Brodogradnja Volume 77 Number 2 (2026) 77205

University of Zagreb Faculty of Mechanical Engineering and Naval Architecture

journal homepage: www.brodogradnja.fsb.hr

Brodogradnja

An International Journal of Naval Architecture and Ocean Engineering for Research and Development

Research on deep learning-based network intrusion detection methods for smart ships

Ziming Lu [1], Hong Zeng [1,2,], Zhilong Zheng [1], Caowei Li [1], Xiao Dong [1]*

[1] *Marine Engineering College, Dalian Maritime University, Dalian 116026, China*
[2] *Dalian Maritime University Smart Ship Limited Company, Dalian 116026*

ARTICLE INFO

Keywords:
Smart Ship
Cybersecurity
Intrusion Detection
Transformer
KAN

ABSTRACT

As the development of intelligent ships rapidly progresses, the importance of maritime network security escalates. Intrusion Detection Systems (IDS) serve as a pivotal defense mechanism against cyberattacks targeting these vessels. To augment the efficacy of IDS in detecting anomalous network traffic, this paper introduces a novel intrusion detection method utilizing both Transformer and Kolmogorov-Arnold Networks (KAN), designated as Transformer-KAN. This approach effectively mitigates the limitations of traditional intrusion detection algorithms, which include inadequate feature extraction capabilities and the excessive parameterization required to delineate complex patterns. The proposed method employs a Transformer encoder to discern long-range dependencies within input sequences, while KAN layers enhance the model's ability to approximate complex patterns through non-linear transformations. This integrative strategy ensures elevated accuracy for the intrusion detection algorithm and proves particularly beneficial in intelligent ship environments, where training data may be sparse or computational resources limited. Experimental results confirm that the Transformer-KAN model attains exceptionally high accuracy in both binary and five-class classification tasks, outperforming the capabilities of standalone Transformer and KAN models. Comparative analyses with conventional algorithms—such as Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), Deep Belief Networks (DBN), and Support Vector Machines (SVM)-further substantiate the superior effectiveness of the proposed method. This research establishes a new theoretical framework and provides a direction for the practical implementation of network security measures in contemporary intelligent ships.

1. Introduction

The 2020 Edition of the "Guidelines on Requirements and Safety Assessment for Ship Network Systems," published by the China Classification Society (CCS), underscores the increasing prominence of cybersecurity within the maritime sector. This rising concern is attributed to the burgeoning integration of digitalization, intelligence, and networking in maritime vessels, which concurrently escalates associated cyber risks [1]. Intelligent ship network systems, emblematic of these technological advancements, are

* Corresponding author.
E-mail address: zenghong@dlmu.edu.cn

https://doi.org/10.21278/brod77205
Received 16 May 2025; Received in revised form 22 September 2025; Accepted 15 October 2025
Available online 5 November 2025
ISSN 0007-215X; eISSN 1845-5859

integral to contemporary maritime operations [2]. The seamless operation of such intelligent vessels is contingent upon the harmonized coordination of various onboard systems. The interconnectivity between these subsystems is predominantly facilitated through networks, encompassing both Ethernet and onboard wireless networks. Moreover, to maintain operational efficacy while at sea, these vessels must also engage in communication with other ships, primarily through radio and satellite links. The network architecture of an intelligent ship, therefore, encompasses both internal subsystems and external communication links, which are imperative for the swift and secure transmission and processing of information. Thus, the accuracy of information transmission is crucial for maintaining the operational integrity of intelligent ships [3].

In recent years, the severity of maritime cyberattacks has markedly increased. For example, in 2016, the administrative systems of the Port of Oakland were compromised by a Denial-of-Service (DoS) attack. Subsequently, in the summer of 2017, the NotPetya ransomware, initially targeting Ukrainian entities, spread globally, significantly affecting A.P. Moller-Maersk terminals across approximately 80 locations and reportedly precipitating financial losses estimated at $300 million. In 2018, Iranian hackers allegedly infiltrated the systems of Austal, an Australian shipbuilding firm that provides naval vessels to both Australia and the United States, extracting ship designs and personnel details. The hackers also engaged in extortion, and the stolen data was subsequently disseminated on the dark web. Additionally, in 2020, Israel was purportedly involved in an attack on Iran’s Shahid Rajaee port, which resulted in extended disruptions to transportation and cargo flows. That same year, the IT systems of the Port of Kennewick succumbed to a ransomware attack, and the French container shipping company CMA CGM was similarly afflicted by Ragnar Locker ransomware, which disrupted several of its offices in China and compelled the shutdown of critical online services, including booking systems [4]. Such incidents of maritime cyberattacks not only inflict substantial economic losses on shipping firms and their personnel but also pose grave threats to maritime safety [5].

Contemporary maritime cybersecurity exhibits several distinct trends. Maritime systems are increasingly integrating advanced technologies such as the Internet of Things (IoT), big data, and artificial intelligence (AI), which are pivotal in the development of smart ports and autonomous vessels [6]. Although the digitalization and integration of AI confer operational advantages, they also considerably enlarge the cyber-attack surface. The integration of Information Technology (IT) and Operational Technology (OT) has rendered traditionally isolated industrial systems vulnerable to threats originating from the information domain [7]. Critical subsystems such as the Global Navigation Satellite System (GNSS) have been specifically identified as high-value targets for cyber-attacks, with their vulnerabilities receiving focused attention. Notably, GNSS spoofing and jamming attacks have emerged as significant threats [8].

Both academic circles and industry sectors acknowledge that cybersecurity monitoring is a crucial strategy for responding to cyber incidents. This approach enables real-time or near-real-time detection of attacks through automated systems [7]. In the realm of autonomous systems, deep learning (DL) models themselves have become targets; their susceptibility to adversarial attacks substantially compromises the reliability and security of intelligent systems. This issue is particularly pertinent to highly autonomous smart ships [9].

Enhancing maritime cybersecurity necessitates the adoption of bespoke protection strategies for intelligent vessels, which are vital for ensuring the safety and stability of the shipping industry [10]. Such strategies encompass comprehensive protection design, data encryption, and the securing of internal components. Network perimeter protection, for instance, the segmentation of ship networks into access buffer zones and service buffer zones, is a critical component of overall protection design [11]. To mitigate maritime threats effectively, it is imperative to establish a robust cybersecurity protection system for vessels. As a fundamental aspect of network boundary protection, intrusion detection systems (IDSs) play an essential role by facilitating the real-time detection of abnormal traffic and intrusive behaviors [12].

Recent advancements in big data, internet technologies, cloud computing, and GPU acceleration have significantly propelled the development of DL, culminating in the formulation of numerous neural network architectures [13, 14]. Presently, DL is extensively employed in the field of intrusion detection. Sarıkaya A.

introduced a novel method known as GRU-GBM, which utilizes the LightGBM gradient boosting framework for feature selection. This method assesses each feature through a secondary LightGBM classifier to ascertain the optimal feature set, employing a threshold-based approach. The features thus selected are subsequently processed by a gated recurrent unit (GRU) model to detect attacks [15]. Tian [16] proposed an intrusion detection method that enhances a deep belief network (DBN) integrated with LightGBM, using the advanced DBN for nonlinear feature extraction prior to classification via LightGBM. Du [17] employed an adaptive synthetic sampling method in conjunction with the Tomek-Links algorithm to resample and balance the training datasets. The study further explored the application of a self-attention mechanism and a bidirectional GRU network for intrusion detection, involving comparative experimentation across multiple models. Alkahtani et al. [18] utilized three sophisticated DL algorithms—convolutional neural network (CNN), long short-term memory (LSTM), and a hybrid CNN-LSTM model—for classifying intrusions. To simplify dataset complexity, particle swarm optimization (PSO) was harnessed for feature selection, with the selected features subsequently being processed through DL algorithms.

These studies demonstrate the capacity of DL methods to autonomously discern complex patterns embedded within data. By constructing nonlinear network models, DL is adept at managing high-dimensional, large-scale datasets and has proven markedly effective in detecting novel and intricate cyber-attacks. When contrasted with conventional intrusion detection methods, DL-based techniques typically exhibit superior accuracy and robustness [19]. Nevertheless, despite the variety of available DL models, the technology for intrusion detection remains predominantly mature in terrestrial implementations. Its precision and adaptability for maritime environments necessitate additional exploration [20, 21].

The direct application of IDS to the specialized context of intelligent vessels continues to encounter numerous gaps and challenges. While IDSs are increasingly recognized as essential components in the conceptual design and advanced applications of intelligent ships, their presence in large-scale practical deployments—especially those utilizing DL—remains relatively scarce and nascent. This rarity significantly contributes to the frequent occurrence of cybersecurity incidents. Furthermore, the data pertaining to attacks within intelligent ship networks are notably imbalanced: normal operational traffic predominates, whereas genuine attacks—particularly targeted assaults on vessel systems—are exceedingly rare yet profoundly destructive. Consequently, the evaluation metrics for many DL algorithms do not effectively translate to the maritime context, highlighting the critical and challenging task of developing an efficient, reliable, and maritime-specific intrusion detection solution.

In response to these challenges, this study introduces a novel fusion model—Transformer-KAN. This model combines the capability of Transformer encoders to capture long-range dependencies with the parameter efficiency and robust nonlinear fitting capacity of Kolmogorov-Arnold Networks (KAN), aiming to strike a balance between high accuracy and operational efficiency while enhancing robustness against imbalanced data. The primary contributions of this study are as follows:

1) The implementation of a DL-based IDS model specifically tailored for intelligent vessels, providing a novel approach to enhance maritime cybersecurity.

2) By replacing traditional MLP with KAN layers, an exceptional balance between performance and efficiency was achieved. The proposal of the Transformer-KAN fusion architecture as an innovative solution for shipboard intrusion detection, which advances beyond conventional multi-algorithm ensembles. By substituting traditional Multilayer Perceptrons (MLP) with KAN, the model not only maintains a strong nonlinear representational capability but also achieves significant improvements in parameter efficiency, rendering it more suitable for resource-limited settings. Through meticulous ablation studies, we validate the effectiveness and necessity of each core component of the model and demonstrate its capacity to detect rare categories of attacks. Comprehensive experiments confirm that the proposed model surpasses mainstream counterparts (e.g., CNN, GRU, DBN, SVM) across both binary and multi-class classification tasks on the NSL-KDD dataset.

3) The model is highly targeted and pragmatic, tailored to tackle specific industrial challenges. The innovation of this design addresses two fundamental challenges associated with deploying DL in real

industrial settings: the trade-off between computational complexity and model performance. The integration of KAN seeks to realize the dual benefit of maintaining or even enhancing detection accuracy while considerably reducing the parameter count and computational burden, thus transforming the model from “deployable in theory” to “deployable in practice”.

4) The model exhibits superior detection performance, with exceptional capability in recognizing rare attacks. The imperative of this study is underscored by its concentration on the distinct challenges posed by intelligent ships. Instead of merely adopting a generic model, this research customizes its approach to cater to the unique requirements of the maritime environment, which include resource constraints, specialized protocols, and a significant class imbalance. The model’s efficiency is attuned to the rigorous demands of onboard edge devices, and its advanced representational learning capability is designed to enhance the detection rate of rare yet critical attacks, thereby significantly mitigating the risk of false negatives—a crucial factor in ensuring maritime safety.

The remainder of this paper is organized as follows: Section 2 reviews related work on intrusion detection and intelligent vessels, and introduces the theoretical basis and overall architecture of the proposed Transformer-KAN model; Section 3 describes the experimental setup, results from hyperparameter tuning, ablation studies, and comparative evaluations, followed by an in-depth discussion of the model’s performance and efficiency; Section 4 summarizes the main conclusions, acknowledges limitations, and suggests potential directions for future work.

2. Methodology

2.1 Overview of IDSs and their application scenarios

An IDS constitutes a sophisticated network security technology aimed at protecting computer assets from security incidents such as unauthorized access, data breaches, and DoS attacks. This protection is facilitated through the real-time monitoring and analysis of network communication traffic, which enables the prompt detection and reporting of suspicious activities [22].

IDSs are typically deployed as a synergistic amalgamation of hardware and software components, serving as a logical enhancement to conventional firewalls. By doing so, they bolster network security by aiding computer systems in mitigating a range of external and a considerable number of internal threats. The primary functions of an IDS are categorized into four main aspects. First, it detects malicious intrusions and cyberattacks, thereby providing timely risk alerts that proactively safeguard the system. Second, it performs real-time monitoring of network communications, which assists in the immediate management of anomalies to preserve network integrity. Third, it possesses the capability to identify activities associated with system vulnerability scans, thus enabling prompt remedial actions and bolstering overall system security [23]. Fourth, an IDS enhances network security management by supplying network administrators with critical data and decision support, thereby raising the overall standard of network security [24].

Figure 1 depicts the network architecture of an intelligent research and training vessel. Initially, a Network-based Intrusion Detection System (NIDS) may be implemented at the juncture between the communication link layer and the internal ship network—that is, positioned subsequent to the router/firewall—to scrutinize all inbound and outbound data traffic in real-time. Such placement facilitates the identification of anomalous traffic patterns and attack strategies emanating from external sources such as satellite communications (VSAT), FBB, VDES, and microwave links, thereby precluding external threats from compromising the integrity of the ship’s internal network. At the border security gateway, the integration of NIDS enables the monitoring of data flows at essential security nodes, encompassing ship-to-shore communications and behavior management servers. This integration permits thorough inspections of critical operations, including ship-to-shore data transfers and remote management, thereby averting malicious commands, data breaches, and lateral movements within the network. Furthermore, NIDS can be positioned at the core switch and the aggregation switches of principal VLANs or subnets to oversee traffic across various business networks, for instance, the living network and the service function network. This deployment aids in the detection of abnormal communications, lateral movements, worm propagation, and other internal threats, thus facilitating the prompt recognition of irregular access across network partitions. On essential servers—such as intelligent navigation systems, ship management systems, data storage servers, and bridge workstations—Host-based Intrusion Detection Systems (HIDS) can be installed to monitor system logs, file modifications, and process activities, thereby identifying local attacks, malware, and privilege escalations that target critical business systems. Additionally, the deployment of NIDS/WIDS (Wireless Intrusion Detection Systems) at wireless access points and terminal access switches is critical for monitoring endpoint device access and ensuring wireless network security. This strategy prevents unauthorized device connections, inadvertent actions by internal personnel, or malicious assaults, thus ensuring the secure isolation of the living area network.

