RESEARCH PAPER

PANDAS: A Pooled Attention and Deep Attention-based Spatial–Temporal Network for Intrusion Detection

Maduri Madhavi [1], Dr.NP Nethravathi[2]

[1]Research scholar, REVA University, School of CSE,India
[2]Professor, REVA University, School of CSE,India
Email: [1] madurimadhavi@gmail.com, [2] nethravathi.np@reva.edu.in

Abstract- The increasing complexity and volume of cyberattacks pose significant challenges to conventional intrusion detection approaches, as they depend on handcrafted features and limited temporal modeling. To overcome these challenges, this study introduces a novel Pooled Attention and Deep Attention-based Spatial–Temporal Network (PANDAS) for multi- category network intrusion detection. The proposed model combines pooled spatial–temporal attention with a 3D convolutional encoder–decoder architecture to effectively capture discriminative patterns in network traffic data. A Pooled Attention Module is employed to jointly emphasize informative feature regions and critical temporal segments, enhancing spatiotemporal feature learning. The model is evaluated on the Multi-Class Network Traffic Attack Dataset (MC-NTAD), comprising normal traffic and multiple flooding-based attack classes. Experiments reveal that the PANDAS model provides outstanding results, achieving more than 99% overall detection and nearly perfect per-class classification**.** Comparative analysis with advanced machine learning(ML) and deep learning(DL) techniques further validates the strength and reliability of the proposed method. These results indicate that PANDAS provides a reliable and scalable approach to detecting network intrusions within dynamic network settings.

Keywords- Pooled Attention Network; Network Traffic Analysis; Cybersecurity; 3D Convolutional Neural Networks; Multi-Class Attack Detection

How to cite this article: Madhavi M, Nethravathi N. PANDAS: A Pooled Attention and Deep Attention-based Spatial–Temporal Network for Intrusion Detection. Int J Drug Deliv Technol. 2026;16(50s): 1611-1620. DOI: 10.25258/ijddt.16.50s.161

I. INTRODUCTION

The Internet has emerged as a crucial aspect of modern life, enabling communication, business operations, cloud services, and large-scale data storage for individuals and organizations [1], [2]. This growing reliance on Internet-based infrastructures has simultaneously increased concerns regarding security and privacy, as modern networks are frequently targeted by cyber-attacks and anomalous activities [3]. Consequently, cybersecurity has received significant interest from academic and industrial sectors, resulting in the deployment of various protection mechanisms to safeguard networked systems.

Typical protective tools, including firewalls, authentication protocols, antivirus applications, and malware detection utilities are widely used as the first line of defense [4]. Although effective against known threats, these techniques are insufficient for protecting networks from contemporary and sophisticated attacks, particularly zero-day and evolving intrusion patterns [5]. To overcome these limitations, Intrusion detection systems(IDSs) serve as a vital security mechanism continuously monitoring network traffic to identify malicious behavior [26].

IDSs are generally categorized as signature-based and anomaly-based detection systems [6], [27]. Signature-driven IDSs utilize established attack patterns and exhibit high accuracy for known attacks; however, they are inherently incapable of identifying novel or previously unseen intrusions [7], [28]. Conversely, anomaly- driven IDSs identify intrusions by recognizing abnormal patterns in network behavior**,** enabling them to uncover previously unseen attacks. [8]. Despite their advantages, anomaly-based approaches often suffer from high false alarm rates and performance degradation when used on large, high-dimensional network traffic datasets [9].

With the accelerated expansion of Internet usage and network complexity**,** ML techniques are increasingly employed to improve IDS effectiveness. Classical ML models, including RF [23], DT [18], NB [19], and SVM [20] have been widely utilized for detecting network intrusions. Nevertheless, these shallow learning models rely extensively on manually engineered features and face difficulties scaling with large, high-dimensional traffic datasets and dynamically evolving attack patterns. Additionally, real-world network traffic data are typically characterized by class imbalance, where normal traffic dominates and minority attack classes are underrepresented, leading to biased learning and poor detection of rare attacks [10], [17].

The latest progress in deep learning offers considerable promise for overcoming the limitations of conventional ML-based IDSs. DL approaches can automatically learn hierarchical and discriminative representations from large datasets, thereby enhancing detection accuracy and robustness [27], [29]. Various DL architectures, including DNNs, CNNs, RNNs, LSTMs, and autoencoders, have found successful applications in network intrusion detection [21], [22]. These models have repeatedly outperformed standard ML methods**,**

*Author for Corresponden madurimadhavi@gmail.com

particularly in complex and large-scale network environments.

Despite these advancements, existing DL-based IDSs still face several critical challenges. First, many approaches fail to adequately model the spatial and temporal patterns present in network traffic, which limits their ability to detect sophisticated and evolving attack behaviors [10]. Second, the persistent problem of class imbalance continues to degrade detection performance for minority attack categories, even in deep learning-based frameworks [11]–[13]. Although data-level techniques such as SMOTE, ADASYN, and GAN-based augmentation aim to resolve this challenge [14]–[16], they may cause redundant data, increase computational demands, or reduce model generalization. Third, most existing IDS models do not jointly model spatial–temporal dependencies, resulting in incomplete feature representation and reduced detection robustness [30][31].

