-incremental learning settings in this paper, and gives an analysis of the basic theory of three-way decision.

Incremental learning

Incremental learning is a machine learning paradigm which aims to learn ongoing tasks continually. It simulates human intelligence in the form of natural intelligence. Since IL learns a sequence of tasks incrementally, it operates under the assumption that the training data D_t, or a significant portion of it, becomes inaccessible once the learning process for a particular task t is completed. IL can be divided into TIL, DIL and CIL (Van de Ven et al 2022; Ke and Liu 2022). In TIL, task identities (task IDs) are provided during both training and testing phases, enabling algorithms to differentiate between classes within the same task. DIL, on the other hand, treats class labels as consistent across all tasks, rendering the provision of task IDs inconsequential. Algorithms in DIL must adapt to the same problem but in varying contexts. In CIL, task IDs are assumed to be unavailable during testing, and class labels are distinct for each task. Algorithms operating in CIL may encounter new classes that were never observed during the training procedure. The explanation of three types of incremental learning can be seen in Fig. 1.

Consider a task set $T = 1...t$ where X represents the input space, Y_t represents the label space



	Input	Expected Output	Explaination
TIL	data stream+task label	global labels in specific tasks	classify different attack types in individual task (1or 2 in task2)
DIL	data stream	attack or normal	class labels are assumed to be the same in all tasks (N or A)
CIL	data stream	global lables	classify all the classes among all tasks (0, 1, 2,...,6)

Fig. 1 Explanation of different settings for incremental learning. In task-specific labels, "N" means normal and "A" means attack

specific to task t, and G denotes the global label space, $G = \{Y_1, Y_2, ..., Y_t\}$. For example, in Fig. 1, $Y_1 = \{N, A\}$, $Y_2 = \{1, 2\}$, $Y_3 = \{3, 4\}$, $Y_4 = \{5, 6\}$, $G = \{0, 1, 2, 3, 4, 5, 6\}$. The training set of task t is $D_t = \left\{\left(x_i^t, y_i^t\right)\right\}_{i=1}^{N_t}$, where x_i^t is the ith data item of task t, y_i^t represents the corresponding label, N_t is the total number of samples in task t. Let $p(X_t)$ be the distribution of input data in task t, it is assumed that $p(X_t) \neq p(X_{t'})$, $\forall t \neq t'$. Specifically, in domain-incremental learning, $Y_1 = Y_2 = ... = Y_t$ if task IDs are not provided. The learning objective for all three types of incremental learning can be formulated as follows:

$$f_{TIL} : X \times T \rightarrow G, \tag{1}$$

$$f_{DIL} : X \rightarrow G, \tag{2}$$

$$f_{CIL} : X \rightarrow T \times G. \tag{3}$$

In this paper, we try to conduct intrusion detection by discerning between normal and attack instances rather than classifying specific attack classes. Consequently, the class labels across different tasks can be assumed to remain consistent. The labels of input data are set to normal, attack and boundary according to the threat level of honeypot system deployed during the activity. The incorporation of a three-way decision framework enhances the analysis of data with boundary label, enabling the effective management of data items characterized by incomplete or insufficient information.

Three-way decision

Two-way decision based methods solely offer the options of accepting or rejecting model outputs, which are lack of flexibility and fail to accommodate misjudgments. To address these limitations, the concept of three-way decision has emerged as an enhancement to traditional two-way decision frameworks. It introduces a third option for deferring decisions when the available evidence is incomplete or insufficiently strong. The theory of three-way decision with probabilistic rough sets was initially formulated by Yao (2010), and subsequent studies (Yao and Zhao 2008; Yao 2008, 2011) further enriched its attributes and demonstrated its superiority. Based on these theories, we adopt the notation C to represent imprecise concepts that can tolerate certain levels of classification errors. This concept serves as a decision boundary to distinguish between normal and attack behaviors:

$$POS(C) \models \{x \in X \big| [x] \subseteq C\}, \tag{4}$$

$$BND(C) \models \{x \in X \big| [x] \subsetneq C \wedge [x] \cap C \neq \emptyset\}, \tag{5}$$

$$NEG(C) \models \{x \in X \big| [x] \cap C = \emptyset\}, \tag{6}$$

where X is the input space, POS means positive (normal) behaviors, NEG means negative (intrusion) behaviors, and BND means boundary behaviors that a decision is hard to make at this moment.

In intrusion detection, we have a state set $\Omega = \{P, N\}$, in which P means positive (normal), N means negative (intrusion). Based on three-way decision, we have an action set $S = \{a_P, a_B, a_N\}$, where a_P, a_B, a_N represent $x \in POS(C)$, $x \in BND(C)$, and $x \in NEG(C)$ respectively. There are some risks to take actions in intrusion prevention, for example, misinterpreting an attack behavior as normal one and allowing it to pass through can have devastating consequences. The risk function of taking actions in different states is given in Table 1.

In Table 1, λ_{PP}, λ_{BP}, λ_{NP} denote the risks introduced by conducting a_P, a_B, a_N respectively when input x is normal, and λ_{PN}, λ_{BN}, λ_{NN} denote the risks for the same decision when x is malicious.

According to probabilistic rough sets (Yao 2010), the expected risk for taking actions on an input x is:

$$R(a_P \mid x) = \lambda_{PP} P(C \mid x) + \lambda_{PN} P(\neg C \mid x), \tag{7}$$

$$R(a_B \mid x) = \lambda_{BP} P(C \mid x) + \lambda_{BN} P(\neg C \mid x), \tag{8}$$

$$R(a_N \mid x) = \lambda_{NP} P(C \mid x) + \lambda_{NN} P(\neg C \mid x), \tag{9}$$

where $P(C \mid x)$ is the probability of $x \in C$. In practice, decisions should be made in minimum risk, that is:

$$(P) \quad \textit{if } R(a_P \mid x) \leq R(a_B \mid x) \textit{ and } R(a_P \mid x) \leq R(a_N \mid x), \textit{ then decide } x \in POS(C), \tag{10}$$

$$(B) \quad \textit{if } R(a_B \mid x) \leq R(a_P \mid x) \textit{ and } R(a_B \mid x) \leq R(a_N \mid x), \textit{ then decide } x \in BND(C), \tag{11}$$

$$(N) \quad \textit{if } R(a_N \mid x) \leq R(a_P \mid x) \textit{ and } R(a_N \mid x) \leq R(a_B \mid x), \textit{ then decide } x \in NEG(C). \tag{12}$$

As λ_{PP}, λ_{BP}, λ_{NP} is the risks of taking actions a_P, a_B, a_N correspondingly when $x \in C$, it can be concluded that:

Table 1 Risk function of taking actions in different states

	$C(P)$: normal	$\neg C(N)$: intrusion
a_P	λ_{PP}	λ_{PN}
a_B	λ_{BP}	λ_{BN}
a_N	λ_{NP}	λ_{NN}

$$\lambda_{PP} \leq \lambda_{BP} < \lambda_{NP}, \tag{13}$$

$$\lambda_{NN} \leq \lambda_{BN} < \lambda_{PN}, \tag{14}$$

because the risk of classifying $x \in POS(C)$ when x is normal is less than or equal to the risk of classifying $x \in BND(C)$, and both of them are absolutely less than the risk of classifying $x \in NEG(C)$. Further, $P(C \mid x)$ is the probability of $x \in C$, $P(\neg C \mid x)$ is the probability of $x \notin C$, so:

$$P(C \mid x) + P(\neg C \mid x) = 1. \tag{15}$$

According to condition (13), (14) and (15), condition (10)-(12) can be re-expressed as:

$$(P) \quad \textit{if } P(C \mid x) \geq \alpha \textit{ and } P(C \mid x) \geq \gamma, \textit{ then decide } x \in POS(C), \tag{16}$$

$$(B) \quad \textit{if } P(C \mid x) \leq \alpha \textit{ and } P(C \mid x) \geq \beta, \textit{ then decide } x \in BND(C), \tag{17}$$

$$(N) \quad \textit{if } P(C \mid x) \leq \beta \textit{ and } P(C \mid x) \leq \gamma, \textit{ then decide } x \in NEG(C). \tag{18}$$

where α, β, γ are computed by the following function (Yao 2010):

$$\alpha = \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}, \tag{19}$$

$$\beta = \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}, \tag{20}$$

$$\gamma = \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}. \tag{21}$$

From equation (19)-(21) we can observe that the decision thresholds are determined only by the risk function. Here we introduce another restricted condition:

$$(\lambda_{PN} - \lambda_{BN})(\lambda_{NP} - \lambda_{BP}) > (\lambda_{BN} - \lambda_{NN})(\lambda_{BP} - \lambda_{PP}), \tag{22}$$

which is proved in Appendix A. From condition (13), (14) and (22), it can be concluded that:

$$0 \leq \beta < \gamma < \alpha \leq 1, \tag{23}$$

the demonstration is shown in Appendix B. According to (16)-(18) and (23), γ makes no effect and can be ignored, so the final decision rules are simplified as following:

$$(P) \quad \textit{if } P(C \mid x) \geq \alpha, \textit{ then decide } x \in POS(C), \tag{24}$$

$$(B) \quad \textit{if } \beta < P(C \mid x) < \alpha, \textit{ then decide } x \in BND(C), \tag{25}$$

$$(N) \quad \textit{if } P(C \mid x) \leq \beta, \textit{ then decide } x \in NEG(C). \tag{26}$$

Methods

We provide a comprehensive description of our proposed method IL-IDS in this section. We commence by presenting the framework of IL-IDS, detailing its overall structure and components. Subsequently, we elaborate on the data pre-processing procedure, elucidating the steps involved in preparing the input data for further analysis. Next, we outline the feature extractor and intrusion detector, systematically explaining their respective methodologies and functionalities. Finally, we delve into the intricacies of conducting a thorough three-way decision on the boundary, describing the decision-making process and its underlying principles.

Framework

In this research, we propose an incremental-learning-based approach for real-world intrusion detection. The data streams under consideration originate from an international activity and arrive in large volumes incrementally. Due to the nature of the data collection process, it is infeasible to gather all the data streams together, resulting in access only to confined data streams on a daily basis. Consequently, an incremental learning method is employed to address this challenge. The input data streams are organized chronologically, and relevant information such as timestamps, IPs, ports, payloads, and descriptions are extracted and concatenated to create data items. A feature extractor based on VAE is trained to learn and compress the feature matrix for subsequent detection tasks.

In the intrusion detection phase, data items are divided into an initial set and rest sets, which are then sequentially fed into the detector. To account for the confines of the data streams, a three-way decision module is incorporated, introducing a boundary for the classification

results. Data falling within the boundary can be deferred or judged by the three-way decision module. A concise representation of our proposed IL-IDS is provided in Algorithm 1. Additionally, the behavior of IPs may evolve over time, wherein an IP initially exhibits normal behavior by gathering information and subsequently launches attacks after several days, once the necessary preparations for intrusion have been completed. This particular scenario poses a great challenge that traditional non-incremental methods are unable to handle.

Algorithm 1 IL-IDS

Input: data stream x;
Parameters: embedding d, decision risk λ_{PP}, λ_{BP}, λ_{NP}, λ_{NN}, λ_{BN}, λ_{PN};
Output: class label y

Initialize VAE and SAE model parameters
Select data fields
Construct init_set
for step in IncrementalStep **do**
  x $\leftarrow$ text_to_vector(data stream x)
  for each epoch $i = 1, 2, ..., 100$ **do**
    z, $h \leftarrow$ VAE(x); // Feature learning
    $L_{vae} \leftarrow MSE + D_{KL}(z|x)$ // Reconstruction loss of VAE
  end for
  for each epoch $i = 1, 2, ..., t$ **do**
    Train classifier SAE
    Finetune SAE with random selected resampling 60%
  end for
  Classification probability $p(C|x) \leftarrow softmax(SAE(h))$
  Calculate the values of α and β according to formula (19)-(21)
  $y \leftarrow$ Three-way decision according to formula (24)-(26)
  if $BND(C)$ is not null **then**
    put $x \in BND(C)$ into the next step
  end if
end for
if $BND(C)$ is not null **then**
  $y \leftarrow$ Threat intelligence of $x \in BND(C)$
end if
return y

Figure 2 illustrates the framework of our proposed approach, which comprises three primary components: the feature extractor, the intrusion detector, and the three-way decision module. In the following subsections, we provide a comprehensive explanation of each component, delving into their respective functionalities, methodologies, and intricacies.

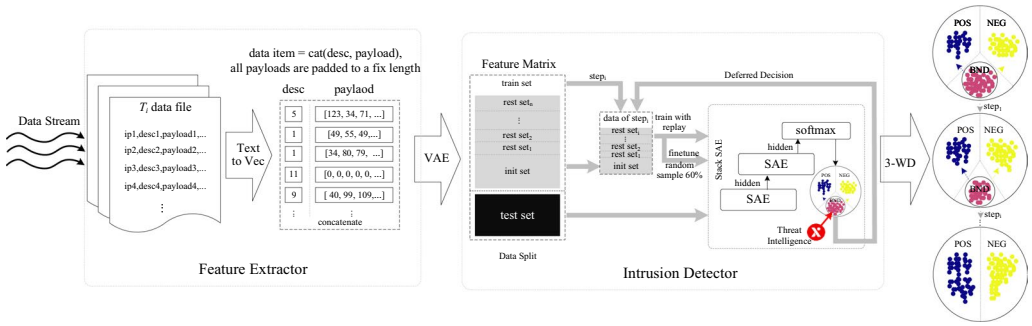


Fig. 2 Framework of the proposed method. "3-WD" means three-way decision

Data pre-processing

In the context of this international activity, a significant volume of data streams arrives incrementally within minutes. During the pre-processing phase, various attributes such as data item identities, timestamps, source IPs, source ports, destination IPs, destination ports, payloads, network protocols, and descriptions are extracted and merged to construct a data file in chronological order. Additionally, we have devised a text-to-vector algorithm specifically tailored for converting the input textual data into vectorized items. This algorithm operates at the character level, parsing the text based on ASCII code (Hieronymus 1993). Notably, our observations indicate that the length of payloads typically ranges from 2 to 1314, with a majority falling within a length of 500 characters. To ensure consistency in the representation, we assign the number 0 as a meaningless value and increase the ASCII code for each character by 1. For instance, the control character "NUL" in ASCII code, which corresponds to 0, is assigned a value of 1 in our proposed method. All data items are padded with 0 to reach the same length of 1314 characters because the input dimensions for VAE needs to be uniform. The decision to parse input data at character level, as opposed to word level commonly used in other natural language processing tasks, is driven by the presence of single characters with special meaning within the payload. Examples of such characters are provided in Table 2.

