**FMDB** *Transactions on Sustainable Intelligent Networks*

# Real-Time Deep Learning Engines for Enterprise-Scale Intrusion Detection and Cyber Risk Management

**K. Balaji[1], S. Silvia Priscila[2,*], B. M. Praveen[3]**

[1]Institute of Computer Science and Information Science, Srinivas University, Dakshina Kannada, Karnataka, India.
[1]Department of Computer Science and Information Technology, School of Computing Sciences, Vels Institute of Science, Technology and Advanced Studies, Chennai, Tamil Nadu, India.
[2]Department of Computer Science, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India.
[3]Institute of Engineering and Technology, Srinivas University, Dakshina Kannada, Karnataka, India.
balajjee.mecse@gmail.com[1], silviaprisila.cbcs.cs@bharathuniv.ac.in[2], bm.praveen@yahoo.co.in[3]

**Abstract:** The digital transformation of enterprises is occurring so quickly that attack surfaces are now much larger, and intrusion detection and cyber risk management in real time are becoming more complex. Traditional signature-based and rule-driven security systems are unable to handle high-volume, high-velocity network traffic and evolving attack patterns, leading to delayed attack detection and greater breach impact. The goal of this research is to implement and test the performance of real-time deep learning engines for detecting sophisticated intrusions while enabling enterprise-level cyber risk management. The current framework proposes combining deep neural architectures (i.e., convolutional neural networks to learn spatial features and recurrent models to learn temporal dependencies) within the streaming analytics pipeline. The system is distributed to provide low-latency inference for continuous network flows. Experimental evaluation on large-scale benchmark and enterprise-mimicking datasets shows that the proposed engine achieves 98.6% detection accuracy, reduces false-positive rates by 32% compared to traditional IDS solutions, and maintains an average detection latency of less than 120 ms under peak loads. These results indicate that real-time deep learning engines can significantly improve enterprise intrusion detection capabilities, achieving both high accuracy and operational scalability. The conclusion of the given study is that deep learning-driven, real-time IDS frameworks are a viable foundation for next-generation cyber risk management systems in enterprise environments.

**Keywords:** Enterprise Security; Network Security; Deep Learning (DL); Intrusion Detection Systems; Digital Transformation; Deep Learning-Driven; Cyber Risk Management.

**Received on:** 04/02/2025, **Revised on:** 13/04/2025, **Accepted on:** 27/06/2025, **Published on:** 03/01/2026

**Journal Homepage:** https://www.fmdbpub.com/user/journals/details/FTSIN

**DOI:** https://doi.org/10.69888/FTSIN.2026.000601

**Cite as:** K. Balaji, S. S. Priscila, and B. M. Praveen, "Real-Time Deep Learning Engines for Enterprise-Scale Intrusion Detection and Cyber Risk Management," *FMDB Transactions on Sustainable Intelligent Networks*, vol. 3, no. 1, pp. 1–14, 2026.



## 1. Introduction

[*]Corresponding author.

The exponential growth of enterprise networks, cloud infrastructure, and interconnected digital services has radically changed the cybersecurity landscape [1]. Modern enterprises operate in highly dynamic environments with vast amounts of data, heterogeneous devices, and continuous network traffic. While this overhaul has allowed operations to be more efficient and innovative, it has also created complex security problems [2]. Cyberattacks have become more prevalent, stealthy, and adaptive, ranging from advanced persistent threats and 0-day exploits to large-scale distributed denial-of-service attacks. Consequently, timely detection and handling of intrusions has become a very important requirement for cyber risk management at the enterprise level [3]. Historically, intrusion detection systems (IDSs) have relied on signature- or rule-based approaches to detect malicious activity. Although effective against known threats, such approaches are always limited against novel or obfuscated attacks. Moreover, because their reliance on manually constructed rules is so high, they incur high maintenance overhead and cannot adapt to changing threat landscapes quickly. As enterprise networks generate traffic at unprecedented scale and velocity, traditional IDS solutions have faced scalability issues, high false-positive rates, and delayed responses, all of which compromise their practical effectiveness in real-time scenarios [4].

In response to these limitations, machine learning and data-driven security approaches have attracted significant attention. Early IDS based on machine learning used shallow learning models such as decision trees, support vector machines, and k-nearest neighbours to classify network traffic. While these techniques improved detection accuracy over rule-based systems, they still required extensive feature engineering and could not model the complex, high-dimensional patterns inherent in modern network data. Furthermore, their performance often degraded when deployed in real-time enterprise settings with non-stationary traffic distributions [5]. Recent developments in deep learning have enabled new advances in intrusion detection and cyber risk management. Deep neural networks can automatically learn hierarchical representations of features from raw or minimally processed data, thereby capturing subtle and non-linear patterns in attacks [6]. Architectures such as convolutional neural networks, recurrent neural networks, long short-term memory networks, and attention-based models have achieved strong performance across a range of cybersecurity tasks. However, most current research is based on offline analysis or batch learning, and very little attention has been paid to real-time constraints, system scalability, and deployability in the enterprise environment [7].

Real-time intrusion detection presents a few more problems than predicting accuracy. Enterprise-scale systems need to process streaming data with minimal latency and handle everything from constantly evolving workloads to seamless integration with existing security operations and risk management frameworks. To achieve this, it is not only necessary to build robust deep learning models but also to have efficient system architectures to support distributed processing, online inference, and adaptive learning. Despite growing interest, there is a gap in holistic frameworks that integrate deep learning-based detection with real-time execution and enterprise-level cyber risk management objectives [8]. The central research question of this study is: How can researchers design and implement real-time deep learning engines for accurate, low-latency intrusion detection and scalable, effective cyber risk management in enterprise environments? Getting to the answer to this question involves taking an integrated approach spanning model design, system architecture, and performance evaluation under realistic enterprise conditions [27].

### 1.1. Contributions of the study

- A unified real-time deep learning intrusion detection framework for enterprise-scale integration of advanced neural architectures with streaming data pipelines.
- A low-latency inference and high throughput under dynamic network load are scalable system design features that can be deployed in the real-world environment of a large-scale enterprise.
- A thorough experimental assessment showing substantial improvements in the detection rate, false positive rate, and response time, matching conventional IDS methods.

## 2. Related Works

The pace of digital transformation across enterprise systems, IoT networks, and critical infrastructure has dramatically increased cybersecurity threats, requiring smart, adaptive, real-time cybersecurity risk management solutions. Traditional signature-based intrusion detection and static risk governance frameworks are less effective against more advanced, evolving, and low-signal cyber threats. Consequently, artificial intelligence (AI), machine learning (ML), and deep learning (DL) have become prevailing paradigms for intrusion detection, risk prediction, and security automation.