To mitigate these challenges, attention mechanisms have recently been introduced into IDS research to enhance feature learning by emphasizing the most relevant regions of the data [24], [25]. Attention-based models improve detection accuracy by suppressing irrelevant features and emphasizing attack-sensitive patterns. However, most existing attention-driven IDSs apply either spatial or temporal attention independently, without effectively exploiting their complementary strengths in spatiotemporal traffic analysis [29].

Motivated by these limitations, this paper proposes a PANDAS (Pooled Attention and Deep Attention-based Spatial–Temporal Network) model for network intrusion detection. The proposed framework integrates 3D convolutional neural networks with Region Attention and Time Attention mechanisms to simultaneously capture spatial and temporal relationships within network traffic. A novel Pooled Attention Module (PAM) is introduced to fuse spatial and temporal attention representations, enabling more discriminative and robust feature extraction. This design improves detection accuracy, reduces false alarms, and enhances the recognition of underrepresented attack types in imbalanced datasets**.**

The primary contributions of this research are outlined as follows**:**

1. A new spatiotemporal DL architecture based on 3D CNNs is developed for network intrusion detection.
2. Region Attention and Time Attention mechanisms are incorporated to enhance spatial and temporal feature learning.
3. A Pooled Attention Module (PAM) is proposed to jointly model spatial–temporal dependencies for improved intrusion detection.
4. Comprehensive experiments highlight the performance advantages of the PANDAS model in comparison with current ML and DL-based IDS approaches.

The rest of the paper is organized as follows**:** Section 2 discusses related studies on ML, DL, and attention-driven intrusion detection systems; Section 3 explains the proposed PANDAS framework in detail; Section 4 outlines the experimental setup and evaluation outcomes; Section 5 provides the conclusions.

II. LITERATURE SURVEY

A. Related Works

Intrusion detection has remained a prominent research area for many years due to the escalating frequency of cyber attacks and the increasing intricacy of modern networks. Several studies have examined different methods for enhancing IDS accuracy, reliability, and flexibility, spanning from rule-based systems to sophisticated deep learning models.

Early IDS research primarily focused on signature-based and rule-driven mechanisms that utilized predefined attack patterns to recognize malicious behavior. These systems demonstrated strong detection performance for previously identified intrusions; however, their inability to recognize new or zero-day attacks limited their effectiveness in dynamic network environments [32,33]. To overcome this drawback, anomaly-based IDSs were introduced, which establish profiles of typical network behavior and identify deviations as possible intrusions[34]. Although anomaly-based approaches can identify previously unseen attacks, they are often sensitive to noise and experience elevated false positive rates, especially in large-scale and diverse networks [35].

ML algorithms such as SVM, DT, NB, KNN, and RF have been extensively applied to intrusion detection owing to their capacity to identify patterns in network traffic data. SVMs [20] are effective in handling high-dimensional data through margin maximization but suffer from high computational cost and sensitivity to parameter selection. Decision Trees [18] offer fast and interpretable classification but are prone to overfitting and poor generalization in noisy and imbalanced datasets. Random Forests [23] improve detection effectiveness and stability by combining multiple decision trees; however, they still rely on handcrafted features and struggle with non-stationary traffic distributions. Naïve Bayes classifiers [19] provide computational efficiency and scalability but are limited by unrealistic feature independence assumptions. KNN-based approaches [36] can achieve good accuracy in well-labeled datasets but incur high inference cost and are highly sensitive to noise and class imbalance, often biasing predictions toward majority traffic classes.

Recent progress in DL has substantially influenced intrusion detection research by facilitating automated extraction of hierarchical and discriminative features from network traffic data. Several studies have explored different DL architectures for improving detection accuracy, robustness, and adaptability to complex attack patterns. In [10], the authors proposed a DNN-based intrusion detection framework that leverages multiple hidden layers to learn nonlinear relationships among network traffic features. These models enhance detection performance over conventional machine learning methods by

automatically extracting high-level feature representations. However, DNN-based IDSs require large labeled datasets and involve intensive computational demands, which can restrict their scalability and suitability for real-time applications. In [37], network traffic features were transformed into two-dimensional matrices and processed using convolutional layers to extract local spatial patterns indicative of attack behaviors. The developed CNN model demonstrated enhanced detection performance relative to classical ML classifiers, particularly for known attack categories. Similarly, the authors in [38] enhanced CNN performance by incorporating multiple convolutional and pooling layers to improve feature abstraction. Despite these improvements, CNN-based IDSs often neglect temporal dependencies between traffic flows and require careful feature reshaping, which may lead to information loss and increased computational overhead.

To address the temporal nature of network traffic, RNN-based IDSs were introduced in [39]. These models improve the detection of time-dependent intrusions such as probing and flooding attacks. However, conventional RNNs suffer from gradient degradation problems, limiting their capacity to capture long-range dependencies.

LSTM networks were explored in [40] to address the shortcomings of conventional RNNs by integrating memory cells with gating functions. IDSs built on LSTM architectures achieved higher detection precision and fewer false alerts. Nevertheless, their high computational cost and long training times hinder deployment in resource-constrained and real-time environments.

Autoencoder-based intrusion detection methods have gained attention for their ability to learn data representations without full label dependence. In [21], an autoencoder was fitted using benign network traffic to extract compact latent representations, and reconstruction errors were used to identify anomalous behavior. These approaches are effective for identifying unknown and previously unseen attacks without the need for labeled datasets. However, their performance is sensitive to threshold selection and may generate false positives for rare but legitimate traffic patterns.