Feature extractor

Following the data pre-processing stage, we proceed to train a feature extractor using a VAE. VAE is a sophisticated model that employs variational inference to learn the probability distribution of input data. It exhibits the ability to generate high-quality reconstructions of the original data and extract underlying feature matrices during the rebuilding process. Given its flexibility in learning data features and reconstructing inputs, VAE proves to be highly suitable for handling the diverse and complex data encountered in real-world activities. Consequently, we choose VAE as our feature extractor over other neural network models. The structure of our feature extractor is symmetric, comprising a two-layer gate-recurrent-unit architecture. For further details on VAE, we refer readers to this paper (Doersch 2016). During the VAE training process, the mean square loss (MSE) (Ye and Barg 2017) and Kullback–Leibler divergence (KL-divergence) (Bu et al 2018) are employed as loss functions. These loss functions can be computed using the following formulas:

Table 2 Examples of some special characters in payload

Payload Sample	Explanation
s:n: select * from users where id=1 or ":[" or 1=1	"*" means all content that matched; ":[" implies that the attacker is trying to inject a string or a command.
a"or/*g*/345=345— #	"/*g*/" is a multi-line comment, "#" is a comment delimiter, which is used to comment out the rest of the line.
XOR(if(now()=sysdate(),sleep(6*6-30),0))OR	"*" means multiply
"apiSelectId":"————","id":"1' or '%1%' like (updatexml(0x3a,concat(1,(select current_user)),1)) or '%%' like '"	A json-like object, "%" presents any number of characters, and ""is used to wrap a string value to pass its wrapped contents as a parameter to the SQL query. ("————" is a substitute character for sensitive information instead of real content)

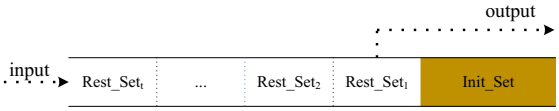


Fig. 3 The queue structure of rest sets

$$MSE = \frac{\sum_{i=1}^{n}(z-x)^2}{n}, \tag{27}$$

$$D_{KL}(z|x) = -\sum_{x} p(x)\log\frac{q(z)}{p(x)}, \tag{28}$$

$$L_{vae} = MSE + D_{KL}(z \mid x) \tag{29}$$

where z is the output of VAE, namely the re-constructed data. $p(x)$ and $q(z)$ are the probability distribution of original data x and re-constructed data z respectively.

Intrusion detector

In order to streamline the intrusion detection model, we simplify the process by reusing a major part of the autoencoder model from the feature extractor. Specifically, we remove the mean-pooling layer located between the encoder and decoder, resulting in a sparse autoencoder (SAE). Additionally, a sigmoid layer and a softmax layer are added to the classification output in the decoder. SAE is a specialized type of autoencoder specifically designed to handle sparse data, such as text and images, where a majority of the elements are zero. By leveraging SAE, our model can effectively learn the characteristics of sparse data and preserve their sparsity. As mentioned in Sect. , the inputs of our model are padded at the character level, and there are inherent correlations among different letters within the same word, resulting in a high level of sparsity. Consequently, training the intrusion detector based on SAE is a reasonable choice.

Moreover, in our approach, we designate the first 5000 items of the input as the initial set for the incremental training of the intrusion detector. The subsequent items are then divided into data blocks of equal size, forming the rest_set$_t$ at timestamp t. These rest_sets, of smaller size compared to the initial set, are stored in a queue data structure and sequentially passed through the intrusion detector, as depicted in Fig. 3. While the initial set is optional, it plays a crucial role in mitigating the issue of catastrophic forgetting, which is a common challenge in incremental learning (Van de Ven et al 2022; Wang et al 2023; Sun et al 2022; Huang and Ma 2023; Sharma and Arora 2021; Rebuffi et al 2017). The intrusion detector consists of a stacked two-layer SAE, where the output of the preceding SAE is served as the input for the subsequent SAE. To further refine the SAE parameters, we conduct parameter fine-tuning by randomly sampling 60% of the training set during each incremental step. The detailed structure of the intrusion detector is presented in the second part of Fig. 2. Notably, our model undergoes updates on an hourly basis.

Three-way decision module

As discussed in Sect. , we introduce a boundary for data samples lacking sufficient and complete information. This approach allows for deferring decisions on challenging examples until additional data with similar features becomes available. By doing so, we address a limitation of the two-way decision framework, wherein decisions must be made even when taking risks. Referring to formula (13) and (14), the values of λ_{PP} and λ_{BP} are observed to be close or equal. In real practice, λ_{PP} is the risk of classifying normal behavior as positive, it should be equal or near to zero because of the right decision. On the other hand, the decision to classify normal behavior as a boundary incurs only a minor delay in making a second decision, so λ_{BP} can be set to a value equal to or slightly larger than λ_{PP}. λ_{NP} represents the risk of incorrect decision where normal behavior is misclassified as malicious. Such misclassifications can lead to the blocking of legitimate users and result in unpleasant user experience. Similarly, λ_{NN} represents the correct decision, while λ_{BN} involves assigning attack behavior to the boundary, which carries potential risks. However, given the potential for significant damage if attack behavior is mistakenly classified as normal behavior, λ_{PN} must be set to a larger value than λ_{NN} and λ_{BN} to prioritize accurate detection and minimize false positives.

Following the formulation in equation (13) and (14), we set the values of λ_{PP}, λ_{BP}, λ_{NP}, λ_{NN}, λ_{BN} and λ_{PN} to 0, 0.1, 0.3, 0, 0.2 and 0.4 respectively. As depicted in Fig. 2 and Algorithm 1, our Intrusion Detector utilizes the data replay training approach (Van de Ven et al 2022), allowing for the re-examination of challenging samples that pose difficulties in reaching conclusive decisions upon their arrival. Moreover, the number of data items in the boundary is less than the number of input data in several orders, so it is convenient for us to integrate threat intelligence into boundary[1]. In real-world deployment, instances that cannot be classified by the intrusion

[1] The threat intelligence sources we used include our internal threat intelligence database and an external provider, Threatbook. (URL: https://x.threatbook.com/)

detector are evaluated by threat intelligence and subsequently incorporated back into the intrusion detector as labeled data. This flexibility is an additional advantage of our proposed method, enabling the incorporation of threat intelligence in the decision-making process.

Experiments and analysis

This section provides an overview of the real-world deployment results of our proposed intrusion detection method. It encompasses a comprehensive examination of model parameters, the training procedure, and the achieved detection outcomes.

