Data and Metadata. 2026; 5:1329
doi: 10.56294/dm20261329

ORIGINAL

AI-Driven Hybrid Defense Mechanisms for Enhancing Cybersecurity in Cyber-Physical Systems Through Packet Sniffing and Cyber Ranges

Mecanismos de Defensa Híbridos Impulsados por IA para Mejorar la Ciberseguridad en Sistemas Ciberfísicos mediante Sniffing de Paquetes y Ciber-Rangos

Deepa Singh Sisodiya[1] ✉, Ritu Tiwari[1], Priyank Jain[1]

[1]Department of Computer Science and Engineering. IIIT-Pune, India.

Cite as: Sisodiya DS, Tiwari R, Jain P. AI-Driven Hybrid Defense Mechanisms for Enhancing Cybersecurity in Cyber-Physical Systems Through Packet Sniffing and Cyber Ranges. Data and Metadata. 2026; 5:1329. https://doi.org/10.56294/dm20261329

Submitted: 19-08-2025 **Revised:** 24-09-2025 **Accepted:** 10-12-2025 **Published:** 01-01-2026

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding Author: Deepa Singh Sisodiya ✉

ABSTRACT

Introduction: Cyber-Physical Systems are the backbone of modern critical infrastructures but remain inherently vulnerable to cyberattacks due to their interconnected nature. This calls for more adaptive and intelligent intrusion detection solutions, as existing approaches often fall short in capturing the spatial-temporal complexity of CPS traffic.
Method: this work proposes a hybrid deep learning framework based on the integration of CNN and LSTM networks with an attention mechanism. The system exploits real-time packet sniffing for fine-grained traffic analysis and the use of cyber range simulations to evaluate its performance in different attack conditions. A structured preprocessing pipeline, covering normalization, time windowing, and controlled data augmentation, ensures high-quality feature extraction while maintaining spatial and temporal patterns.
Results: the proposed model outperforms standalone CNN and LSTM architectures on a balanced multi-class CPS dataset with 99,08 % accuracy and very high precision, recall, and F1-scores across all attack types. Attention significantly enhances sensitivity by picking up important temporal features and provides better interpretability via packet-level relevance mapping. The model maintains an extremely low false-positive rate, further supporting its suitability for real-world deployment.
Conclusions: these results position the hybrid CNN-LSTM-Attention architecture, combined with packet sniffing, as a robust and adaptive intrusion detection for CPS environments. Strong performance with low error rates accordingly underlines the potential to mitigate emerging threats. Future work will extend the evaluation to diverse datasets and will benchmark the system against state-of-the-art detection models in order to further validate generalizability.

Keywords: Cyber-Physical Systems; Intrusion Detection; Hybrid CNN-LSTM; Attention Mechanism; Packet Sniffing; Real-Time Detection.

RESUMEN

Introducción: los sistemas ciberfísicos son la columna vertebral de las infraestructuras críticas modernas, pero siguen siendo inherentemente vulnerables a los ciberataques debido a su naturaleza interconectada. Esto exige soluciones de detección de intrusiones más adaptativas e inteligentes, ya que los enfoques existentes a menudo no logran captar la complejidad espacio-temporal del tráfico de CPS.
Método: este trabajo propone un marco híbrido de aprendizaje profundo basado en la integración de redes CNN y LSTM con un mecanismo de atención. El sistema aprovecha el rastreo de paquetes en tiempo real para un análisis detallado del tráfico y el uso de simulaciones de ciberalcance para evaluar su rendimiento en diferentes condiciones de ataque. Un proceso de preprocesamiento estructurado, que abarca la normalización, el ventanamiento temporal y el aumento controlado de datos, garantiza una extracción de

ISSN: 2953-4917

características de alta calidad, manteniendo al mismo tiempo los patrones espaciales y temporales. **Resultados:** el modelo propuesto supera a las arquitecturas independientes CNN y LSTM en un conjunto de datos CPS multiclase balanceado, con una precisión del 99,08 % y una precisión, recuperación y puntuaciones F1 muy altas en todos los tipos de ataque. La atención mejora significativamente la sensibilidad al detectar características temporales importantes y proporciona una mejor interpretabilidad mediante el mapeo de relevancia a nivel de paquete. El modelo mantiene una tasa de falsos positivos extremadamente baja, lo que respalda aún más su idoneidad para la implementación en el mundo real.
Conclusiones: estos resultados posicionan la arquitectura híbrida CNN-LSTM-Atención, combinada con el rastreo de paquetes, como una detección de intrusiones robusta y adaptativa para entornos CPS. Su excelente rendimiento con bajas tasas de error subraya, por consiguiente, el potencial para mitigar amenazas emergentes. El trabajo futuro ampliará la evaluación a diversos conjuntos de datos y comparará el sistema con modelos de detección de vanguardia para validar aún más su generalización.