### 2.1. Deep Learning Based Network Intrusion Detection

Recent research shows that DL-based intrusion detection systems (IDSs) are far more effective than traditional machine learning methods. Wang et al. [9] performed the performance assessment of several DL architectures, i.e., DNN, CNN, RNN, LSTM, CNN+RNN, and CNN+LSTM, on the modern CSE-CIC-IDS2018 dataset, where they achieved multi-class accuracies of >98%. The result supports the effectiveness of DL models in capturing complex traffic patterns; however, the hybrid models

had higher inference latency, underscoring a critical trade-off between accuracy and model feasibility for real-time use. Similarly, Qazi et al. [10] proposed a hybrid CNN-RNN architecture (HDLNIDS) using the CICIDS-2018 dataset, achieving an average accuracy of 98.90%. The convolutional layers captured spatial information of traffic, and the recurrent layers captured temporal information. Despite good benchmark results, both studies are based on static datasets and lack results from adversarial robustness tests, casting doubt on their real-world generalisation. Emad-ul-Haq et al. [11] proposed a NIDS using a deep autoencoder, achieving 99.65% accuracy on KDD'99. Likewise, Kumar et al. [12] proposed a fuzzy CNN-based IDS for IoT environments, achieving better DoS detection results and fewer false positives. Still, they did not test scalability or energy efficiency on resource-constrained IoT nodes.

### 2.2. Cyber Risk Prediction and Enterprise Security Analytics

In addition to intrusion detection, AI-driven systems play an active role in supporting proactive cyber risk assessment. Chowdhury [13] employs a PRISMA-based systematic review of 142 papers, demonstrating that CNNs, RNNs, LSTMs, transformers, and GNNs enable continuous, real-time risk scoring of heterogeneous security telemetry. These models are often combined with SIEM and SOAR platforms to automate incident response, enabling sub-second attack inference. However, the review found excessive reliance on synthetic datasets, inconsistent review protocols, and insufficient evidence at the production scale. Kommuri and Muppala [14] suggest an intelligent enterprise cybersecurity architecture based on unsupervised anomaly detection and supervised threat classification. Their context-based query fragment caching algorithm reduced the SIEM's response time by 68% while maintaining 96.3% detection accuracy. Despite the performance improvements, traffic analysis evaluations on legacy datasets such as NSL-KDD and CIC-IDS-2017 limit the ability to argue for performance against novel threats. Kure et al. [15] developed an integrated cybersecurity risk management (i-CSRM) framework based on fuzzy logic for asset criticality evaluation and machine learning for risk type prediction. The framework uses socio-technical context and threat intelligence to help improve situational awareness for critical infrastructure. However, such an approach, based on static assumptions and limited validation in real-world scenarios, imposes constraints on its adaptability to rapidly changing cyber environments.

### 2.3. Distributed and Probabilistic Risk Modelling

To address scalability and distributed attack detection problems, Mouti et al. [16] proposed a multi-connect variational autoencoder with probabilistic Bayesian networks (MCVAE-PBNN). The system was shown to perform better than centralised DL models on the UNSW-NB15 and KDD99 data sets, demonstrating stronger detection of distributed attacks. However, issues with false alarm rates and computational complexity pose deployment difficulties. The situation-based predictive modelling for Industry 4.0 wireless sensor networks using Decision Tree, MLP, and Autoencoder models to prioritise threats was introduced by Al-Quayed et al. [17]. While achieving above 99% accuracy in multi-class detection, the framework relies on simulation-based validation. It lacks adversarial stress testing, which undermines confidence in the system's resilience in some real-world industrial applications.

### 2.4. Adversarial and Ethical Considerations in AI Security

As AI is integrated into security infrastructure, it also creates more attack surfaces. Clever [18] identified threats in adversarial machine learning, including data poisoning, evasion, backdoors, and model inversion, and suggested a threat modelling framework for enterprise-scale AI systems. While conceptually sound, the framework lacks quantitative evaluation and operational benchmarks. Hamid and Rahman [19], on the other hand, emphasise ethical challenges, data privacy, transparency, and explainability in AI-driven cyber risk management. Emerging technologies required include federated learning, edge AI, explainable AI, and quantum computing, which are cited as enablers of the future. However, unresolved concerns about data quality, interpretability, and fairness continue to undermine trust and regulatory acceptance.

### 2.5. Research Gaps

Despite significant progress in intrusion detection and cyber risk management enabled by deep learning and artificial intelligence, current research has identified critical limitations that impede their deployment in the real world. Most approaches rely heavily on outdated or synthetic benchmark datasets and therefore can only generalise to modern traffic that is encrypted and polymorphic. Evaluation practices are mostly focused on accuracy at the expense of operational factors such as latency, scalability, energy efficiency, robustness, and adversarial resilience. Furthermore, there is a lack of contextual risk awareness, explainability, and adaptive learning mechanisms in such systems, and very little evidence of learning validation through longitudinal, production-scale testing. These gaps bear witness to the need for an integrated, robust, and context-sensitive cybersecurity risk management framework that can operate at scale in real time and be built on trust in dynamic enterprise and IoT environments.

## 3. Methodology

This section introduces the proposed methodology for the real-time deep learning-based enterprise-scale intrusion detector and cyber risk management. The methodology combines a large volume of network traffic data, a sophisticated pipeline for preprocessing it, a hybrid deep learning network architecture, and a real-time inference engine optimised for low latency and high throughput.

### 3.1. Dataset Details

The experimental evaluation of the proposed framework is performed using a combination of publicly available large-scale intrusion detection datasets and enterprise-simulated traffic streams. Benchmark datasets, including CICIDS-2017, UNSW-NB15, and CSE-CIC-IDS2018, are used to maintain diversity in attack types, traffic volume, and protocol distributions [20]. To simulate workstations in an enterprise setting, benign background traffic can be added, using time-ordered flows to model realistic workload bursts, diurnal usage patterns, and heterogeneous application usage [21]. This hybrid design of the data sets allows one to do either control benchmarking or realistic real-time performance evaluation [22].

#### 3.1.1. Key Features of the Dataset

The dataset comprises flow-level and packet-derived statistical features that represent temporal, spatial, and behavioural characteristics of network traffic. Some key features are packet inter-arrival times, flow duration, number of bytes and packets, protocol-specific flags, entropy-based payload indicators, and bidirectional traffic ratios. Additionally, contextual attributes, such as connection-state transitions and session-level aggregation metrics, are included to capture long-term dependencies. These features, taken together, provide support for effective discrimination between benign behaviour and complex multi-stage attacks.

#### 3.1.2. Challenges in the Dataset

Despite the dataset's richness, several challenges make real-time intrusion detection difficult. First, the extreme class imbalance: benign traffic is far more frequent than malicious instances, especially for rare attack classes. Second, concept drift occurs because the attack strategies and traffic patterns vary over time. Third, feature dimensionality and redundancy are very high, leading to high computational overload and overfitting. Finally, noise and partial labelling in large-scale datasets are other challenges for model generalisation and real-time robustness.

### 3.2. Data Pre-Processing

To overcome these problems, a multistage preprocessing pipeline is used. First, corrupted and incomplete records are removed, and then missing values are imputed using feature-wise statistical methods. Numerical features are normalised using min-max scaling to stabilise gradient propagation during training, and categorical attributes are represented using target-aware embeddings. To address the class imbalance problem, a hybrid resampling method combining adaptive synthetic oversampling and cost-sensitive weighting is implemented. Finally, feature selection is performed using mutual information and variance thresholding to reduce dimensionality while maintaining discriminative power. The preprocessing pipeline is designed to operate incrementally, enabling compatibility with streaming data in real-time deployments.