Data analysis

The data streams are coming incrementally from honeypots deployed for the activity. The honeypots are simulated web pages of every entrance into different part of official websites designed for the activity, and the only difference between the entrance of the official websites and our deployed honeypots is the URL. The honeypots are invisible to normal users, which can only be found by intranet scanning. In the context of incrementally arriving data streams, we extract significant features and organize them into JSON files. These extracted features encompass various aspects, including the identity of data items, timestamps, source IPs, source ports, destination IPs, destination ports, payloads, network protocols, and descriptions. The threat level is served as the training label, with threat levels labeled as low, medium, or high. Low threat level corresponds to normal behavior, while high threat level indicates attack behavior. Medium threat level is designated as the boundary category, which may be normal or malicious. According to our statistic, on the final day of the international activity, 80.3% of the data items are with a low threat level, 18.8% of the data items are with a medium threat level, and 0.9% of the data items are with a high threat level. Additionally, there were 21.8% incomplete data items lacking one or more essential features. Throughout the activity, our objective is to discern whether the source IPs are associated with malicious behavior or not, based on the available confined data streams.

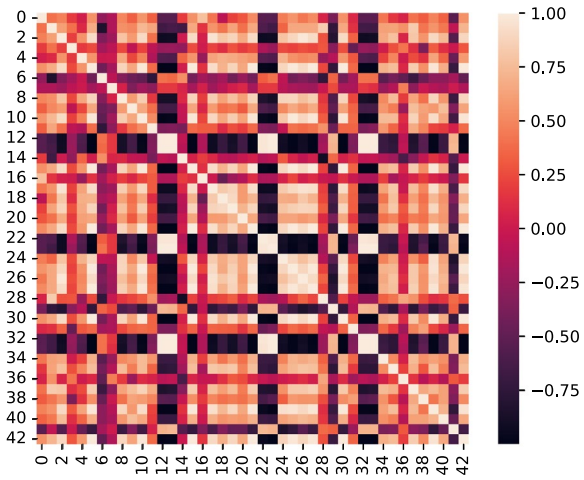


Fig. 5 Relation heatmap of feature matrix. Note that only the first 42 columns of feature matrix are shown because of its large volume

Results of feature learning

A symmetric VAE is employed to extract hidden features from the input data streams through a training process spanning 100 epochs. The learning rate is initialized at 3.5e−4 and linearly increased to 3.5e−3. At the 40*th* epoch, the learning rate is reset to 3.5e−4, and further decreased to 3.5e−5 at the 70*th* epoch. The hidden dimension of VAE is configured as 256, and a batch size of 128 is utilized. The loss function comprises both mean square loss and Kullback-Leibler divergence, as shown in formula (29). Figure 4 provides an illustration of the reconstruction loss,

