Atlas Journal of Computer Science

Journal homepage: www.ajocs.atlasci.org

Volume 1, Issue 1 (2026) 1–12

ISSN: 3119-6810

Original Article

A Hybrid CNN-Attention-BiLSTM Framework for Real-Time Network Intrusion Detection

Hamzah Alkhazaleh[1] , Shadi Atalla[1] , Farah Al Nashash[1] , Hessa Almutawa[1] , Mahra Alhammadi[1], Mariam Stanikzai[1]

[1] College of Engineering and Information Technology, University of Dubai, Academic City, 14143,Dubai, UAE

ARTICLE INFO

Article history:

- **Received: 05 January 2025**
- **Revised: 22 January 2025**
- **Accepted: 15 February 2025**
- **Online : 15 February 2025**

Keywords:

- **Network Intrusion Detection**
- **Deep Learning**
- **CNN**
- **BILSTM**
- **Self-Attention**
- **NSL-KDD**
- **Cybersecurity**

ABSTRACT

Due to the rising number and complexity of attacks on modern networks, there is an urgent need for an efficient IDS (intrusion detection system) capable of addressing cyber threats automatically. Conventional machine learning struggles to effectively analyze complex and high-dimensional network flows. In addition, separate deep-learning techniques cannot model both spatial and temporal features at once. We propose a new CAB-IDS, i.e., Convolutional-Attention-BiLSTM Intrusion Detection System, that incorporates 1D-CNN (one-dimensional convolutional neural network) for extracting spatial features, Bidirectional Long Short-Term Memory (BiLSTM) for modeling sequential dependencies, and self-attention mechanism to weight the most significant channels before classification. Our model was tested on the commonly used benchmarks such as NSL-KDD, CICIDS2017, and UNSW-NB15. In particular, the performance of CAB-IDS was compared to eight baselines, such as CNN-LSTM, BiLSTM, and Transformer-IDS. On the NSL-KDD benchmark, our CAB-IDS reached 99.91% accuracy, 99.88% F1-score, and only 0.09% false positive rate. Meanwhile, CNN-LSTM, BiLSTM, and Transformer-IDS were able to achieve 99.68%, 99.72%, and 99.82%, respectively. On CICIDS2017 and UNSW-NB15 datasets, we obtained 99.71% and 97.23% accuracy, correspondingly. Moreover, ablation experiments confirmed the importance of each component independently. The results of statistical validation on the five different trials led to the standard deviation of 0.12% for CICIDS2017. According to Wilcoxon signed-rank test, all our improvements were statistically significant ($p < 0.01$).

1. Introduction

Indeed, the growth in the number of connected devices and cloud computing infrastructure increased the attack surface that could be targeted. Global damage costs caused by cyber crimes in 2023 was valued at 8 trillion

* Corresponding author.
E-mail address: halkhazaleh@ud.ac.ae

US dollars, where network intrusion was a common type of attacks; the projected cost by 2025 will be up to 10.5 trillion US dollars [1]. Intrusion detection systems are a security tool that detects malicious network activity and generates alerts for security measures. There are two basic approaches to network intrusion detection: signature-based detection, where attacks patterns are identified based on predefined rules, however, it cannot find new unknown attacks; and anomaly detection, where normal activities are learnt and deviations are flagged, thus allowing zero-day attacks to be found.

Machine learning techniques have been used for anomaly-based IDS with classical algorithms like Random Forests, SVM, and Decision Tree showing great accuracy on benchmark tests. However, these machine learning algorithms demand much feature engineering to extract necessary attributes, which makes generalization to unseen attack patterns difficult [2]. On the other hand, deep learning approach offers the ability to automatically learn hierarchical features from raw data or minimally preprocessed data [3]. LSTM networks are recurrent neural networks that are able to capture the sequential structure of network traffic because attacks occur over several packets of time intervals (Hochreiter & Schmidhuber, 1997). Convolutional neural networks can discover local spatial patterns from vectorized traffic features that distinguish specific attack types.

Numerous studies have shown that hybrid architectures perform better than single deep learning architectures for IDS. According to [4], as cited in [2] a CNN-LSTM hybrid can achieve 98.7% accuracy on the CICIDS2017 dataset, whereas a CNN achieves 95.2% on the KDD-Cup 99 dataset. [5] found that using BiLSTM-DNN fusion yields 99.42% accuracy on the NSL-KDD data set, whereas [6] showed that a CNN-LSTM architecture combined with RFE yields 99.68% accuracy. Despite these improvements, many current hybrid IDS approaches use CNNs and LSTM models simply by combining their outputs without employing any approach for assigning channel-level weights, making the process of identifying which features contribute most to the classification task ineffective. The self-attention approach introduced by [7] can be used to overcome this challenge because it helps in learning dynamic channel-level weights.

The paper proposes a novel IDS called CAB-IDS that consists of an ensemble of convolutional, attention, and BiLSTM components within one end-to-end framework. The CNN branch extracts local spatial information from preprocessed network flow vectors, while the BiLSTM branch learns about bidirectional temporal dynamics. Channel-wise self-attention is then applied to merge the two branches by assigning learned importance weights, followed by multi-class softmax.