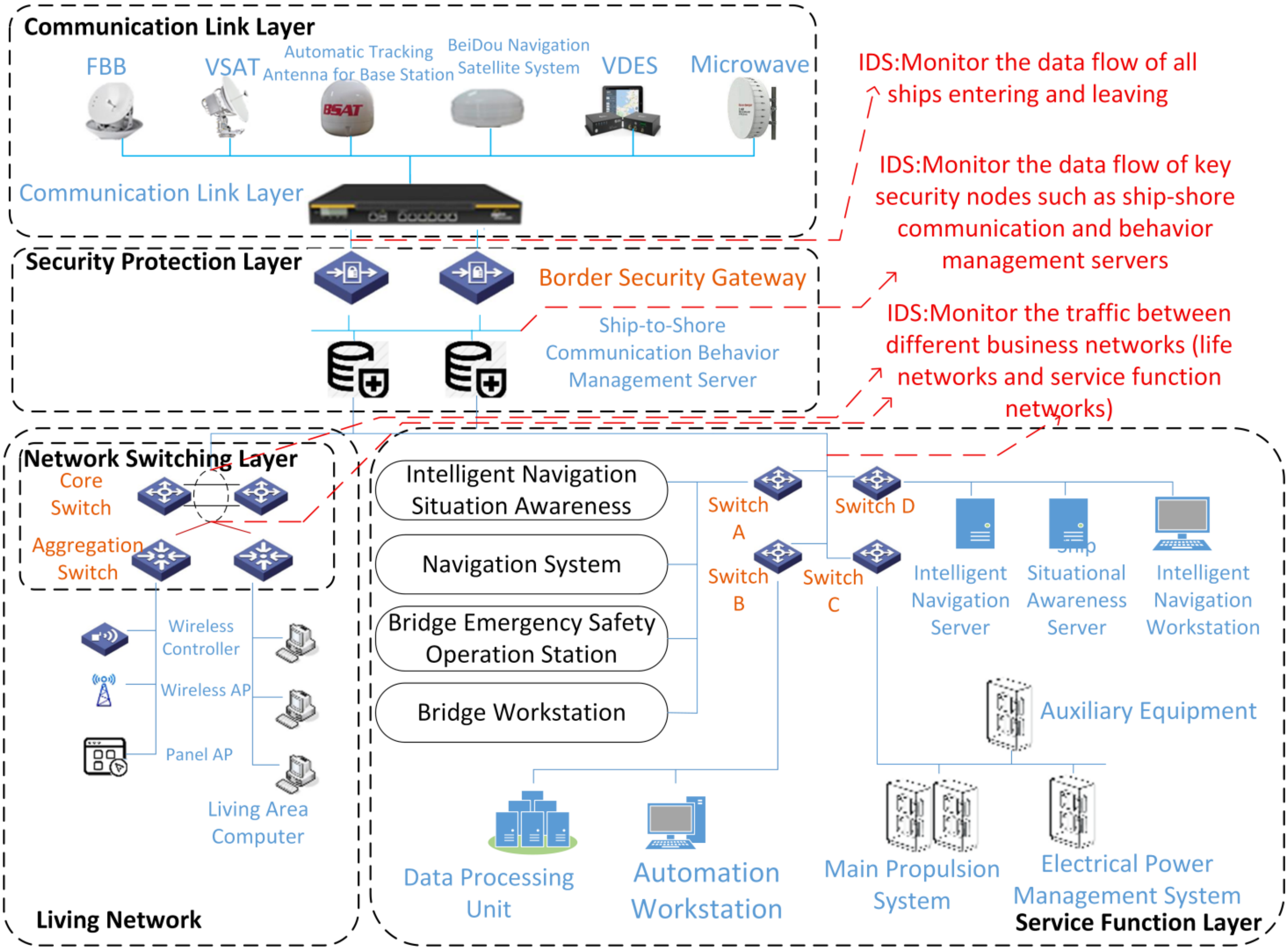


Fig. 1 Network structure of intelligent research and training dual purpose ship

2.2 Transformer

The Transformer, a DL model architecture pioneered by the research team at Google [25], was specifically developed for sequence-to-sequence (Seq2Seq) tasks. It has exhibited remarkable efficacy across a range of applications such as machine translation, text summarization, and dialogue generation.

The architecture of the Transformer model consists of two primary components: the Encoder and the Decoder. The Encoder processes the input sequence, while the Decoder is tasked with generating the output sequence. Central to the Transformer is the Self-Attention Mechanism, which allows the model to allocate varying attention weights to each position within the sequence during processing. This feature enables the model to discern and capture the dependencies among different positions within the sequence more effectively, thereby facilitating a more precise recognition of patterns, including those indicative of malicious network traffic. Additionally, the Transformer utilizes a Multi-Head Attention mechanism. By computing attention weights in parallel across multiple "heads," the model can capture semantic information at various hierarchical levels and from diverse contextual viewpoints. This capability significantly augments the model's representational power and its performance in generalization tasks [26].

Conventional sequence modeling approaches, such as Recurrent Neural Networks (RNNs), encounter difficulties with long sequences. In RNNs, each timestep's computation depends on the previous timestep's state, necessitating a sequential processing of data elements. This dependency hampers the ability to process inputs in parallel, thus leading to slower training speeds. Furthermore, RNNs are plagued by constraints related to their "short-term memory," which limits their capacity to process long sequences effectively and makes them prone to issues such as gradient vanishing or exploding. Although variants like LSTM and GRU networks address the problem of long-range dependencies to a certain degree, they do not completely eliminate it. Particularly for lengthy sequences, the problems associated with long-tail effects persist, and parallel computation is still impracticable [27].

In stark contrast, the Transformer architecture abandons the sequential, time-iterative structure characteristic of RNNs. It introduces the Multi-Head Self-Attention mechanism, which effectively overcomes the long-range dependency challenge. Importantly, this design enables each input element at any timestep to directly connect with elements at all other timesteps. As a result, the Transformer not only manages long sequences efficiently but also utilizes the Multi-Head Self-Attention mechanism to calculate attention values for the entire sequence in parallel. This parallel processing capability markedly improves training efficiency, providing the Transformer with enhanced speed and superior overall performance compared to RNNs [28].

The Transformer model functions solely through attention mechanisms, as depicted in Figure 2. It comprises an Encoder and a Decoder, both of which share a similar structure. The primary distinction between them lies in the implementation of a mask before the Multi-Head Self-Attention mechanism in the Decoder to prevent label leakage or the premature utilization of future information in the prediction process. Within the Transformer model, the Encoder consists of multiple identical layers stacked to progressively extract features from the input sequence. Each encoder layer primarily includes two sub-layers: a Multi-Head Self-Attention layer and a fully connected feed-forward network (FFN) layer. Similarly, the Decoder consists of multiple identical layers, each containing three main sub-layers: a masked Multi-Head Self-Attention layer, an Encoder-Decoder attention layer, and a fully connected FFN layer. The Decoder aims to generate the target sequence, achieving this by applying self-attention to both the output of the Encoder and the previously generated partial sequence [29].

Architectures such as CNN, RNN, and the Transformer build upon the MLP, which acts as a foundational element for most DL models. However, the intrinsic architecture of MLPs—characterized by linear transformations followed by fixed nonlinear activation functions—exhibits significant limitations. A critical issue is their parameter inefficiency, as MLPs require extensive breadth to attain sufficient expressive capacity, necessitating vast arrays of neurons and, consequently, large weight matrices. This requirement leads to a surge in the number of parameters, causing inflated model sizes that are computationally expensive and have substantial memory demands.

In contrast, the KAN redefines the design principles of MLPs by substituting linear weight matrices with learnable 1D spline functions. This innovative modification provides numerous benefits, including markedly enhanced parameter efficiency. Unlike traditional MLPs, KAN does not depend on a wide network structure; it achieves comparable or superior performance with significantly fewer parameters. Through the utilization of adaptable spline functions capable of executing complex transformations, KAN obviates the necessity for numerous neurons, thereby reducing the need to augment representational capacity through extensive neural layering. Consequently, integrating the Transformer with KAN in this study presents a more compact architecture, enhancing deployability on computationally limited edge devices, such as those employed in intelligent maritime systems. This integration addresses a fundamental requirement of contemporary maritime applications.

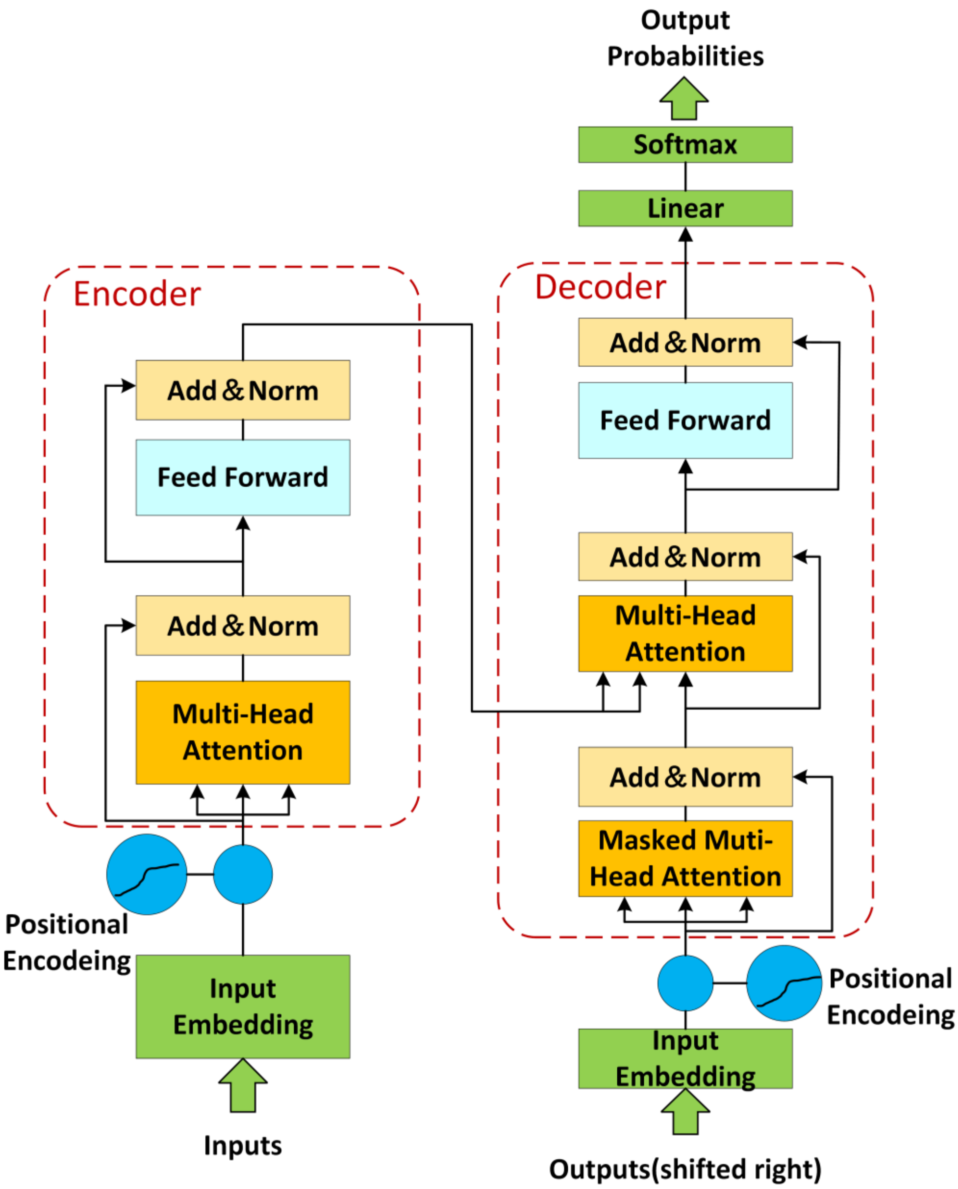


Fig. 2 Transformer network architecture

2.3 KAN

The KAN is a DL network model inspired by the Kolmogorov-Arnold representation theorem. This model is devised to approximate and represent complex multivariate continuous functions using suitable neural network architectures [30]. KAN provides a robust theoretical framework and an innovative network structure for tackling the challenge of function approximation, marking a significant theoretical advancement in the field of DL.

The foundational research on KAN establishes that any multivariate continuous function defined within a bounded domain can be articulated as a finite composition of continuous univariate functions and summation operations. This relationship is mathematically expressed as follows:

$$f(X)=f(x_1,x_2,\ldots,x_n)=\sum_{q=1}^{2n+1}\varphi_q\left(\sum_{p=1}^{n}\varphi_{q,p}(x_p)\right) \tag{1}$$

where $\varphi_{q,p}$ denotes univariate functions that map each input variable (x_p), such that $\varphi_{q,p}$:[0,1]→R and φ_q : R → R. These 1D functions are parameterized using B-spline curves, incorporating learnable coefficients corresponding to the local B-spline basis functions. Each layer of KAN comprises these parametrized one-dimensional functions:

$$\varphi=\{\varphi_{q,p}\}, \quad p=1,2,\ldots,n_{\text{in}}, \quad q=1,2,\ldots,n_{\text{out}} \tag{2}$$

where $\varphi_{q,p}$ are the parametrized functions with learnable parameters, n_{in} represents the number of inputs, and n_{out} signifies the number of outputs. Each function, previously denoted as $\varphi_{q,p}$, is parameterized as a B-spline, which are piecewise polynomial functions known for their defined degree of continuity. By utilizing knot vectors and basis functions, B-splines extend Bézier curves and afford superior local control over the shape of the function. Each B-spline is composed as a linear combination of basis spline functions, thereby enhancing the network's ability to model complex data patterns [31].

The overall architecture or shape of the KAN is specified using an array of integers, $[n_0,n_1,...,n_L]$. For a visual representation and further intuition on this structure, refer to Figure 3.

Within this architecture, the quantity of neurons (nodes) in the *i*-th layer is denoted by n_{in}. The *i*-th neuron in layer l is indexed as (l,i), and its corresponding activation value is denoted by $x_{l,i}$. Between the l-th layer and the $(l+1)$-th layer, there are $n_l n_{l+1}$ learnable spline activation functions:

$$\phi_{l,j,i}, \quad l=0,\ldots,L-1, \quad i=1,\ldots,n_l, \quad j=1,\ldots,n_{l+1} \tag{3}$$

where $\phi_{l,j,i}$ represents the activation function that links neuron (l,i) to neuron $(l+1,j)$.