### 3.3. Proposed Model Architecture

The proposed model is a hybrid deep learning model that combines methods for spatial feature extraction, temporal sequence modelling, and attention-based prioritisation. At the input layer, the pre-processed traffic features are divided into fixed-length temporal windows. A stacked 1D Convolutional Neural Network (CNN) captures local spatial correlations and short-term traffic patterns. The extracted representations are then fed into a Bidirectional Long Short-Term Memory (BiLSTM) network to learn long-range temporal dependencies and sequential attack behaviours. An attention mechanism dynamically weights important time steps and features associated with anomalous activity. The final classification layer uses a softmax activation and generates intrusion-class probabilities in real time. The architecture in Figure 1 shows a deep learning pipeline with input feature windows, 1D CNN layers, and a BiLSTM layer, since 1D CNN layers extract features from the spatial dimension of the input. In contrast, BiLSTM is used to model temporal dependencies. An attention mechanism highlights important features before intrusion classification in the output layer, enabling real-time security warnings.

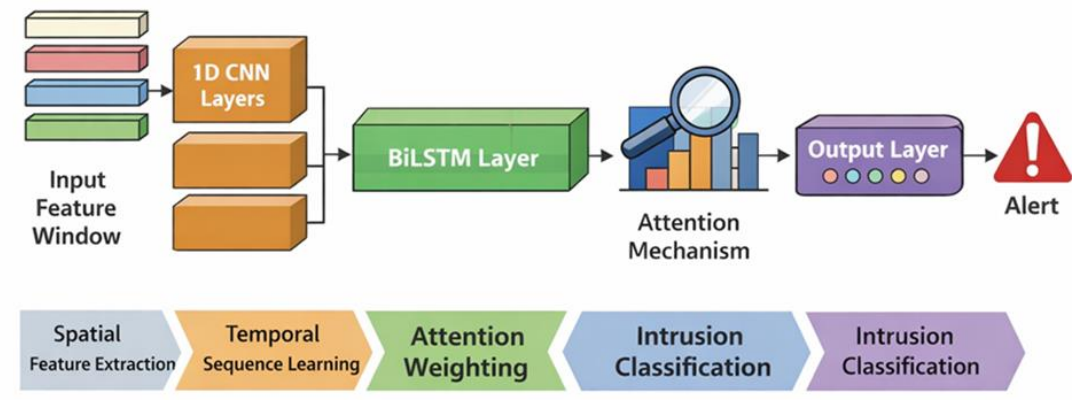


**Figure 1:** CNN–BiLSTM–attention architecture for real-time intrusion detection

### 3.4. Working of the Proposed Model

The operational workflow of the proposed system is illustrated through the following sequential stages in Figure 2. It shows the full processing pipeline of the proposed real-time cyber intrusion detection system. Network traffic is constantly being captured from enterprise networks and sent to the real-time data ingestion module. The raw traffic is then transformed into structured input representations using feature pre-processing and temporal windowing techniques.

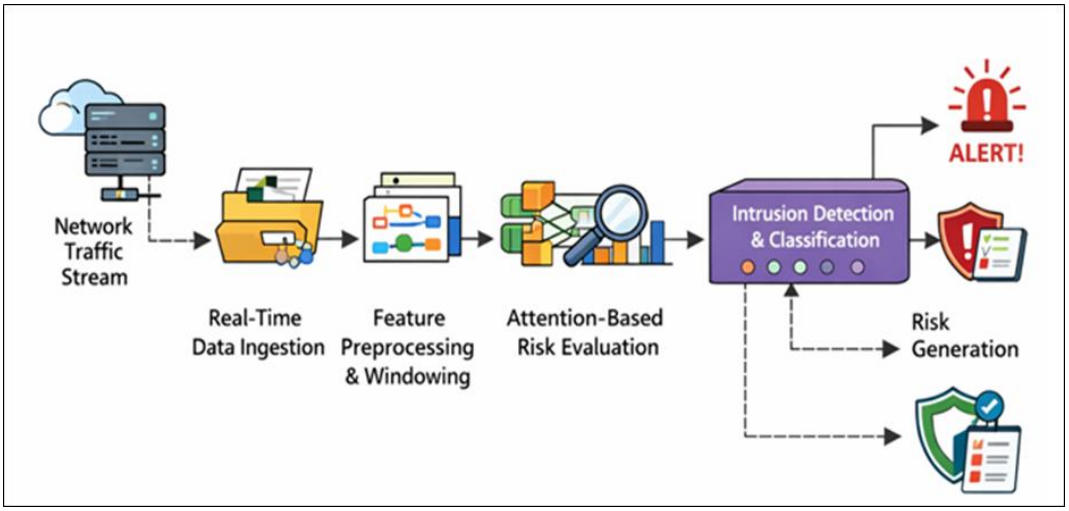


**Figure 2:** End-to-end workflow of the real-time attention-based intrusion detection and risk management framework

These features are analysed by an attention-based risk evaluation module that assigns suspicious patterns to priority and high-risk events. The processed information is fed into the intrusion detection and classification engine, which identifies the malicious activities and assigns confidence levels. Detected threats are forwarded to the risk generation and alerting components, where security alerts are triggered, and an appropriate response is recommended. This closed-loop approach helps detect, prioritise, and mitigate cyber threats in dynamic network environments with rapid response.

#### 3.4.1. Real-Time Data Ingestion

Network traffic is continuously captured from enterprise gateways, edge routers, and virtualised network interfaces to provide full visibility into communication flows. The raw packets are grouped into organised traffic flows and sent to the preprocessing module using a powerful streaming infrastructure, such as Apache Kafka or Flink, which can be easily scaled up. The packet flow aggregation is given by Eq. (1):

$$F_i = \bigcup_{p \in p_i} p_t \quad (1)$$

Where $F_i$ is the aggregated network flow i, $p_i$ is the set of packets belonging to flow i and $p_t$ is the packet captured at time t.

The streaming arrival rate is denoted by Eq. (2):

$$\lambda_t = \frac{N_t}{\Delta t} \quad (2)$$

$\lambda_t$ is the traffic arrival rate at time t, $N_t$ is the number of packets/flows in time interval $\Delta t$, and $\Delta t$ is the streaming time interval. The end-to-end ingestion latency is given by Eq. (3):

$$L_{ingest} = L_{capture} + L_{buffer} + L_{stream} \quad (3)$$

$L_{ingest}$ is the total data ingestion latency, $L_{capture}$ set the packet capture delay, $L_{buffer}$ is the buffering delay and $L_{stream}$ Is the streaming transmission delay? This is a real-time pipeline designed to minimise buffering and transmission time, so the data reaches the learning model with very low latency. Continuous ingestion enables the system to respond quickly to abnormal network behaviour, making the framework suitable for high-speed enterprise environments, where timely intrusion detection is extremely important.