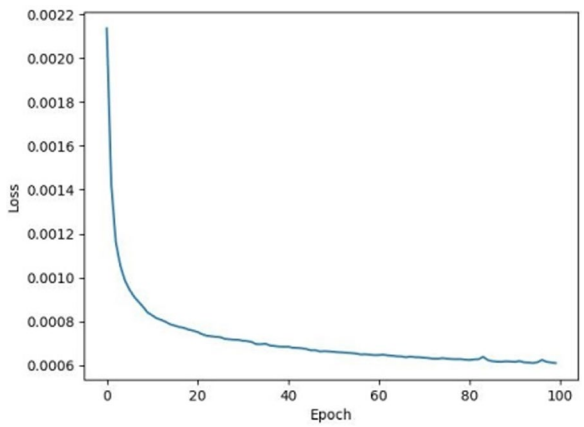


Fig. 4 Reconstruction loss of feature extractor

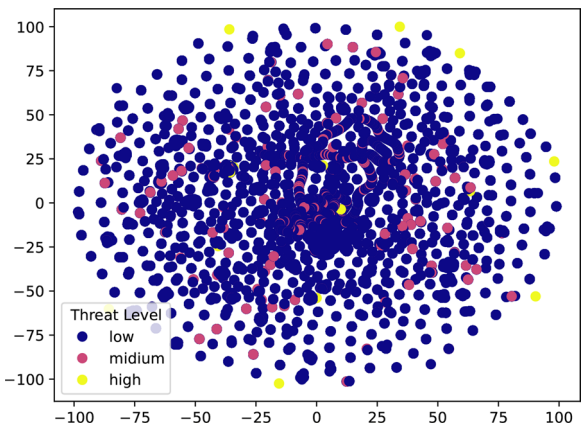


Fig. 6 2-D t-SNE visualization of input data

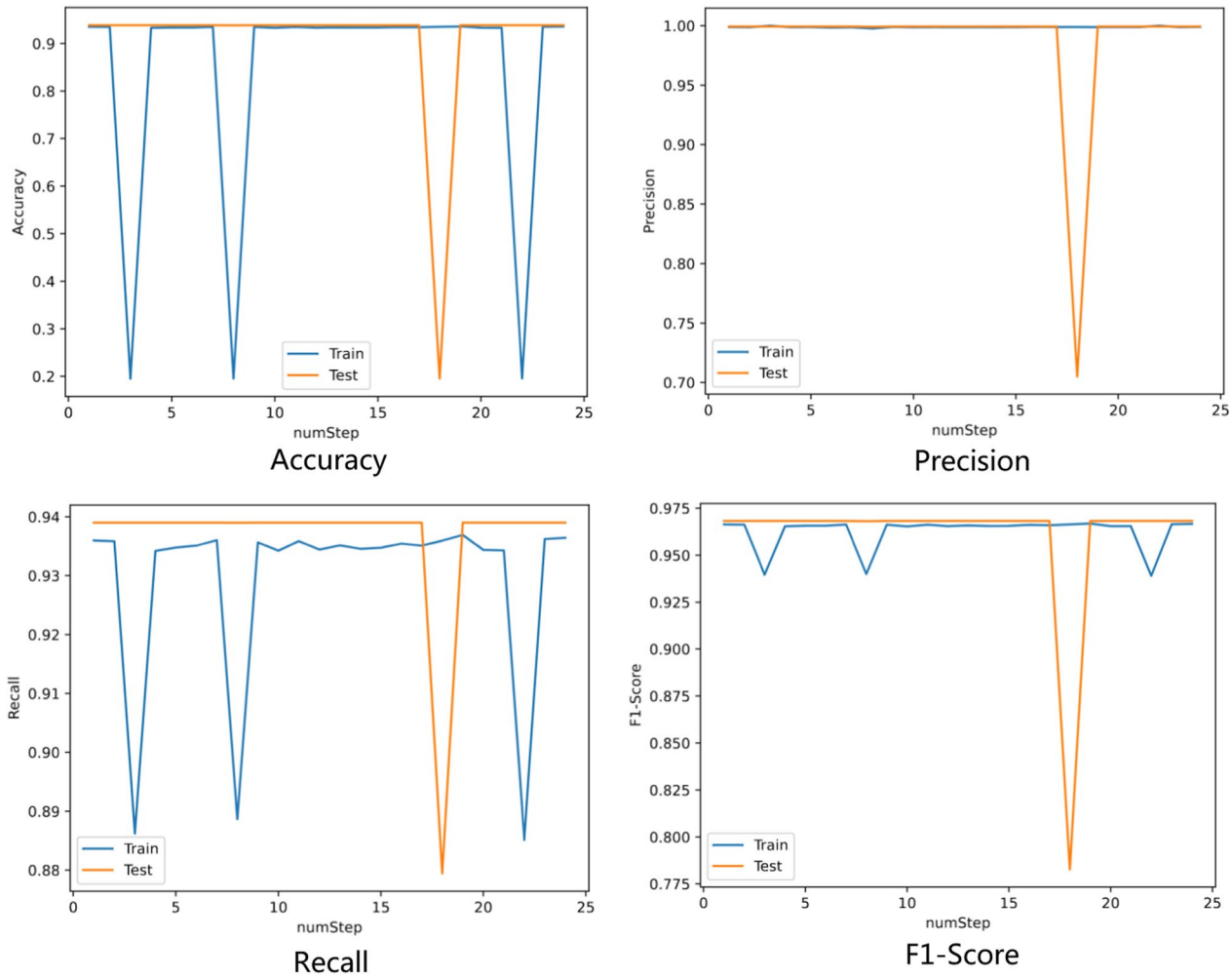


Fig. 7 Real deployment results of our method

demonstrating that the reconstruction loss is sufficiently low, indicating that the hidden layer of VAE can effectively capture input data features because the reconstructed data is rebuild from the output of hidden layer. Figure 5 presents the correlation heatmap of the feature matrix. Notably, the correlations among those features are minimal, indicating satisfactory compression effects. Additional pair plot examples can be found in Appendix C. In order to gain a comprehensive understanding of the input data, a two-dimensional t-SNE embedding result is visualized in Fig. 6. As can be observed from Fig. 6, the data points representing medium threat levels exhibit confusion with benign and attack data, posing a significant challenge for intrusion detection.

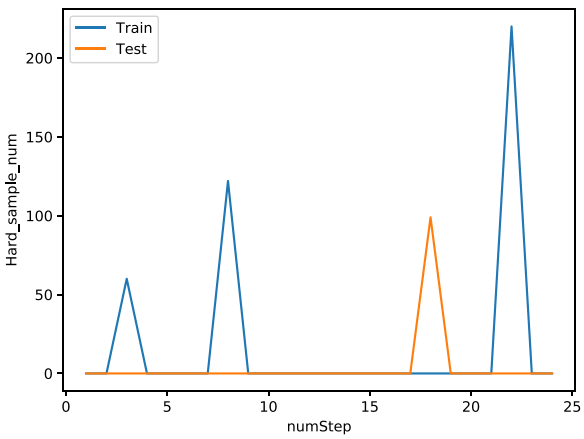


Fig. 8 Hard example number. Hard examples are those data items that need to refer to threat intelligence

Results of intrusion detection

During the evaluation of our model, we employ four commonly used evaluation metrics in the field of attack detection: accuracy, precision, recall and F1-score, their computational formula are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{30}$$

Table 3 Comparison results with Deeplog

	Accuracy	Precision	Recall	F1-score
Deeplog	**0.9521**	0.9466	**0.9521**	0.9424
IL-IDS(ours)	0.9384	**0.9993**	0.9390	**0.9683**

$$Precision = \frac{TP}{TP + FP}, \tag{31}$$

$$Recall = \frac{TP}{TP + FN}, \tag{32}$$

$$F1\text{- }score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{33}$$

where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative. The results of real deployment of IL-IDS is shown in Fig. 7. As can be seen that the evaluation matrix displays high scores, which can effectively help us find out attack behaviors. However, fluctuations in the results are observed. These fluctuations can be attributed to two main factors. Firstly, the presence of unseen data in the data stream leads to the

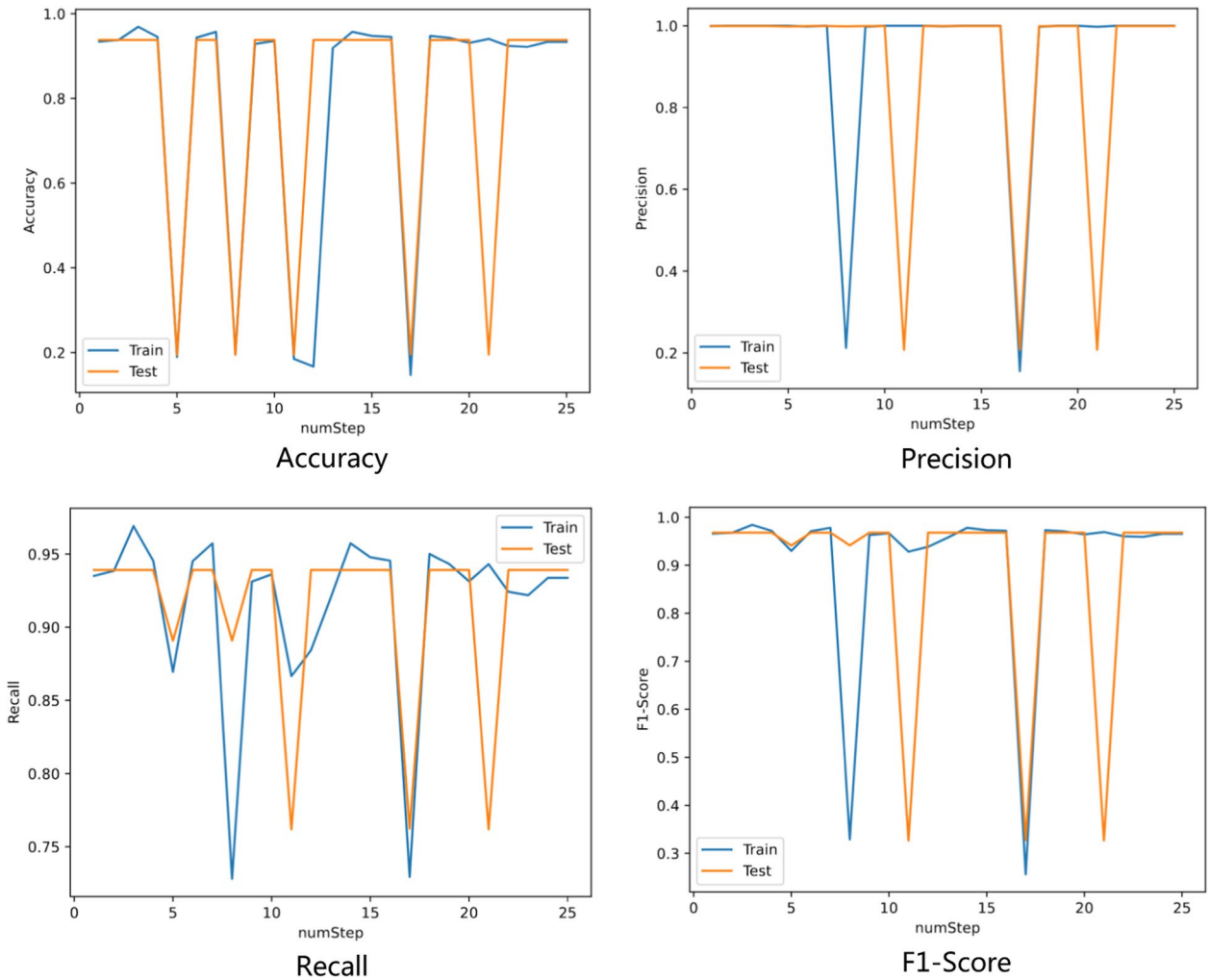


Fig. 9 Comparison results without initial set

shift of data distributions, which necessitates some time for the model to adapt to. As depicted in Fig. 7, the intrusion detector demonstrates the ability to quickly adapt to novel data within a single step, typically within an hour. Secondly, the phenomenon of catastrophic forgetting in incremental learning contributes to the fluctuations. Addressing catastrophic forgetting is beyond the scope of this article and is intended as a future direction for our research.