The KAN is structurally analogous to an MLP, comprising stacked layers. Unlike MLPs, which employ simple linear transformations followed by non-linear activation functions, KAN integrates complex, learnable function mappings [32].

$$\text{KAN}(x)=(\Phi_{L-1}\circ\Phi_{L-2}\circ\cdots\circ\Phi_1\circ\Phi_0)x \tag{4}$$

where $\Phi_1\circ\Phi_0 = \Phi_1(\Phi_0(x))$ represents the composition of functions.

The computational procedure within a KAN network consisting of L layers, which processes the input x_l from layer l to produce the input x_{l+1} for the next layer $l+1$, is described by the following equation:

$$\mathbf{x}_{l+1}=\begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_l}(\cdot) \end{pmatrix}\mathbf{x}_l \tag{5}$$

where ϕ_l signifies the functional matrix corresponding to the l-th layer of the KAN. The symbol $(\cdot)$ serves as a placeholder for the input variable of the function, signifying that this position is designed to receive a variable as input.

To enhance the optimization of KAN, the implementation details of a residual activation function are delineated. The activation function $\phi(x)$ is conceptualized as the sum of a basis function $b(x)$ and a spline function:

$$\phi(x)=\omega_1 b(x)+\omega_2\text{spline}(x) \tag{6}$$

where $b(x)$ is identified as the fundamental function, typically implemented as SiLU. The weights ω_1 and ω_2 are parameters for the respective components:

$$b(x) = \text{silu}(x) = \frac{x}{1+e^{-x}} \tag{7}$$

The linear combination of B-splines, represented as $\text{spline}(x)$, constitutes the learnable spline function:

$$\text{spline}(x) = \sum_i c_i B_i(x) \tag{8}$$

The primary advantages of the KAN model include its robust theoretical foundation rooted in established mathematical theorems, its efficacy in approximating multidimensional functions, its potent expressive capabilities, and its potential to reduce reliance on extensive datasets compared to conventional architectures. Consequently, KAN is particularly apt for applications that demand accurate representation of complex, high-dimensional relationships. It has proven effective in managing high-dimensional data and addressing complex function approximation challenges [33].

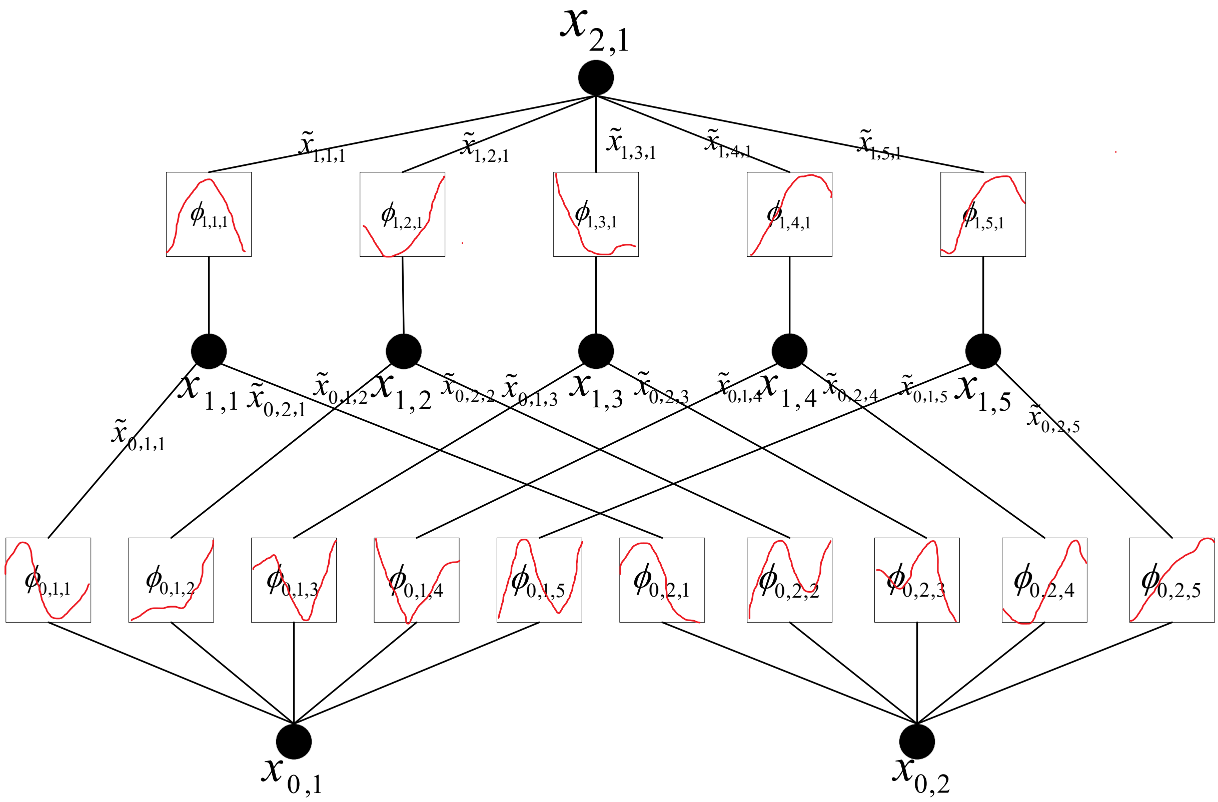


Fig. 3 Notations of activations that flow through the network

2.4 Algorithmic process

The increasing diversity and frequency of network attacks have rendered traditional firewall systems increasingly inadequate for addressing the complexities of modern network security. In response to the enhanced intricacy of network environments, researchers have developed NIDS. Although contemporary mainstream intrusion detection algorithms often employ machine learning (ML) techniques, the inherent constraints of conventional ML approaches often result in various challenges. These challenges include, but are not limited to, suboptimal detection accuracy and efficiency, as well as the labor-intensive task of manual feature selection and engineering.

To address these issues, this study introduces a novel intrusion detection model that utilizes both temporal and spatial information, designated as the Transformer-KAN model. The architecture of this proposed model is depicted in Figure 4. The model fundamentally comprises three key phases: a data preprocessing module, a training phase, and a testing phase. The algorithm's flowchart is presented in Figure 5.

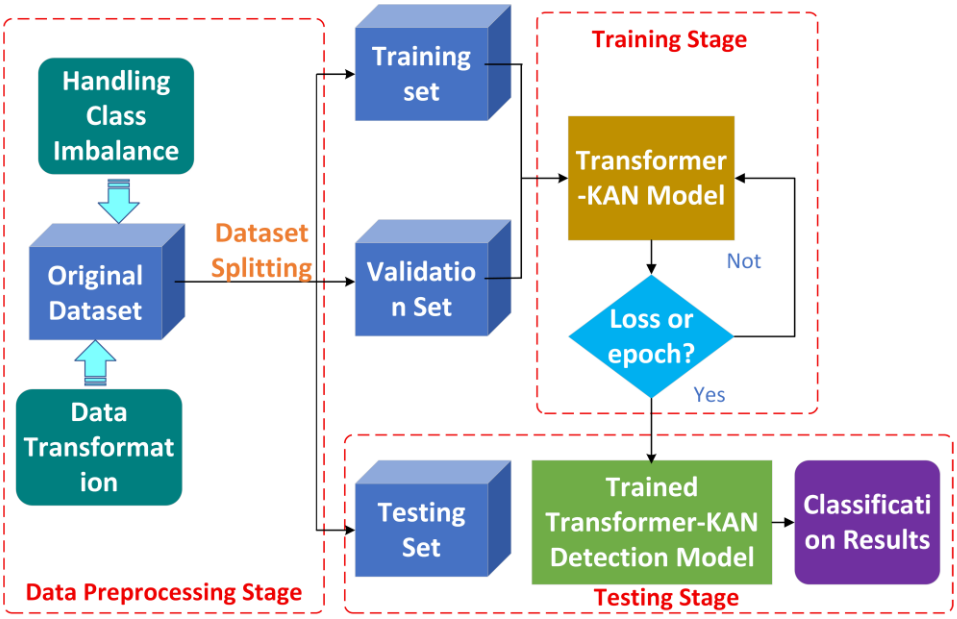


Fig. 4 Transformer-KAN architecture diagram

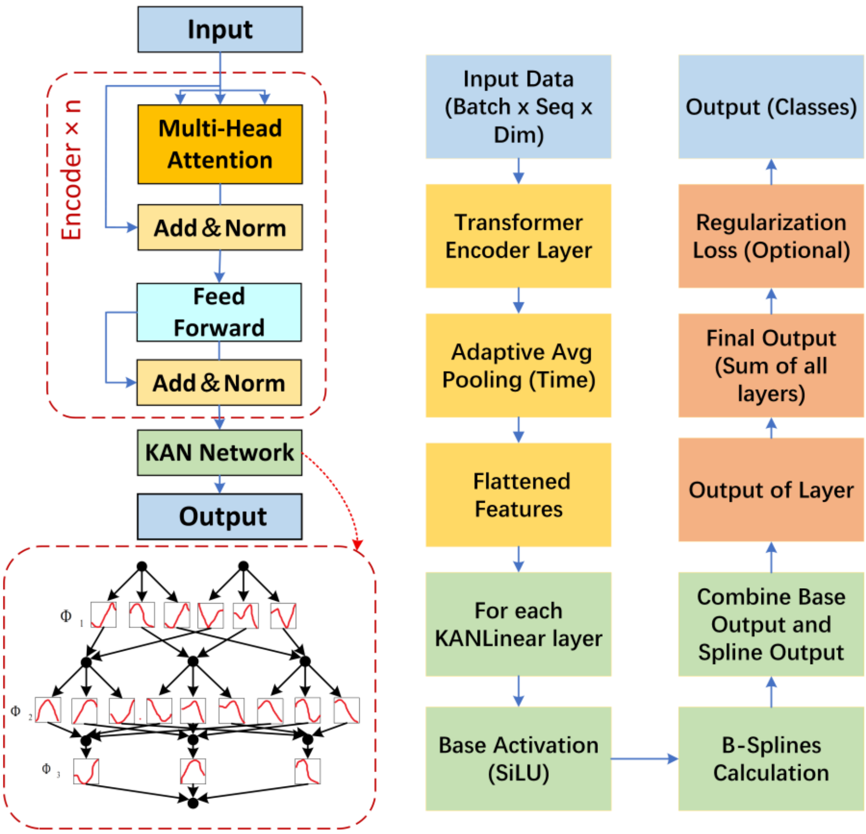


Fig. 5 Algorithm workflow diagram

During the forward pass of the model, the input data initially undergoes appropriate dimensionality transformation and standardization to align with the input specifications of the Transformer encoder. Specifically, the input sequences are reformatted to be compatible with the Transformer encoder. In this study, each input sequence contains multiple features. The model initially adjusts these sequences to the configuration (batch_size,seq_len,input_dim) before introducing them into the Transformer encoder for further processing.

Following the preprocessing of the input data, the data advances to the Transformer encoder layers. The primary function of the Transformer encoder is to discern long-range dependencies within the input sequence through the self-attention mechanism. An encoder layer typically consists of the following essential components: a self-attention mechanism and a position-wise feed-forward neural network, usually organized in a multi-layer stack. The self-attention mechanism assesses the relationships among various positions in the input sequence, creating a representation for each position via a weighted summation. Input vectors are projected into Q (Query), K (Key), and V (Value) representations, and for each attention head, the ensuing computation is performed:

$$\mathrm{Attention}(Q,K,V)=\mathrm{Softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{9}$$

where the matrix Q represents the Query matrix, encapsulating a set of queries that delineate the current focus of each output element. The matrix K is referred to as the Key matrix, containing keys that characterize the identity and content of each input element for retrieval purposes. The matrix V is the Value matrix, which holds the actual content or information from each input element that is to be aggregated. The term d_k signifies the dimensionality of the queries and keys. The scaling factor $\frac{1}{\sqrt{d_k}}$ is employed to mitigate the excessive growth in magnitude of the inner products $QK^{\top}$, which could otherwise drive the softmax function into regions characterized by exceedingly small gradients, thus addressing the issue of vanishing gradients.

Subsequently, each position independently undergoes processing through a two-layer FFN:

$$\mathrm{FFN}(x)=\mathrm{ReLU}(xW_1+b_1)W_2+b_2 \tag{10}$$

where x represents the input vector to the layer, emanating from the output of the self-attention mechanism at a specific sequence position. The matrix W_1 serves as the weight matrix for the first linear transformation, projecting the input into a higher-dimensional space. The vector b_1 is the bias vector incorporated following the first linear transformation. The Rectified Linear Unit activation function, denoted as ReLU(z)=max(0,z), introduces non-linearity by setting all negative values within the intermediate vector to zero. The matrix W_2 is the weight matrix for the second linear transformation, which reprojects the activated intermediate features back to the original model dimensionality. The vector b_2 is the bias vector introduced after the second linear transformation, yielding the final output for this position.

To further augment the representational capacity, both the Self-Attention and FFN modules include:

$$\mathrm{LayerNorm}(x+\mathrm{SubLayer}(x)) \tag{11}$$

where SubLayer(x) is executed by the current sub-layer, typically either the multi-head self-attention mechanism or the position-wise FFN. The Layer Normalization function, LayerNorm(·), is applied to the outcome of the residual sum. This normalization technique stabilizes the training process by independently normalizing the values across the feature dimension for each sample to achieve zero mean and unit variance, subsequently incorporating learnable scaling and shifting parameters. This approach effectively mitigates the "covariate shift" within the network, often resulting in accelerated and more stable training regimes.

Residual connections are instrumental in circumventing the vanishing gradient problem, while Layer Normalization (LayerNorm) facilitates expedited convergence.

By employing multiple layers of Transformer encoders, the model incrementally extracts higher-level abstract representations from rudimentary features. The three-dimensional tensor output from the encoder is subjected to dimensionality reduction to create a unified feature representation, which is subsequently input into the KAN layer. This output encapsulates the contextual information of the input data, thereby offering rich feature representations for downstream processing.

The Transformer encoder, in conjunction with its integrated FFNs, forms a deep nonlinear transformation architecture. This architecture is adept at abstracting input features sequentially across layers and deriving more discriminative high-dimensional representations. These representations not only preserve the original information but also assimilate a comprehensive contextual understanding. This integration facilitates more precise nonlinear transformations and predictions in subsequent modules, such as the KAN layer. Owing to the self-attention mechanism, the Transformer exhibits substantial parallelism during both training and inference phases, allowing for the concurrent processing of all temporal steps. This characteristic markedly enhances training efficiency and scalability compared to sequential models like RNNs, rendering it particularly advantageous for processing large-scale data.