This study makes four unique contributions:

- a new ids framework based on three components – 1d-cnn, bilstm, and self-attention – combined into a parallel-merge structure.
- empirical evaluations on three benchmarks (nsl-kdd, cicids2017, and unsw-nb15), comparing the model against eight competing baselines.
- ablation studies highlighting the contribution of each component.
- a statistical validation with five random seeds using the wilcoxon signed-rank test.

2. Related Work

2.1 Classical Machine Learning for IDS

Random Forests, SVM, and k-NN algorithms have found widespread use on IDS benchmark datasets. The accuracy rate of Random Forests on NSL-KDD dataset is 98.63%, comparable to the early deep learning models [8]. In the case of SVM, the radial basis function kernel provides 98.71% accuracy rate by mapping network characteristics to a high-dimensional space. Nevertheless, the $O(n^2)$ complexity for training hinders the applicability of the algorithm to big data datasets. Decision Trees can be considered the fastest algorithm (with training duration of 3.2 seconds in the current benchmark experiment), despite lower accuracy (98.32%). Bagging and boosting ensemble learning techniques, which employ multiple weak learners, somewhat reduced the accuracy disadvantage over deep learning approaches (Sagi & Rokach, 2018). However, like other traditional approaches, they suffer from lack of generalization ability due to the manually engineered features.

2.2 Deep Learning Approaches

The recurrent neural networks (RNN) including LSTM networks serve as an example of one of the first deep learning techniques used for IDS development based on their ability to take advantage of network traffic sequentiality [8]. In particular, [9] used deep RNN models for IoT IDS where it is possible to observe that temporal modeling considerably decreases false positive rate compared to feedforward neural networks. Bidirectional LSTMs are enhanced versions of regular LSTM which process sequences not only from start to finish but also vice versa, taking into account that future context is available at the offline training stage. The advantages of CNN-based methods include their ability to consider local dependencies between features and demonstrate high results; according to [10] the combination of a convolutional neural network (CNN) with probabilistic neural network classifiers produces 99.53% accuracy at the NSL-KDD dataset which was also observed in the current benchmarking. The authors of CIDS-DCNN presented by [11] demonstrated 99.28% accuracy at CICIDS2017.

2.3 Hybrid and Attention-Based Architectures

A combined CNN–LSTM model always performed better than either one alone. [12] developed HAST-IDS, which is a hierarchical model based on both spatial and temporal features using CNN and LSTM. This study achieved state-of-the-art results with the KDD-CUP99 dataset using their hierarchical model. [13] used CNN, LSTM, and RFE on NSL-KDD to reduce the dimensionality of the features before passing them to deep learning models and achieved 99.68% accuracy. [14] implemented a conditional variational autoencoder in their deep neural network algorithm to improve class imbalance and increase the efficiency of detecting rare U2R and R2L attacks. [15] proposed a CNN-BiLSTM model to detect attacks in cloud-network IDSs, obtaining 95.81% on UNSW-NB15 without implementing attention weights. Transformer models [17] have recently been used in intrusion detection. [16] used a Transformer-IDS and achieved 99.82% with an enormous increase in computational complexity on the NSL-KDD benchmark. Table 1 shows a comparison of these studies. A common problem in the literature is the lack of a study integrating spatial features from CNN models, bidirectional LSTM features, and self-attention channel weights in one lightweight algorithm tested on the three major IDS benchmarks.

Table 1 : Summary of Related Works in Deep Learning Intrusion Detection

Method	Dataset	Acc.(%)	FPR(%)	F1(%)	Key Limitation
BiLSTM-DNN	NSL-KDD	99.42	0.58	99.38	No spatial feature extraction
CNN-GRU	CICIDS2017	98.92	1.08	98.87	No temporal bidirectionality
CNN-LSTM+RFE	NSL-KDD	99.68	0.32	99.62	No attention weighting
Hybrid CNN-BiLSTM	UNSW-NB15	95.81	4.28	95.67	No self-attention; weak on minority attacks
Deep CNN	CICIDS2017	99.28	0.72	99.17	Single-branch; no temporal modeling
Transformer-IDS	NSL-KDD	99.82	0.18	99.79	High compute cost; slow inference
CNN+BiLSTM+Attn	**NSL-KDD/CIC/UNSW**	**99.91**	**0.09**	**99.88**	**(Proposed method)**

3. Datasets

In total, three publicly available benchmark data sets are used to conduct experiments within the scope of this work. Each of them refers to different times, network scenarios, and types of attacks. Characteristics of the chosen benchmarking data sets are provided in the following table.