Class imbalance remains a significant issue in the field of IDs, as normal traffic and frequent attack types dominate most benchmark datasets, while rare but critical attacks appear infrequently. This imbalance leads to classifiers biased toward dominant classes, resulting in poor detection of minority attack types. To mitigate this problem, approaches at the data level, including oversampling, under sampling, SMOTE, and ADASYN have been proposed. The SMOTE was proposed to alleviate the limitations of random by creating synthetic samples based on the closest neighboring instances [42]. However, SMOTE does not consider class boundaries explicitly and may generate overlapping or noisy samples, which can negatively affect model generalization. Adaptive Synthetic Sampling (ADASYN) enhances SMOTE through adaptive synthesis of more samples that are difficult to recognize [41]. In intrusion detection applications, ADASYN-based approaches enhanced detection performance for infrequent and complex attacks by focusing on hard-to-classify regions of the feature space.

Recent research has explored GANs to produce realistic synthetic data representing underrepresented attack categories [43]. GAN-based IDS frameworks demonstrated superior data diversity and improved minority-class detection compared to traditional oversampling techniques. However, GAN training is computationally intensive and sensitive to model configuration, and insufficiently trained generators might generate low-fidelity or duplicate samples, adversely impacting detection robustness. Alternatively, cost-aware learning approaches impose greater penalties for misclassifying minority classes, guiding the system to prioritize rare but important attacks [44]. However, effectively handling class imbalance while maintaining detection robustness remains an open research problem.

Attention mechanisms have recently emerged as a promising direction in IDS research due to their ability to emphasize informative features and suppress irrelevant information. Attention-based IDS models exhibit superior detection results and better interpretability against to conventional deep learning techniques[29]. Since network traffic inherently contains both spatial and temporal characteristics, spatiotemporal modeling has gained increasing importance. Hybrid architectures integrating CNNs to extract spatial features with RNNs or LSTMs sequential dependencies has been explored to capture complex traffic patterns [45]. Nevertheless, many existing models treat spatial and temporal dimensions independently, limiting their ability to capture joint dependencies. Furthermore, single-attention mechanisms may fail to fully exploit complementary spatial–temporal relationships present in network traffic data.

B. Limitations of Existing Methods and Research Motivation

Although significant progress has been made in intrusion detection using ML and DL techniques, several limitations remain unresolved. Many IDS models lack a unified framework for jointly modeling spatial and temporal dependencies. Class imbalance remains a barrier to detecting minority attacks, and existing attention-based methods often apply spatial or temporal attention in isolation. Additionally, the increasing complexity of network environments demands robust and scalable architectures capable of learning discriminative spatiotemporal representations.

These limitations motivate the necessity of advanced IDS frameworks that integrate deep spatiotemporal feature extraction with pooled attention mechanisms to improve detection accuracy, robustness, and generalization in real-world network scenarios.

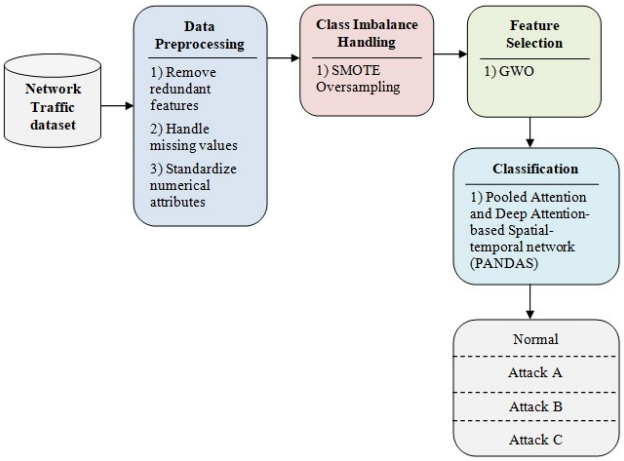

Fig. 1 Configuration of the proposed framework for intrusion identification

III. PROPOSED METHODOLOGY

Fig. 1 show the proposed intrusion detection framework follows a structured workflow that begins with data preprocessing, where redundant and non-informative features are eliminated, missing data are filled in, and numerical attributes are standardized. Class imbalance is addressed through SMOTE to generate extra instances for underrepresented attack categories. Relevant features are then selected using Grey Wolf Optimization (GWO) combined with mutual information ranking to reduce the number of features while preserving key discriminative details. The extracted features are reshaped into tensors suitable for region–time modeling and fed into the proposed Pooled Attention and Deep Attention-based Spatial-temporal network (PANDAS), which incorporates a Pooled Attention Module (PAM) combining region and time attention to enhance feature learning. The attention-enhanced representations are processed through a 3D convolutional encoder–decoder network, with fully connected layers producing multi-class predictions.

A. Data Preprocessing

Initially, the acquired network traffic data is loaded into the model for further processing. Data preprocessing is performed in three stages: data encoding**,** normalization**,** and class imbalance handling**.**

(a) Data Encoding: To represent categorical attributes in the dataset, including network protocols and traffic types, are transformed using one-hot encoding (OHE). This method transforms each categorical variable into a binary vector, where each column corresponds to a distinct category and only one element per row is set to "1," to denote the corresponding category. The one-hot encoding can be mathematically formulated as:

$$OHE(cat) = \{y_1, y_2, \cdots, y_K\}$$

where $y_K \in \{0,1\}$ denotes the encoded value for the K^{th} category.