In addiction, we conduct a statistical analysis of the challenging data samples that our model is unable to classify due to incomplete or insufficient information. These samples are estimated and labeled by referring to threat intelligence. Figure 8 illustrates that the quantity of data which requires manual handling is within an acceptable range. Moreover, the occurrence of these challenging samples is limited to a few steps, representing a significantly smaller proportion in several orders of magnitude when compared with the original input data.

Comparison

Deeplog (Du et al 2017) represents a state-of-the-art approach in intrusion detection, assuming complete data availability. In order to assess the performance of our proposed method, we compare it with Deeplog using data from the final day of the international activity, as presented in Table 3. Examining the results in Table 3, it is evident that our approach achieves higher precision by incorporating threat intelligence through a three-way decision module. Additionally, our method yields comparable results to Deeplog in terms of accuracy and recall, and even outperforms Deeplog in terms of F1-score. Considering that Deeplog is trained on the entire dataset, our objective is to design an incremental learning method for real-world scenarios that can achieve similar or superior results to traditional approaches.

Furthermore, we conducted comparative experiments to assess the performance of the intrusion detector when trained without an initial set. The results of these experiments are presented in Fig. 9. In comparison to the results depicted in Fig. 7, it becomes evident that the detection model exhibits greater stability when trained with an initial set. This finding underscores the significance of an initial set in an incremental learning setting, particularly when confronted with incoming data without prior knowledge or information.

Conclusion and future work

In this paper, we propose a novel approach for real-world intrusion detection within the context of an international activity. Diverging from traditional two-way decision methods, our approach leverages incremental processing of diverse data streams and utilizes a three-way decision framework. To preprocess the data, we develop a text-to-vector algorithm that operates at the character level, allowing for effective parsing of the input data. Subsequently, we employ a VAE to learn and compress the feature matrix. For classification purposes, we utilize a two-layer SAE as the primary classifier, which is augmented with threat intelligence through the boundary of three-way decision. To demonstrate the effectiveness of our method, we present the results obtained from its actual deployment and provide comparisons with existing approaches. The experimental findings highlight the accuracy and efficiency of our proposed approach, particularly in scenarios with confined data streams, thereby establishing its efficacy for real-world intrusion detection.

In future work, research on real-time intrusion detection in real-word deployment will be carried out. On the basis of intrusion detection, the attack categories are classified in detail to provide defense reference for network security workers. Further, we devote ourselves to the study of unknown attack detection based on incremental learning, which aims to dealing with the intricate and dynamic network intrusions.

Appendix A Proof of formula (22)

To simplify, Let $A = \lambda_{PP}$, $B = \lambda_{BP}$, $C = \lambda_{NP}$,$D = \lambda_{NN}$, $E = \lambda_{BN}$, $F = \lambda_{PN}$, then formula (22) can be written as:

$$(F - E)(C - B) > (E - D)(B - A). \tag{A1}$$

As analyzed in Section ., in practical usage, A is slightly smaller than B, but A and B are strictly smaller than C, the margin between A and B is smaller than the margin between B and C. The relationships between D, E and F are the same as the relationships between A, B and C. So another relationship within these six parameters can be concluded:

$$0 \le A \le B < C \le 1,\ \ 0 \le D \le E < F \le 1, \tag{A2}$$

$$(C - B) > (B - A),\ (F - E) > (E - D), \tag{A3}$$

$$(C - B) \approx (F - E),\ (B - A) \approx (E - D),\ (C - A) \approx (F - D). \tag{A4}$$

Let $f(x) = x^2, x_1 = (C - B), x_2 = (F - E), x_3 = (E - D)$, $x_4 = (B - A)$, according to (A2)-(A4), (A1) can be approximated as:

$$(F - E)(C - B) = x_1 x_2 \approx x_1^2, \tag{A5}$$

$$(E - D)(B - A) = x_3 x_4 \approx x_3^2. \tag{A6}$$

$x_1^2 > x_3^2$ because $x_1 > x_3$ and $f(x) = x^2$ is an increasing function, so (A1) is proved.

Note that (A5) and (A6) is not strictly mathematically true, but it is the case in the real use of three-way decision.

Appendix B Proof of formula (23)

Under the same settings as Appendix A, formula (19)-(21) can be re-written as:

$$\alpha = \frac{F-E}{(F-E)+(B-A)}, \tag{A7}$$

$$\beta = \frac{E-D}{(E-D)+(C-B)}, \tag{A8}$$

$$\gamma = \frac{F-D}{(F-D)+(C-A)}. \tag{A9}$$

Therefore, we can prove formula (23) $0 \le \beta < \gamma < \alpha \le 1$ step by step:

Step 1: Prove $0 \le \beta < 1$:

As proved in Appendix A, since $0 \le A \le B < C \le 1$ and $0 \le D \le E < F \le 1$, we have:

$$0 \le (E-D) < (F-D), \tag{A10}$$

$$0 < (C-B) \le 1, \tag{A11}$$

therefore, both the numerator and denominator of β are non-negative, implying $0 \le \beta \le 1$, and because $(C-B) > 0$, so $\beta < 1$.

Step 2: Prove $0 < \gamma < 1$:

Since $0 \le A \le B < C \le 1$ and $0 \le D \le E < F \le 1$, we have:

$$0 < (F-D) < 1, \tag{A12}$$

$$0 < (C-A) < 1, \tag{A13}$$

therefore, both the numerator and denominator of γ are positive, so $0 < \gamma$, and because $(C-A) > 0$, so the denominator of γ is larger than the numerator of γ, so $\gamma < 1$, thus $0 < \gamma < 1$.

Step 3: Prove $0 < \alpha \le 1$:

Again, since $0 \le A \le B < C \le 1$ and $0 \le D \le E < F \le 1$, we have:

$$0 < (F-E) < 1, \tag{A14}$$

$$0 \le (B-A) < 1, \tag{A15}$$

therefore, both the numerator and denominator of α are positive, so $0 < \alpha$, and because $(B-A) \ge 0$, when $(B-A) = 0, \alpha = 1$, ensuring $0 < \alpha \le 1$.