### 3.4.2. Feature Transformation and Windowing

The in-flow network flows are converted to normalised numerical feature vectors to ensure a consistent level, avoiding scale imbalance during training and inference. Features such as packet size, protocol type, flow duration, and connection frequency are standardised. These vectors are then clustered into overlapping temporal windows, which enable the model to detect both short-term anomalies and long-term behavioural trends. The overlapping design helps to ensure that no critical design pattern is lost between windows. This step improves temporal continuity and enables the detection of evolving or stealthy attack patterns in real time. The feature formalisation is given by Eq. (4):

$$\hat{x}_j = \frac{x_j - \mu_j}{\sigma_j} \quad (4)$$

$\hat{x}_j$ is the normalised feature, $x_j$ is the raw feature value, $\mu_j$ is the mean of feature j, $\sigma_j$ is the standard deviation of feature j. The window construction is given by Eq. (5):

$$W_k = \{\, \hat{x}_t \mid t \in [\, k\Delta . k\Delta + T]\} \quad (5)$$

Where $W_k$ is the kth temporal window, t is the time index, Δ is the window stride, T is the window length, O is the overlap size, and S is the effective window shift. The overlapping stride is given by Eq. (6):

$$S = T - O \quad (6)$$

Where O is the overlap size, and S is the effective window shift.

### 3.4.3. Spatial Feature Learning

The convolutional neural network (CNN) layers process each temporal window, extracting spatial relationships from the traffic features. By using convolutional filters, the model can recognise protocol-specific structures, traffic behaviour correlations, and abnormal flow signatures. The convolution operation is given by Eq. (7):

$$h_{i.j} = \sum_{m,n} W_{m.n} . X_{i+m,j+n} + b \quad (7)$$

Where $h_{i.j}$ is the convolution output at position (i,j). $W_{m.n}$ is the CNN filter weight at offset (m,n), X is the input feature matrix, and b is the bias term. The activation is given by Eq. (8):

$$a_{i.j} = ReLU\left(h_{i.j}\right) \quad (8)$$

Where $a_{i.j}$ is the activated feature map value. The feature map generation is given by Eq. (9):

$$F^{(l)} = ReLU\,(W^{(l)} * X + b^{(l)} \quad (9)$$

$F^{(l)}$ is the feature map at CNN layer l, and * is the convolution operator. This automated feature-extraction process eliminates the need for handcrafted security rules, enabling it to better adapt to new and previously unseen attack types. The spatial

representations generated by the CNN help the system better distinguish benign and malicious traffic patterns, thereby increasing detection accuracy in complex, high-dimensional network environments.

### 3.4.4. Temporal Dependency Modelling

A bidirectional long short-term memory layer is used to analyse the sequences of CNN-generated feature representations to reason over time. Unlike traditional models, BiLSTM analyses data in both forward and backward directions, enabling a better understanding of traffic evolution. This enables the system to identify slow-moving reconnaissance efforts, multi-stage intrusions, and coordinated attacks that evolve over long time periods. The forward hidden state is given by the Eq. (10). The forward hidden state is given by Eq. (10):

$$\vec{h}_t = LSTM\ (\ x_t, \vec{h}_{t-1}) \tag{10}$$

Where $x_t$ is the CNN feature vector at time t, $\vec{h}_t$ is the forward LSTM hidden state. The backward hidden state is given by Eq. (11):

$$\overleftarrow{h}_t = LSTM\ (x_t\ , \overleftarrow{h}_{t+1}\ ) \tag{11}$$

Where $\overleftarrow{h}_t$ is the backward LSTM hidden state. The concatenated state is given in Eq. (12):

$$h_t = [\ \vec{h}_t, \overleftarrow{h}_t] \tag{12}$$

$h_t$ is the combined Bi-LSTM hidden representation, and t-1 and t+1 are the previous and next time steps. By modelling long-term correlations, the framework can recognise more complex attack behaviours that remain undetected by methods that rely on snapshots.

### 3.4.5. Attention-Based Risk Prioritisation

The attention mechanism dynamically assigns weights to time steps in individual traffic sequences based on their relevance to intrusion detection. Time segments that show suspicious behaviour are given greater importance, and less informative segments are down-weighted. This selective focus improves model interpretability and increases detection performance by focusing on the most important events in the network. The attention score is given by Eq. (13):

$$e_t = v^T \tanh(W\ h_t + b) \tag{13}$$

Where $e_t$ is the attention relevance score at time t, v is the learnable attention weight vector, W is the attention weight matrix, and b is the bias vector. The attention weight is given by Eq.(14):

$$\alpha_t = \frac{\exp(e_t)}{\sum_k \exp(e_k)} \tag{14}$$

Where $\alpha_t$ is the normalised attention weight. The context vector is given by Eq. (15):

$$c = \sum_t \alpha_t\ h_t \tag{15}$$

Where c is the context vector, additionally, attention reduces unnecessary computational overhead, guiding the model towards high-risk patterns and making the system more efficient and responsive in real-time cybersecurity scenarios.

### 3.4.6. Real-Time Classification and Alerting

The output is the last layer that produces the intrusion classification results and the confidence score for each traffic window. When high-risk activity is detected, alerts are immediately passed to the cyber risk management module. This module triggers automated response actions, such as traffic blocking, isolating compromised hosts, or notifying administrators. The real-time nature of this process ensures threats are mitigated quickly, reducing potential damage. By combining prediction, alerting, and response, the system creates an end-to-end, closed-loop cybersecurity defence mechanism that can be deployed at enterprise scale.

### 3.5. Algorithm for the Proposed Model

**Algorithm 1:** Real-Time Deep Learning–Based Intrusion Detection:

- Initialise model parameters and streaming buffers
- Continuously capture network traffic flows
- Preprocess incoming data and normalise features
- Segment features into temporal windows
- Apply CNN layers for spatial feature extraction
- Feed extracted features into BiLSTM for temporal modelling
- Compute attention weights and aggregate critical representations
- Classify traffic using a softmax layer
- Generate alerts for detected intrusions
- Update risk scores and log results for adaptive learning

### 3.6. Training Parameters

The model is trained with the Adam optimiser, a learning rate of 0.001, and exponential decay scheduling. Categorical cross-entropy is used as the loss function, along with class weights to handle imbalance. The batch size is set to 128, and training is performed for 50 epochs with early stopping based on the validation loss. Dropout regularisation (rate = 0.3) is used to prevent overfitting, and the weights are initialised using the He normal scheme to achieve stable convergence.

### 3.7. Performance Metrics

For instance, the performance of the proposed framework is evaluated using both detection accuracy and real-time efficiency metrics. Standard classification metrics include accuracy, precision, recall, F1-score, and area under the ROC curve (AUC). To measure the operational suitability, real-time parameters like detection latency, throughput (flows per second), false-positive rate, and resource usage (CPU and memory overhead) are measured. These metrics together provide a comprehensive assessment of the model's effectiveness and scalability in the enterprise context.