(b) Data Normalization: After encoding, feature values are normalized using Z-score to achieve zero mean and unit variance across all features. This method enhances model stability and improves convergence during training. The Z-score normalization is given by:

$$y_K^{'} = \frac{y_K - \mu_K}{\sigma_K}$$

where μ_K and σ_K are the mean and standard deviation for the K^{th} feature, and $y_K^{'}$ is the normalized value. Standardization guarantees that variables with wider numeric ranges do not overshadow other features during training contribute proportionally during model training.

(c) Class Imbalance: To mitigate class imbalance, the work utilizes SMOTE, a well-established method for data augmentation in ML [42]. In this approach, minority class instances are first identified, and for each such instance, its K-nearest neighbors are determined. Artificial samples are created by combining a minority instance with its closest neighbors. These generated instances are then merged with the original data to form a balanced dataset.

The SMOTE process can be mathematically expressed as:

$$y_{new} = y_K + \Re \times (y_K - y_i)$$

where y_K and y_i denoting the K-nearest neighbor and the minority class feature vectors, respectively, and $\Re$ being a random variable in the interval [0,1] that determines the interpolation factor. By scaling $\Re$ appropriately and repeating the process, multiple synthetic instances are generated, creating a balanced dataset suitable for training the intrusion detection model.

B. Feature Selection Using Grey Wolf Optimization (GWO)

The GWO employs swarm intelligence and draws inspiration from the social structure and cooperative hunting strategies of grey wolves [46]. Wild wolves form packs divided into four hierarchical roles: Alpha (leader), Beta (advisors), Delta (support and sentinels), and Omega (lowest rank). This hierarchy directs the optimization procedure, where alpha, beta, and delta

*Author for Corresponden madurimadhavi@gmail.com

wolves correspond to the top three solutions, while remaining wolves update their positions according to these leaders to explore the search space effectively.

The GWO algorithm mimics three main behaviors: encircling prey, hunting, and position updating. The encircling mechanism is formulated mathematically as:

$$D = | K.Q_p - Q(t) |, \quad Q(t+1) = Q(t) - A \cdot .D$$

where Q_p denotes the prey's position vector, Q represents the wolf's position vector, and A and K are defined coefficient vectors:

$$A = 2a \cdot rand_1 - a, \quad K = 2 \cdot rand_2$$

Here, a linearly declines from 2 to 0, while $rand_1$, $rand_2$ are random vectors in the interval [0,1].

The hunting behavior relies on the three leading wolves (Alpha, Beta, and Delta) to guide the pack. The positions of other wolves are updated as:

$$Q(t+1) = \frac{1}{3}(Q_1 + Q_2 + Q_3)$$

Where

$$Q_1 = Q_\alpha - A_1 \cdot D_\alpha, \; Q_2 = Q_\beta - A_2 \cdot D_\beta, \; Q_3 = Q_\delta - A_3 \cdot D_\delta$$

And

$$D_\alpha = | K_1 - Q_\alpha - Q |, \; D_\beta = | K_2 - Q_\beta - Q |, \; D_\delta = | K_3 - Q_\delta - Q |$$

This ensures that all wolves update their positions based on the leaders, maintaining a balance between exploration and exploitation in the search space.

For feature selection, the continuous GWO outputs are converted into binary values to represent inclusion (1) or exclusion (0) of features:

$$Z_k = round(| y_k \bmod 2 |) \bmod 2$$

Where y_k is the continuous GWO solution and $Z_k \in \{0,1\}$ determines whether a feature is selected.

The GWO approach is computationally efficient, requires minimal parameter tuning, and balances exploration and exploitation efficiently, enabling the selection of informative features from complex, high-dimensional network traffic data.

C. Pooled Attention and Deep Attention-based Spatial-temporal network(PANDAS)

After feature selection and balancing, the optimized feature vectors are reshaped into multi-dimensional tensors suitable for region–time modeling. This transformation facilitates the model's ability to extract both spatial correlations among features and temporal patterns across network traffic. The resulting spatiotemporal tensors are then fed into the proposed Pooled Attention Network for Detection of Attacks System (PANDAS)**.**

The main part of the proposed framework is the PANDAS architecture as show in Fig. 2, which integrates a Pooled Attention Module (PAM) to enhance feature learning. The PAM combines Region Attention and Time Attention mechanisms to selectively emphasize informative feature regions and critical temporal segments. Region attention highlights discriminative spatial feature patterns, while time attention captures important temporal dynamics in network traffic. By pooling these complementary attention representations, PAM enables joint spatial–temporal feature refinement, the model's capability to recognize complex and evolving attack behaviors.

The attention-enhanced feature maps are subsequently processed through a deep 3D convolutional encoder–decoder network. The encoder employs a series of 3D convolutional layers with batch normalization and pooling operations for extracting hierarchical spatiotemporal representations, while progressively reducing feature dimensionality. The decoder uses 3D transpose convolution layers to reconstruct salient features from the compressed representations, ensuring that important spatial and temporal information is retained for classification.

Finally, the decoder output is flattened and passed through dense layers with a subsequent softmax activation to produce predictions for multiple intrusion classes. Training is performed using sparse categorical cross-entropy, with optimization via the Adam optimizer to ensure stable and efficient convergence. This integrated methodology enables robust intrusion detection by jointly addressing feature redundancy, class imbalance, and spatiotemporal dependency modeling within a unified deep learning framework.