Furthermore, the Transformer encoder is distinguished by its high modularity, which facilitates seamless integration with other components, such as the KAN layer in this study. Its hyperparameters-including the number of layers, number of attention heads, and dimensions of hidden layers-can be flexibly adjusted to accommodate diverse task requirements, thus demonstrating its strong practical utility. Through the integration of residual connections and layer normalization, the Transformer maintains gradient stability within deep architectures, effectively mitigating issues related to gradient explosion or vanishing and thereby enhancing training robustness and expediting convergence.

Upon receiving the dimensionally fixed feature vectors from the Transformer encoder, the KAN layer initially performs preprocessing to confine the data within a specified grid range. This preprocessing ensures that the input adheres to the operational domain of the B-spline basis functions, thereby guaranteeing numerical stability during computations. At the initialization stage, the KAN layer constructs a grid based on parameters such as grid size, spline order, and range, which determine the resolution and complexity of the piecewise polynomials.

The KAN layer initiates its process with a basic linear transformation, which serves a function analogous to that of a traditional fully connected layer, albeit with a distinct approach to the integration of activation functions. This initial transformation is instrumental in capturing linear features and elementary patterns within the input data. Subsequently, the KAN layer models nonlinear relationships among input features utilizing B-spline basis functions. As piecewise polynomials, B-splines facilitate smooth interpolations within intervals, thereby aiding in the detection of more intricate patterns. Utilizing these basis functions, the KAN constructs high-dimensional nonlinear feature representations, which subsequently undergo another linear transformation through a spline weight matrix. The final output of the KAN is derived by aggregating the outcomes of the initial linear transformation and the B-spline transformation, thereby encapsulating both linear and nonlinear attributes.

To mitigate overfitting, the KAN layer incorporates L1 regularization on the spline weights, effectively constraining the complexity of the model. This array of mechanisms equips the KAN with enhanced nonlinear modeling capabilities relative to traditional linear layers, enabling it to capture global dependencies and fit local patterns with high precision. This enhancement not only augments the model's expressive capacity but also its generalization performance.

In the broader context of the Transformer-KAN model, the KAN layer processes the output from the encoder as previously detailed, culminating in the generation of results for classification or regression tasks. This model synergistically combines the Transformer's proficiency in capturing long-range dependencies with the KAN's robust nonlinear fitting capabilities, rendering it particularly effective for tasks that involve multi-scale features. By leveraging learnable B-spline-based activation functions, the KAN frequently attains superior accuracy with fewer parameters compared to traditional MLPs, while also improving interpretability and visualization. Additionally, its regularization strategies enhance generalization, rendering the model suitable for deployment in resource-limited scenarios and areas necessitating explainable decisions, such as in scientific research.

3. Experimental results and analysis

3.1 Experimental data and preprocessing

The KDD Cup 1999 (KDDCUP99) dataset is one of the most extensively utilized datasets in intrusion detection research. Originally released through a collaboration between the MIT Lincoln Laboratory and the Defense Advanced Research Projects Agency (DARPA), this dataset comprises detailed network traffic records. These were meticulously gathered from a simulated real-world network environment, providing a substantial corpus with approximately 41,000 training instances and 11,000 testing instances.

Addressing certain limitations inherent in the original KDDCUP99 dataset, Tavallaee et al. [34] introduced an enhanced version known as the NSL-KDD dataset. The primary enhancement involved the removal of redundant records from the KDDCUP99 data. Furthermore, the NSL-KDD provides standardized training and testing sets, thereby establishing a benchmark to facilitate comparative analyses across different research endeavors. The dataset categorizes records into five main classes: one Normal class and four distinct attack types: DoS, Probe, Remote-to-Local (R2L), and User-to-Root (U2R). Specifically:

Probe: Activities such as port scanning or network surveillance.

DoS (Denial-of-Service): Attacks designed to exhaust network resources and prevent legitimate users from accessing services.

R2L (Remote-to-Local): Unauthorized access attempts originating from a remote machine aimed at gaining local user privileges.

U2R (User-to-Root): Attempts by an unauthorized local user to obtain superuser (root) privileges, typically through privilege escalation exploits.

The present study employs the NSL-KDD dataset for model evaluation. Each record within this dataset consists of 41 features (attributes) and one class label. The dataset identifies a total of 38 specific attack subtypes. The standard training set contains instances belonging to 22 of these attack subtypes, while the corresponding standard test set includes instances from 20 subtypes present in the training set, along with instances from an additional 17 novel attack subtypes not represented in the training data. These 38 distinct attack subtypes are categorized under the four primary attack categories previously mentioned (DoS, Probe, R2L, U2R). This experiment utilizes a processed version of the NSL-KDD dataset to assess the proposed model’s performance. Detailed descriptions of the dataset characteristics are provided in Tables 1 and 2.

Table 1 Dataset sample distribution

Category	Normal	DOS	U2R	R2L	Probe
Training set	67334	45927	67	995	11656
Testing set	9711	7638	200	2754	2421

Table 2 Attack type

Attack type	Description	Twenty-two types of attacks
DOS	Distributed Denial-of-Service attack	ping-of-death, syn flood, smurf,etc.
R2L	Unauthorized access from a remote host	guessing password
U2R	Unauthorized local root access	buffer overflow attacks,etc.
Probe	Port monitoring or scanning	port-scan, ping-sweep,etc.

To enhance the efficacy of iterative model training and parameter updates, several data preprocessing steps are customarily executed prior to the initiation of the training phase. These operations commonly encompass standardizing data formats, imputing missing values, outlier removal, numericalization, and normalization. Due to the necessity of addressing class imbalance within the dataset, this study incorporated an additional step of data augmentation. Therefore, the preprocessing procedure employed in this research comprises three primary stages: numericalization, addressing class imbalance, and standardization. A detailed description of each stage is provided below:

(1) Numericalization: The NSL-KDD dataset includes three categorical features-protocol_type, service, and flag-which are non-numeric. As the model architecture mandates numerical inputs, these categorical features were transformed into numerical representations utilizing the LabelEncoder tool available in the scikit-learn library. Similarly, the label column, which comprises categorical string data, was converted into a numerical format. The results of these conversions are elaborated in Table 3.

Table 3 Numericalization

Attack type	Meaning	Conversion Result	
		Binary Classification	Multi-class Classification
Normal	Normal Record(s)	0	0
Dos	Distributed Denial-of-Service attack	1	1
U2R	Unauthorized local root access		2
R2L	Unauthorized access from a remote host		3
Probe	Port monitoring or scanning		4

(2) Addressing Class Imbalance: The NSL-KDD dataset presents a significant challenge due to class imbalance, particularly concerning the U2R and R2L attack categories. These classes are disproportionately represented, with a higher occurrence in the test set than in the training set, where they represent a minimal fraction. This imbalance may cause models trained on this dataset to perform inadequately in distinguishing these minority attack types. To counter this issue, both feature data and corresponding labels were converted into the DMatrix format, which is specifically designed for use with the XGBoost algorithm. DMatrix is an internal data structure that optimizes memory efficiency and computational speed within XGBoost and supports sparse matrices. Notably, XGBoost includes hyperparameters, such as scale_pos_weight, that are specifically tailored to manage class imbalance. Adjusting this parameter can significantly enhance the model's proficiency in recognizing and accurately classifying instances of minority classes.

(3) Standardization: Disparities in numerical ranges across different feature dimensions within the dataset can hinder model training, potentially resulting in delayed convergence and suboptimal accuracy. To mitigate this issue, feature standardization was conducted utilizing the MinMaxScaler technique. This method scales and translates each feature individually to ensure it lies within the range [0, 1]. Such normalization promotes more stable and efficient dynamic parameter updates during the iterative training process. The formula for MinMaxScaler is as follows:

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{12}$$

where x represents the original feature value, $x_{\min}$ is the minimum value for that feature across the dataset, and $x_{\max}$ is the maximum value for that feature.

3.2 Intrusion detection evaluation metrics

To effectively evaluate intrusion detection algorithms, it is essential to employ both valid and practical experimental assessment methods, as well as to utilize established criteria for measuring the model's generalization capabilities, commonly referred to as evaluation metrics.

These metrics are pivotal in assessing the performance of algorithms, providing a comprehensive and accurate reflection of the model's behavior. The performance evaluation of IDS incorporates multiple indicators, which are instrumental in determining the system's effectiveness, accuracy, and applicability.

Table 4 Evaluation metrics

Confusion Matrix		Predicted Value(s)	
		Positive	Negative
Actual Label(s)	Positive	TP	FP
	Negative	FN	TN

The confusion matrix, detailed in Table 4, employs specific terms to evaluate the classification performance for each type of network traffic: True Positive (TP) denotes the number of samples correctly predicted as belonging to the particular traffic type under consideration. False Positive (FP) indicates the number of samples that, while belonging to other traffic types, were incorrectly classified as belonging to the type under review. Conversely, True Negative (TN) represents the number of samples from other traffic types that were correctly identified as not belonging to the type under consideration. False Negative (FN) refers to the number of samples that, although actually belonging to the specific traffic type, were erroneously classified as belonging to other types.

The diagonal elements of the confusion matrix reflect the degree of concordance between the model's predictions and the actual, or ground truth, labels of the data. Higher values along the diagonal indicate a greater number of samples that the model has correctly classified for each respective category.

The metrics utilized for evaluating the results of this research are detailed subsequently. Accuracy is defined as the ratio of the total number of correctly predicted samples to the total number of samples, as presented in Equation (13):

$$Accuracy = \frac{TN + TP}{TP + FP + TN + FN} \tag{13}$$

Precision, also known as the positive predictive value, serves as an evaluation metric that is derived from prediction outcomes. It quantifies the proportion of instances that are correctly identified as positive among all instances predicted to be positive. Specifically, precision is calculated as the ratio of True Positives (TP) to the total number of instances predicted as positive, which includes both True Positives (TP) and False Positives (FP), as depicted in Equation (14):

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

Recall, alternatively referred to as Sensitivity or the True Positive Rate (TPR), assesses the model's performance by focusing on the actual positive samples. It is defined as the proportion of all instances that truly belong to the positive class and are correctly identified as such. Recall is determined by the ratio of True Positives (TP) to the total number of actual positive instances, encompassing both True Positives (TP) and False Negatives (FN), as shown in Equation (15):

$$Recall = \frac{TP}{FN + TP} \tag{15}$$

The F1 Score, a composite evaluation metric, considers both Precision and Recall and is commonly used to gauge the efficacy of binary classification models. The F1 Score is the harmonic mean of Precision and Recall, crafted to provide a balanced view of these metrics, as delineated in Equation (16):

$$F1\,score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{16}$$

By integrating these evaluation metrics, ideally structured within an assessment framework [or an evaluation matrix, if deemed appropriate], a comprehensive understanding of the performance of intrusion detection algorithms across various dimensions is facilitated. This integrative approach aids in the selection of suitable algorithms and the adjustment of parameters to enhance overall effectiveness.

3.3 Hyperparameter tuning experiment

The experimental simulation environment was configured as outlined in Table 5.

Table 5 System configuration

Configuration Name	Environment Setup
Operating System	windows10
Processor (CPU)	Intel(R) Core(TM) i9-13900H CPU
Memory (RAM)	16G
Software Framework	Pytorch2.3.0
Programming Language	pyhton3.9
IDE	pycharm
Key Libraries	Scikit-learn, NumPy, Pandas. etc

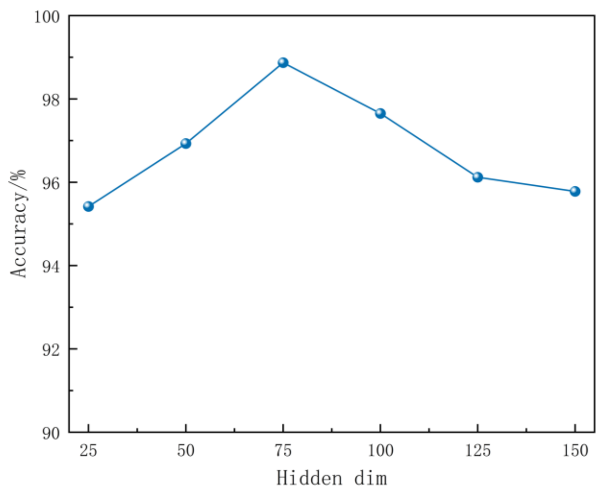
This section is dedicated to validating the performance of the proposed Transformer-KAN model. Given that the model comprises multiple parameters, it is imperative to assess their impact on the model's performance to ascertain optimal settings for future experiments. The parameters evaluated include grid size (grid_size), spline polynomial order (spline_order), spline scaling factor (scale_spline), noise scaling factor (scale_noise), base scaling factor (scale_base), grid range (grid_range), grid epsilon (grid_eps), input dimension (input_dim), attention dimension (hidden_dim), number of Transformer encoder layers (num_layers), number of multi-head attention heads (num_heads), dropout rate (dropout_rate), and the number of output classes (num_classes).

Based on the foundational concepts of the model, the parameters predicted to exert the most substantial influence on performance—grid size (grid_size), number of Transformer encoder layers (num_layers), and attention dimension (hidden_dim)—were selected for in-depth comparative analysis. Initial investigations focused on understanding the correlation between alterations in these parameters and the accuracy trends. This was followed by a systematic grid search aimed at identifying the optimal configuration. The experiments utilized a combined dataset, which included both the training set (KDDTrain+) and the test set (KDDTest+) from the NSL-KDD dataset. By amalgamating these datasets, the total sample size available for the model's internal partitioning into training, validation, and testing segments was increased, thereby enhancing the robustness of the training process. The experiments were structured around a five-class classification task, facilitating a detailed comparative analysis.