Step 4: Prove $\gamma < \alpha$:

Express $\alpha - \gamma$, subtracting the expression for α and γ, we get:

$$\begin{aligned}\alpha - \gamma &= \frac{(F-E)\times[(F-D)+(C-A)] - (F-D)\times[(F-E)+(B-A)]}{[(F-E)+(B-A)]\times[(F-D)+(C-A)]} \\ &= \frac{(F-E)(C-A)-(F-D)(B-A)}{[(F-E)+(B-A)]\times[(F-D)+(C-A)]},\end{aligned} \tag{A16}$$

because $(F-E)>0$, $(F-D)>0$ and $(C-A)>0$, the denominator of $\alpha-\gamma$ is positive, combined with (A2)-(A4), the numerator of $\alpha-\gamma$ can be rearranged as:

$$\begin{aligned}&(F-E)(C-A)-(F-D)(B-A)\\&>(E-D)(C-A)-(F-D)(B-A)\\&\approx(B-A)\times[(C-A)-(F-D)]\quad //\because(E-D)\approx(B-A)\\&\approx 0\quad \#\because(C-A)\approx(F-D)\end{aligned}\tag{A17}$$

namely $(F-E)(C-A)-(F-D)(B-A)>0$, thus $\alpha-\gamma>0$, meaning $\gamma<\alpha$.

Step 5: Prove $\beta<\gamma$:

Similarly, express $\gamma-\beta$, subtracting the expressions for γ and β, we get:

$$\begin{aligned}\gamma-\beta&=\frac{(F-D)\times[(E-D)+(C-B)]-(E-D)\times[(F-D)+(C-A)]}{[(F-D)+(C-A)]\times[(E-D)+(C-B)]}\\&=\frac{(F-D)(C-B)-(E-D)(C-A)}{[(F-D)+(C-A)]\times[(E-D)+(C-B)]}\end{aligned}\tag{A18}$$

because $(F-D)>0$, $(C-A)>0$ and $(C-B)>0$, the denominator of $\gamma-\beta$ is positive, combined with (A2)-(A4), the numerator of $\gamma-\beta$ can be rearranged as:

$$\begin{aligned}&(F-D)(C-B)-(E-D)(C-A)\\&>(F-D)(B-A)-(E-D)(C-A)\\&\approx(B-A)\times[(F-D)-(C-A)]\quad //\because(E-D)\approx(B-A)\\&\approx 0\quad \#\because(F-D)\approx(C-A)\end{aligned}\tag{A19}$$

namely $(F-D)(C-B)-(E-D)(C-A)>0$, thus $\gamma-\beta>0$, meaning $\gamma>\beta$.

Appendix C Feature matrix

The pair plot of learned feature matrix is presented in Fig. 10. It can be seen that the distribution of features vary from each other, and there is almost no correlation between those features.

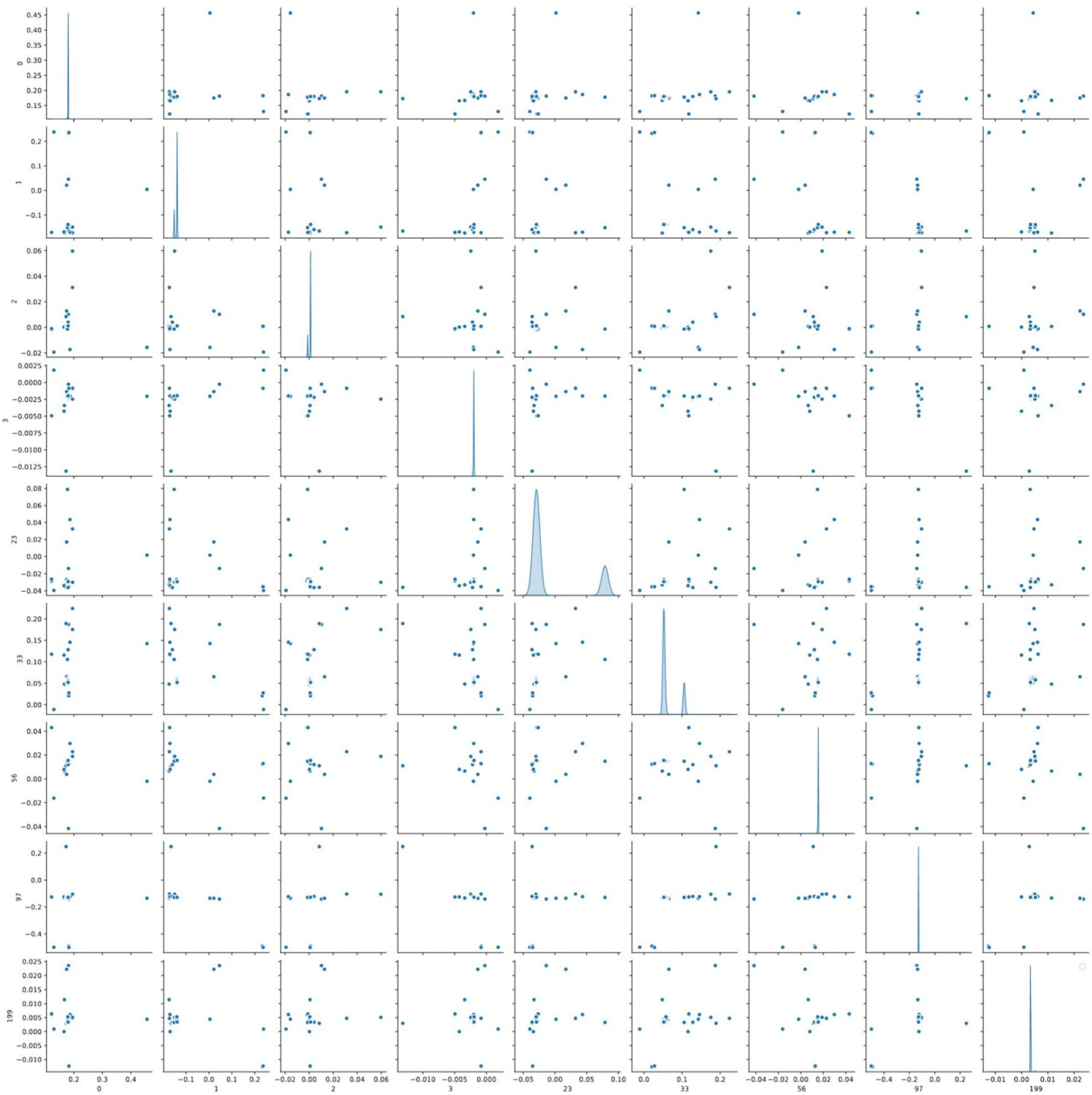


Fig. 10 Pair plot of learned feature matrix. Note that only the random selected columns of feature matrix are shown because of its large volume

Abbreviations

IL: Incremental Learning
IDS: Intrusion Detection Systems
VAE: Variation Autoencoder
TIL: Task-incremental Learning
DIL: Domain-incremental Learning
CIL: Class-incremental Learning
SAE: Sparse Autoencoder

Authors' contributions

Conceptualization, methodology, software, implementation, validation, writing: Jianming Li, Ye Wang. Revising, validation: Liyi Zeng, Wenying Feng. Environment deployment: Xiao Jing, Cui Luo. Conceptualization, supervision, attack analysis: Yan Jia, Zhaoquan Gu, Binxing Fang. All authors read and approved the final manuscript.

Funding

This work has received financial support from grants of the Natural Science Foundation of China (No. 61902082) and Major Key Project of PCL (Grant Nos. PCL2024A05; PCL2024AS102).