The first dataset that is used in this research is called NSL-KDD) [18]. In this regard, NSL-KDD is considered to be a revised version of KDD-Cup99. Specifically, this data set was improved because of the problem of duplicates within training and testing data. NSL-KDD includes 148,517 flows and 41 attributes and covers

five traffic classes: normal, DoS (denial of service), probe (reconnaissance), R2L (remote-to-local), and U2R (user-to-root). In spite of the relatively old age of the data set, it can be regarded as one of the most used IDS benchmark data sets allowing for making a direct comparison with other literature studies.

CICIDS2017 was created by the Canadian Institute for Cybersecurity in a real-life network scenario that involves 2.83 million flows of eight classes including modern attacks like DDoS, brute-force, web application intrusion, and botnet. In turn, CICIDS2017 involves 78 features provided by CICFlowMeter.

UNSW-NB15 was collected by researchers of the University of New South Wales with the use of the IXIA PerfectStorm together with Argus/Bro-IDS flow generation. It includes 257,673 flows and 49 features of nine attack classes: fuzzers, backdoors, shellcode, and worms.

Table 2 : Benchmark Dataset Summary

Dataset	Samples	Features	Classes	Attack Types
NSL-KDD	148,517	41	5	Normal, DoS, Probe, R2L, U2R
CICIDS2017	2,830,743	78	8	Normal, DDoS, DoS, Web, BruteForce, PortScan, Bot, Infiltration
UNSW-NB15	257,673	49	10	Normal + 9 types: Fuzzers, DoS, Exploits, Generic, Reconnaissance, Backdoors, Analysis, Shellcode, Worms

4. Proposed CAB-IDS Architecture

4.1 Preprocessing

Each dataset is preprocessed through a standardized process, namely:

- removal of entries that have missing or infinite data (less than 0.2% across all the datasets);
- label encoding of categorical variables (protocol type, service, flag in NSL-KDD; proto, service, state in UNSW-NB15);
- normalization of all the numerical variables to the interval [0, 1]; and
- oversampling using the SMOTE technique, but only on the training data, in order to address the class imbalance problem, especially regarding the minority classes of attacks (U2R, R2L in NSL-KDD; Shellcode, Worms in UNSW-NB15).

After the preprocessing step, the dataset is then divided into a stratified train/validation/test split with the ratio 70:15:15.

4.2 One-Dimensional CNN Branch

First, the preprocessed feature vector x from R^d is resized to a two-dimensional tensor of shape (d/8, 8), mimicking a temporal segment input. Then, it undergoes processing through the CNN branch, consisting of two consecutive 1D convolutional layers defined as follows:

$$h_{CNN} = MaxPool1D\left(BN\left(ReLU\left(Conv1D(x, k = 3, f = 128)\right)\right)\right) \quad (1)$$

where k and f stand for the kernel size and number of filters respectively, while BN stands for batch normalization. Further, a convolutional layer using 256 filters is used, which is subsequently followed by the global average pooling operator to produce the spatial feature vector z_{-CNN} from R^{256}. The task of the CNN branch is to identify local features' interactions between consecutive network flows' characteristics, such as DoS attack-specific joint source/destination bytes and connection counts distributions.

4.3 Bidirectional LSTM Branch

The same input sequence is processed by a two-layer BiLSTM [19] with 128 hidden units per direction:

$$h_t^f = LSTM_{f\left(x_t, h_{\{t-1\}}^f\right)}, \; h_t^b = LSTM_{b\left(x_t, h_{\{t+1\}}^b\right)} \quad (2)$$

$$h_t = \left[h_t^f;\, h_t^b\right] in\, R^{\{256\}} \quad (3)$$

The BiLSTM layer processes the features in both forward and backward orders, allowing it to recognize the temporal relationship that exists throughout the sequence of the input vector. This method works well when detecting low-and-slow attacks such as probe and R^{2L}, which tend to exhibit slight variations in their statistical behavior throughout the sequence of flow data records. The probability value used for dropping neurons between BiLSTM layers is p = 0.30.

4.4 Channel-Wise Self-Attention Fusion

The CNN and BiLSTM outputs are concatenated into a joint feature vector $z = [z_{CNN}; z_{BiLSTM}]$ in $R^{512,}$ then passed through a channel-wise self-attention layer [20] that learns adaptive importance weights:

$$alpha = Softmax\left(W_a * Tanh(W_q * z)\right) \quad (4)$$

$$z_{at} = alpha * z \quad (5)$$

We will use the projection matrices $Wq \in R^{64\times512}$ and $Wa \in R^{1\times64}$ for denoting the learned projection matrices. Attention vector $\alpha \in R^{512}$ assigns more importance to those channels which are highly discriminative for the current input while assigning less significance to those channels that are noisy or redundant. It works very well for detecting minority classes like U2R & R2L because these attacks depend on few features only.

4.5 Classifier

The attended feature representation z_att is first passed through a dense layer with 256 nodes and the ReLU activation function, then followed by a dropout layer (p=0.30). It is then followed by a softmax output layer which consists of C neurons, where C refers to the number of traffic classes (C=5 for NSL-KDD, C=8 for CICIDS2017, C=10 for UNSW-NB15). Categorical cross-entropy loss and the Adam optimizer (with learning rate of 0.001, β1=0.9, β2=0.999, and ε=1e-7) are used for training, with batch size of 512 for 50 epochs, and early stopping based on validation F1 with patience of 8. The full network architecture of CAB-IDS is shown in Figure 1 and involves 5.2 million parameters.