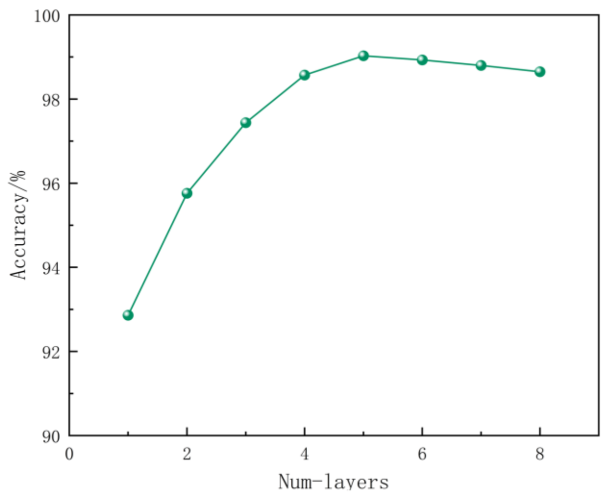
Throughout the tuning process, all parameters except the one under investigation were maintained constant. The fixed parameters were established as follows: spline_order = 3, scale_noise = 0.1, scale_base = 1.0, scale_spline = 1.0, grid_range = [-1, 1], num_heads = 1, and batch_size = 128. The model underwent training for 50 epochs with a learning rate of 0.0004, utilizing the Adam optimizer.

1. Attention Dimension (hidden_dim)

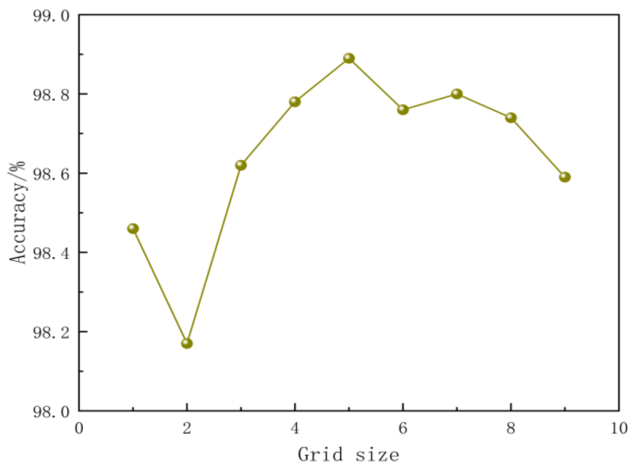
The selection of the attention dimension critically influences model performance, with potential impacts whether it is excessively large or small. Consequently, a parameter search was conducted within the range of 0 to 200. For this experiment, we commenced with an attention dimension of 25, subsequently increasing it incrementally to 50, 75, 100, 125, and 150. The number of encoder layers was consistently held at 3, and the grid size was established at 3. The experimental outcomes are depicted in Figure 6 (a).



(a) Attention Dimension Optimization Plot



(b) Encoder Layer Number Optimization Plot



(c) Grid Size Optimization Plot

Fig. 6 Hyperparameter optimization plot

As indicated by the plot, the accuracy of the model demonstrates an upward trajectory as the hidden_dim increases from 25 to 75, attaining a peak performance level of 98.87% at hidden_dim = 75. However, further elevation of hidden_dim to 100, 125, and 150 results in a subsequent decline in accuracy.

When the hidden_dim is relatively modest, the model lacks sufficient capacity to effectively encapsulate the complex patterns inherent in the data, thereby leading to underfitting and a consequent reduction in accuracy. As hidden_dim escalates, the augmented capacity of the model facilitates improved adaptation to data features, thereby enhancing accuracy. Conversely, an excessively large hidden_dim may confer an unwarranted expansion of model capacity, precipitating overfitting by incorporating noise present in the training data. This phenomenon results in diminished test accuracy, thereby impairing generalization.

2. Number of Encoder Layers (num_layers)

With the attention dimension established at 75 and the grid size set at 5, we systematically varied the number of encoder layers from 1 to 8. The results of these variations are depicted in Figure 6 (b).

The data suggest that the quantity of encoder layers in the Transformer-KAN model exerts a considerable influence on classification accuracy. Optimal performance, achieving a peak accuracy of 99.03%, was observed with five layers. While an initial increase in the number of layers enhances performance, a subsequent decline is noted when more than five layers are used, potentially due to overfitting or elevated computational complexity.

Mechanistically, augmenting the number of layers initially enhances the model's capacity, facilitating the learning of more intricate patterns, which, in turn, enhances performance. Within the Transformer-KAN architecture, each additional encoder layer exploits the self-attention mechanism inherent in the Transformer component and the non-linear transformations facilitated by the KAN layers to distill and represent progressively abstract features. However, the observed decrement in accuracy beyond five layers indicates a point of diminishing returns. Potential contributing factors to this decline include an increased risk of overfitting, the challenges of vanishing or exploding gradients, and greater computational demands.

3. Grid Size (grid_size)

With the attention dimension maintained at 75 and the number of encoder layers fixed at 7, we sequentially altered the grid size from 1 to 9. These experimental outcomes are illustrated in Figure 6 (c).

Incremental increases in the grid size from 1 to 5 resulted in a rise in classification accuracy from 98.46% to 98.89%, indicating that enlarging the grid size bolsters the model's expressive capability and facilitates a better fit to the data. Within the KAN layers, the grid_size parameter regulates the density and interpolation accuracy of the B-spline basis functions. Smaller grid sizes, such as 1 or 2, are associated with lower resolution and consequently, a reduced number of basis functions. This limitation hampers the model's ability to adequately capture subtle non-linear patterns in the data, leading to diminished accuracy. As the grid_size is increased to 3 and 4, the density of basis functions and interpolation accuracy improve, enabling the model to more accurately approximate complex relationships, with accuracies rising to 98.62% and 98.78% respectively.

The peak accuracy, reaching 98.89%, was achieved when the grid size was set at 5. This configuration appears to provide the KAN layers with adequate resolution to effectively interact with the Transformer's capacity for modeling long-range dependencies. As a result, the system can accurately represent the non-linear characteristics of the data. However, increasing the grid size beyond this point resulted in a marked increase in the number of B-spline basis functions, which in turn elevated the model's complexity. Such complexity heightened the risk of overfitting, as the model began to interpret noise in the training data as significant features, rather than identifying traits that are generalizable.

Table 6 Model parameters

Name	Setting
Grid_size	5
Spline_order	3
Scale_noise	0.1
Scale_base	1.0
Grid_range	[-1, 1]
Scale_spline	1.0
Grid_eps	0.02
Input_dim	41
Hidden_dim	75
Transformer num_layers	5
Num_heads	1
Dropout_rate	0.5
Num_classes	Category-Specific Value
Epoch	50
Batch_size	32（Binary Classification）/64（Multi-class Classification）
Learn_rate	0.0003

Based on preliminary experiments, we assessed the impact of various settings of three parameters on accuracy. Subsequently, we undertook a comprehensive grid search across the entire range of these parameters. The search space encompassed six candidate values for the attention dimension (25, 50, 75, 100, 125, 150), six candidate values for the number of encoder layers (2, 3, 4, 5, 6, 7), and five candidate values for the grid size (1, 3, 5, 7, 9). After evaluating all 180 combinations, the optimal configuration was identified as an attention dimension of 75, five encoder layers, and a grid size of 5, which achieved the highest accuracy of 99.03%. Consequently, we established these model parameters, as detailed in Table 6, and employed this configuration for all subsequent experiments involving the Transformer-KAN model in this study.

3.4 Experiments on binary and five-class classification of the main model

Through parameter tuning experiments, we determined the impact of the three most influential parameters on model performance. These experiments utilized the optimal values identified in the previous section to maximize accuracy. Employing these parameters, we conducted both binary and multi-class (five-class) classification experiments to evaluate the corresponding performance metrics.

1. Binary classification experiment

The accuracy and loss curves of the Transformer-KAN model during its training phase are illustrated in Figure 7. It is evident that after approximately 30 epochs, both the training and testing accuracies stabilized, exceeding 0.98. Furthermore, the loss curve demonstrates a smooth and steady decline in loss values. By around the 10th epoch, both accuracy and loss had reached a state of stability, indicating high learning efficiency and rapid convergence. The consistency in high performance across training and validation phases suggests an excellent classification capability without significant indications of overfitting, thus confirming that the model generalizes effectively to unseen data.

The confusion matrix, depicted in Figure 9 (c), reveals the model achieved the following performance metrics:

Accuracy: 98.96%,

Precision: 98.96%,

Recall: 98.96%,

F1-score: 98.96%.

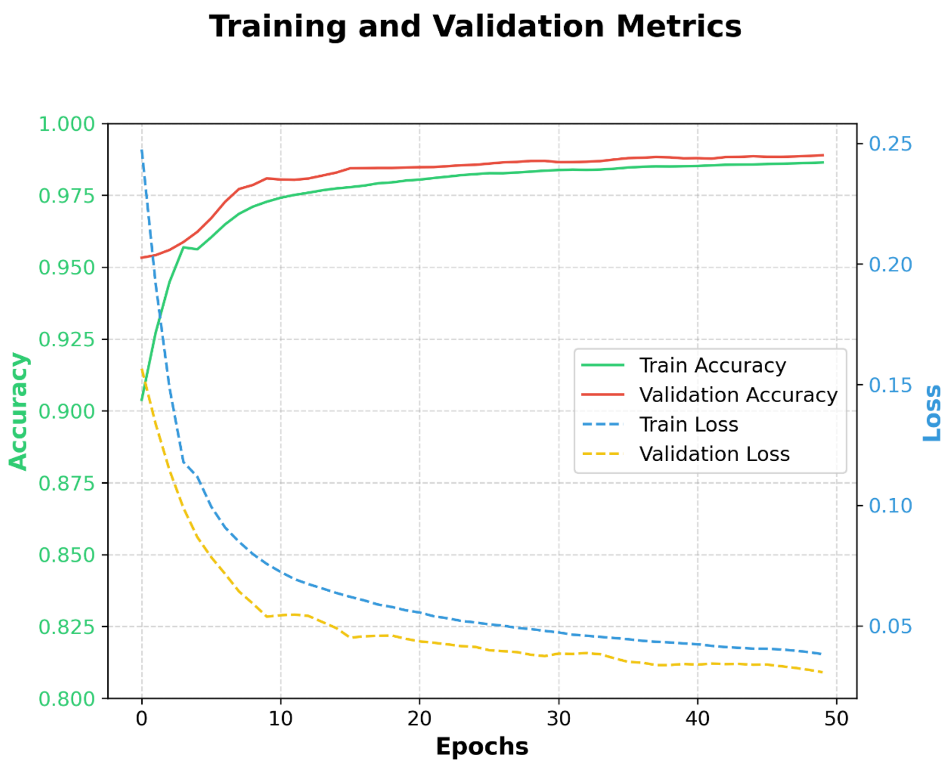


Fig. 7 Transformer-KAN binary classification accuracy and loss curves

2. Five-class classification experiment

The accuracy and loss trajectories of the model during its training are displayed in Figure 8. This training process showcases excellent stability with minimal fluctuations, as evidenced by both training and validation losses decreasing smoothly. The confusion matrix, shown in Figure 10 (c), indicates the model achieves the following performance metrics:

Accuracy: 99.03%,

Precision: 99.02%,

Recall: 99.03%,

F1-score: 99.01%.

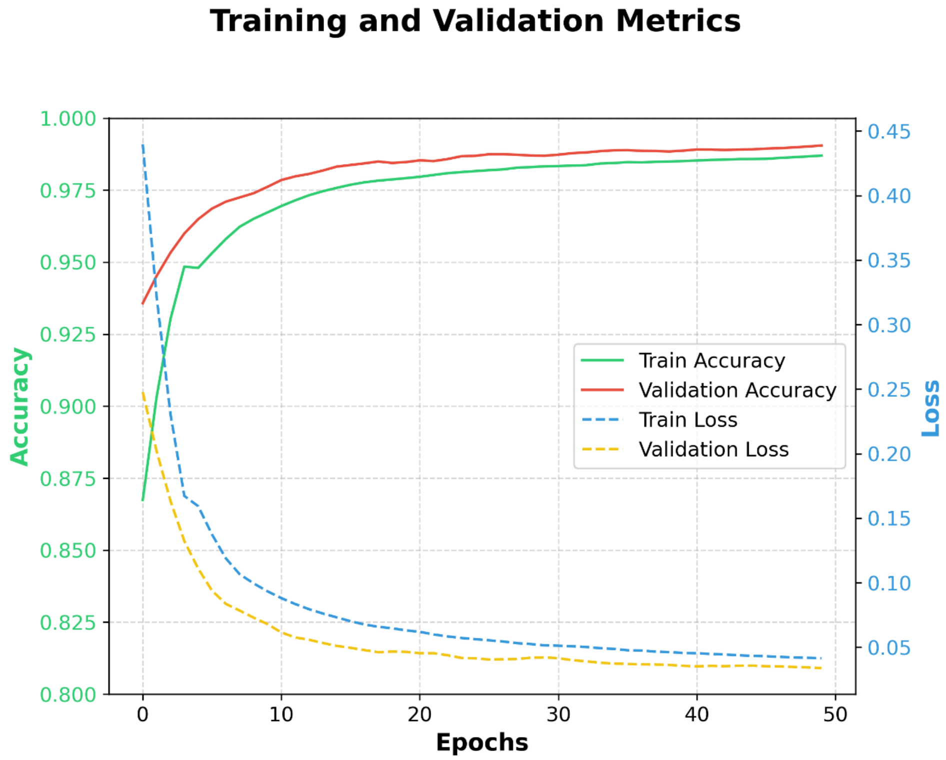


Fig. 8 Transformer-KAN five-class classification accuracy and loss curves

3.5 Ablation experiment

To assess the essentiality and contribution of each principal component within the proposed Transformer-KAN architecture, an ablation study was undertaken. This involved sequentially removing key modules to quantitatively evaluate the effect of individual components on overall model performance. The experiments were conducted under both binary and five-class classification frameworks.

1. Binary classification experiment

In this experiment, the layer parameters for the three models under comparison were standardized. The dataset, previously restructured, had its labels consolidated by amalgamating categories 1, 2, 3, and 4 into a single class labeled as ‘1’, designated as anomalous traffic. The classification metrics for the three models are detailed in Table 7.