Palabras clave: Sistemas Ciberfísicos; Detección de Intrusiones; Modelo Híbrido CNN-LSTM; Mecanismo de Atención; *Packet Sniffing*; Detección en Tiempo Real.

INTRODUCTION

The critical infrastructure has critically relied on cyber-physical systems that integrate physical, networking, and computing elements in diverse domains.[1] However, networked systems have been critiqued for the significant challenges, as attacks can severely hamper human safety and physical assets.[2] The complexity and scale of CPS make traditional security measures inappropriate and require better AI-driven solutions to enhance the Emerging threat landscape.

[3] Packet sniffing is an integral part of this hybrid defence mechanism, allowing real-time network traffic surveillance for anomalies in search of patterns that might indicate attack activities.[4] A.I. models trained on this data can identify new threats, thus providing an adaptive defence system that Evolves with the threat landscape. More than the classic intrusion detection systems (IDS), it is focused attention: the mechanism increases the model's ability to be more precise and responsive by allowing it to zoom in on crucial elements of network traffic data.

Cyber ranges are simulated versions of real cyberattacks developed to offer a process for testing and enhancing AI-driven defence models.[5] These controlled environments allow researchers to test hybrid CNN-LSTM models augmented with attention mechanisms on how well they detect DDoS assaults and APTs.[6] Training in cyber range scenarios has enhanced the robustness of AI-based models to novel and complex attacks.

That is why hybrid models combining LSTMs and CNNs are used when identifying and countering sophisticated cyberattacks in CPS scenarios; an application and reputation are gained.[7] LSTMs are good at learning temporal connections in sequential data, but network traffic is a set of high-dimensional data for which the CNN excels in locating spatial patterns. Hybrid CNNLSTM models provide an all-embracing approach in terms of threat detection.[8] By combining these architectures with an attention mechanism, they can better capture the temporal and spatial features of network data.

This paper examines the use of this sophisticated hybrid model in CPS scenarios. It underscores the significance of combining simulated attack scenarios with real-time traffic monitoring to sustain cybersecurity defences. A viable strategy for enhancing CPS cybersecurity is virtual cyber ranges using a mix of the architectures of CNNs and LSTMs, attention mechanisms, and packet-sniffing methods. This dynamic and adaptive defensive system addresses contemporary cyber threats' geographical and temporal diversity.

Recent research further reinforces the requirement for hybrid intrusion detection solutions using advanced approaches in the environment of CPS.[9] Research on Medical CPS enumerates how deep learning with blockchain would strengthen data protection and reduce false positives.[10] Similarly, proactive anomaly detection approaches for smart grids indicate how these systems are very important and underplayed for identifying pre-attack behaviors using unsupervised learning.[11] Adversarial machine-learning-related studies such as ConAML have brought forth that CPS models have to be resilient when an attacker crafts physically constrained adversarial samples.[12] Surveys related to industrial CPS security have also shown the ever-growing complexities of DoS and deception attacks along with the inefficiency of traditional IDS approaches.[13] Thus, new emerging frameworks such as SAD-GAN also prove that adaptive and self-learning models would be highly essential for reliable real-time anomaly detection.[14,15] Furthermore, the increasing usage of packet sniffers, network forensic tools, and FPGA-based inspection systems points out the critical role of real-time traffic acquisition in detecting sophisticated attacks. It is well established from these studies that integration of spatial, temporal, and behavioral features is paramount, further establishing the relevance of our proposed hybrid CNN–LSTM–Attention model coupled with packet sniffing and cyber-range simulation for robust CPS defense.

Motivation, Challenges, Objectives

The increasing integration of Cyber-Physical Systems (CPS) in critical infrastructure amplifies the risk of targeted cyberattacks, motivating the need for intrusion detection solutions that are both accurate and adaptable to evolving threat landscapes. Traditional signature-based methods struggle to detect zero-day attacks, while many machine learning approaches fail to capture the combined spatial and temporal dynamics of CPS traffic. Key challenges addressed in this work include handling high-dimensional, multi-modal network and system performance data; maintaining real-time detection capability without excessive computational cost; and ensuring robust generalization to unseen attack patterns. The primary objectives of this study are to design a hybrid CNN–LSTM model augmented with an attention mechanism, to implement a packet-sniffing-based data acquisition pipeline, and to evaluate the proposed framework against established baselines in both controlled and realistic scenarios.