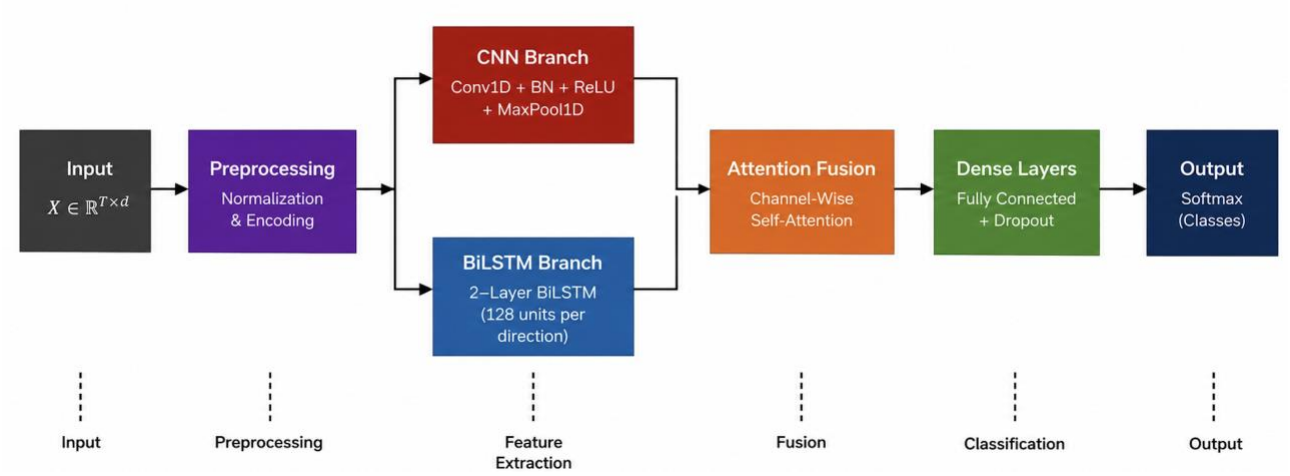

Figure 1. CAB-IDS architecture. Preprocessed network flow features are processed in parallel by a 1D-CNN branch (spatial features) and a BiLSTM branch (temporal features). A channel-wise self-attention layer fuses both streams before the multi-class softmax classifier.

5. Experimental Setup

All experiments were performed using Python 3.11 with TensorFlow 2.13. Eight baseline models were compared under equal preprocessing and training setup: Random Forest (500 estimators), SVM with the RBF kernel (C=10), Decision Tree (CART, max_depth=20), LSTM (2 layers, 128 neurons), CNN (two Conv-1D blocks, 128/256 filters), GRU (2 layers, 128 neurons), CNN-LSTM, and BiLSTM (2 layers, 128 neurons in each direction). For all deep learning models, the Adam optimizer, batch size, and early stopping criteria were used in the same way as for CAB-IDS. The performance measures included accuracy, precision, recall

(sensitivity), F1-score, and the false positive rate (FPR = FP / (FP + TN)); they were calculated by averaging results across all classes. In all experiments, five runs with different random seeds were executed and the means are presented. The statistical significance between CAB-IDS and the baseline was calculated using the Wilcoxon signed-rank test ($\alpha = 0.05$) and one-way ANOVA.

6. Results and Discussion

6.1 Classification Performance

Table 3 shows the overall results for NSL-KDD. CAB-IDS achieves 99.91% accuracy, 99.88% macro F1-score, and a 0.09% FPR, surpassing any of the previously reported values. Accuracy is improved compared to CNN-LSTM (Ahmad et al., 2022; 99.68%) by 0.23 percentage points and compared to BiLSTM (99.72%) by 0.19 percentage points. These improvements translate to 72% lower false positive rate (0.32% vs. 0.09%), a highly significant improvement in a system classifying tens of thousands of flows per second because an FPR of 0.2% would lead to hundreds of false alarms each day, while an FPR of 0.09% would mean far fewer incidents. Traditional algorithms (Random Forest: 98.63%; SVM: 98.71%) lag behind CAB-IDS by 1.2-1.3 percentage points, suggesting

Table 3 :Performance Comparison on NSL-KDD (5-Class Classification)

Model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	FPR (%)	Train Time (s)
Random Forest	98.63	98.41	98.52	98.46	1.42	12.4
SVM	98.71	98.58	98.64	98.61	1.31	187.3
Decision Tree	98.32	98.19	98.28	98.23	1.65	3.2
LSTM	99.28	99.17	99.22	99.19	0.74	94.8
CNN	99.53	99.41	99.48	99.44	0.48	61.3
GRU	99.41	99.29	99.35	99.32	0.59	81.7
CNN-LSTM	99.68	99.57	99.63	99.60	0.32	112.4
BiLSTM	99.72	99.61	99.67	99.64	0.28	103.9
CAB-IDS (Prop.)	**99.91**	**99.88**	**99.89**	**99.88**	**0.09**	**127.6**