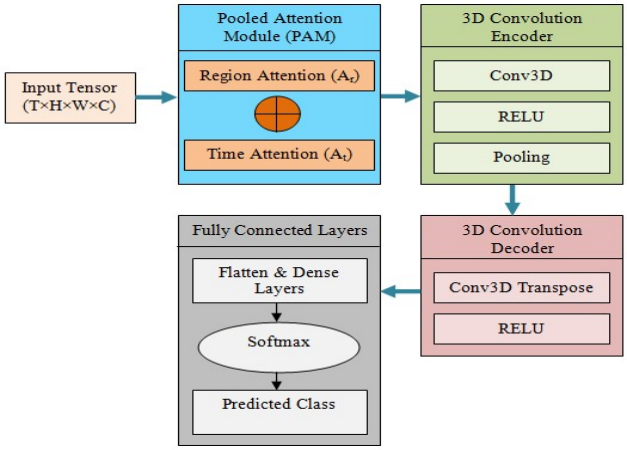

Fig 2. PANDAS Model architecture

1) Pooled Attention Module (PAM)

The Pooled Attention Module integrates Region Attention and Time Attention to enhance spatiotemporal feature learning. Given an input tensor $X \in \mathrm{R}^{T\times H\times W\times C}$,

where T denoting the temporal dimension (time steps or traffic windows), H and W denote spatial dimensions corresponding to reshaped feature regions, and C indicating the total number of feature channels. The objective of PAM is to adaptively reweight this tensor such that salient spatial and temporal features contribute more strongly to subsequent learning stages.

Region Attention Block (RAB): The Region Attention Block is designed to capture spatial relevance by identifying discriminative regions within the feature maps. In the context of intrusion detection, certain combinations of traffic attributes or feature groups are more indicative of malicious behavior than others. RAB assigns higher importance to such regions while suppressing less informative or redundant spatial features.

To achieve this, global average pooling (GAP) is first applied across the spatial dimensions H and W, aggregating region-wise information while preserving temporal and channel-wise characteristics. This operation produces a compact representation that summarizes the spatial distribution of features:

$$GAP_{H,W}(X) = \frac{1}{H\times W}\sum_{h=1}^{H}\sum_{w=1}^{W} X(t,h,w,c)$$

The pooled features are subsequently fed into a 3D convolution layer to model interdependencies among channels and temporal steps, followed by a sigmoid activation generating normalized attention weights:

$$A_r = \sigma(Con3D(GAP_{H,W}(X))),$$

where $\sigma(\cdot)$ ensures that attention weights lie in the range [0,1]. The resulting attention map A_r encodes the relative importance of different spatial regions for intrusion detection.

The region-enhanced feature tensor is obtained through element-wise multiplication:

$$X_r = X \otimes A_r$$

This operation enhances spatial regions that are strongly correlated with attack patterns while suppressing irrelevant background features. Consequently, the model becomes more sensitive to localized spatial anomalies that may indicate intrusions.

Time Attention Block (TAB): While spatial attention captures region-wise importance, network intrusions also exhibit strong temporal characteristics, such as bursty traffic, repeated access attempts, or abnormal sequence patterns. The Time Attention Block is specifically designed to model such temporal dependencies by assigning adaptive weights along the temporal axis.

Similar to RAB, TAB begins by applying global average pooling, but this time across the temporal dimension T. This operation effectively summarizes temporal dynamics while retaining spatial and channel-wise information:

$$GAP_T(X) = \frac{1}{T}\sum_{t=1}^{T} X(t,h,w,c).$$

The pooled temporal representation is processed using a 3D convolution layer followed by a sigmoid activation to generate temporal attention weights:

$$A_t = \sigma(Con3D(GAP_T(X)))$$

These weights reflect the relative importance of different time steps in identifying intrusions. The time-attended feature tensor is then computed as:

$$X_t = X \otimes A_t$$

By emphasizing critical temporal segments, TAB enables the model to focus on suspicious traffic intervals while reducing the influence of benign or repetitive patterns.

Attention Pooling and Feature Fusion: After deriving the region-enhanced and time-enhanced representations, PAM integrates both attentions to form a unified spatiotemporal feature representation. Rather than processing spatial and temporal attentions independently, PAM performs a pooling-based fusion to jointly capture their complementary effects:

$$X_{PAM} = X_r + X_t$$

This additive fusion approach maintains the contributions of both attention mechanisms while avoiding excessive parameter growth. The resulting feature tensor emphasizes discriminative spatial regions that are also temporally significant, facilitating robust detection of complex and evolving attack behaviors.

2) 3D Convolutional Encoder

The encoder consists of a series of stacked 3D convolutional layers that operate across temporal and spatial dimensions simultaneously. Each layer applies a set of learnable 3D kernels to extract localized spatiotemporal features, enabling the model to capture complex interactions between feature regions over time. Formally, the output of a 3D convolution can be expressed as:

$$Y = f\left(\sum_{k=1}^{K} W_k * X + b\right)$$

where W_k denotes the convolutional kernels, b is the bias term, f(·) is a nonlinear activation function, and ∗*∗ represents the 3D convolution operation.

Batch Normalization layers are employed after each convolution to stabilize gradient flow and accelerate convergence by normalizing intermediate feature distributions. Nonlinear activation functions, such as ReLU, introduce nonlinearity and improve the network's capacity to model complex attack patterns. Max pooling layers are interleaved with convolutional layers to downsample feature maps, reduce spatial dimensions, and retain the most salient features:

$$Y_{pool} = \max_{(i,j,k)\in\Omega} Y(i,j,k)$$

where Ω denotes the pooling region.