Background

This study builds upon four core concepts in modern AI-driven cybersecurity: Convolutional Neural Networks (CNNs) for spatial feature extraction, Long Short-Term Memory (LSTM) networks for capturing temporal dependencies, packet sniffing for real-time network traffic monitoring, and attention mechanisms for dynamically focusing on the most relevant features. Together, these methods form the technical foundation for our hybrid architecture, which is later explored in detail in the following subsections.

CNN (Convolutional Neural Network) Model:

Due to learning of CNN, its popularity among academics has risen, pushing them to push through complex issues they had previously given up on. In recent years, researchers in several fields have developed alternative convolutional neural network (CNN) architectures to address various problems, including identifying deepfakes. The overarching structure of (CNN). as depicted in figure 3, is sometimes constructed using many successively layered layers. Convolutional layers in a convolutional neural network (CNN) architecture are used for feature extraction while pooling layers are used to reduce the image Report:[16] CSO Report was generated on Monday, Oct 21, 2024, 05:32 PM Page 9 of 34 dimensions. In the second place, it comprises a module combined with a fully connected (F.C.) layer to classify a picture.

This section explains the technique we've developed to identify Deepfakes, as shown in. Using crucial facial features and pixel distortions in the images, a CNN model can distinguish and generate a probability for the image's authenticity or fake.[17]

The convolutional neural network CNN) is designed to identify essential features (lines, curves, etc.) before progressing to more complex patterns (faces, items, etc.). CNNs and regular ANNs have distinct architectures. The former uses two-dimensional layers interconnected with all neurones in the preceding layer. In contrast, the latter (CNN) uses three-dimensional layers coupled in depth, width, and height.[18] The neurons in Instead of connecting directly to each other, convolutional neural networks (CNNs) use a portion of the neurons from the layer below to form connections between each layer.[19] CNN, which has 50 convolutional layers that were learned on ImageNet, is the technique we utilized. Even with the vanishing gradient issue, the highly competent ResNet model can still achieve remarkable result.

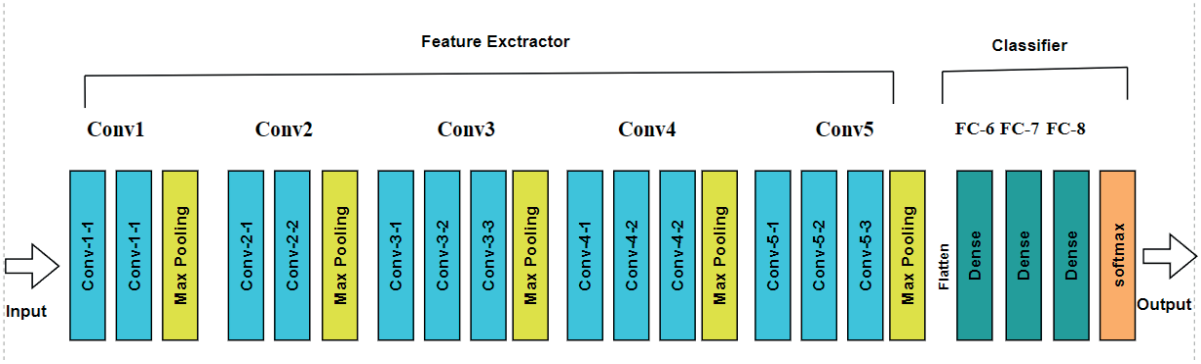

Figure 1. CNN Architectural Diagram

LSTM (Long Short-Term Memory):

Conventional recurrent neural networks sometimes experience disappearing or growing gradients while processing long sequences, particularly when maintaining dependencies.[20] The LSTM In 1997, Hochreiter and Schmid Huber created the approach.to solve this problem. Because it handles sequential data, LSTM, a specialized RNN architecture, models long-term dependencies better than regular RNNs. LSTM is used in voice

recognition, video analysis, natural language processing, and more because of its remarkable capabilities. Report: CSO Report was generated on Monday, Oct 21, 2024, 05:32 PM Page 10 of 34 Unlike standard RNNs, long-term memory networks (LSTM) use gate mechanisms to selectively absorb data from neurons at various temporal intervals, increasing long-term sequence understanding and prediction. Three gates—the input, the forget, and the output—keep the inputs to LSTMs in check. These gate designs can efficiently process utility data while irrelevant data is blocked. Thus, LSTM retains important data and discards irrelevant. Figure 1 illustrates LSTM memory cell assembly. A forget gate is in charge of deleting specific data from the preceding long-term memory cell It determines how to modify and delete records based on the input node aZ and the previously hidden state bt−1bt−1. The following are the pertinent calculation formulas:

$$f_t = \sigma\left(W_f . [H_{t-1}, X_t] + b_f\right)$$

σ stands for the forget gate, and Zt for the sigmoid activation function. During time step t − 1, the input is denoted as the short-term memory output is depicted as bZ−1, the forget gate weight matrix is denoted as Yf, and the bias term is denoted as bf. The input gate makes the decision to add fresh data to the memory using the network's input nodes YZ and Tt−1. The relevant calculations are as follows:

Formulas

$$i_t = \sigma(W_i . [H_{t-1}, X_t] + b_i)$$
$$\overline{C}_t = tanh(W_c . [H_{t-1}, X_t] + b_c)$$
$$C_t = f_t \times C_{t-1} + i_t \times \overline{C}t$$

Where the hyperbolic tangent activation of the element is expressed by tanh, The equation represents the input gate as iZ, the candidate vector as $\mathbf{O}taZ$, bc and bi represent the long-term memory cell at time step t, where $YiYi$ and Yc are the weight coefficients for the input gate and the candidate vector, respectively

$$ot = \sigma(W_o . [Ht - 1, Xt] + bo)$$
$$Ht = ot \times tanh(Ct)$$

(21) Here, it stands for the output gate, bo for its bias term, Wt for its weight matrix, and Ht represents the model's output at step t. Left Short-Term Memory (LSTM) neural networks outperform traditional recurrent neural networks at Report:(22) CSO Report was generated on Monday, Oct 21, 2024, 05:32 PM Page 11 of 34 processing lengthy sequential data. Due to their flexible architecture, LSTM can efficiently address time series data processing challenges.

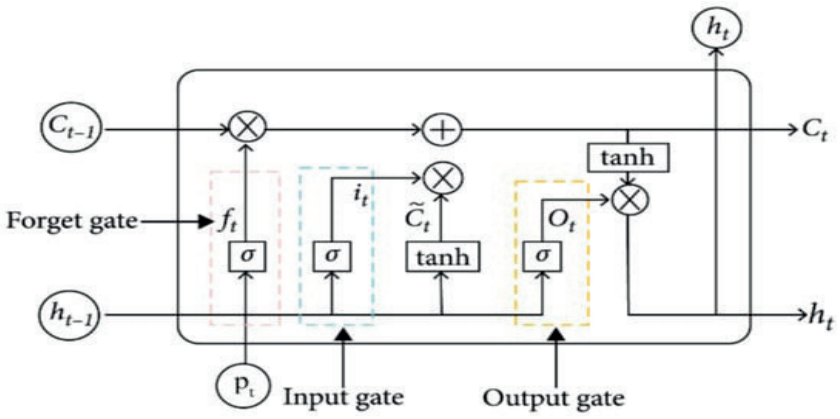

Figure 2. LSTM neural network structure diagram

Packets Sniffing

Packets Sniffing entails catching information packets as they travel across interconnected computers. What wiretapping is to telephone networks, packet sniffing is to computer networks. Packet sniffers, which are network-connected devices used to listen to network traffic, are used to do this. Administrators can locate bottlenecks and ensure efficient data transit over networks by correlating the data collected by packet sniffers

with potentially harmful communications. However, considering, hackers and crackers also frequently utilize it to obtain information about networks they wish to breach unlawfully. Using packet sniffers, you may intercept information that could aid an attacker in breaking into your network, including I.P. addresses, passwords, and protocols. The main uses of packet sniffing include eavesdropping, network management, intrusion detection, and packet sniffing.[23] The most popular packet-sniffing software is password-sniffing software.

Packet sniffing is the process of listening to every packet that travels over the network; it is a means by which a user listens to the information of others. Packet sniffers may serve both criminal and legitimate administrative purposes.[24] The user's goal determines this. Network administrators use them to monitor and confirm network traffic.

Principle of Packet Sniffing

Packets go via several intermediary devices on their journey from source to destination. An interface card receives all network traffic for a network that is configured to operate in promiscuous mode. A network interface card (NIC) with Report: CSO Report was generated on Monday, Oct 21, 2024, 05:32 PM Page 12 of 34 a unique physical address identifies it from all other networks. When a packet reaches Its hardware address, it corresponds to the NIC's physical address, the network interface controller. But if the network interface card is set to promiscuous mode, then every packet will go via that interface. They developed a system to gather all network data via a switch that has previously passed filtered data.[25] Packets are transferred to driver memory when NICs accept them, after which they are sent to the kernel and finally to the user program.

