ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

Network Intrusion Detection Method Based on Adaptive Synthetic Sampling and an Improved Convolutional Neural Network

[1] N. SAI RAHUL, [2] C. SUMANA SRI , [3] P. ROHAN

[4] Ms. D. PRATHYUSHA, [5] Dr. K.VASANTH KUMAR [6] Mr. LALAM RAMU

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING - INTERNET OF THINGS

MALLA REDDY ENGINEERING COLLEGE (MREC) - HYDERABAD, TELANGANA

Corresponding author: N. Sai Rahul (mkssps42@gmail.com)

ABSTRACT

The differing qualities of organize assaults postures serious challenges to interruption location frameworks (IDSs). Conventional assault acknowledgment strategies more often than not receive mining information affiliations to recognize inconsistencies, which has the impediments of a tall untrue caution rate (Distant), moo acknowledgment exactness (ACC) and destitute generalization capacity. To enhance the comprehensive capabilities of IDS and reinforce organize security, we propose a novel interruption location strategy based on the versatile manufactured testing (ADASYN) calculation and an moved forward convolutional neural organize (CNN). To begin with, we utilize the ADASYN strategy to adjust the test conveyance, which can viably avoid the demonstrate from being touchy to huge tests and overlook little tests. Moment, the progressed CNN is based on the part convolution module (SPCCNN), which can increment the differences of highlights and dispose of the affect of interchannel data excess on show preparing. At that point, an AS-CNN demonstrate blended with ADASYN and SPC-CNN is utilized for interruption discovery errands. At long last, the standard NSL-KDD dataset is chosen to test AS-CNN. The reenactment outlines that the precision is 4.60% and 2.79% higher than that of the conventional CNN and RNN models, and the location rate (DR) compared with the two models

Keywords - Intrusion detection, adaptive synthetic sampling, AS-CNN, NSL-KDD.

I. INTRODUCTION

The swift adoption of 5G technology has led to its deployment in increasingly sophisticated scenarios. However, the inherent openness and distributed nature of wireless networks make them prime targets for attacks, hence the heightened focus on network security. Standard security measures include firewalls, encryption, and access controls, but attack identification systems become critical when these defenses are compromised. Common types of network attacks are user to root (U2R), denial of service (DoS), remote to local (R2L), and probe attacks. An effective Intrusion Detection System (IDS) needs to accurately classify these attacks based on learned behaviors, enabling early anomaly detection and preventative actions.

Traditionally, attack identification utilizes machine learning techniques, which involve data preprocessing, feature selection, and classification. Data preprocessing enhances feature recognition, with feature encoding being a prevalent technique. The level of feature discretization varies with the encoding method used, often impacting classification performance positively. A novel character-level encoding has been proposed to explore feature interrelations better.

The process of feature analysis and selection is pivotal to the accuracy and efficiency of IDS. Techniques like PCA have been used for selecting crucial features, boosting IDS efficiency significantly. Additionally, feature reduction and augmentation methods such as optimized LDA, CCS, and edge density ratio conversion have been applied to generate superior feature sets.

At the core of IDS are the classifiers that categorize network traffic post-preprocessing. Various classifiers have been developed, including naïve Bayes, decision trees, artificial neural networks, and support vector machines. To harness the strengths of multiple classifiers, a mixed approach using decision trees and k-means, as well as a combination of SVM and k-means, have been explored to enhance attack identification capabilities. These hybrid classifiers aim to integrate the benefits of various methodologies to

improve detection rates and reduce false positives. By combining techniques such as decision trees, which are effective for handling categorical data, with k-means clustering, which can identify patterns and anomalies in unlabeled data sets, the resulting system can provide more robust and accurate detection of complex threats.

Moreover, integrating SVM with k-means allows for a balance between the robust classification capabilities of SVM and the unsupervised learning advantages of k-means. This synergy can enhance the classifier's ability to discern subtle distinctions between normal behavior and potential threats, particularly in vast and dynamic network environments where new types of attacks emerge frequently.

The evolution of IDS solutions must continually adapt to the changing landscape of network threats facilitated by advancements in technology like 5G. The combination of advanced preprocessing techniques, strategic feature selection, and innovative classifier designs are essential to developing systems that can effectively anticipate and mitigate these evolving security challenges. As such, ongoing research and development in this area are crucial to maintaining the integrity and security of modern wireless networks.

II. RELATED WORK

The evolution of attack recognition technology has progressed through three primary stages: pattern matching algorithms, machine learning (ML) algorithms, and deep learning (DL) algorithms. Initially, intrusion detection tasks employed pattern matching algorithms based on feature matching. Improved versions of classical algorithms significantly enhanced the timeliness of Intrusion Detection Systems (IDS). Further developments applied additional pattern matching algorithms to these tasks, assessing their execution efficiency. However, due to the diversity of network attacks, these algorithms s t ruggled to adapt to contemporary network environments.

Machine learning-based attack recognition algorithms later supplanted traditional pattern matching techniques, demonstrating superior performance in IDS tasks. Supervised learning models like support vector machines (SVM) were adapted for attack recognition, with enhancements like chi-square feature selection leading to notable improvements in model accuracy and efficiency. Novel systems combined relevant feature screening with decision trees, and dynamic recursive feature selection algorithms extended with convolutional neural networks proposed intelligent algorithms that improved the detection of unknown attacks. Enhanced IDS using Bayesian networks and feature selection algorithms also showed increased accuracy. Despite their successes, these ML algorithms required extensive feature engineering and complex parameter adjustments.