Through successive convolution and pooling operations, the encoder progressively abstracts low-level traffic characteristics into high-level spatiotemporal representations, capturing both short-term fluctuations and long-range temporal dependencies associated with intrusion behaviors.

3) Decoder and Feature Reconstruction

To preserve discriminative information and improve classification robustness, the compressed representations produced by the encoder are fed into a decoder block. The decoder utilizes 3D transpose convolution (deconvolution) layers to upsample feature maps and reconstruct critical spatiotemporal patterns from the encoded representations. The transpose convolution operation is defined as:

$$X' = Conv3DTranspose(Y)$$

where Y is the encoded feature tensor. This reconstruction process helps retain critical spatial and temporal details that may be lost during pooling, thereby enhancing the expressiveness of the learned features.

The encoder–decoder structure also acts as a regularization mechanism, forcing the network to learn compact yet informative representations, which improves generalization and reduces overfitting.

4) Feature Aggregation and Classification

After decoding, the feature maps are vectorized and processed through dense layers to aggregate global spatiotemporal information. These dense layers further refine feature representations by learning complex decision boundaries between normal and malicious traffic classes. Dropout regularization is applied to mitigate overfitting by temporarily disabling neurons at random during training.

The last layer uses softmax activation to yield probabilities for each intrusion category:

$$\overline{y_c} = \frac{e^{z_c}}{\sum_{k=1}^{C} e^{z_k}}$$

where z_c denotes the logit corresponding to class c, and C represents the total number of classes. The class with the largest probability is assigned as the model's output.

IV. RESULT AND DISCUSSION

The dataset utilized in this study is the Multi-Class Network Traffic Attack Dataset (MC-NTAD)**,** which consists of 80,000 packet-level network traffic records labeled to represent normal behavior and multiple network attack types. Each record corresponds to a single network packet and includes 21 features covering temporal information, protocols, IP addresses, ports, packet length, and cumulative byte statistics. The dataset is categorized into five classes: Normal traffic (40,000 samples) and four attack classes—ICMP Flood, UDP Flood, TCP SYN Flood, and LAND Flood—with 10,000 samples each. Owing to its detailed feature set and balanced class distribution, the dataset is well suited for network intrusion detection, traffic analysis, and machine learning–based attack classification. Data transformation is conducted through label encoding, which converts categorical data into numerical format, facilitating further analysis. An 80-20 partitioning is applied, allocating 80% of the data to the training set and 20% to the testing set.

Table 1 Overall Performance of the Proposed PANDAS Model

Metric	Value (%)
Accuracy(acc)	99.98
Precision(pre)	99.95
Recall(rec)	99.97
F1-score(f1)	99.96

A) Proposed Evaluation

To evaluate the effectiveness of the proposed framework, standard metrics including accuracy(acc), precision(pre), recall(rec), and F1-score(f1) are employed. The experimental results shown in Table I indicate that the PANDAS model attains an overall

accuracy The experimental results shown in Table I indicate that the PANDAS model attains an overall accuracy of 99.98%, with corresponding precision(pre), recall(rec), and F1-score(f1) values of 99.95%, 99.97%, and 99.96%, respectively. of 99.98%, with corresponding precision, recall, and F1-score values of 99.95%, 99.97%, and 99.96%, respectively. These outcomes demonstrate that the proposed method achieves robust IDs with a well-maintained trade-off between false positives(FPs) and false negatives(FNs). The high recall value is particularly important for security applications, as it confirms the model's capability to identify malicious activities while minimizing missed detections.

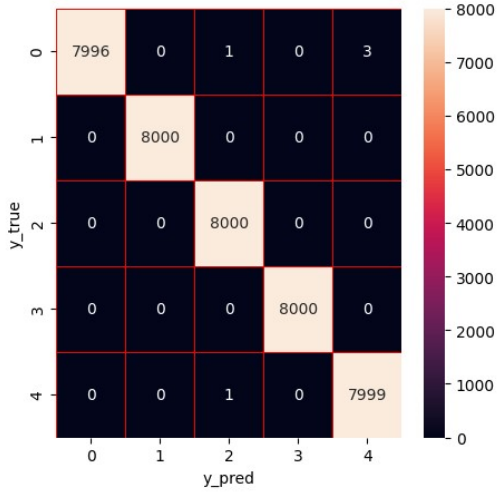

Fig. 3 Confusion Matrix

Based on the confusion matrix as shown in fig. 3, the proposed PANDAS model achieves near-perfect class-wise performance in percentage terms. Normal traffic (Class 0) achieves 99.95% accuracy, with 0.05% of samples misclassified, mainly confused with the LAND Flood class. ICMP Flood (Class 1)**,** UDP Flood (Class 2)**,** and TCP SYN Flood (Class 3) are all classified with 100% accuracy**,** showing no false positives or false negatives. LAND Flood (Class 4) achieves 99.99% accuracy, with only 0.01% of samples misclassified as UDP Flood. Overall, the confusion matrix reflects an average class-wise accuracy above 99.98%**,** confirming the strong robustness and reliable multi-class discrimination capability of the proposed PANDAS intrusion detection framework.