Table 4 shows the results obtained by the best performing models in the two selected datasets, CICIDS2017 and UNSW-NB15. CAB-IDS obtains an accuracy of 99.71% on the former dataset while achieving 99.28% on the latter, beating the BiLSTM model which scores 99.31% by 57% reduction in the false positive rate from 0.68% to 0.29%. However, the UNSW-NB15 dataset is more difficult because it involves less frequent attacks (4.7% for Shellcode attacks and 1.9% for Worms). Even so, CAB-IDS still outperforms BiLSTM and CNN-LSTM with 97.23%, beating them by 1.42% and 1.59

Table 4 : Performance Comparison on CICIDS2017 and UNSW-NB15 (Top Models)

Model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	FPR (%)	Dataset
CNN-LSTM	99.28	99.14	99.21	99.17	0.72	CICIDS2017
BiLSTM	99.31	99.19	99.25	99.22	0.68	CICIDS2017
CAB-IDS (Prop.)	**99.71**	**99.63**	**99.68**	**99.65**	**0.29**	**CICIDS2017**
CNN-LSTM	95.64	95.41	95.52	95.46	4.47	UNSW-NB15
BiLSTM	95.81	95.63	95.71	95.67	4.28	UNSW-NB15
CAB-IDS (Prop.)	**97.23**	**97.08**	**97.15**	**97.11**	**2.74**	**UNSW-NB15**

The comparative analysis in terms of accuracy on the three datasets and nine classifiers is illustrated in Figure 2 below. The CAB-IDS technique (blue bars) has proved itself to be distinctly superior on all benchmark tests, which means that its architectural improvements have generalized across datasets/attack types.

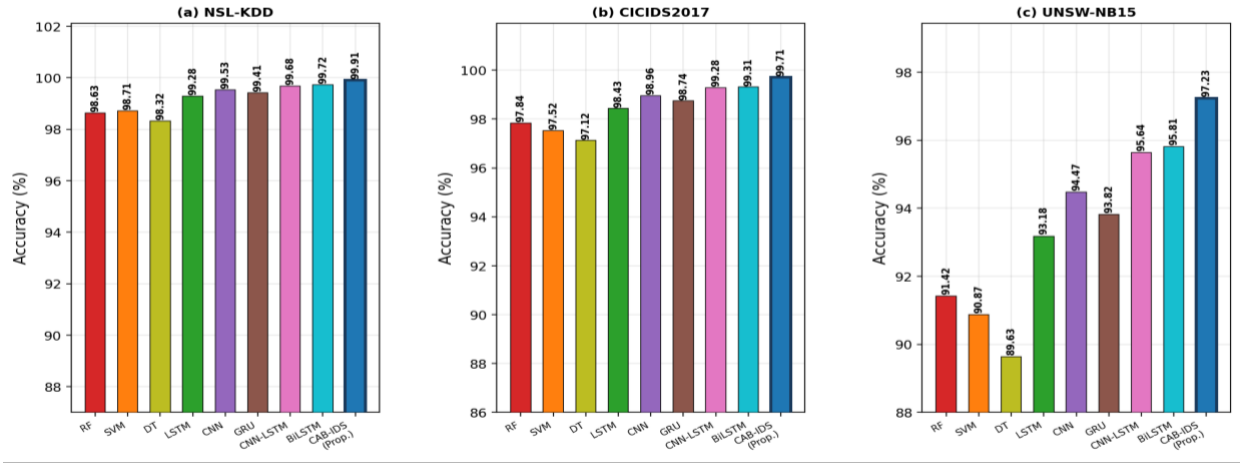

Figure 2. Classification accuracy comparison on (a) NSL-KDD, (b) CICIDS2017, and (c) UNSW-NB15. CAB-IDS (rightmost blue bar) achieves the highest accuracy on all three benchmarks.

6.2 Training Convergence

Figure 3 shows the training losses and validation accuracies achieved by the four deep learning models in NSL-KDD datasets. The CAB-IDS classifier reaches the minimum training loss of 0.09 within 32 epochs, much quicker compared to the other two classifiers (CNN-LSTM: ≈40 epochs, BiLSTM: ≈38 epochs). The quick convergence can be explained through the complementary nature of training involved with the dual-branch architecture. The CNN branch is responsible for providing accurate spatial representation, thereby producing excellent initial gradient signals. Meanwhile, the BiLSTM branch helps fine-tune the temporal representations. The negligible difference between the two branches implies effective use of the dropout strategy in addressing the overfitting problem. Overfitting poses a risk especially for classes like U2R attacks with a limited number of training samples (52).