As the focus shifted towards deep learning, the need for intricate feature engineering diminished. DL algorithms could autonomously abstract features from basic network traffic. For instance, classifiers using long short-term memory (LSTM) with optimization techniques effectively mined temporal feature associations. Models combining attention mechanisms with bidirectional LSTM (BLSTM) extracted and refined features for network anomaly detection. Enhancements like particle swarm optimization further improved the anomaly detection capabilities of deep belief networks (DBN). Methods integrating association rules with improved deep neural networks (DNN) also saw enhanced recognition accuracy due to effective feature association mining. Despite these a d v a n c e m e n t s , s o m e m o d e l s i n c r e a s e d computational complexity due to feature enhancement methods.

Convolutional neural networks (CNN) have been effectively employed in intrusion detection, extracting network traffic characteristics more efficiently. Simpler CNN models that converted data into a 2D format demonstrated significant improvements in detection efficiency. Models like the multistage feature-based CNN utilized outputs from all convolutional layers to enhance the model’s expressive ability. Cross-layer aggregated CNN models also significantly boosted feature expression capabilities. Although CNN-based attack recognition algorithms enhanced detection accuracy, they often overlooked the redundancy of inter-channel information in convolution layers, leading to challenges in discerning which channels might be redundant.

To address the issue of interchannel information redundancy, a new approach was proposed that split the convolution layer into two parts: a representative part and a potentially redundant part, with hierarchical processing applied thereafter. This concept inspired the development of a CNN equipped with this new mechanism, applied to attack recognition tasks. The performance of this

new CNN model showed clear advantages over traditional models. Before deployment, data augmentation techniques were used to reduce model sensitivity to large sample biases. A hybrid model combining this new CNN approach with data augmentation techniques exhibited superior performance compared to using the innovative CNN alone.

III. PROPOSED SYSTEM

A. RESEARCH CONCEPT

Figure 1 outlines the structure of the enhanced IDS, which is divided into four main components:

Part 1: Data Preprocessing. This stage involves three key processes: numerical transformations, normalization, and ADASYN data augmentation. These adjustments help modify the original data format, reduce disparities between different feature measurements, and ensure a more balanced distribution of samples, significantly enhancing the overall effectiveness of the CNN model.

Part 2: Network Design. In this segment, the SPC-CNN model is introduced to extract multiscale features and effectively address the issue of interchannel information redundancy. This model incorporates both advanced methods and fundamental data processing techniques.

Part 3: Model Training and Testing. This phase includes the training activities where network parameters are continuously tweaked to achieve model convergence. Additionally, testing is conducted after each training phase to determine if the model should be finalized based on the test outcomes.

Part 4: Evaluation. This section demonstrates the practicality and effectiveness of the new model. The performance of the optimization method is impartially assessed using metrics such as False Acceptance Rate (FAR), Accuracy (ACC), and Detection Rate (DR).

B. DATA ANALYSIS AND PREPROCESSING

The NSL-KDD dataset is commonly utilized as a standard benchmark for assessing the detection capabilities of IDS. This dataset is divided into three subsets: KDDTrain, KDDTest-21, and KDDTest+. Each entry in the dataset is a connection vector featuring one label and 41 attributes, which includes 38 numerical attributes and 3 categorical attributes. Table 1 displays the