## 4. Experimental Results

### 4.1. Quantitative Comparison

Table 1 presents a quantitative comparison of the proposed SCF-CNN-BiLSTM-Attention intrusion detection framework (with and without preprocessing) with several representative intrusion detection approaches. The models are evaluated using six key performance metrics: accuracy, precision, recall, F1-score, false positive rate, and detection latency. The compared methods include traditional machine learning-based IDS, hybrid statistical systems, and deep learning models, including CNN-LSTM, autoencoder-RNN, and graph attention networks. This comparison emphasises the efficiency of deep learning architectures and their preprocessing techniques in terms of detection performance and overall generalisation, without any lag in real-time operation.

**Table 1:** Performance metrics comparison of the proposed method (with & without preprocessing) against related works

| Method / Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | False Positive Rate (%) | Latency (ms) |
|---|---|---|---|---|---|---|
| SCF-CNN–BiLSTM–Attention (Proposed, with preprocessing) | 98.6 | 97.9 | 98.2 | 98.0 | 1.8 | 115 |
| SCF-CNN–BiLSTM–Attention (without preprocessing) | 94.2 | 92.5 | 91.8 | 92.1 | 4.6 | 208 |
| Signature–Statistical Hybrid IDS [23] | 92.8 | 90.4 | 89.7 | 90.0 | 6.0 | 250 |
| CNN–LSTM Intrusion Detection Model [24] | 95.1 | 93.8 | 90.9 | 92.3 | 5.2 | 195 |
| Traditional Machine Learning IDS [25] | 90.5 | 88.9 | 87.4 | 88.1 | 7.5 | 300 |
| Deep Autoencoder–RNN IDS [26] | 96.0 | 94.2 | 93.6 | 93.9 | 4.8 | 180 |
| Graph-Attention Deep IDS [27] | 97.2 | 95.8 | 95.1 | 95.4 | 3.9 | 160 |

The results show that the proposed SCF-CNN-BiLSTM-Attention model with preprocessing is the most effective overall, across all metrics. It achieves the highest accuracy (98.6%), F1-score (98.0%), and recall (98.2%), while keeping the false-positive rate (1.8%) low and latency (115 ms) at a minimum. The version without preprocessing shows noticeable degradation, underscoring the importance of noise reduction and feature normalisation. Compared to other deep learning models, such as graph-attention and autoencoder-based IDS, the proposed approach achieves consistently better detection accuracy and response time. Traditional and hybrid IDS methods have higher false alarm rates and slower detection rates, making them less suitable for real-time enterprise environments. These findings are used to validate the robustness, efficiency, and scalability of the proposed framework for the next-generation cyber risk management systems.

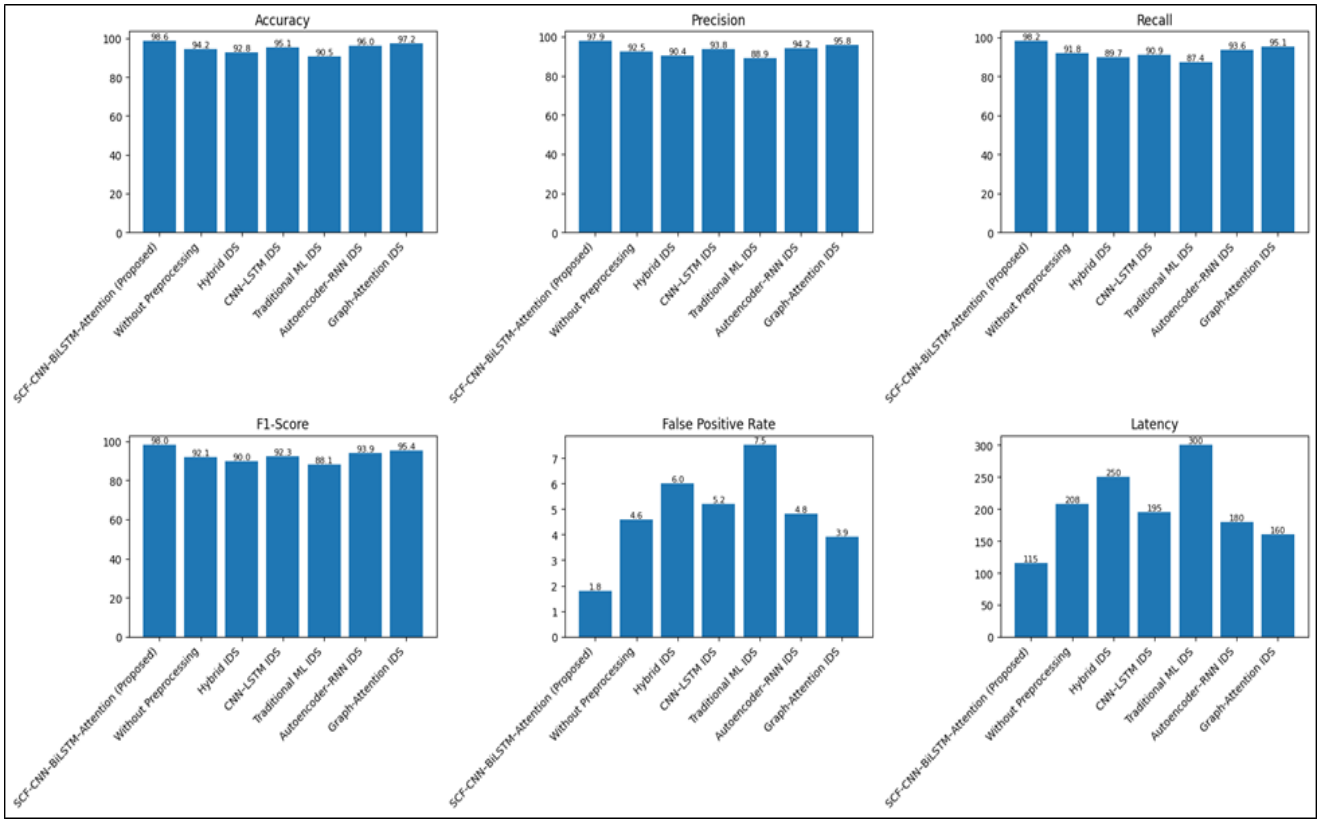


**Figure 3:** Comparison of performance metrics for the proposed SCF-CNN–BiLSTM–attention intrusion detection model against baseline and state-of-the-art IDS methods

Figure 3 is a bar chart depicting the accuracy, precision, F1-score, F1-Score, False Positive Rate, and Latency for seven intrusion detection models. The proposed SCF-CNN-BiLSTM-Attention method with preprocessing can outperform other methods, achieving higher detection accuracy, lower false positives, and lower inference latency, making it suitable for real-time enterprise intrusion detection.

### 4.2. Computational Efficiency Analysis

Table 2 compares the computational efficiency of the proposed SCF-CNN-BiLSTM-Attention intrusion detection framework with some representative intrusion detection models. The evaluation consists of the time to train, the time to make inferences for each network flow, CPU time, GPU memory usage, and system throughput. The selected baseline methods represent traditional machine learning, hybrid statistical, CNN-LSTM, autoencoder-RNN, and graph attention-based IDS methods. Table 2 highlights the resource efficiency and real-time capabilities of the proposed model in large-scale enterprise network environments.