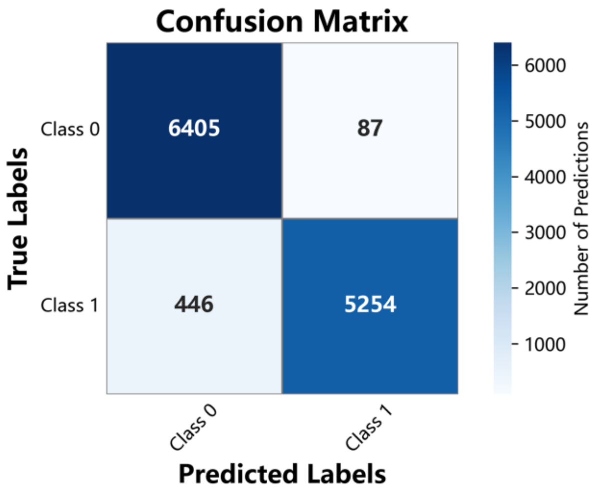
The Transformer-KAN model demonstrated superior performance over both the standalone Transformer and KAN models across all evaluated metrics. Notably, it achieved an accuracy that was 3.33% higher than that of the Transformer model and a 2.51% improvement relative to the KAN model. Additionally, the model exhibited enhanced performance in terms of both precision and recall, registering increases of 3.19% in precision and 3.33% in recall compared to the Transformer. These enhancements indicate a significant reduction in false positives (as evidenced by the increased precision) and an improved detection rate of positive cases (as reflected by the heightened recall). Moreover, the F1-score reached 98.96%, indicating that the model not only identifies positive cases with high accuracy but also effectively minimizes missed detections, thus achieving an optimal balance between precision and recall.

Table 7 Binary classification metrics

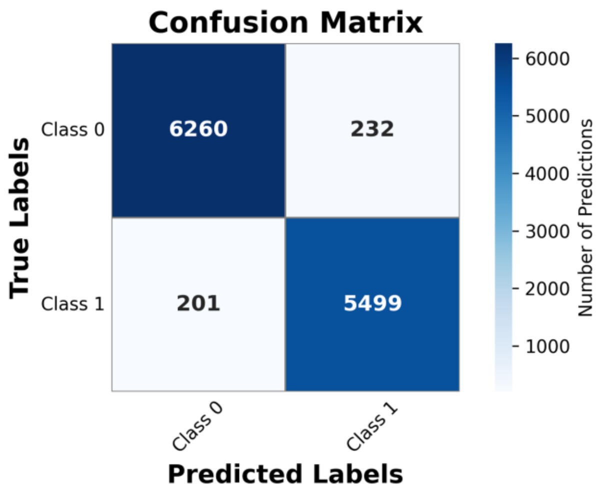
Model	Accuracy/%	Precision/%	Recall/%	F1-score /%
Transformer	95.63	95.77	95.63	95.62
KAN	96.45	96.45	96.45	96.45
Transformer-KAN	98.96	98.96	98.96	98.96

The confusion matrices for the three models—Transformer, KAN, and Transformer-KAN-are depicted in Figure 9, respectively. Analysis of these matrices elucidates distinct behavioral patterns among the models.

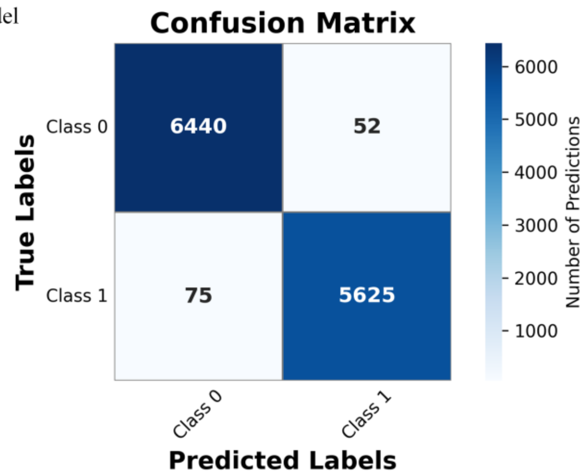
The Transformer model, adept at capturing global information and long-range dependencies, nevertheless displayed a notable rise in misclassifications for Class 1. Specifically, 446 samples assigned to Class 1 were incorrectly predicted as Class 0 (as shown in Figure 9.a). This issue may be attributed to the Transformer's propensity to learn global relationships extensively, which can lead to a bias toward the majority class (i.e., Class 0) in imbalanced datasets. Such a bias could consequently diminish the predictive accuracy for the minority class (Class 1).



(a) Binary Classification Confusion Matrix for the Transformer Model



(b) Binary Classification Confusion Matrix for the KAN Model



(c) Binary Classification Confusion Matrix for the Transformer-KAN Model

Fig. 9 Binary classification confusion matrix for the models

Conversely, the KAN model, adept at extracting local features, exhibits limitations in processing global information. Although its predictions for Class 0 are relatively accurate, it still results in a non-negligible number of misclassifications; specifically, 232 samples from Class 0 were misclassified as Class 1 (Figure 9.b). This issue likely stems from the KAN model's lack of robust global sequence modeling capabilities, which are characteristic of the Transformer. This deficiency may cause the model to overlook certain global patterns within complex datasets, thereby diminishing its predictive efficacy. Moreover, the KAN model's misclassifications for Class 1 are even more pronounced, with 201 samples erroneously assigned to Class 0 (Figure 9.b). This indicates that the performance of the KAN model on the minority class may be impeded by its strong reliance on local features.

The integrated Transformer-KAN model, by synergistically combining the global sequence modeling capabilities of the Transformer with the local feature extraction prowess of the KAN, achieves a more effective balance in predicting both majority and minority classes. In this integrated architecture, the Transformer component primarily addresses the capture of global long-range dependencies, while the KAN component augments the learning of local information, potentially through mechanisms such as convolutional layers and attention (as inferred from the architectural context, although the mechanisms within the KAN itself focus on learnable activation functions). This synergy results in enhanced overall predictive performance, as evidenced by a reduction in misclassifications across both classes compared to the performance of the standalone models (Figure 9.c).

2. Five-class classification experiment

The classification metrics for three models in the five-class experiment are delineated in Table 8. The Transformer model exhibited consistent performance in terms of both precision and recall, albeit it was slightly outperformed by both the KAN and Transformer-KAN models. Its marginally lower precision indicates a potential propensity for misclassification across certain categories.

The KAN model surpassed the Transformer model, particularly in aspects of precision and recall, underscoring its efficacy in feature extraction and sample classification. Nevertheless, it did not reach the performance levels of the Transformer-KAN model.

The Transformer-KAN model achieved the most exemplary results, sustaining high levels of accuracy, recall, and overall performance. This substantiates its superiority in managing complex multi-class classification tasks characterized by significant inter-class variances.

Table 8 Five-class classification metrics

Model	Accuracy/%	Precision/%	Recall/%	F1-score /%
Transformer	95.76	95.96	95.76	95.84
KAN	96.35	96.24	96.35	95.94
Transformer-KAN	99.03	99.02	99.03	99.01

The confusion matrices for the three models-Transformer, KAN, and Transformer-KAN-in this classification experiment are illustrated in Figures 10, respectively. Within these matrices, C1-C5 denote the five classes, corresponding to the categories utilized in numerical processing as detailed in Table 9. The names mentioned subsequently in the text also correlate with these respective classes. The analysis of these matrices yields several insights:

Table 9 Confusion matrix class correspondence

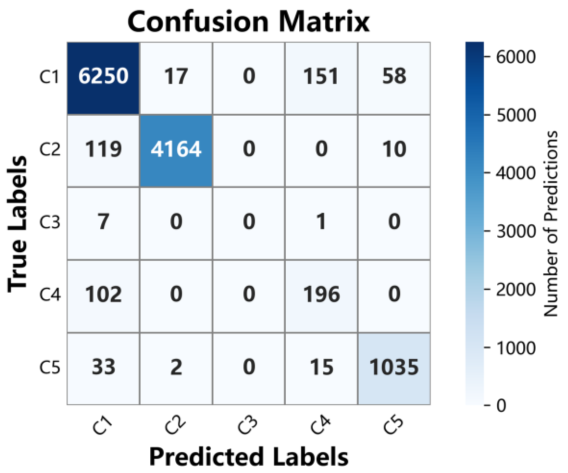
Category	Name
C1	Normal
C2	Dos
C3	U2R
C4	R2L
C5	Probe

The Transformer model displays relatively pronounced misclassifications for C1 (Normal) and C2 (DoS). Notably, a significant misclassification event involved 151 instances of C1 (Normal) being erroneously classified as R2L (Figure 10.a). This observation indicates limitations in the Transformer's capacity to navigate complex inter-class dynamics, likely due to its primary reliance on global sequence modeling, which may not effectively capture intricate non-linear decision boundaries.

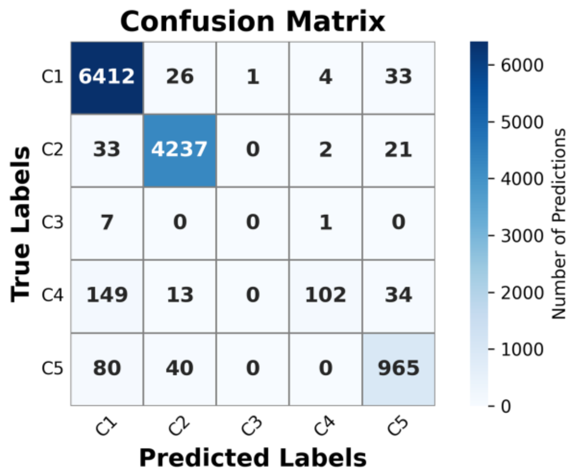
In contrast, the KAN model exhibits significant misclassification errors particularly for classes C4 (R2L) and C5 (Probe). Specifically, it misclassified 149 instances of C4 (R2L) as C1 (Normal), and 80 instances of C5 (Probe) were similarly misclassified as Normal (Figure 10.b). These results suggest that despite its robust non-linear modeling capabilities, the KAN model lacks efficacy in capturing global context and long-range dependencies.

Both the Transformer and KAN models demonstrated suboptimal performance in classifying C3 (U2R). The underlying causes appear to be multifaceted: (1) The Transformer model, despite its emphasis on global information, may neglect essential local features that are critical for accurately identifying C3 (U2R), thereby compromising its classification accuracy. (2) The KAN model, while possessing strong capabilities for non-linear fitting, does not adequately model global sequence dependencies. Consequently, it struggles to represent the distribution patterns of C3 (U2R) within the global feature space, resulting in poor discrimination, particularly when the feature distribution of U2R is complex or substantially overlaps with other classes.

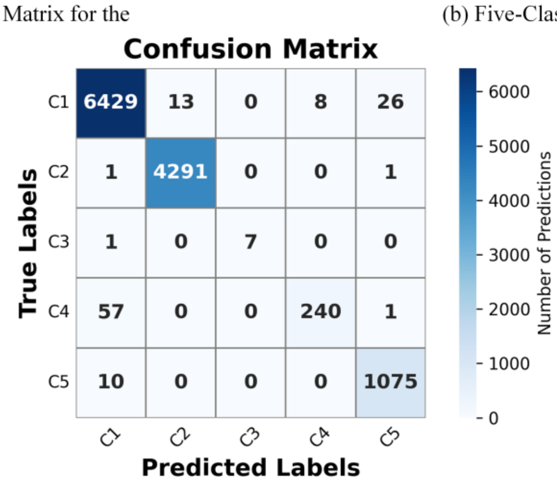
The hybrid Transformer-KAN model, which amalgamates the strengths of both the Transformer (global context capture) and the KAN (complex non-linear boundary fitting), demonstrates enhanced performance in classifying C3 (U2R). Additionally, this integrated model also shows improved accuracy in other classes, with a higher number of correctly classified instances compared to the standalone models (Figure 10.c).



(a) Five-Class Classification Confusion Matrix for the Transformer Model



(b) Five-Class Classification Confusion Matrix for the KAN Model



(c) Five-Class Classification Confusion Matrix for the Transformer-KAN Model

Fig. 10 Five-class classification confusion matrix for the models

3.6 Contrast experiment

To ascertain the superiority of the proposed Transformer-KAN-based intrusion detection methodology, a comparative analysis was undertaken against established DL and ML techniques. The efficacy of the Transformer-KAN approach was evaluated relative to conventional methods employing CNNs, GRUs, Deep Belief Networks (DBNs), and Support Vector Machines (SVMs). Each model underwent training under uniform dataset conditions, with optimization of their respective hyperparameters to ensure optimal performance, thereby enabling an equitable comparison.

The outcomes of the comparative experiments for the binary classification task are depicted in Figure 11. The findings indicate that the Transformer-KAN model outperformed all baseline models in accuracy, thereby affirming its enhanced capability for precise classification. Notably, the Transformer-KAN model exhibited exceptional performance metrics, achieving nearly identical precision and recall rates of 98.96%. This equilibrium underscores its robust capacity to minimize both false positives and false negatives, thus ensuring dependable identification of the positive class, ostensibly abnormal traffic. Consequently, the F1-score for the Transformer-KAN model, which represents a harmonic mean of precision and recall, achieved an apex of 98.96%. This figure substantially exceeds those recorded for other models, such as 97.45% for CNNs, underscoring its efficacy in accurately discerning relevant instances whilst minimizing misclassifications.

In conclusion, within the realm of binary classification, the Transformer-KAN model significantly enhances performance compared to traditional CNN, GRU, DBN, and SVM methodologies. This underscores its substantial promise for practical deployment in IDS.

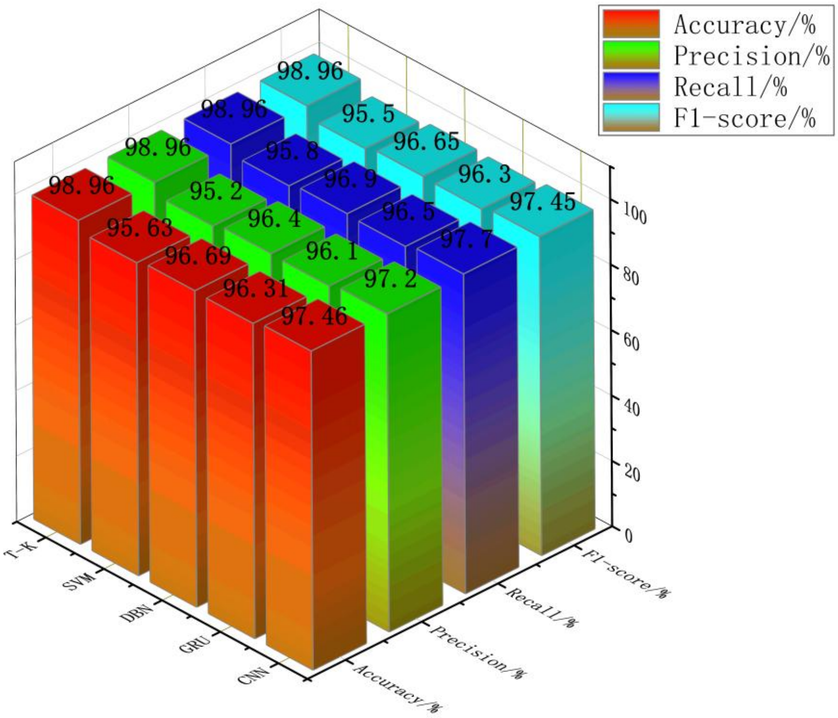


Fig. 11 Comparison of binary classification metrics

The experimental outcomes for the five-class classification task, as delineated in Table 10 and Figure 12, further substantiate the superior classification prowess of the Transformer-KAN model. In this regard, the accuracy of the Transformer-KAN model surpasses that of competing models by approximately 1.5% to 3.7%. Such results highlight its advanced capability to correctly classify a higher number of samples, thereby diminishing the likelihood of classification errors. Relative to traditional models like CNNs, GRUs, DBNs, and SVMs, the Transformer-KAN model not only achieves higher accuracy but also demonstrates a more favorable balance between precision and recall. This attribute renders the model markedly effective in managing complex multi-class challenges, thereby affirming its status as a highly efficient and formidable architectural choice.