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

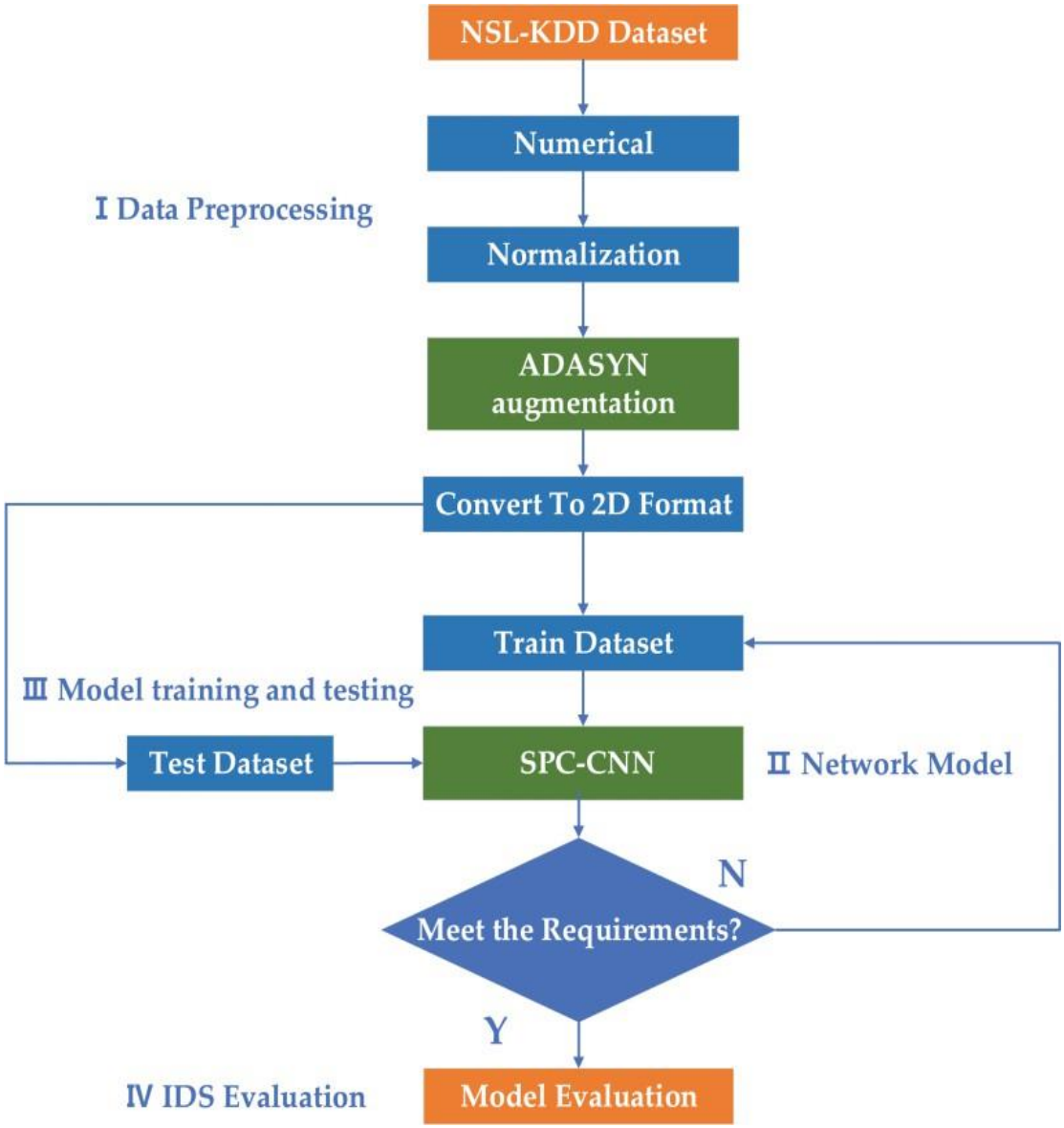

FIGURE 1. The flowchart of AS-CNN model. (AS-CNN is composed of ADASYN and SPC-CNN).

distribution of categories within the KDDTrain subset. The NSL-KDD dataset comprises five categories of samples: Dos, Probe, R2L, U2R, and Normal. The four non-normal categories are further broken down into 39 specific types of attacks, with 22 types present in the training set and 17 types not previously seen in the testing set. This split is designed to evaluate the model's ability to generalize. The KDDTest-21 and KDDTest+ datasets include different types of abnormal samples, with KDDTest-21 often serving as an additional benchmark alongside KDDTest+ to test the model's generalization performance.

1) NUMERICAL

The dataset initially includes three types of categorical features: 3 protocol types, 70 service types, and 11 connection states. However, the input requirement for the SPC-CNN is a standardized 2D numerical matrix. To accommodate this, a one-hot encoding technique is applied to these categorical features to transform them into a numerical format.

TABLE 1. Category distribution of the KDDTrain dataset.

Category	Number	Ratio
Dos	45927	36.46%
Probe	11656	9.52%
R2L	995	0.79%
U2R	52	0.04%
Normal	67347	53.46%
Total	**125973**	**1.00**

to convert the categorical features into a digital format, enhancing the model's ability to accurately interpret these features. For example, the three categorical protocol features—ICMP, TCP, and UDP—are transformed into numerical representations as (0, 0, 1), (0, 1, 0), and (1, 0, 0) respectively, as detailed in Table 2. This encoding process ultimately expands the original set of 42 features into 122-dimensional vectors.

2) NORMALIZATION

The range of values across the 122 features in the NSL-KDD dataset varies significantly. For instance, the feature Src_Bytes ranges from [0, 1379963888], while srv_serror_rate is confined within [0, 1]. This variation can lead to disparities in how the features influence the model, necessitating normalization to ensure uniformity in feature scaling.

TABLE 2. One-hot encoding.

Attribute	Encoding	Bit
3 protocol types	(0.1.0)	3
70 types of network services	(0……1……0)	70
11 types of link states	(1…0…0)	11

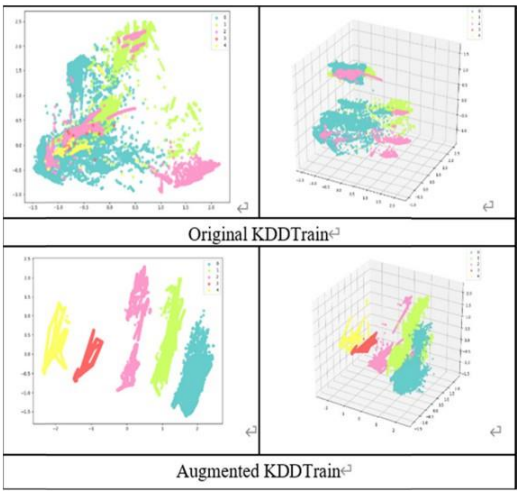

FIGURE 2. Dataset distribution.

C. REDESIGN OF THE CONVOLUTIONAL NEURAL NETWORK

The enhanced CNN model is depicted in Figure 3. Initially, the original k × k matrix undergoes convolution by Conv1, producing L feature maps, with the visual outcomes displayed in Figure 4. These feature maps reveal notable similarities, yet they are not identical, suggesting a degree of interchannel information redundancy. Traditional CNN-based attack recognition models often overlook this redundancy, simply processing all feature maps in subsequent convolution layers without further analysis. This redundancy tends to diminish both the detection capabilities and efficiency of the network. The challenge lies in the fact that it is difficult to definitively identify whether similar features might contain critical differing details, and therefore, they cannot be straightforwardly discarded.