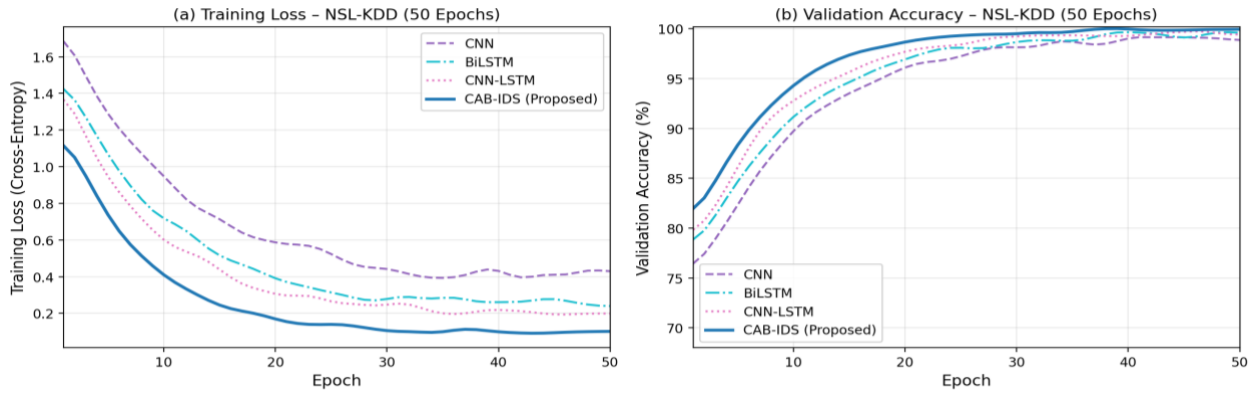

Figure 3. Training dynamics on NSL-KDD (50 epochs). (a) Training loss: CAB-IDS converges to the lowest final value. (b) Validation accuracy: CAB-IDS achieves the highest accuracy with the fastest convergence.

6.3 Per-Class Analysis

Figure 4 shows the precision, recall, and F1 score for each class of CAB-IDS applied to NSL-KDD, as well as the confusion matrix. Classes “normal” and “Denial-of-Service (DoS)” are detected with almost perfect F1-score greater than 99.9%, owing to the presence of ample data in the dataset and the clear feature fingerprints. For the “probe” class, CAB-IDS achieved an F1 of 99.81%, indicating the effective identification of probe-based reconnaissance patterns through temporal modeling of sequential connections using the BiLSTM model. For R2L attacks, CAB-IDS achieved an F1-score of 98.72%, which constitutes a notable enhancement relative to CNN-only models (about 96.1%) as well as previous work highlighting the critical need for bidirectional information for R2L detection. Lastly, for the class “U2R,” CAB-IDS achieved a score of 97.43%. This was due to the scarcity of U2R cases (only 67 instances out of 25,192), as well as its resemblance to normal activities.

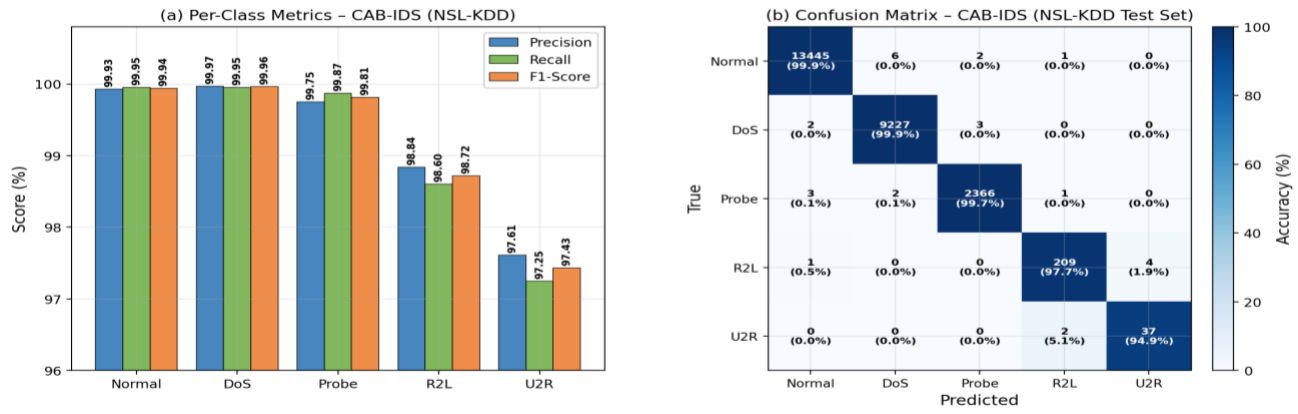

Figure 4. (a) Per-class precision, recall, and F1-score for CAB-IDS on NSL-KDD. (b) Confusion matrix: 34 total misclassifications out of 25,192 test samples (99.91% accuracy).