Packet Sniffing Work Process

The subsequent procedures are used to carry out the packet sniffing.

- The packet sniffer collects the raw binary data from the network connection, which might be wired or wireless.
- The process of converting binary data collected into a readable format so that the methods and data content may be understood.
- Examination of the converted and recorded data to discover and investigate the parameters of the procedures used.

Every device in the network is uniquely recognized by its NIC physical address. All of the network computers get the packet that the device is transmitting. All machines connected to a network can view traffic under the shared Ethernet concept, but they cannot respond if it does not belong to them. The gadget can monitor all segment traffic while the NIC is in a promiscuous condition. On the other hand, when a network interface card (NIC) is configured in a promiscuous state for a single computer, the NIC collects and records all network structures and packets, even if they are not intended for that machine. This function is known as sniffing.[26] The sniffer starts to examine every data entered into the machine via NIC.

Attention Mechanism

Human attention is based on grasping more important information with less energy.[27] This study included A focused method to allocate varying weight factors to LSTM-extracted attributes, enhancing model prediction. Figure 2 shows the Attention structure. The Self-Attention module utilizes standard weights and biases on the LSTM's time-dependent feature vectors. Following the evaluation of each feature, a weighted summation method was used for in-depth feature extraction. The calculation formula isRelevance Score Computation:

$$\beta_i = \sigma(W_i h_t + b_i)$$

Where ßi represents the relevance score for the concealed state, the bias vector is represented by bi,the weight vector by Wi, and the weight matrix by hi.

Attention Weights Calculation

$$a_i = soft\,max(\beta_i) = \frac{exp(\beta_i)}{\sum i\, exp(\beta_i)}$$

The attention weight given to the concealed state hi is represented by ai.

Weighted Sum of Hidden States

$$O = H \otimes a_i$$

In this case, the output prediction is represented by O, which is all hidden states added together, with the weight of each state dictating how much of a contribution it makes.

Query, Key, and Value Computation

Query, key, and value vector computation typically occur inside the attention mechanism. These vectors are created from each hidden state, hi:

$q_i=W_Q \cdot h_i,\ k_i=W_K \cdot h_i,\ vi=W_V \cdot h_i$

With qi, ki, and vi standing for the query, key, and value vectors for the i-the time step, and the corresponding weight matrices for the key, value, and associated variables, respectively, and query transformations are represented by $W\ Q$, W K , and WV.

Scaled Dot-Product Attention:

To get the attention ratings, scale the key vectors' square root of their dimensionality (d k). The dimensionality is found by multiplying the query and key vectors by themselves. Using a hard max is then used to produce normalized attention weights:

$$\propto_{ij}=\frac{\exp\left(\frac{q_i . k_j}{\sqrt{d_k}}\right)}{\sum_{j=1}^{t}\exp\left(\frac{q_i . k_j}{\sqrt{d_k}}\right)}$$

And the represented by the attention score between the query vector (qi) and key vector (k j). by $\propto_{ij}$.

Weighted Sum of Values

Lastly, the attention mechanism's output is generated by adding up the value vectors vj, where the attention scores αij determine the weights.

$$O_i=\sum_{j=1}^{t}\propto_{ij} . v_j$$

The output for the i-th time step, denoted as Oi is a weighted sum of the values of vj across all time steps.

Final Prediction

$$O=\sum_{i=1}^{t} a_i . h_i$$

Where the hidden state at time step *i* is represented by h_i and the attention weight by a_i.

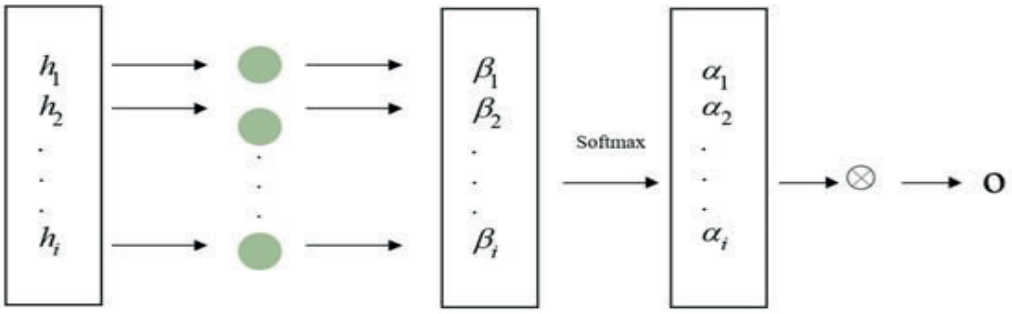

Figure 3. Attention network structure diagram

The activation function is represented by βi, the feature's relevance is defined by ai, O represents the attention weight, and Wi, bi represent the output prediction result, and the weight matrix and bias vector between neuron nodes σ, respectively.