To more effectively extract multiscale features and reduce interchannel redundancy, we have developed an improved CNN incorporating the SPConv module. As illustrated in Figure 3, the SPC-CNN architecture comprises seven layers: an input layer, one convolution layer, two SPConv modules, a fully connected layer, a softmax layer, and an output layer. The SPConv module is segmented into four key components: a channel splitting block, two convolution blocks, and a feature fusion block. The operational flow of SPC-CNN is outlined as follows:

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

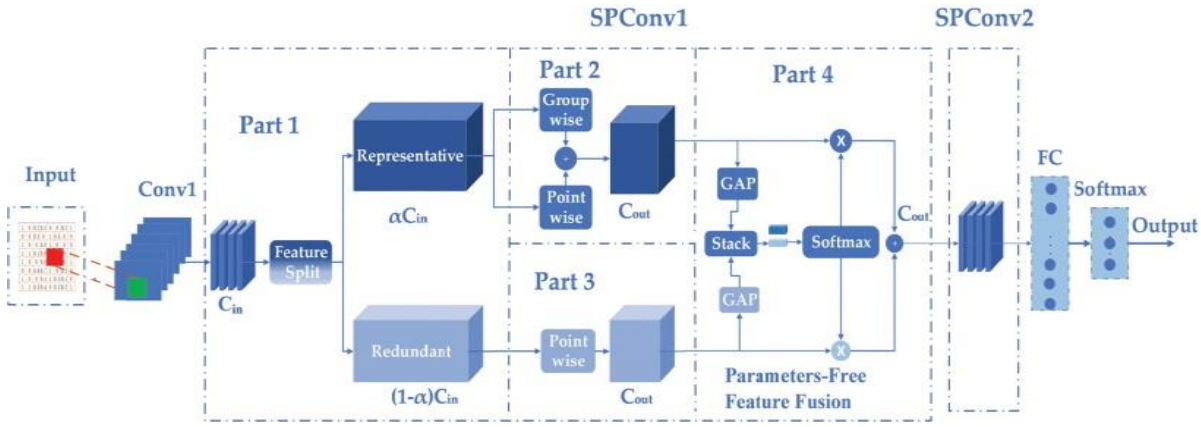

FIGURE 3. The improved convolutional neural network.

1) In Part 1, L feature maps produced by Conv1 are segmented into representative and redundant parts by the channel splitting module.
2) In Parts 2 and 3, the SPConv module processes the representative and redundant parts with differing levels of feature extraction, respectively.
3) Part 4 employs a soft attention module to amalgamate features from various channels, enhancing the rational utilization of features across different channels.

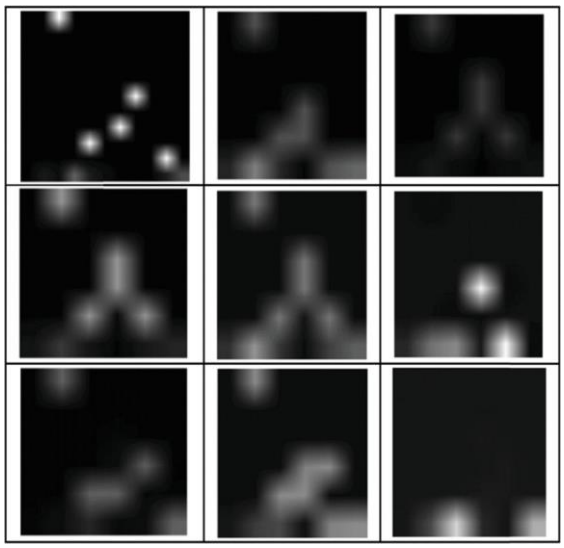

FIGURE 4. Visualization of Conv1 output feature maps.

4) The model's convergence is facilitated by calculating the loss from the softmax layer output and optimizing network parameters through error backpropagation.

D. MODEL TRAINING

The training samples are converted into a standard image format, which serves as the specific input for the SPC-CNN. Conv1 activates using the ReLU function and produces L feature maps that are then forwarded to the SPC module.

1) FEATURE SPLITTING

Consider X ∈ R^(L*h*w) and Y ∈ R^(M*h*w) as the input and output tensors of the SPConv module, respectively. The SPConv module splits the L input channels into two groups based on the ratio α: (1−α), targeting the representative and redundant parts. The 3×3 convolution kernel processes the representative part to extract essential information, while a more cost-effective 1×1 convolution kernel

$$\begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_M \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1,\alpha L} \\ W_{21} & W_{22} & \cdots & W_{2,\alpha L} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{M,\alpha L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{,\alpha L} \end{bmatrix} + \begin{bmatrix} w_{1,\alpha L+1} & w_{1,\alpha L+2} & \cdots & w_{1,L} \\ w_{2,\alpha L+1} & w_{2,\alpha L+2} & \cdots & w_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M,\alpha L+1} & w_{M,\alpha L+2} & \cdots & w_{M,L} \end{bmatrix} \begin{bmatrix} x_{\alpha L+1} \\ x_{\alpha L+2} \\ \vdots \\ x_L \end{bmatrix}$$

processes the redundant part, optimizing the extraction process. The operation within the SPConv module is given by:

where wij, j ∈ [αL + 1, L] denotes the weight parameter of economical 1 × 1 pointwise convolution kernels on (1−α)L redundant channels. Wij, j ∈ [1, αL] denotes the weight parameter of αL representative channels.