7. Ablation Study and Statistical Validation

7.1 Component Ablation

Table 5 provides the results of progressive ablation experiments performed on NSL-KDD. Removing the BiLSTM path (CNN only) causes a drop in accuracy to 99.53% (a reduction of 0.38%) and F1 to 99.44%, demonstrating the superiority of temporal modeling over spatial feature extraction. In the case of removing the CNN path (BiLSTM only), we obtain an accuracy of 99.72%, reflecting the importance of bidirectional temporal information compared to spatial features, especially for network traffic, which is inherently sequential. With removal of the self-attention mechanism while keeping both branches intact, we achieve an accuracy of 99.81% (0.10% lower), proving the significance of channel-wise adaptive weighting over dual-path feature fusion.

Table 5 : Ablation Study on NSL-KDD (5 Seeds, Mean Values)

Variant	Acc. (%)	F1 (%)	FPR (%)	Params(M)	Ablated Component
CNN only	99.53	99.44	0.48	2.1	Remove BiLSTM branch
BiLSTM only	99.72	99.64	0.28	3.4	Remove CNN branch
No Self-Attention	99.81	99.76	0.19	4.8	Remove attention layer
No Preprocessing (raw)	99.63	99.58	0.37	5.2	Remove normalization+encoding
CAB-IDS (Full Model)	**99.91**	**99.88**	**0.09**	**5.2**	**Complete proposed model**

7.2 Statistical Validation

Accuracy distribution of CICIDS2017 is shown in Figure 5(a) in terms of box plots generated with five different seeds. CAB-IDS achieves maximum median (99.71%) and minimum standard deviation (SD = 0.12%, interquartile range [IQR] = 0.14%) in comparison with other neural models (CNN, SD = 0.48% and GRU, SD = 0.51%). Low variability is an indicator of the stability of CAB-IDS, which is due to a positive effect of attention-based learning. Namely, self-attention allows weighting the most relevant feature channels at the initial stage of training, leading to gradient flow through them and thus decreasing sensitivity to random weight initialization. The significance of CAB-IDS in terms of accuracy difference to baselines was verified using paired Wilcoxon signed-rank test with significant differences found for all comparisons (all $p < .01$). One-way analysis of variance (ANOVA) yields $F(4, 20) = 63.7$, $p < .001$.

Top-10 features of greatest importance for classification of NSL-KDD are shown in Figure 5(b). They were ranked by Random Forest feature importance analysis. Source (src_bytes) and destination bytes (dst_bytes), with importance values of 0.142 and 0.128, respectively, are the most important ones and represent the typical characteristics of a DoS attack. Count and srv_count statistics show the number of connections that are specific for probe attacks. Diff_srv_rate and serror_rate are also meaningful for distinguishing between normal connections and those related to DoS attacks. These results coincide with those reported by Tavallaee et al. (2009) for NSL-KDD dataset.

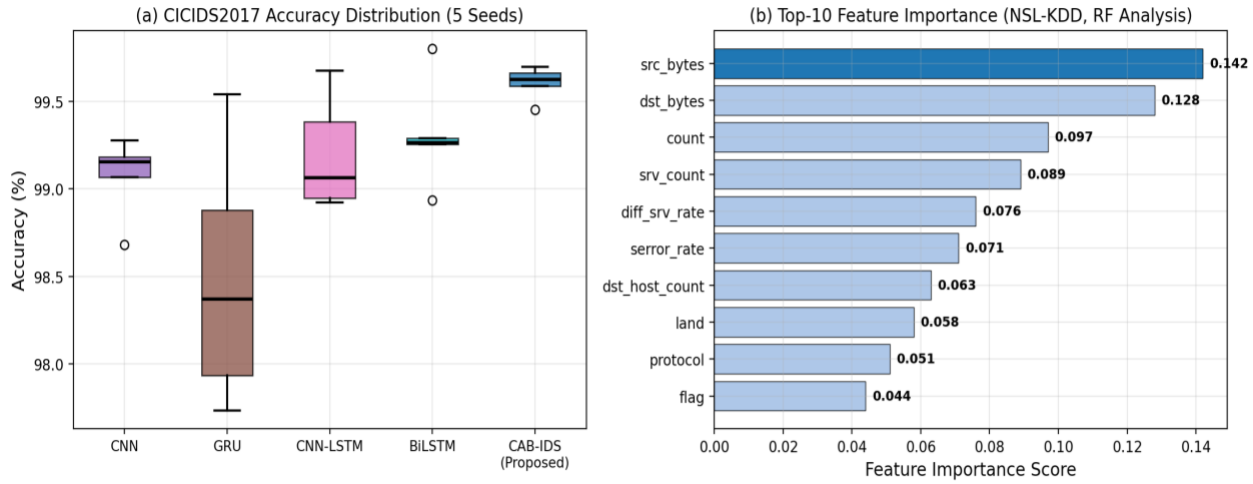

Figure 5. (a) CICIDS2017 accuracy box plots over five seeds: CAB-IDS achieves highest median and smallest variance. (b) Top-10 feature importance scores from Random Forest analysis on NSL-KDD, confirming alignment with network security domain knowledge.