Table 10 Five-class classification results

Type	Precision/%	Recall/%	F1-score/%
Normal	98.94	99.27	99.11
DOS	99.70	99.95	99.83
U2R	100.00	87.50	93.33
R2L	96.77	80.54	87.91
Probe	97.46	99.08	98.26

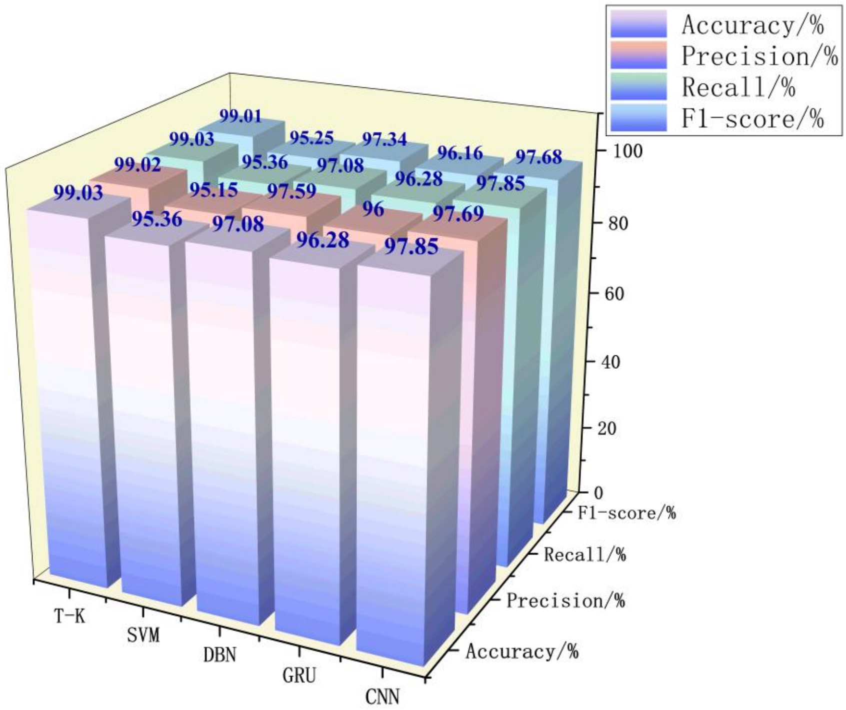


Fig. 12 Comparison of five-class classification metrics

The subsequent analysis explores the efficacy of the Transformer-KAN model in a five-class classification task, utilizing specific metrics to compare its precision, recall, and F1-score against other established methods.

Figure 13 elucidates the precision scores of the evaluated models within this task. Precision serves as an indicator of the accuracy of positive predictions made by a model, defined as the proportion of instances correctly identified as positive out of all positive predictions made. The Transformer-KAN model exhibited a precision of 99.02%, outperforming the baseline methods included in this analysis. This high precision score underscores the model’s exceptional capability in minimizing false positives-instances erroneously classified as belonging to a particular class. Thus, relative to alternative methodologies, the Transformer-KAN model exhibits superior reliability in accurate class assignment.

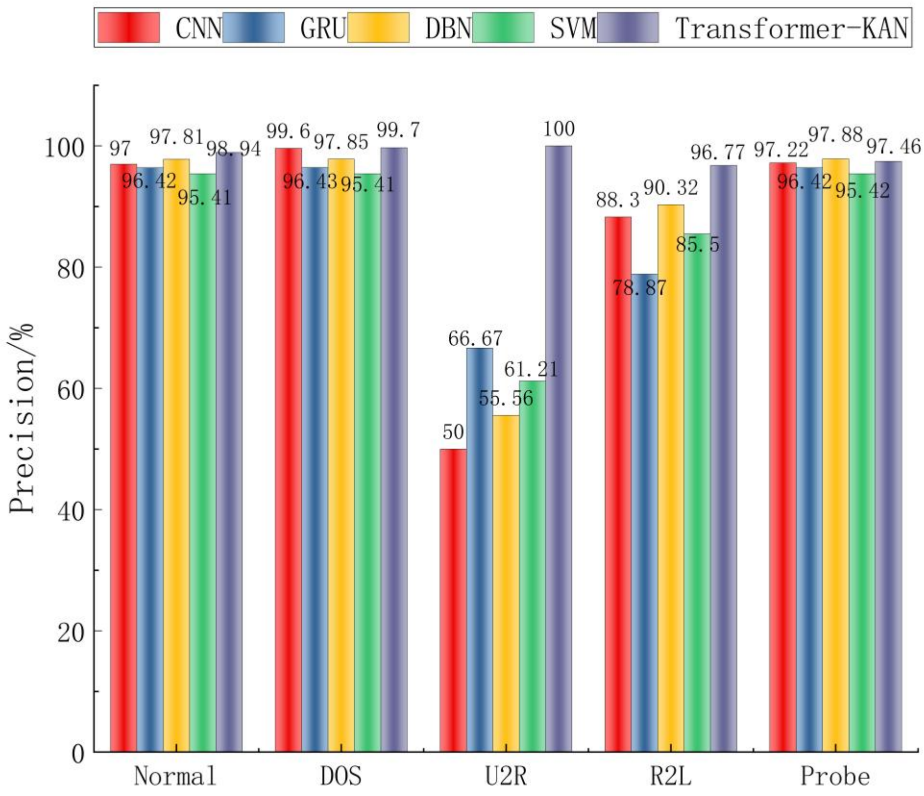


Fig. 13 Precision comparison

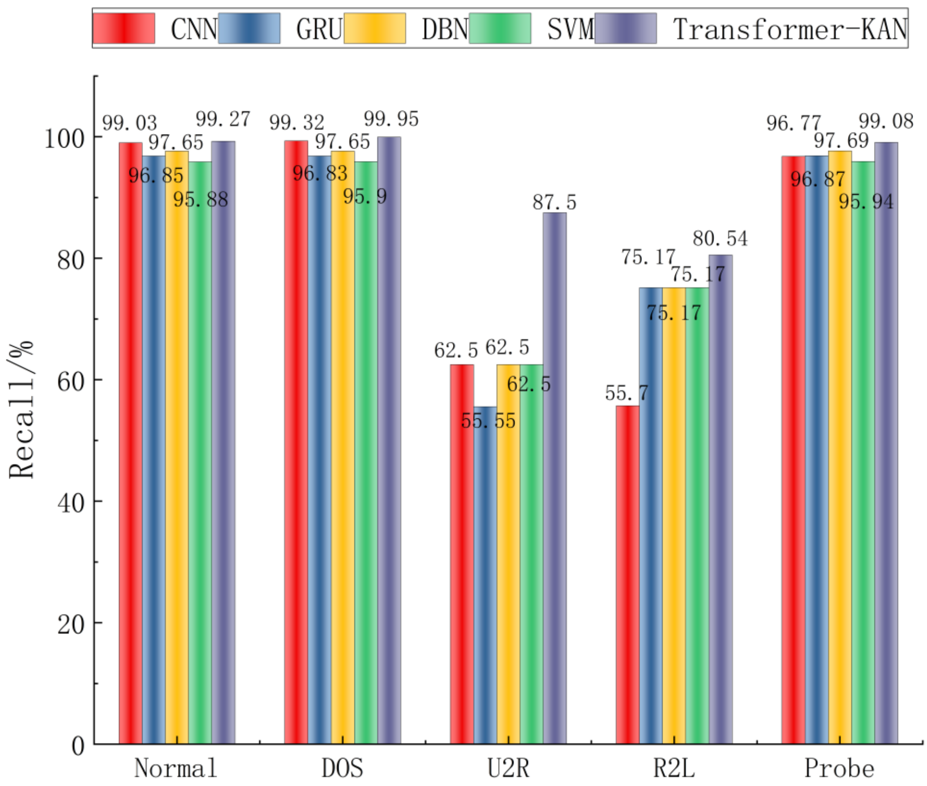


Fig. 14 Recall comparison

Figure 14 delineates the comparative results for recall-also referred to as sensitivity or the true positive rate-achieved by the Transformer-KAN model and the baseline methods. Recall measures the proportion of actual positive instances within the test set that were correctly identified by the model. The Transformer-KAN model achieved a recall score of 99.03%, significantly surpassing the recall metrics of other evaluated models.

This outstanding recall performance highlights the model's adeptness in precisely identifying the majority of actual positive samples, a crucial attribute as it indicates a significant reduction in false negatives-instances that truly belong to the positive class but are mistakenly classified as negative.

The F1-score, computed as the harmonic mean of precision and recall, serves as a robust, single metric for appraising overall model performance, offering a more intuitive assessment than analyzing precision and recall in isolation. Figure 15 exhibits the comparative F1-scores attained by the Transformer-KAN model and other DL methods, including CNNs. As a composite metric that effectively amalgamates both precision and recall, the F1-score reflects the model's balanced proficiency in minimizing both false positives and false negatives.

The Transformer-KAN model achieved an F1-score of 99.01%, highlighting its exceptionally balanced performance in the multi-class classification task. This score demonstrates the model's capacity to consistently achieve high precision-indicating that its class predictions are likely correct-and high recall-indicating effective identification of a large proportion of actual class instances. This synergy significantly enhances the model's overall classification performance.

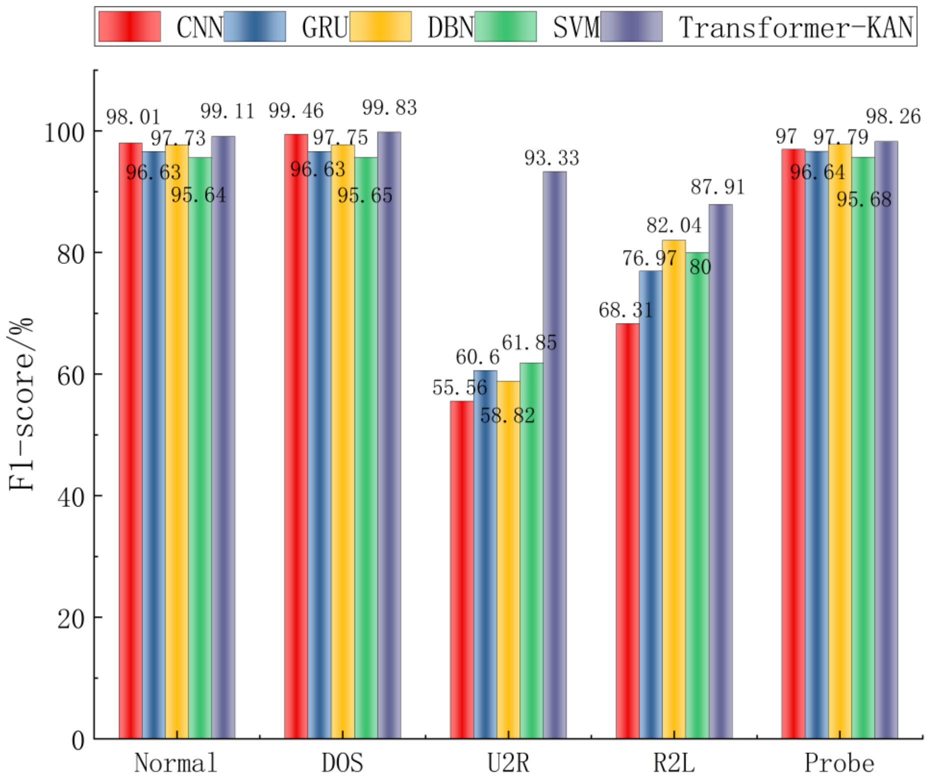


Fig. 15 F1-score comparison

3.7 Discussion

The development of the novel Transformer-KAN model effectively addresses the inherent limitations associated with utilizing either the Transformer or KAN architecture independently. This model ameliorates the constraints characteristic of standard Transformers. Although Transformers are adept at capturing long-range dependencies and global context through self-attention mechanisms, their reliance on fixed activation functions, such as ReLU, within the FFN layers may limit their ability to model complex, non-linear relationships among features adeptly. Moreover, the model confronts the limitations inherent in standalone KANs. KANs, which feature learnable, often spline-based activation functions, possess superior theoretical potential for approximating intricate functions and enhancing model interpretability. Nonetheless, KANs may lack robust mechanisms, which are intrinsic to Transformers, for capturing sequential dependencies or global contextual information. By synergistically integrating these two architectures, the Transformer-KAN model harnesses the powerful sequence modeling and context aggregation capabilities of the Transformer framework while concurrently benefiting from the enhanced adaptive learning and the ability to represent complex non-linear feature interactions afforded by KAN layers.

Comparative experiments have rigorously validated the effectiveness of the Transformer-KAN architecture against established baseline models (CNN, GRU, DBN, SVM) in both binary and five-class classification tasks. The results underscore the superior performance of the Transformer-KAN across all key evaluation metrics.

These outcomes represent a significant performance improvement margin compared to traditional DL and ML models. Notable advantages include:

Enhanced Classification Accuracy: The Transformer-KAN model demonstrated significantly higher accuracy scores, indicating superior overall instance classification capability relative to baseline methods. The absolute improvement in accuracy ranged from approximately 1.5% to 3.7%, highlighting a substantial enhancement.