2) ADDITIONAL REDUCTION IN THE REPRESENTATIVE PART

Although the input channels are already separated into representative and redundant sections, there might still be excess redundancy within the representative channels. To address this, groupwise convolution (GWC) is employed on the representative channels to minimize this redundancy further. GWC operates similarly to standard convolution but uses sparse block diagonal convolution kernels, each targeting a specific channel segment. This approach, however, can lead to some loss of information since the blocks operate independently. To mitigate this

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

information loss, pointwise convolution (PWC) is implemented in parallel during the same stage. The outputs from both GWC and PWC are then combined directly, ensuring that the processed features from the representative channels are comprehensive and retain necessary details without redundancy.

$$\begin{bmatrix} W_{11}^{P} & 0 & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & W_{GG}^{P} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_G \end{bmatrix} + \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1,\alpha L} \\ w_{21} & w_{22} & \cdots & w_{2,\alpha L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{M,\alpha L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{,\alpha L} \end{bmatrix}$$

Therefore, the representative part of formula can be expressed as formula. where G represents the number of groups in GWC and each group contains g channels. WP vv represents the weight parameter of the GWC kernel in group v.

3) FEATURE FUSION SPC-CNN provides a nonparametric feature fusion module to better fuse features from different channels. As shown in Part 4 of Figure 3, GAP (global average pooling) is used to generate channelwise statistics (CS), CS1,CS3 ∈ R C, and then CS is output by compressing the spatial dimension. The c-th parameter of CS is as follows:

The channel statistics CS1,CS3 are stacked together and processed by softmax to output the important vector β, γ ∈R c . Its c-th parameter is as follows:

$$\gamma_c = \frac{e^{S_{1c}}}{e^{S_{3c}} + e^{S_{1c}}}, \quad \beta_c + \gamma_c = 1$$

The redundant part and the representative part are organically fused according to the important parameters β and γ , and the comprehensive feature O is obtained by cross-channel fusion.

$$O = \gamma T1 + \beta T3$$

In summary, the output of the SPConv1 module is shown in formula.

$$CS_{kc} = F_{gap}(T_{kc}) = \frac{1}{H * W} \sum_{i=1}^{H} \sum_{j-1}^{W} T_{kc}(i,j), \quad k \in [1,3]$$

$$O = W'X \approx \beta \begin{bmatrix} W_{11}^{P} & 0 & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & W_{GG}^{P} \end{bmatrix} \begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_G \end{bmatrix} + \beta \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1,\alpha L} \\ w_{21} & w_{22} & \cdots & w_{2,\alpha L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{M,\alpha L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{,\alpha L} \end{bmatrix} + \gamma \begin{bmatrix} w_{1,\alpha L+1} & w_{1,\alpha L+2} & \cdots & w_{1,L} \\ w_{2,\alpha L+1} & w_{2,\alpha L+2} & \cdots & w_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M,\alpha L+1} & w_{M,\alpha L+2} & \cdots & w_{M,L} \end{bmatrix} \begin{bmatrix} x_{\alpha L+1} \\ x_{\alpha L+2} \\ \vdots \\ x_L \end{bmatrix}$$

IV. SIMULATION AND ANALYSIS

A. ENVIRONMENT AND PARAMETER SETTING

The specific configuration details are provided in Table 3. The learning rate for the network is established at 0.1, and a dropout rate of 0.5 has been determined to optimize detection performance based on various tests. Furthermore, the number of epochs and batch size have been set at 200 and 50, respectively. Table 4 lists the hyperparameters for the SPC-CNN model.

TABLE 3. Configuration information.

Item	Configuration
OS	Ubuntu16.04 (X64)
GPU	4* GTX 1080Ti
Python	3.6

TABLE 4. Parameter setting of SPC-CNN.

Layer	Attribute	Size	Strides	Active Function
L_1	$Conv_1$	2*2*8	1	Relu
L_{21}	GWC_1	3*3*16	1	Relu
L_{22}	PWC_1	1*1*16	1	Relu
L_{23}	PWC_2	1*1*16	1	Relu
L_{31}	GWC_2	3*3*32	1	Relu
L_{32}	PWC_3	1*1*32	1	Relu
L_{33}	PWC_4	1*1*32	1	Relu
-	GAP	2*2	1	Relu
L_4	FC	-	-	Dropout
L_5				Softmax

B. EVALUATION CRITERIA

We select three criteria of ACC, FAR and DR to evaluate the performance of IDS. The related

parameters: TP (true positive) and TN (true negative) represent the number of attacks and normal samples correctly classified, respectively, while FP (false positive) and FN (false negative) represent the number of misclassifications. A summary of the metrics is as follows.

$$ACC = \frac{TN + TP}{TN + FN + TP + FP}$$

$$DR = \frac{TP}{FN + TP}$$

$$FAR = \frac{FP}{TN + FP}$$

ACC represents the proportion of samples accurately identified by the IDS relative to the total number of samples. DR denotes the ratio of correctly identified attack samples to the overall number of abnormal samples, reflecting the system's ability to recognize attacks. FAR indicates the error rate of the IDS in distinguishing between normal and abnormal samples. Therefore, an effective intrusion detection system aims to achieve higher ACC and DR while maintaining a lower FAR.

C. SIMULATION

The KDDTest+ and KDDTest-21 datasets are utilized to evaluate the effectiveness of the SPC-CNN model discussed in this paper. The evaluation metrics, ACC, DR, and FAR, are derived from the output of a five-dimensional confusion matrix. The simulation outcomes are detailed in Table 6. According to these results, when compared to a traditional CNN model referenced in [23], the SPC-CNN shows improvements on the KDDTest+ dataset with increases of 4.35% in ACC and 5.95% in DR, and a reduction in FAR by 5.49%.

TABLE 5. Simulation of SPC-CNN and traditional CNN.