B) Comparison Results

Results in Table II indicate that the PANDAS model surpasses both traditional ML and advanced DL approaches across all evaluation metrics. Classical machine learning models like SVM, DT, RF, and KNN show relatively lower performance due to their dependence on handcrafted features and capability of representing temporal patterns present in network traffic data. Although Random Forest improves detection accuracy through ensemble learning, it still lacks adaptability to evolving attack patterns.

Deep learning models, including CNN and LSTM, achieve higher accuracy by learning hierarchical spatial features and temporal dependencies, respectively. The hybrid CNN–LSTM model further enhances performance by combining spatial and temporal learning. However, these models treat all features and time steps uniformly, which limits their capacity to attend to the most significant regions and key time intervals regions and critical time intervals.

Table II Performance Comparison with Existing ML and DL Techniques

Model	Accuracy (acc) (%)	Precision (pre) (%)	Recall (rec) (%)	F1-Score (f1) (%)
SVM[20]	92.40	91.85	92.10	91.97
Decision Tree [18]	90.75	89.90	90.30	90.10
Random Forest[23]	95.60	95.10	95.25	95.17
KNN [36]	91.85	91.20	91.40	91.30
CNN [38]	96.90	96.45	96.70	96.57
LSTM [40]	97.40	97.10	97.25	97.17
CNN–LSTM [47]	98.20	97.95	98.05	98.00
Proposed PANDAS	**99.98**	**99.95**	**99.97**	**99.96**

In contrast, the proposed PANDAS framework achieves the highest accuracy(acc) of 99.98% and an F1-score(f1) of 99.96%**,** demonstrating superior detection capability. This performance improvement is primarily attributed to the integration of the Pooled Attention Module, which jointly emphasizes discriminative spatial regions and significant temporal segments, and the 3D convolutional encoder–decoder architecture, which effectively captures complex spatiotemporal correlations. The near-perfect class-wise results observed in the confusion matrix further demonstrate the resilience and reliability of the proposed model. Overall, the comparative evaluation confirms that PANDAS provides a more accurate and robust solution for multi-class intrusion detection compared to existing ML and Dl models, making it well suited for deployment in modern network security environments.

V. CONCLUSION

This paper introduced PANDAS, an attention-enhanced DL framework for multi-class network intrusion detection. By integrating pooled spatial and temporal attention with 3D convolutional encoder–decoder architecture, the proposed model effectively captures complex spatiotemporal patterns in network traffic. Experimental evaluation on the MC-NTAD dataset demonstrated high detection accuracy, balanced class-wise performance, and minimal misclassification across all attack categories. Comparative results confirmed that PANDAS outperforms conventional ML and existing DL models. The strong performance underscores the effectiveness of pooled attention in improving feature discrimination and intrusion detection reliability. Future research will aim to expand the framework for real-time deployment, analysis of encrypted traffic, and intrusion detection.

References

[1] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," Expert Systems with Applications, vol. 36, no. 10, pp. 11 994–12 000, 2009.

[2] A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, "Student engagement level in e learning environment: Clustering using k-means," American Journal of Distance Education, vol. 34, no. 2, 2019.

[3] M. Injadat, F. Salo, and A. B. Nassif, "Data mining techniques in social media: A survey," Neurocomputing, vol. 214, pp. 654 – 670, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092523121630683X

[4] M. B. Salem, S. Hershkop, and S. J. Stolfo, "A survey of insider attack detection research," in Insider Attack and Cyber Security. Springer, 2008, pp. 69–90.

[5] W. Bul'ajoul, A. James, and M. Pannu, "Improving network intrusion detection system performance through quality of service configuration and parallel technology," Journal of Computer and System Sciences, vol. 81, no. 6, pp. 981–999, 2015.

[6] S. M. H. Bamakan, B. Amiri, M. Mirzabagheri, and Y. Shi, "A new intrusion detection approach using pso based multiple criteria linear programming," Procedia Computer Science, vol. 55, pp. 231–237, 2015.

[7] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," Applied soft computing, vol. 10, no. 1, pp. 1–35, 2010.

[8] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, 2013.

[9] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 4, pp. 70–73, 2014.

[10] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, ''Survey of intrusion detection systems: Techniques, datasets and challenges,'' Cybersecurity, vol. 2, no. 1, pp. 1–22, Dec. 2019.

[11] H. Chindove and D. Brown, ''Adaptive machine learning based network intrusion detection,'' in Proc. Int. Conf. Artif. Intell. Appl., Dec. 2021, pp. 1–6.

[12] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, ''Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives,'' in Proc. IEEE 3rd Int. Conf. Comput., Commun. Secur. (ICCCS), Oct. 2018, pp. 1–8.

[13] T. Liu, X. Zhu, W. Pedrycz, and Z. Li, ''A design of information granulebased under-sampling method in imbalanced data classification,'' Soft Comput., vol. 24, no. 22, pp. 17333–17347, Nov. 2020.

[14] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, ''An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset,'' Comput. Netw., vol. 177, Aug. 2020, Art. no. 107315.

[15] J. Liu, Y. Gao, and F. Hu, ''A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM,'' Comput. Secur., vol. 106, Jul. 2021, Art. no. 102289.

[16] Y. Lu, ''Intrusion detection classification method based on generative adversarial networks,'' in Proc. 3rd Int. Conf. Frontiers Electron., Inf. Comput. Technol. (ICFEICT), May 2023, pp. 344–349.