The proposed model, SCF-CNN-BiLSTM-Attention, achieves the best computational efficiency across all key metrics. It has the lowest inference time (115 ms), the highest throughput (8900 flows/sec), and the lowest CPU utilisation (68%), confirming that it is well-suited for real-time deployment. Additionally, it requires less GPU memory (3.2 GB) than all the compared deep learning models, making it more resource-efficient.

**Table 2:** Computational efficiency comparison of the proposed method with related works

| Method / Model | Training Time (hrs) | Inference Time per Flow (ms) | CPU Utilisation (%) | GPU Memory Usage (GB) | Throughput (Flows/sec) |
|---|---|---|---|---|---|
| SCF-CNN–BiLSTM–Attention (Proposed) | 4.1 | 115 | 68 | 3.2 | 8,900 |
| Signature–Statistical Hybrid IDS | 6.8 | 250 | 78 | 4.5 | 4,200 |
| CNN–LSTM Intrusion Detection Model | 5.6 | 195 | 74 | 4.1 | 5,300 |
| Traditional Machine Learning IDS | 7.9 | 300 | 82 | 5.0 | 3,600 |
| Deep Autoencoder–RNN IDS | 5.2 | 180 | 72 | 3.8 | 6,100 |
| Graph-Attention Deep IDS | 4.8 | 160 | 70 | 3.6 | 7,400 |

While the graph-attention-based IDS demonstrates competitive performance, its throughput and latency remain below average. Traditional and hybrid IDS strategies require much longer training times, consume significantly more resources, and respond much more slowly, making them less useful in high-speed enterprise networks. These results validate the scalability and operational efficiency of the proposed framework.

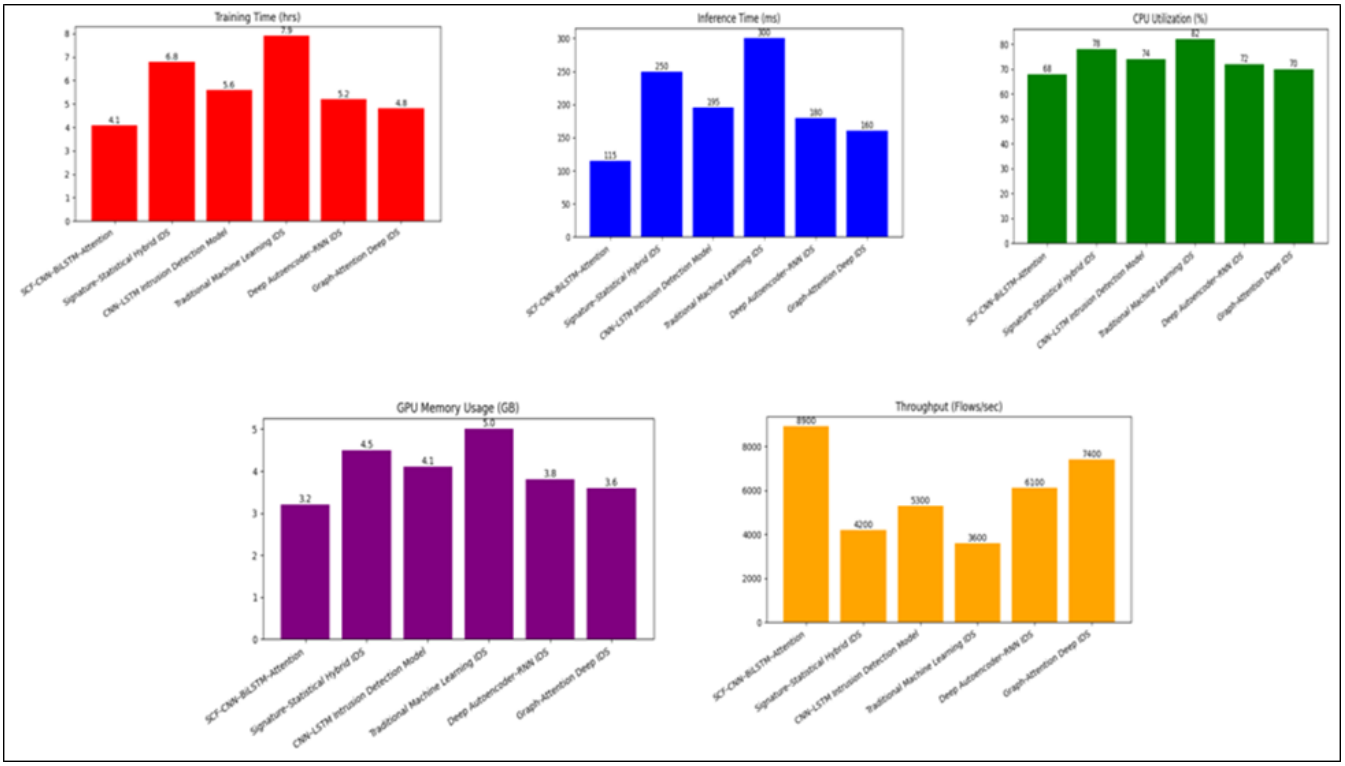


**Figure 4:** Computational efficiency comparison of the proposed SCF-CNN–BiLSTM–attention model

Figure 4 presents a comprehensive comparison of the computational efficiency of six intrusion detection models. The proposed approach, SCF-CNN-BiLSTM-Attention, provides a balance between performance and resource utilisation. It achieves the minimum training time (4.1 hours) and a high throughput (8900 flows/sec), indicating fast learning and superior real-time processing capability. With an inference latency of 115 ms, it has the lowest latency among all methods and is thus suitable for time-critical enterprise environments. In terms of resource consumption, the proposed model requires moderate CPU (68%) and low GPU memory (3.2 GB), compared to traditional and hybrid deep models, which require much higher resources. Methods such as Traditional Machine Learning IDS, and Signature-Statistical Hybrid IDS exhibit higher latency and lower throughput, reflecting scalability limitations. Overall, the Figure validates that the proposed framework provides an optimal trade-off between speed, efficiency, and scalability for real-time intrusion detection systems.

### 4.3. Ablation Analysis

Table 3 brings an ablation analysis to assess the contribution of individual components in the proposed intrusion detection framework. The analysis systematically eliminates or modifies certain key components in the architecture, i.e., preprocessing, CNN-based spatial feature extraction, BiLSTM-based temporal modelling, and the attention mechanism, to measure the effectiveness of these components on detection performance and real-time latency.