Dataset	Model	ACC(%)	DR(%)	FAR(%)
KDDTest+	CNN	79.48	68.66	27.90
	SPC-CNN	**83.83**	**74.61**	**22.41**
KDDTest-21	CNN	60.71	58.47	71.88
	SPC-CNN	**69.42**	**66.44**	**60.17**

Actual \ Predicted	Normal	Probe	Dos	U2R	R2L
Normal	**1453**	254	145	41	259
Probe	68	**1876**	300	10	148
Dos	552	111	**3085**	32	562
U2R	95	18	0	**34**	53
R2L	506	14	9	77	**2148**

TABLE 6. Simulation of AS-CNN.

Moreover, the results from Table 5 indicate that the SPC-CNN also significantly enhances detection performance on the KDDTest-21 dataset compared to the traditional CNN. The SPC-CNN effectively addresses inter-channel information redundancy and enhances feature diversity, significantly boosting the model's recognition performance and generalization capability.

Further illustrating the combined effect of ADASYN data augmentation and the SPC-CNN model on IDS, the augmented KDDTrain dataset was used to train the SPC-CNN, after which an optimal model was saved. This model was then tested using the KDDTest+ and KDDTest-21 datasets. The outcomes, shown in Table 6, reveal that the hybrid AS-CNN model outperforms the standalone SPC-CNN model, as indicated in Table 5. Specifically, the AS-CNN model increased DR by 5.39% and 7.21% on the KDDTest+ and KDDTest-21 datasets, respectively, while FAR decreased by 10.09% and 14.51%. These improvements demonstrate that the ADASYN algorithm helps balance sample distribution and refine category discretization, significantly enhancing the IDS's classification performance.

Tables 7 and 8 display the confusion matrices for tests conducted with the AS-CNN model using the standard KDDTest+ and KDDTest-21 datasets, respectively. In these tables, the numbers along the main diagonal indicate the count of correctly identified samples. Based on

Dataset/Criteria	ACC	DR	FAR
KDDTest+	84.08	80.00	12.32
KDDTest-21	72.54	73.65	45.66

TABLE 7. KDDTest+ classification confusion matrix of AS-CNN

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

Actual \ Predicted	Normal	Probe	Dos	U2R	R2L
Normal	**8690**	314	209	52	446
Probe	68	**1891**	304	10	148
Dos	552	112	**6192**	32	570
U2R	95	18	0	**34**	53
R2L	506	14	9	77	**2148**

calculated and are presented in Figure 5 and Table 9.

Tables 7 and 8 show the results from testing the

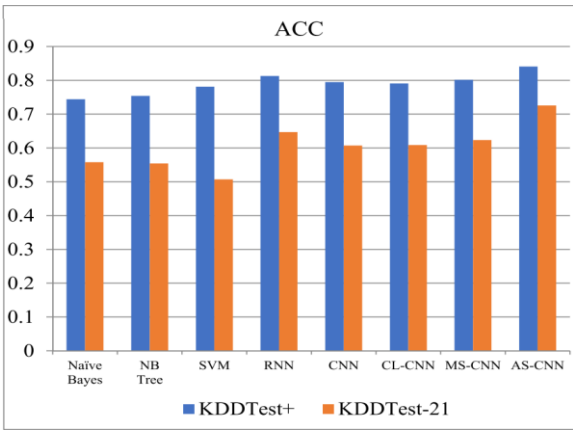

FIGURE 5. ACC of multiple models.

AS-CNN model with the standard KDDTest+ and KDDTest-21 datasets. In these tables, figures on the main diagonal represent the number of correctly identified samples. Subsequently, the evaluation metrics ACC, DR, and FAR for the AS-CNN model are calculated based on the confusion matrices, as shown in Figure 5 and Table 9.

Many classic intrusion detection algorithms have been implemented in various tasks, including Bayesian, NBTree, SVM, CL-CNN, RNN, CNN, and MS-CNN. Figure 5 depicts the ACC for these models using the same KDDTest+ and KDDTest-21 datasets. It is evident that deep learning (DL) methods generally achieve higher ACC than traditional machine learning (ML) methods. Although the detection performance of the traditional 1D CL-CNN model and 2D CNN model are comparable, the CL-CNN, which utilizes feature coding to increase feature dimensions and employs 1D convolution, is less efficient than the 2D CNN model. Both models share the simplicity of their CNN structures.

TABLE 9. DR and FAR for multiple models.

Criteria/ \ Model	RNN	CNN	CL-CNN	AS-CNN
DR(%)	69.73	68.66	68.56	**80.00**
FAR(%)	26.89	27.90	25.10	**12.32**

The MS-CNN model's structure differs by obtaining multistage features of network traffic through altered connections in the convolution layers. This diversity in features enhances the CNN's ability to characterize intrusion samples, thereby improving the classifier's performance. The AS-CNN model proposed in this article, different from the MS-CNN model, not only captures diversified features but also effectively reduces feature redundancy between channels through channel splitting and applies a soft attention mechanism for rational use of multi-level features. Consequently, the overall performance of AS-CNN is significantly enhanced compared to traditional CNN and MS-CNN models. For instance, on the KDDTest+ dataset, the ACC of AS-CNN increased by 2.79% and 3.95% compared to the classic RNN and MS-CNN models, respectively, and by 7.87% and 10.22% on the KDDTest-21 dataset.

Moreover, Table 9 highlights DR and FAR as key indicators for evaluating the comprehensive capabilities of various intrusion detection models, showing that AS-CNN outperforms in both metrics. Specifically, the DR of AS-CNN has increased by 11.34%, and FAR has decreased by 15.58% compared to the traditional CNN model. Overall, the proposed AS-CNN model demonstrates significantly improved performance and generalization capabilities compared to other DL models.