Balanced Precision and Recall: Experimentally, the model's ability to concurrently achieve exceptionally high and well-balanced precision and recall scores was a critical advantage. This robustness minimizes both false positives and false negatives, thereby generating reliable and trustworthy predictions.

Effective Handling of Complexity: Particularly evident in the more challenging five-class task, the Transformer-KAN architecture exhibited proficiency in managing complex classification scenarios characterized by multiple, potentially overlapping classes. This capability is likely attributable to the enhanced non-linear modeling capabilities provided by the integrated KAN layers within the Transformer framework.

In summary, the Transformer-KAN model successfully merges the sequence processing capabilities of Transformers with the adaptive function approximation benefits of KANs. This integrative approach effectively surmounts the limitations inherent in the individual components, resulting in a model distinguished by its notably enhanced accuracy, balanced performance metrics, and an augmented ability to model complex data relationships. This claim is robustly substantiated by the exceptional outcomes observed in the comparative experiments presented.

4. Conclusions

This study addresses the operational demands of network intrusion detection for intelligent vessels by introducing a novel DL fusion model, termed Transformer-KAN. The model amalgamates the long-range dependency capturing ability of the Transformer encoder with the potent nonlinear fitting capacity and parameter efficiency of the KAN, specifically designed to address the principal challenges of limited computational resources and the prevalence of highly imbalanced attack data in maritime network environments.

Experimental findings affirm that the efficacy and necessity of each component within the proposed architecture were corroborated through ablation studies, revealing a synergistic effect engendered by the integrated design. In both binary and multi-class classification tasks during comparative experiments, the model surpassed several established ML and DL models in terms of overall accuracy, recall, and F1-score, thereby demonstrating its superior comprehensive performance.

The Transformer-KAN model exhibits significant potential to function as a pivotal component in future intelligent ship cybersecurity frameworks. Owing to its high accuracy and inherent suitability for lightweight deployment, it is ideally positioned for use in resource-limited onboard edge computing devices, facilitating real-time localized threat detection and substantially bolstering the proactive defense capabilities of intelligent vessels. The deployment of such an efficient and pragmatic network IDS constitutes an effective strategy for enhancing the overarching security landscape of maritime cybersecurity.

However, this study has several limitations.

1) Limitations in dataset relevance and timeliness

The NSL-KDD dataset, while widely adopted as a benchmark, does not accurately represent the attack types and network traffic characteristics found in modern intelligent shipping systems. These systems utilize specialized protocols such as NMEA 2000, Modbus TCP/IP, and satellite communication protocols within OT environments. Additionally, prevalent cyber threats like spoofing attacks targeting vessel AIS and GPS systems are absent from the NSL-KDD dataset. Although there are public intrusion detection datasets in other domains, such as the VeReMi dataset for vehicular networks, a dedicated open dataset for maritime intrusion detection does not exist.

Therefore, the high accuracy achieved by the model on the NSL-KDD dataset may not be a reliable indicator of its performance in actual intelligent ship network environments, significantly limiting the generalizability of our conclusions.

2) Trade-off between model complexity and computational efficiency

The paper outlines the benefits of the proposed method in resource-constrained scenarios, but this assertion requires a more objective evaluation. The self-attention mechanism in Transformers, for example, is inherently computationally demanding. Although KANs are more parameter-efficient, the overall computational load and inference speed of the integrated Transformer-KAN model necessitate thorough assessment.

Given that embedded systems and edge computing nodes on intelligent vessels usually face severe computational limits, the complexity and inference latency of the proposed model could impede its capability to fulfill real-time intrusion detection requirements, thereby limiting its practical utility. This issue represents a crucial trade-off that needs careful consideration.

3) Imbalanced detection capability across specific attack types (issues with recall and precision)

As evidenced by the five-class classification results in Table 10, the model demonstrates low recall in detecting U2R and R2L attacks. This indicates a propensity toward missed detections, such as misclassifying U2R or R2L attacks as normal traffic or other attack types. The primary reason for this is the extreme scarcity of samples from these categories in the NSL-KDD dataset, which impedes the model's ability to learn their complex and nuanced feature patterns.

The Transformer architecture is highly effective at capturing long-range dependencies, and KANs demonstrate a strong representational capacity. However, when dealing with highly imbalanced data distributions, models like these tend to prioritize performance optimization for dominant categories, such as DoS attacks. This issue is a widespread challenge in DL-based approaches.

To overcome the limitations mentioned above, this paper proposes several potential directions for future research. While the dataset employed in this study generally satisfies the training requirements for intrusion detection models, using a domain-specific dataset for validation would be more suitable. The primary task involves constructing or adopting a realistic network traffic dataset tailored for intelligent vessels, which should encompass modern shipboard network protocols and actual attack samples that target maritime systems. Evaluating the performance of the Transformer-KAN model on such a dataset is a crucial step towards its practical application.

In terms of model lightweighting and deployment optimization for real-time performance, future research could focus on improving efficiency. For example, applying knowledge distillation could transfer capabilities from a large model to a smaller, more compact one. Additionally, techniques such as pruning and quantization could be used to enhance the model's compatibility with edge computing devices. The actual throughput and latency of the deployed model should be empirically evaluated.

To address the trade-offs and limitations in detecting specific attack categories, the introduction of advanced loss functions such as Focal Loss could help alleviate class imbalance. Alternatively, employing data augmentation techniques to create high-quality synthetic samples for minority attack classes would also be beneficial. Another promising approach is cost-sensitive learning, which during training assigns different misclassification penalties to different classes, thereby steering the model's optimization towards more security-critical objectives.

Furthermore, the robustness of the model against adversarial attacks needs thorough investigation to ensure it cannot be easily compromised by maliciously crafted traffic.

FUNDING

This research was funded by High Technology Ship Research and Development Program, project "Development of liquid cargo and electromechanical simulation operation system for LNG ship", grant number CBG3N21-3-3.

REFERENCES

[1] Guo, J., Guo, H., 2023. Real-time risk detection method and protection strategy for intelligent ship network security based on cloud computing. *Symmetry*, 15(5), 988. https://doi.org/10.3390/sym15050988

[2] Ma, X., Shen, J., Liu, Y., Qiao, W., 2020. A methodology to evaluate the effectiveness of intelligent ship navigational information monitoring system. *IEEE Access*, 8, 193544-193559. https://doi.org/10.1109/ACCESS.2020.3032543

[3] Li, G., Deng, X., Zhou, M., Zhu, Q., Lan, J., Xia, H., Mitrouchev, P., 2020. Research on data monitoring system for intelligent ship. *Advanced Manufacturing and Automation IX,* IWAMA 2019, Lecture Notes in Electrical Engineering, vol 634, Springer Singapore, 234-241. https://doi.org/10.1007/978-981-15-2341-0_29

[4] Erstad, E., Hopcraft, R., Harish, A V., Tam, K., 2023. A human-centred design approach for the development and conducting of maritime cyber resilience training. *WMU Journal of Maritime affairs*, 22(2), 241-266. https://doi.org/10.1007/s13437-023-00304-7

[5] Karim, M. S., 2022. Maritime cybersecurity and the IMO legal instruments: Sluggish response to an escalating threat? *Marine Policy*, 143, 105138. https://doi.org/10.1016/j.marpol.2022.105138.

[6] Erbas, M., Khalil, S. M., Tsiopoulos, L., 2024. Systematic literature review of threat modeling and risk assessment in ship cybersecurity. *Ocean Engineering*, Volume 306, 118059, ISSN00298018. https://doi.org/10.1016/j.oceaneng.2024.118059.

[7] Vaarandi R., Tsiopoulos L., Visky G., Rehman M Ur., Bahşi H., 2025. A Systematic Literature Review of Cyber Security Monitoring in Maritime. *IEEE Access*, vol 13, 85307-85329. https://doi.org/10.1109/ACCESS.2025.3567385

[8] Ben Farah, M. A., Ukwandu, E., Hindy, H., Brosset, D., Bures, M., Andonovic, I., Bellekens, X., 2022. Cyber Security in the Maritime Industry: A Systematic Survey of Recent Advances and Future Trends. *Information*, 13, 22. https://doi.org/10.3390/info13010022

[9] Girdhar, M., Hong, J., Moore, J., 2023. Cybersecurity of Autonomous Vehicles: A Systematic Literature Review of Adversarial Attacks and Defense Models. *IEEE Open Journal of Vehicular Technology*, 4, 417-437. https://doi.org/10.1109/OJVT.2023.3265363

[10] Shin, G., Yang, H., 2025. Deep reinforcement learning for integrated vessel path planning with safe anchorage allocation. *Brodogradnja*, 76(3), 76305. https://doi.org/10.21278/brod76305

[11] Xin, X., Liu, K., Loughney, S., Wang, J., Li, H., 2023. Graph-based ship traffic partitioning for intelligent maritime surveillance in complex port waters. *Expert Systems with Applications*, 231, 120825. https://doi.org/10.1016/j.eswa.2023.120825

[12] Abdulganiyu, O. H., Ait Tchakoucht, T., Saheed, Y. K., 2023. A systematic literature review for network intrusion detection system (IDS). *International journal of information security*, 22(5), 1125-1162. https://doi.org/ 10.1007/s10207-023-00682-2

[13] Ozsari, I., 2023. Predicting main engine power and emissions for container, cargo, and tanker ships with artificial neural network analysis. *Brodogradnja*, 74(2), 77-94. https://doi.org/10.21278/brod74204

[14] Tang, H., Zhu, R., Wan, Q., Ren, D., 2025. Short-term prediction of trimaran load based on data driven technology. *Brodogradnja*, 76(1), 76101. https://doi.org/10.21278/brod76101

[15] Sarıkaya, A., Günel Kılıç, B., Demirci, M. 2022. GRU-GBM: A combined intrusion detection model using LightGBM and gated recurrent unit. *Expert Systems*, 39(9), e13067. https://doi.org/10.1111/exsy.13067

[16] Tian, Q., 2022. An Intrusion Detection Approach Based on Improved Deep Belief Network and LightGBM. *International Symposium on Computer Science and Intelligent Control (ISCSIC)*, 40-44. https://doi.org/10.1109/ISCSIC57216.2022.00020

[17] Du, X., Gan. G., 2024. Network Intrusion Detection Based on Self-Attention Mechanism and BIGRU. *International Conference On Mobile Internet, Cloud Computing and Information Security (MICCIS)*, 236-241. https://doi.org/10.1109/MICCIS63508.2024.00045

[18] Alkahtani, H., Aldhyani, T. H. H., 2021. Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms. *Complexity*, 2021(1), 5579851. https://doi.org/ 10.1155/2021/5579851

[19] Zhang, W., Zhang, Y., 2023. Research on classification and navigational risk factors of intelligent ship. *Brodogradnja*, 74(4), 105-128. https://doi.org/10.21278/brod74406

[20] Ahmad Z., Shahid Khan A., Wai Shiang C., Abdullah J., Ahmad F., 2021. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), e4150. https://doi.org/10.1002/ett.4150

[21] Lan, D., Xu, P., Nong, J., Song J., Zhao J., 2024. Application of Artificial Intelligence Technology in Vulnerability Analysis of Intelligent Ship Network. *International Journal of Computational Intelligence Systems*, 17, 147. https://doi.org/10.1007/s44196-024-00539-z

[22] Patil, S., Varadarajan, V., Mazhar, S. M., Sahibzada, A., Ahmed, N., Sinha, O., Kumar, S., Shaw, K., Kotecha, K., 2022. Explainable artificial intelligence for intrusion detection system. *Electronics*, 11(19), 3079. https://doi.org/10.3390/electronics11193079

[23] Zhang, C., Liu, J., Zhi, J., Zhang, X., Wang, J., Wu, Z., 2023. Oceancl-ids: A continual learning-based intrusion detection system for ocean-going ship-satellite communication network. *IEEE*, 1150-1155. https://doi.org/10.1109/ictis60134.2023.10243816

[24] Fakiha B., 2021. Business organization security strategies to cyber security threats. *International Journal of Safety and Security Engineering*, 11(1), 101-104. https://doi.org/10.18280/ijsse.110111

[25] Kummerow, A., Abrha, E., Eisenbach, M., Rösch, D., 2024. Unsupervised Anomaly Detection and Explanation in Network Traffic with Transformers. *Electronics*, 13(22), 4570. https://doi.org/10.3390/electronics13224570

[26] Ullah, S., Ahmad, J., Khan, M. A., Alshehri, M. S., Boulila, W., Koubaa, A., Jan, S U., Ch, M. I., 2023. TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT Networks. *Computer Networks*, 237, 110072. https://doi.org/ 10.1016/j.comnet.2023.110072

[27] Kim, T., Pak, W., 2023. Deep learning-based network intrusion detection using multiple image transformers. *Applied Sciences*, 13(5), 2754. https://doi.org/10.3390/app13052754

[28] Yao, R., Wang, N., Chen, P., Ma, D., Sheng, X., 2023. A CNN-transformer hybrid approach for an intrusion detection system in advanced metering infrastructure. *Multimedia Tools and Applications*, 82(13), 19463-19486. https://doi.org/10.1007/s11042-022-14121-2

[29] Wang, M., Yang, N., Weng, N., 2023. Securing a smart home with a transformer-based iot intrusion detection system. *Electronics*, 12(9), 2100. https://doi.org/ 10.3390/electronics12092100

[30] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., Tegmark, M., 2024. Kan: Kolmogorov-arnold networks. *arXiv Is Hiring a DevOps Engineer*, arXiv:2404.19756. https://doi.org/10.48550/arXiv.2404.19756

[31] Meng, L., Gan, H., Liu, H., Lu, D., 2025. Deep learning-based research on fault warning for marine dual fuel engines. *Brodogradnja*, 76(3), 76303. https://doi.org/10.21278/brod76303

[32] Cherednichenko, O., Poptsova, M., 2025. Kolmogorov–Arnold networks for genomic tasks. *Briefings in Bioinformatics*, 26(2), bbaf129. https://doi.org/10.1093/bib/bbaf129

[33] Amouri, A., Al Rahhal, M. M., Bazi, Y., Alaparthy, V. T., Fotopoulos, G., 2024. A Hybrid based Kolmogorov-Arnold Network for Intrusion Detection Systems in IoT. *IEEE*, Middle East Conference on Communications and Networking (MECOM). IEEE: 247-250. https://doi.org/10.1109/MECOM61498.2024.10881369

[34] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A., 2009. A detailed analysis of the KDD CUP 99 data set. *IEEE* https://doi.org/10.1109/cisda.2009.5356528