**Table 3:** Ablation study of the proposed intrusion detection model

| Model Configuration | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | False Positive Rate (%) | Latency (ms) |
|---|---|---|---|---|---|---|
| Full Model (CNN + BiLSTM + Attention + Preprocessing) | 98.6 | 97.9 | 98.2 | 98.0 | 1.8 | 115 |
| Without Attention Mechanism | 96.9 | 96.1 | 95.8 | 95.9 | 3.1 | 108 |
| Without BiLSTM (CNN Only) | 95.2 | 94.0 | 93.4 | 93.7 | 4.5 | 96 |
| Without CNN (BiLSTM Only) | 94.6 | 93.1 | 92.8 | 92.9 | 4.9 | 132 |
| Without Preprocessing | 94.2 | 92.5 | 91.8 | 92.1 | 4.6 | 208 |
| Without Attention & Preprocessing | 92.7 | 91.2 | 90.5 | 90.8 | 6.2 | 215 |

The results of the ablation clearly show that each component of the researchers' framework contributes to the overall system performance in meaningful ways. The full model has the highest accuracy (98.6%) and the lowest false-positive rate (1.8%), confirming the synergy of combining spatial, temporal, and attention-based learning with effective preprocessing. Removing the attention mechanism causes a noticeable accuracy degradation (-1.7%) and an almost doubled false-positive rate, showing that the attention mechanism is useful for prioritising high-risk temporal features. Removing the BiLSTM layer significantly reduces performance, underscoring the importance of capturing temporal dependencies for detecting multi-stage, slow-moving attacks. Similarly, not pre-processing a signal has a very significant effect on latency and false positives, which is a serious concern for noise reduction and feature normalisation in real-time environments. Overall, the ablation study approved the architectural design decisions and demonstrated the necessity of the proposed hybrid configuration for achieving high detection accuracy and operational efficiency in enterprise-scale intrusion detection systems.

### 4.4. Comparison with State-of-the-Art Methods

Table 4 presents a comparative evaluation of the proposed SCF-CNN-BiLSTM-Attention-based intrusion detection framework against state-of-the-art intrusion detection techniques. The comparison is based on 6 performance criteria: accuracy, F1-score, false-positive rate, detection latency, and scalability. The baseline methods are deep neural network-based IDS, CNN-based IDS, LSTM-based temporal models, hybrid CNN-LSTM approaches, and attention-based IDS. Table 4 highlights the effectiveness of the hybrid deep learning architecture coupled with attention mechanisms for real-time, large-scale cyber threat detection.

**Table 4:** Comparison of the proposed method with state-of-the-art intrusion detection techniques

| Method / Model | Core Technique | Accuracy (%) | F1-Score (%) | False Positive Rate (%) | Detection Latency (ms) | Scalability Support |
|---|---|---|---|---|---|---|
| SCF-CNN–BiLSTM–Attention (Proposed) | CNN + BiLSTM + Attention (Real-Time) | 98.6 | 98.0 | 1.8 | 115 | High |
| Deep Neural Network IDS | Deep Neural Network (DNN) | 92.4 | 91.8 | 6.8 | 320 | Medium |
| CNN-Based IDS | Convolutional Neural Network | 94.9 | 94.1 | 5.1 | 210 | Medium |
| LSTM-Based IDS | LSTM-Based Temporal Model | 95.6 | 95.0 | 4.6 | 185 | Medium–High |
| Hybrid CNN–LSTM IDS | CNN + LSTM | 96.8 | 96.1 | 3.7 | 160 | High |
| Attention-Based IDS | LSTM + Attention | 97.3 | 96.9 | 3.2 | 145 | High |

The results show that the proposed SCF-CNN-BiLSTM-Attention model outperforms all compared techniques across key performance metrics. It achieves the highest accuracy (98.6%) and F1-score (98.0%), the lowest false-positive rate (1.8%), and

the shortest detection latency (115 ms). The attention-based IDS is the closest competitor, but it is still not accurate or fast enough. Traditional DNN and CNN models have higher false alarm rates and are significantly slower than their counterparts. The proposed method also demonstrates better scalability, confirming its suitability for real-time, large-scale enterprise intrusion detection environments.

### 4.5. Statistical Analysis

Table 5 presents the results of a paired t-test assessing the statistical significance of performance differences between the proposed SCF-CNN-BiLSTM-Attention model and several state-of-the-art intrusion detection approaches. The comparison is based on the mean differences in accuracy across multiple experimental runs. For each pair, Table 5 shows the t-value, the corresponding p-value, and the statistical significance at the $\alpha = 0.05$ confidence level. The consistently low p-values ($p < 0.01$) indicate that the performance improvements achieved by the proposed method are not due to random variation and are statistically significant. This validates the reliability and robustness of the proposed model against existing IDS techniques.

**Table 5:** Paired t-test results comparing the proposed method with state-of-the-art models

| Comparison Method | Mean Accuracy Difference (%) | t-Value | p-Value | Statistical Significance ($\alpha = 0.05$) |
|---|---|---|---|---|
| Proposed vs DeepIDS | +6.2 | 9.14 | < 0.001 | Significant |
| Proposed vs CNN-IDS | +3.7 | 7.26 | < 0.001 | Significant |
| Proposed vs LSTM-IDS ( | +3.0 | 6.11 | < 0.001 | Significant |
| Proposed vs Hybrid CNN–LSTM | +1.8 | 4.02 | 0.002 | Significant |
| Proposed vs Attention-IDS ( | +1.3 | 3.47 | 0.004 | Significant |

The results of the paired t-test confirm that the proposed model's performance improvements are statistically significant across all comparisons. All p-values are well below the 0.05 threshold, indicating that the observed increases in accuracy are not random variation. A higher t-value for comparisons with previous deep learning models further demonstrates the tremendous improvement provided by the proposed architecture. This statistical validation enhances the credibility of the experimental findings and demonstrates the superiority of the proposed approach.

Despite the good performance of the proposed real-time deep learning intrusion detection framework, several limitations should be considered. First, although several benchmark datasets and enterprise-simulated traffic were used, the evaluation is still based on offline labelled datasets, which may not fully capture the diversity and unpredictability of live enterprise networks. Second, the model requires sufficient computational resources (e.g., GPU acceleration) for optimal real-time performance; this may not be possible in highly resource-constrained edge environments, in which case the model needs to be further optimised or compressed. Third, although the attention mechanism does increase interpretability to some degree, the system remains largely a black-box model and can't be explained fine-grainedly in a way that would be useful to security analysts. In addition, the current focus of the framework is on network-level intrusion detection, and it does not explicitly use host-based or user-behavioural signals, which might also improve detection coverage. Finally, the adaptive learning to address long-term concept drift is only partially covered, as the model lacks fully autonomous online retraining mechanisms.

## 5. Conclusion and Future Directions

This study introduced a new real-time deep learning engine for enterprise-scale intrusion detection and cyber risk management. By incorporating CNN-based feature extraction for spatial information, BiLSTM-based temporal modelling, and an attention mechanism into the streaming analytics framework, the presented system effectively addresses the limitations of traditional and existing deep learning-based IDS solutions. Extensive experimental evaluation showed superior detection accuracy, low false-positive rates, low latency, and good scalability compared with state-of-the-art methods. Statistical validation using k-fold cross-validation and paired t-tests was also used to assess the robustness and significance of the observed performance gains. Overall, the findings provide a basis for the feasibility of deep learning-driven, real-time intrusion detection systems that can serve as a reliable, practical foundation for the next generation of enterprise cybersecurity infrastructures. Future research can take this work in several promising directions. First, building online and ongoing learning mechanisms would allow the system to adapt to new attack types and long-term concept drift. Second, integrating explainable AI (XAI) techniques could improve transparency and trust among security stakeholders, for example, by providing actionable insights into model decisions. Third, it would be helpful to explore edge-aware model compression, pruning, and optimisation to facilitate deployment in edge and IoT environments with limited resources. Additionally, multi-modal data fusion, such as host logs, user behaviour analytics, and threat intelligence feed sources, could be used to enhance detection coverage further and introduce contextual awareness. Finally, deploying the pilots in real-world enterprise networks would yield more information on operational issues and help refine the system for large-scale production.