The detection rates of specific attack types by various models are also crucial for evaluating IDS performance. Figure 6 illustrates the detection rates for different attack categories across multiple models. The traditional CNN model employs a weight-loss method to address unbalanced sample distributions, resulting in notable DR improvements for minority samples. The DR for the Probe and U2R attack types is better in the CL-CNN model than in the traditional CNN model due to the use of character-level encoding, which

discretizes features. Additionally, the cross-layer aggregated structure in the MS-CNN model enhances feature diversity, improving its ability to recognize Probe and U2R attacks compared to CNN and CL-CNN. Unlike the traditional models, the AS-CNN utilizes the ADASYN data augmentation algorithm to mitigate the impact of unbalanced sample distribution on model training, further enhancing its performance.

The AS-CNN model adeptly addresses the challenges of feature redundancy and the fusion of multi-scale features across channels, issues that were not effectively handled by the MS-CNN model. Incorporating elements from both ADASYN and SPC-CNN, the AS-CNN model is designed specifically for intrusion detection tasks. As detailed in this paper, the AS-CNN

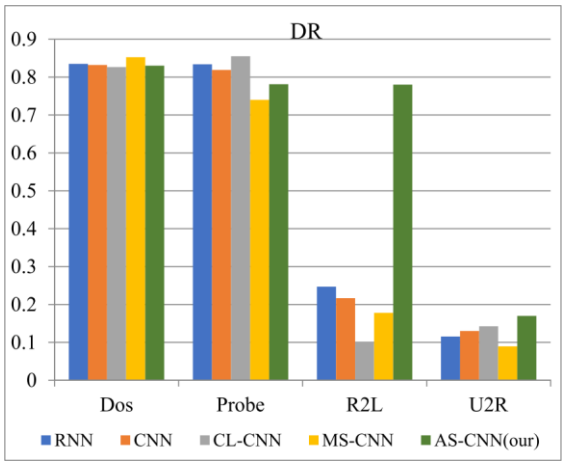

FIGURE 6. DR of multiple models to different attacks.

demonstrates strong detection rates (DR) for most attack categories, except for probes. It notably excels in identifying R2L and U2R attack samples, which have historically presented significant challenges in the field. Figure 6 highlights the substantial improvements in the DRs for R2L and U2R attacks achieved by the hybrid AS-CNN model.

The advantages of our deep learning-based system are manifold. Firstly, it operates in real-time, capable of processing videos at a speed of 22 frames per second, facilitating the filtering of live-captured content. Secondly, it offers practical applications for video-sharing platforms, enabling the automatic removal or obscuring of unsafe content segments. Additionally, it holds promise for the development of parental control solutions, allowing for the automatic filtration of child-inappropriate content on the internet.

Importantly, our methodology for detecting inappropriate content remains robust against attempts to manipulate video metadata by malicious uploaders. Moving forward, we aim to enhance our model's performance by integrating temporal stream analysis with optical flow frames and further expanding the classification labels to target various types of inappropriate children's content found in YouTube videos. This ongoing research will continue to advance our understanding and capabilities in combating inappropriate content online.

V. CONCLUSION

We aim to tackle the challenges of unbalanced data distribution and interchannel information redundancy, which are often overlooked by current CNN-based intrusion detection systems. To address this, we employ the ADASYN technique to equalize the sample distribution, helping to reduce the model's oversensitivity to larger samples while enhancing its responsiveness to smaller ones. Additionally, our newly designed split-based SPC-CNN model offers three key advantages: 1) it enables the extraction of multiscale features via varied convolution operations; 2) it effectively reduces interchannel redundancy through its complementary processing approach; and 3) it incorporates a soft attention mechanism, allowing the SPC-CNN to utilize multiscale features more effectively, thereby enhancing the model's expressive capabilities. We use a hybrid AS-CNN model, which combines ADASYN and SPC-CNN features, for intrusion detection tasks. The simulation results show that this hybrid model significantly improves performance across multiple classification metrics. However, there remains ample scope to further enhance the recognition accuracy for smaller samples and improve operational efficiency. Future efforts will focus on boosting the identification capabilities of IDS by integrating a simplified residual network.

REFERENCES

[1] ‘‘A survey on wireless security: Technical challenges, recent advances, and future trends,’’ Proc. IEEE, vol. 104, no. 9, pp. 1727–1765, Sep. 2016, doi: 10.1109/JPROC. 2016.2558521.

[2] ‘‘An overview of wireless network security,’’ in Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud), New York, NY, USA, Jun. 2017, pp. 306–309, doi: 10.1109/CSCloud.2017.45.