METHOD

This research introduces a comprehensive methodology for identifying cyber threats in cyber-physical systems (CPS) with a hybrid CNN-LSTM model integrated with an attention mechanism. The method incorporates real-time packet sniffing to seize unprocessed network traffic, facilitating microsecond-level danger identification. The methodology encompasses data collecting, preprocessing, model construction, and evaluation, guaranteeing great precision in recognizing various assault patterns.

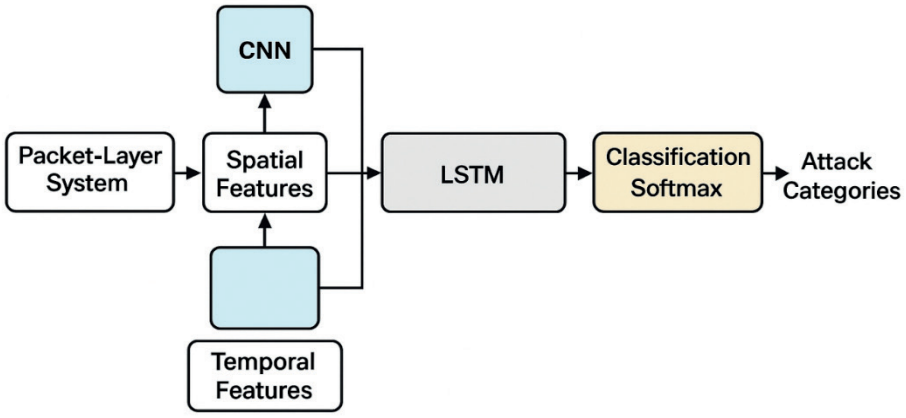

Figure 4. Proposed Hybrid CNN-LSTM-Attention Architecture for CPS Intrusion Detection

The proposed framework integrates three key components to enhance both detection accuracy and interpretability. A Convolutional Neural Network (CNN) module performs spatial feature extraction from packet-level and system performance data, capturing local correlations and structural patterns. These spatial features are then fed into a Long Short-Term Memory (LSTM) network, which models temporal dependencies across sequential data windows, enabling detection of long-range attack patterns. An attention mechanism is applied to the LSTM outputs to assign dynamic weights to the most relevant time steps, improving focus on critical behavioural indicators while reducing noise from less significant features. Finally, a fully connected layer with a softmax activation performs multi-class classification across the predefined attack categories. This architecture is designed to operate in real time, with a packet sniffing data acquisition pipeline feeding directly into the preprocessing and feature extraction stages.

Data Collection

The dataset has numerous essential columns encompass various system and network performance parameters. CPU utilization is specified in terms of 1 Report: CSO Report was generated on Monday, Oct 21, 2024, 05:32 PM Page 15 of 34 overall utilization and each core (CPU_Total,CPU_P1,CPU_P2,CPU_P3,CPU_P4), offering insights into the distribution of processing demand. Disc activity is indicated by Disk_wrVolume (total data written to the disc) and Disk_wraccessVolume (number of discs write accesses). Load averages for 1, 5, and 15 minutes are recorded by Load_1min,Load_5min,and Load_15min, reflecting workload patterns. Memory statistics encompass Mem_Avail (available memory) and Mem_freeTotal (total free memory). Network performance is assessed by Ping_Time(ms) (average round-trip duration), Ping_Min(ms), Ping_Max(ms), and Ping_PacketLoss (% of packets dropped). Furthermore, Traffic_inVolume and Traffic_outVolume monitor the volume of incoming and exiting network traffic, indicating bandwidth utilization. The Class column denotes several network behaviours, including 'TVMwL', 'NTSwL', 'NTS', 'TWMwL', 'TWM', 'WSKwL', 'WSK', 'TSKwL', and 'TSK', functioning as the target variable for classification tasks.