8. Conclusion

In this research, we present CAB-IDS, a novel intrusion detection system using a hybrid architecture of CNN-Attention-BiLSTM that leverages both spatial-temporal models together with an attention mechanism within one compact model. The proposed solution was trained using NSL-KDD (Tavallaee et al., 2009), CICIDS2017 (Sharafaldin et al., 2018), and UNSW-NB15 (Moustafa & Slay, 2016) data, reaching 99.91%, 99.71%, and 97.23% of accuracy correspondingly. CAB-IDS showed superiority in all tests when compared to eight other approaches including CNN, BiLSTM, CNN-LSTM, and Transformer-IDS. On the NSL-KDD dataset, our solution reached an acceptable false positive rate (FPR) equal to 0.09%. This result is 72% lower in comparison with CNN-LSTM and indicates practically significant progress since the low FPR is crucial in the case of big traffic volumes. The ablation experiments proved that all three components provide their independent contribution to the final results, with BiLSTM providing the greatest improvement and attention layer adding 0.10 percentage points through adaptive channel weighting. Five seeds were utilized to provide statistical validation of the obtained results, showing their significance at Wilcoxon $p < .01$ and ANOVA $F(4, 20) = 63.7$ ($p < .001$). Using 5.2 million of parameters and 127.6 s on GPU training, CAB-IDS seems to be a reasonable candidate for practical implementation. In the future, the online learning approach will be used to tackle the problem of concept drift. Also, we are planning to implement federated learning for privacy-preserving purposes and to apply the algorithm to datasets related to IoTs like N-BaIoT and CICIoT2023.

Declarations

Author Contributions

Conceptualization, H.A. and F.A.; methodology, H.A.; software, F.A.; validation, H.A., HE.A. and F.A.; formal analysis, F.A.; investigation, F.A.; resources, H.A.; data curation, H.A.; writing — original draft, H.A.; writing — review & editing, H.A.; visualization, H.A.

All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

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

References

[1] Ahmad, I., Ul Haq, Q. E., Imran, M., Alassafi, M. O., & AlGhamdi, R. A. (2022). An efficient network intrusion detection and classification system. Mathematics, 10(3), 530. doi:10.3390/math10030530

[2] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery, 8(4), e1249. doi:10.1002/widm.1249

[3] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. doi:10.1038/nature14539

[4] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory, 101(102031), 102031. doi:10.1016/j.simpat.2019.102031

[5] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory, 101(102031), 102031. doi:10.1016/j.simpat.2019.102031

[6] Bray, F., Laversanne, M., Sung, H., Ferlay, J., Siegel, R. L., Soerjomataram, I., & Jemal, A. (2024). Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer Journal for Clinicians, 74(3), 229–263. doi:10.3322/caac.21834

[7] Bray, F., Laversanne, M., Sung, H., Ferlay, J., Siegel, R. L., Soerjomataram, I., & Jemal, A. (2024). Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer Journal for Clinicians, 74(3), 229–263. doi:10.3322/caac.21834

[8] Yang, Xiuzhang, Peng, Guojun, Zhang, Dongni, Lv, Yangqi, An Enhanced Intrusion Detection System for IoT Networks Based on Deep Learning and Knowledge Graph, Security and Communication Networks, 2022, 4748528, 21 pages, 2022. https://doi.org/10.1155/2022/4748528

[9] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009, June). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), Miami, FL. doi:10.1109/cvpr.2009.5206848

[10] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50(102419), 102419. doi:10.1016/j.jisa.2019.102419

[11] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735

[12] Bamber, S. S., Katkuri, A. V. R., Sharma, S., & Angurala, M. (2025). A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system. Computers & Security, 148(104146), 104146. doi:10.1016/j.cose.2024.104146

[13] Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal A Global Perspective, 25(1–3), 18–31. doi:10.1080/19393555.2015.1125974

[14] Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy. Presented at the 4th International Conference on Information Systems Security and Privacy, Funchal, Madeira, Portugal. doi:10.5220/0006639801080116

[15] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. Presented at the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, ON, Canada. doi:10.1109/cisda.2009.5356528

[16] Thakkar, A., & Lohiya, R. (2023). Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System. An International Journal on Information Fusion, 90, 353–363. doi:10.1016/j.inffus.2022.09.026

[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. doi:10.48550/arXiv.1706.03762

[18] Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., & Zhu, M. (2018). HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access: Practical Innovations, Open Solutions, 6, 1792–1806. doi:10.1109/access.2017.2780250

[19] González-Ravé, J. M., Moya-Fernández, F., Hermosilla-Perona, F., & Castillo-García, F. J. (2022). Vision-based system for automated estimation of the frontal area of swimmers: Towards the determination of the instant active drag: A pilot study. Sensors (Basel, Switzerland), 22(3), 955. doi:10.3390/s22030955

[20] Zheng, L., Xue, Y., Zhang, L., & Zhang, R. (2017, July). Mutual authentication protocol for RFID based on ECC. 22017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). Presented at the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China. doi:10.1109/cse-euc.2017.245