[3] ''Intrusion detection alert management for high-speed networks: Current researches and applications,'' Secur. Commun. Netw., vol. 8, no. 18, pp. 4362–4372, Dec. 2015.
[4] ''A survey of data mining and machine learning methods for cyber security intrusion detection,'' IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016, doi: 10.1109/COMST.2015.2494502.
[5] ''Character-level intrusion detection based on convolutional neural networks,'' in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8488987.
[6] ''Firefly algorithm based feature selection for network intrusion detection,'' Comput. Secur., vol. 81, pp. 148–155, Mar. 2019.
[7] ''Research on SVM network intrusion detection based on PCA,'' Inf. Netw. Secur., vol. 2, pp. 15–18, Feb. 2015.
[8] ''An improved LDA-based ELM classification for intrusion detection algorithm in IoT application,'' Sensors, vol. 20, no. 6, p. 1706, Mar. 2020.
[9] ''The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems,'' Sensors, vol. 20, no. 9, p. 2559, Apr. 2020.
[10] ''An effective intrusion detection framework based on SVM with feature augmentation,'' Knowl.-Based Syst., vol. 136, pp. 130–139, Nov. 2017.
[11] ''Bayesian model averaging of Bayesian network classifiers for intrusion detection,'' in Proc. IEEE 38th Int. Comput. Softw. Appl. Conf. Workshops, Västerås, Sweden, Jul. 2014, pp. 128–133, doi: 10.1109/COMPSACW.2014.25.
[12] ''A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems,'' Expert Syst. Appl., vol. 42, no. 5, pp. 2670–2679, Apr. 2015.
[13] ''A deep learning based artificial neural network approach for intrusion detection,'' in Proc. Int. Conf. Math. Comput. (ICMC), Haldia, India, Jan. 2017, pp. 44–53.
[14] ''An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization,'' Neurocomputing, vol. 199, pp. 90–102, Jul. 2016.
[15] ''Effective discriminant function for intrusion detection using SVM,'' in Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Jaipur, India, Sep. 2016, pp. 1148–1153, doi: 10.1109/ICACCI.2016.7732199.
[16] ''Mixed intrusion detection algorithm based on k-means and decision tree,'' Comput. Modernization, pp. 12–16, Dec. 2017.
[17] ''Hybrid machine learning technique for intrusion detection system,'' in Proc. 5th Int. Conf. Comput. Informat. (ICOCI), Istanbul, Turkey, Aug. 2015, pp. 2289–3784.
[18] ''Intrusion detection of UAVs based on the deep belief network optimized by PSO,'' Sensors, vol. 19, no. 24, p. 5529, Dec. 2019.
[19] ''Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark,'' IEEE Access, vol. 6, pp. 59657 – 59671 , 2018 , doi: 10 . 1109 / ACCESS.2018.2875045.
[20] ''A deep learning approach for intrusion detection using recurrent neural networks,'' IEEE Access, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.
[21] ''Malware traffic classification using convolutional neural network for representation learning,'' in Proc. Int. Conf. Inf. Netw. (ICOIN), Da Nang, Vietnam, 2017, pp. 712–717, doi: 10.1109/ICOIN.2017.7899588.
[22] ''Network intrusion detection model based on convolutional neural network,'' J. Inf. Secur. Res., pp. 990–994, Nov. 2017.
[23] ''A novel intrusion detection model for a massive network using convolutional neural networks,'' IEEE Access, vol. 6, pp. 50850–50859, 2018, doi: 10.1109/ACCESS.2018.2868993.
[24] ''The research and amelioration of pattern-matching algorithm in intrusion detection system,'' in Proc. IEEE 14th Int. Conf. High Perform. Comput. Commun. IEEE 9th Int. Conf. Embedded Softw. Syst., Liverpool, U.K., Jun. 2012, pp. 1712–1715, doi: 10.1109/HPCC.2012.256.
[25] ''Analysis of pattern matching algorithms in network intrusion detection systems,'' in Proc. Int. Conf. Adv. Comput., 2016, pp. 1–5.
[26] ''Intrusion detection model using fusion of chi-square feature selection and multi class SVM,'' J. King Saud Univ.-Comput. Inf. Sci., vol. 29, no. 4, pp. 462–472, Oct. 2017.
[27] ''Decision tree based intrusion detection system for NSL-KDD dataset,'' in Proc. Int. Conf. Inf. Commun. Technol. Intell. Syst., Ahmedabad, India, 2017, pp. 207–218, doi: 10.1007/978-3-319-63645-0_23.
[28] ''Intrusion detection using dynamic feature

selection and fuzzy temporal decision tree classification for wireless sensor networks,'' IET Commun., vol. 14, no. 5, pp. 888–895, Mar. 2020, doi: 10.1049/iet-com.2019.0172.

[29] ''Intrusion detection system using Bayesian network and feature subset selection,'' in Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC), Coimbatore, India, Dec. 2017, pp. 1–5, doi: 10.1109/ICCIC.2017.8524381.

[30] ''An effective intrusion detection classifier using long short-term memory with gradient descent optimization,'' in Proc. Int. Conf. Platform Technol. Service (PlatCon), Busan, South Korea, Feb. 2017, pp. 1–6, doi: 10.1109/PlatCon.2017.7883684.

[31] ''BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset,'' IEEE Access, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[32] ''An optimization method for intrusion detection classification model based on deep belief network,'' IEEE Access, vol. 7, pp. 87593–87605, 2019, doi: 10.1109/ACCESS.2019.2925828.

[33] ''Malicious network traffic detection based on deep neural networks and association analysis,'' Sensors, vol. 20, no. 5, p. 1452, Mar. 2020.

[34] ''Applying convolutional neural network for network intrusion detection,'' in Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Udupi, India, Sep. 2017, pp. 1222–1228, doi: 10.1109/ICACCI.2017.8126009.

[35] ''Intrusion detection system for NSL-KDD dataset using convolutional neural networks,'' in Proc. 2nd Int. Conf. Comput. Sci. Artif. Intell. (CSAI), 2018, pp. 81–85.

[36] ''Wireless network intrusion detection based on improved convolutional neural network,'' IEEE Access, vol. 7, pp. 64366–64374, 2019, doi: 10.1109/ACCESS.2019.2917299.

[37] ''Split to be slim: An overlooked redundancy in vanilla convolution,'' 2020, arXiv:2006.12085. [Online]. Available: http://arxiv.org/abs/2006.12085

[38] ''A study on NSL-KDD dataset for intrusion detection system based on classification algorithms,'' Int. J. Adv. Res. Comput. Commun. Eng., vol. 4, no. 6, pp. 446–452, 2015."