Availability of data and materials

Because the original data and materials are related to the real traffic during the international activity, it is unavailable for security and privacy reasons. The processed data will be made available on request.

Declarations

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Received: 9 August 2024 Accepted: 6 January 2025
Published online: 29 October 2025

References

Bansal A, Mahapatra S (2017) A comparative analysis of machine learning techniques for botnet detection. In: Proceedings of the 10th international conference on security of information and networks, pp 91–98
Bu Y, Zou S, Liang Y et al (2018) Estimation of kl divergence: optimal minimax rate. IEEE Trans Inf Theory 64(4):2648–2674
Clegg BA, DiGirolamo GJ, Keele SW (1998) Sequence learning. Trend Cognit Sci 2(8):275–281
Devecsery D, Chow M, Dou X, et al (2014) Eidetic systems. In: 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14), pp 525–540
Doersch C (2016) Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908
Du M, Li F, Zheng G, et al (2017) Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pp 1285–1298
Duchi J, Mackey L, Wauthier F (2024) Anomaly detection for asynchronous and incomplete data
Farrukh YA, Khan I, Wali S et al (2022) Payload-byte: A tool for extracting and labeling packet capture files of modern network intrusion detection datasets. 2022 IEEE/ACM International Conference on Big Data Computing. IEEE, Applications and Technologies (BDCAT), pp 58–67
Geng Y, Li Y, Zhang S (2021) Research on multi-granularity intrusion detection algorithm based onsequential three-way decision. In: Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering, pp 1154–1160
Han D, Wang Z, Chen W, et al (2021) Deepaid: Interpreting and improving deep learning-based anomaly detection in security applications. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp 3197–3217
Han D, Wang Z, Chen W, et al (2023) Anomaly detection in the open world: Normality shift detection, explanation, and adaptation. In: NDSS
Hassan M, Haque ME, Tozal ME et al (2021) Intrusion detection using payload embeddings. IEEE Access 10:4015–4030
Hieronymus JL (1993) Ascii phonetic symbols for the world's languages: Worldbet. J Int Phonet Assoc 23(3):12–54
Huang Y, Ma M (2023) Ill-ids: an incremental lifetime learning ids for vanets. Comput Secur 124:102992
Ji Y, Lee S, Fazzini M, et al (2018) Enabling refinable {Cross-Host} attack investigation with efficient data flow tagging and tracking. In: 27th USENIX Security Symposium (USENIX Security 18), pp 1705–1722
Ke Z, Liu B (2022) Continual learning of natural language processing tasks: A survey. arXiv preprint arXiv:2211.12701
Khoury J, Upthegrove T, Caro A, et al (2020) An event-based data model for granular information flow tracking. In: 12th International Workshop on Theory and Practice of Provenance (TaPP 2020)
Kim S, Jung C, Jang R, et al (2023) A robust counting sketch for data plane intrusion detection. In: 30th Annual Network and Distributed System Security Symposium, NDSS 2023, The Internet Society
Latif S, Boulila W, Koubaa A et al (2024) Dtl-ids: an optimized intrusion detection framework using deep transfer learning and genetic algorithm. J Network Compute Appl 221:103784
Lippmann R, Haines JW, Fried DJ et al (2000) The 1999 darpa off-line intrusion detection evaluation. Comput Network 34(4):579–595
Lippmann RP, Fried DJ, Graf I, et al (2000b) Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. In: Proceedings DARPA Information survivability conference and exposition. DISCEX'00, IEEE, pp 12–26
Moustafa N, Slay J (2015) Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 military communications and information systems conference (MilCIS), IEEE, pp 1–6
Moustakidis S, Karlsson P (2020) A novel feature extraction methodology using siamese convolutional neural networks for intrusion detection. Cybersecurity 3(1):16
Nauman M, Azam N, Yao J (2016) A three-way decision making approach to malware analysis using probabilistic rough sets. Inf Sci 374:193–209
Patcha A, Park JM (2007) Network anomaly detection with incomplete audit data. Comput Network 51(13):3935–3955
Ramezany S, Setthawong R, Tanprasert T (2022) A machine learning-based malicious payload detection and classification framework for new web attacks. 2022 19th International Conference on Electrical Engineering/Electronics. Computer, Telecommunications and Information Technology (ECTI-CON), IEEE, pp 1–4
Rebuffi SA, Kolesnikov A, Sperl G, et al (2017) icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 2001–2010
Rizvi S, Scanlon M, McGibney J, et al (2022) Deep learning based network intrusion detection system for resource-constrained environments. In: International Conference on Digital Forensics and Cyber Crime, Springer, pp 355–367
Sharafaldin I, Lashkari AH, Ghorbani AA et al (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp 1:108–116
Sharma N, Arora B (2021) Review of incremental and online learning methods for network anomaly detection. Turkish Online Journal of Qualitative Inquiry 12(7)
Sun Q, Lyu F, Shang F et al (2022) Exploring example influence in continual learning. Adv Neural Inf Process Syst 35:27075–27086
Tavallaee M, Bagheri E, Lu W, et al (2009) A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications, Ieee, pp 1–6
Van de Ven GM, Tuytelaars T, Tolias AS (2022) Three types of incremental learning. Nature Mach Intell 4(12):1185–1197
Wang L, Zhang X, Su H, et al (2023) A comprehensive survey of continual learning: Theory. Method and Application 2302
Wang Z, Li Y, Shen L, et al (2024) A unified and general framework for continual learning. arXiv preprint arXiv:2403.13249
Wu Z, Gao P, Cui L et al (2021) An incremental learning method based on dynamic ensemble rvm for intrusion detection. IEEE Trans Network Service Managem 19(1):671–685
Yang J, Chen X, Chen S et al (2021) Conditional variational auto-encoder and extreme value theory aided two-stage learning approach for intelligent fine-grained known/unknown intrusion detection. IEEE Trans Inf Foren Secur 16:3538–3553
Yao Y (2008) Probabilistic rough set approximations. Int J Approx Reason 49(2):255–271
Yao Y (2009) Three-way decision: an interpretation of rules in rough set theory. In: Rough Sets and Knowledge Technology: 4th International Conference, RSKT 2009, Gold Coast, Australia, July 14-16, 2009. Proceedings 4, Springer, pp 642–649
Yao Y (2010) Three-way decisions with probabilistic rough sets. Inf Sci 180(3):341–353
Yao Y (2011) The superiority of three-way decisions in probabilistic rough set models. Inf Sci 181(6):1080–1096
Yao Y, Zhao Y (2008) Attribute reduction in decision-theoretic rough set models. Inf Sci 178(17):3356–3373
Ye M, Barg A (2017) Asymptotically optimal private estimation under mean square loss. arXiv preprint arXiv:1708.00059
Ye Q, Wu X, Yan B (2010) An intrusion detection approach based on system call sequences and rules extraction. In: 2010 2nd International Conference on E-business and Information System Security, IEEE, pp 1–4
Zhang S, Li Y, Du X (2020) An intrusion detection approach based on autoencoder and three-way decisions. In: Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering, pp 495–500
Zhou D, Wang F, Ye H et al (2023) A review of class incremental learning algorithms based on deep learning. (in chines). J Comput 8(46):1577–1605

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.