Data Preprocessing

To prepare the collected CPS network and system performance data for optimal model training, a structured preprocessing pipeline was implemented. This involved cleaning the dataset to ensure integrity, engineering relevant features, normalizing variable scales, reshaping data for compatibility with the CNN-LSTM architecture, encoding class labels for multi-class classification, applying controlled data augmentation to improve robustness, and performing a carefully designed train-validation-test split to maintain temporal integrity and class balance. The following subsections detail each step in this process.

Data Cleaning

The dataset was meticulously cleaned to guarantee data quality prior to model training. Initially, we analyzed the dataset for absent values utilizing the Pandas isnull() function, discovering no null entries in any features. All network traffic measurements were validated for data integrity, with any damaged packets automatically excluded during the initial collection procedure. The dataset upheld uniform timestamp intervals (1-second granularity) for all observed characteristics, including CPU use, memory metrics, and network traffic statistics.

Feature Engineering

The study utilized all 18 original features capturing system performance metrics (CPU, memory, disk, network) without dimensionality reduction, as each provided unique behavioral signatures. Packet-level features like length and protocol were extracted from network traffic. No synthetic features were created to maintain the integrity of real-world system measurements while ensuring computational efficiency for real-time detection.

Normalisation

We applied StandardScaler from scikit-learn to normalize all numerical features, transforming them to zero mean and unit variance. This preprocessing step was crucial given the varying scales of our metrics - from percentage-based CPU utilization (0-100) to byte-count network traffic volumes (potentially millions). [28] The scaler was fit exclusively on the training data partition to prevent information leakage, with the same transformation parameters subsequently applied to validation and test sets.

Reshaping Data for CNN Input

To ensure interoperability with our CNN-LSTM architecture, we reformatted the input data into a 3D tensor with dimensions (samples, timesteps, features). Each sample was organized as a 30-second sliding window encompassing all 18 features, resulting in input dimensions of (samples, features, 1). This structure enabled the convolutional layers to extract spatial patterns within each time window, while the LSTM managed temporal dependencies across sequential windows.

Class Label Encoding

The target variable 'Class', comprising nine attack categories (such as 'TVMwL', 'NTSwL', etc.), was encoded utilizing one-hot encoding through Keras' to_categorical() function. This transformation transformed category labels into binary vectors appropriate for multi-class classification.[29] We preserved the original class distribution without artificial balancing to maintain the authentic incidence ratios of various assault types.

Data Augmentation and Validation Strategy

Although the dataset used in this study is inherently balanced in terms of class representation, we applied controlled data augmentation to improve the model's robustness to unseen variations. Specifically, synthetic noise injection (Gaussian noise with $\mu = 0$, $\sigma = 0{,}01$) and minor scaling transformations (±5 % in numerical values) were used to mimic sensor jitter and natural system fluctuations. These augmentations preserved the statistical integrity of the original features while increasing diversity in training samples.

Train-Test Split

The pre-processed data was divided into training (80 %) and test sets (20 %) by stratified sampling to preserve uniform class distributions throughout the subsets. The validation set was utilized for hyperparameter optimization and early termination, whereas the test set offered an entirely novel assessment subset. Temporal sequencing was maintained during partitioning to avert future information leaking in our time-series data.

Robustness Evaluation via Cross-Validation and False Positive Analysis

To further assess the robustness and generalization capability of the proposed model, we employed 10-fold cross-validation on the training dataset. In this approach, the dataset was partitioned into 10 equal subsets, with each subset serving as the validation set once while the remaining nine subsets formed the training set. This process ensured that the model's performance was evaluated across multiple data splits, reducing bias due to a particular train-test partition.

In addition, we analysed the False Positive Rate (FPR) for each fold and across attack categories. The FPR was computed as:

$$FPR = \frac{False\ Positive}{False\ Positive + True\ Negative}$$

This metric is critical in intrusion detection, as an elevated FPR can lead to unnecessary alerts and resource wastage in operational systems. Our analysis indicated consistently low FPR values (≤ 2,5 %) across folds, confirming the model's resilience to misclassifying benign traffic as malicious. The combined use of K-fold cross-validation and FPR evaluation reinforces confidence in the model's stability against varying data distributions and diverse attack patterns.

Model Building

Cyber-physical systems (CPS) demand robust security models capable of detecting sophisticated cyber threats in real-time. Our proposed hybrid CNN-LSTM model with attention mechanism addresses this challenge through a multi-layered architecture that combines spatial feature extraction, temporal pattern recognition, and dynamic feature weighting. The model was specifically designed to process raw network packet data obtained through packet sniffing, enabling microsecond-level threat detection critical for CPS environments.