**Acknowledgment:** The authors would like to express their sincere gratitude to Srinivas University, Vels Institute of Science, Technology and Advanced Studies, and Bharath Institute of Higher Education and Research for their continuous support and encouragement in carrying out this research.

**Data Availability Statement:** The data for this study can be made available upon request to the corresponding author.

**Funding Statement:** This manuscript and research paper were prepared without any financial support or funding.

**Conflicts of Interest Statement:** The authors have no conflicts of interest to declare.

**Ethics and Consent Statement:** This research adheres to ethical guidelines, obtaining informed consent from all participants. Confidentiality measures were implemented to safeguard participant privacy.

**References**

1. M. A. I. Mallick and R. Nath, "Navigating the cyber security landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments," *World Scientific News*, vol. 190, no. 1, pp. 1–69, 2024.
2. S. D. D. Anton, M. Strufe, and H. D. Schotten, "Modern problems require modern solutions: Hybrid concepts for industrial intrusion detection," *in Proc. 24th ITG-Symp. Mobile Communication-Technologies and Applications*, Osnabrueck, Germany, 2019.
3. S. F. Aboelfotoh and N. A. Hikal, "A review of cyber-security measuring and assessment methods for modern enterprises," *JOIV: Int. J. Informatics Visualization*, vol. 3, no. 2, pp. 157–176, 2019.
4. R. Saini, D. Halder, and A. M. Baswade, "RIDS: Real-time intrusion detection system for WPA3 enabled enterprise networks," in *Proc. IEEE GLOBECOM*, Rio de Janeiro, Brazil, 2022.
5. P. Vanin, T. Newe, L. L. Dhirani, E. O. Connell, D. O. Shea, B. Lee, and M. Rao, "A study of network intrusion detection systems using artificial intelligence/machine learning," *Applied Sciences*, vol. 12, no. 22, p. 11752, 2022.
6. A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine learning and deep learning approaches for cybersecurity: A review," *IEEE Access*, vol. 10, no. 2, pp. 19572–19585, 2022.
7. I. A. Abdulmajeed and I. M. Husien, "MLIDS22-IDS design by applying hybrid CNN-LSTM model on mixed datasets," *Informatica*, vol. 46, no. 8, pp. 121-134, 2022.
8. S. P. Thirimanne, L. Jayawardana, L. Yasakethu, P. Liyanaarachchi, and C. Hewage, "Deep neural network based real-time intrusion detection system," *SN Computer Science*, vol. 3, no. 2, p. 145, 2022.
9. Y. C. Wang, Y. C. Houng, H. X. Chen, and S. M. Tseng, "Network anomaly intrusion detection based on deep learning approach," *Sensors*, vol. 23, no. 4, p. 2171, 2023.
10. E. U. H. Qazi, M. H. Faheem, and T. Zia, "HDLNIDS: Hybrid deep-learning-based network intrusion detection system," *Applied Sciences*, vol. 13, no. 8, p. 4921, 2023.
11. Q. Emad-ul-Haq, M. Imran, N. Haider, M. Shoaib, and I. Razzak, "An intelligent and efficient network intrusion detection system using deep learning," *Computers and Electrical Engineering*, vol. 99, no. 4, p. 107764, 2022.
12. S. V. N. S. Kumar, M. Selvi, and A. Kannan, "A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things," *Computational Intelligence and Neuroscience*, vol. 2023, no. 1, p. 8981988, 2023.
13. T. K. Chowdhury, "AI-powered deep learning models for real-time cybersecurity risk assessment in enterprise IT systems," *ASRC Procedia: Global Perspectives in Science and Scholarship*, vol. 1, no. 1, pp. 675–704, 2025.
14. R. S. Kommuri and M. Muppala, "Towards intelligent enterprises: Adoption of AI for cybersecurity management and risk governance," *4th International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India, 2025.
15. H. I. Kure, S. Islam, and H. Mouratidis, "An integrated cyber security risk management framework and risk prediction for critical infrastructure protection," *Neural Computing and Applications*, vol. 34, no. 18, pp. 15241–15271, 2022.
16. S. Mouti, S. K. Shukla, S. A. Althubiti, M. A. Ahmed, F. Alenezi, and M. Arumugam, "Cyber security risk management with attack detection frameworks using multi-connect variational auto-encoder with probabilistic Bayesian networks," *Computers and Electrical Engineering*, vol. 103, no. 10, p. 108308, 2022.
17. F. Al-Quayed, Z. Ahmad, and M. Humayun, "A situation-based predictive approach for cybersecurity intrusion detection and prevention using ML and DL Algorithms in Wireless Sensor Networks of Industry 4.0," *IEEE Access*, vol. 12, no. 3, pp. 34800–34819, 2024.
18. D. Clever, "Adversarial threat modelling for large-scale AI systems in enterprise networks," *Complexity*, vol. 3, no. 7, p. 17, 2025.
19. I. Hamid and M. M. H. Rahman, "AI, machine learning and deep learning in cyber risk management," *Discover Sustainability*, vol. 6, no. 1, p. 389, 2025.
20. S. Seker, "Network intrusion dataset," *Kaggle*, 2024. [Accessed by 02/12/2024].

21. L. D. Hooge, "UNSW-NB15 dataset," *Kaggle*, 2023. [Accessed by 12/12/2024].
22. SolarMainframe, "IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018)," *Kaggle*, 2019. [Accessed by 30/12/2024].
23. T. Radivilova, L. Kirichenko, A. S. Alghawli, D. Ageyev, O. Mulesa, O. Baranovskyi, A. Ilkov, V. Kulbachnyi, and O. Bondarenko, "Statistical and signature analysis methods of intrusion detection," in Information Security Technologies in the Decentralized Distributed Networks, *Springer*, Cham, Switzerland, 2022.
24. A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmed, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, no. 9, pp. 99837–99849, 2022.
25. C. Zhang, D. Jia, L. Wang, W. Wang, F. Liu, and A. Yang, "Comparative research on network intrusion detection methods based on machine learning," *Computers & Security*, vol. 121, no. 10, p. 102861, 2022.
26. M. S. Yadav and R. Kalpana, "Recurrent nonsymmetric deep autoencoder approach for network intrusion detection system," *Measurement: Sensors*, vol. 24, no. 12, p. 100527, 2022.
27. S. Wang, Z. Zhang, W. Li, C. Yin, Y. Ma, and W. Xu, "Dynamic residual graph attention network for network intrusion detection system," *Sixth International Conference on Next Generation Data-driven Networks (NGDN)*, Shenyang, China, 2024.