[17] Q. Xu, S. Lu, W. Jia, and C. Jiang, ''Imbalanced fault diagnosis of rotating machinery via multi-domain feature extraction and cost-sensitive learning,'' J. Intell. Manuf., vol. 31, no. 6, pp. 1467–1481, Aug. 2020.

[18] L. E. Jim and J. Chacko, ''Decision tree based AIS strategy for intrusion detection in MANET,'' in Proc. IEEE Region 10th Conf. (TENCON), Oct. 2019, pp. 1191–1195.

[19] B. G. Narendrasinh and D. Vdevyas, ''FLBS: Fuzzy lion Bayes system for intrusion detection in wireless communication network,'' J. Central South Univ., vol. 26, no. 11, pp. 3017–3033, Nov. 2019.

[20] H. Zhang, Y. Li, Z. Lv, A. K. Sangaiah, and T. Huang, ''A real-time and ubiquitous network attack detection based on deep belief network and support vector machine,'' IEEE/CAA J. Autom. Sinica, vol. 7, no. 3, pp. 790–799, May 2020.

[21] S. Longari, D. H. N. Valcarcel, M. Zago, M. Carminati, and S. Zanero, ''CANnolo: An anomaly detection system based on LSTM autoencoders for controller area network,'' IEEE Trans. Netw. Service Manage., vol. 18, no. 2, pp. 1913–1924, Jun. 2021.

[22] G. C. Fernández and S. Xu, ''A case study on using deep learning for network intrusion

detection,'' in Proc. IEEE Mil. Commun. Conf. (MILCOM), Nov. 2019, pp. 1–6.

[23] Y. Duan, X. Li, X. Yang, and L. Yang, ''Network security situation factor extraction based on random forest of information gain,'' in Proc. 4th Int. Conf. Big Data Comput., 2019, pp. 194–197.

[24] O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, ''EIDM: Deep learning model for IoT intrusion detection systems,'' J. Supercomput., vol. 79, no. 12, pp. 13241–13261, Aug. 2023.

[25] P. Choobdar, M. Naderan, and M. Naderan, ''Detection and multi-class classification of intrusion in software defined networks using stacked autoencoders and CICIDS2017 dataset,'' Wireless Pers. Commun., vol. 123, no. 1, pp. 437–471, Mar. 2022

[26] B. A. Tama and K.-H. Rhee, ''An extensive empirical evaluation of classifier ensembles for intrusion detection task,'' Comput. Syst. Sci. Eng., vol. 32, no. 2, pp. 149–158, 2017

[27] Y. LeCun, Y. Bengio, and G. Hinton, ''Deep learning,'' Nature, vol. 521, no. 7553, pp. 436–444, May 2015

[28] R. Sommer and V. Paxson, "Enhancing byte-level network intrusion detection signatures with context," in Proceedings of the 10th ACM conference on Computer and communications security, 2003, pp. 262– 271

[29] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE transactions on emerging topics in computational intelligence, vol. 2, no. 1, pp. 41–50, 2018.

[30] Y. Yu and N. Bian, ''An intrusion detection method using few-shot learning,'' IEEE Access, vol. 8, pp. 49730–49740, 2020, doi: 10.1109/ACCESS. 2020.2980136.

[31] F. H. Botes, L. Leenen, and R. D. L. Harpe, ''Ant colony induced decision trees for intrusion detection,'' in Proc. Eur. Conf. Cyber Warfare Secur., 2017, pp. 53–62.

[32] J. P. Anderson, "Computer security threat monitoring and surveillance," *Technical Report*, 1980.

[33] D. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, 1987.

[34] S. Axelsson, "Intrusion detection systems: A survey and Taxonomy," *Technical Report*, 2000.

[35] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *IEEE Symposium on Security and Privacy*, 2010.

[36] Liao, Y., & Vemuri, V. R. (2002). Use of K-Nearest Neighbor classifier for intrusion detection. *Computers & Security*, *21*(5), 439–448. https://doi.org/10.1016/S0167-4048(02)00514-X

[37] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[38] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "HAST-IDS: Learning hierarchical spatial–temporal features using deep neural networks," *IEEE Access*, vol. 6, pp. 1792–1806, 2018

[39] Hossain, M. D., Inoue, H., Ochiai, H., Fall, D., & Kadobayashi, Y. (2020). LSTM-Based Intrusion Detection System for In-Vehicle Can Bus Communications. *IEEE Access*, *8*, 185489–185502. https://doi.org/10.1109/ACCESS.2020.3029307

[40] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access*, *5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

[41] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *IEEE International Joint Conference on Neural Networks*, pp. 1322–1328, 2008.

[42] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[43] Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: Generative adversarial networks for attack generation against intrusion detection," *IEEE Access*, vol. 8, pp. 48327–48340, 2020.

[44] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[45] Wang, Z., & Ghaleb, F. A. (2023). An Attention-Based Convolutional Neural Network for Intrusion Detection Model. *IEEE Access*, *11*, 43116–43127. https://doi.org/10.1109/ACCESS.2023.3271408

[46] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis, *Grey Wolf Optimizer*, *Advances in Engineering Software*, Volume 69, Pages 46–61, 2014.

[47] Altunay, H. C., & Albayrak, Z. (2023). A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks. *Engineering Science and Technology, an International Journal*, *38*, 101322. https://doi.org/10.1016/j.jestch.2022.101322