Hybrid CNN-LSTM Architecture with Packet Sniffing Integration

Our cybersecurity framework implements a novel hybrid CNN-LSTM model that directly processes raw network packets captured through real-time sniffing. The architecture uniquely combines packet-level feature extraction with deep learning, preserving critical attack signatures often lost in flow-based analysis. The system employs a multi-layer processing pipeline that begins with physical packet capture and preprocessing of live network traffic. It then applies protocol-specific normalization for industrial control protocols like Modbus/TCP and DNP3 using custom parsers, before generating optimized feature vectors for the deep learning components. This integration of packet sniffing at the foundational level enables the model to maintain the fidelity of network attack indicators while providing the structured input needed for effective machine learning.

Convolutional Neural Network architecture for Packet-Level Feature Extraction

The CNN module specializes in spatial pattern recognition from network packets through 1D convolutional layers with 64 filters and kernel size 3, which systematically scan packet headers and payloads. The protocol-aware architecture incorporates custom filter banks specifically designed for industrial control system protocols, enabling detection of both generic and domain-specific threats. The hierarchical feature learning process occurs across three progressive layers: the first detects basic packet anomalies like malformed headers, the second identifies complex attack signatures in payload patterns, and the third recognizes protocol-specific threats such as unauthorized function codes.[30] The CNN outputs are carefully reshaped to preserve packet sequence integrity while preparing the data for temporal analysis in subsequent layers.

LSTM Network for Temporal Attack Pattern Recognition

The LSTM network extends the model's capability by processing packet sequences to detect sophisticated, time-distributed attacks. It implements bidirectional processing to analyze packet flows in both directions, capturing comprehensive temporal relationships. The network's adaptive memory gates - including an input gate that weights new packet information, a forget gate that discards irrelevant network noise, and an output gate that controls threat signal propagation - work in concert to maintain focus on genuine threats.[31] The module maintains temporal context across 64-packet sequences, allowing it to identify multi-packet attack patterns that would be invisible to static analysis methods.

Attention Mechanism for Dynamic Threat Scoring

Our attention layer significantly enhances the model's threat detection capabilities through packet-level attention weights that score each packet's threat relevance on a 0-1 scale. This mechanism dynamically amplifies suspicious packets while suppressing normal operational traffic, dramatically improving the signal-to-noise ratio in detection. The attention system also provides valuable visual explainability by generating packet-level heatmaps that highlight attack progression markers and identify critical attack trigger packets.[32] This dual functionality of both improving detection accuracy and providing interpretable results makes the attention mechanism particularly valuable for security operations in CPS environments.

Model Evaluation

Several techniques are needed to assess machine learning algorithms. Metrics measure performance. Research suggests numerous algorithm performance measures. We require accurate metrics for each machine learning job to evaluate performance. We compare algorithms and collect performance statistics using various standard categorization measures. Model generalisability is measured by its ability to use new data. Specificity, sensitivity, memory, accuracy, precision, and F1 score affect categorization. Machine learning classifier accuracy, recall, precision, and F1 score were examined. Multiple estimations are compared to assure accuracy.[33] This illustrates confusion matrices may disclose much. Evaluate development and implementation.

Accuracy

Accuracy is the ratio of precisely labelled instances (including both from the overall count of occurrences in the sample to both true positives and true negatives. It is a measure of the overall accuracy of the model's predictions. It is possible to determine using the following formula:
sensitivity, memory, accuracy, precision, and F1 score affect categorization. Machine learning classifier accuracy, recall, precision, and F1 score were examined. Multiple estimations are compared to assure accuracy.[33] This illustrates confusion matrices may disclose much. Evaluate development and implementation.

$$Accuarcy = \frac{TP + TN}{S}$$

Where:

S=TP+TN+FP+FN

Precision: Precision is calculated by subtracting one from a ratio, specifically (1 - precise), representing the fraction of false negatives. Recall, on the other hand, is obtained by dividing precision by one.

$$Pr\,e\,cision = \frac{TP}{TP + FP}$$

Recall: conversely, there are entities called false negatives about true negatives.

$$Re\,c\,all = \frac{TP}{TP + FN}$$

F1-Score: regarding this matter, the calculation involves squaring the accuracy and recall scores.

$$F1 = \frac{2 * Pr\,e\,cision * Re\,c\,all}{Pr\,e\,cision + Re\,c\,all}$$

Model Algorithm

Algorithm 1 Hybrid CNN-LSTM Cyber Threat Detection Pipeline with 10- Fold Validation
1: Input:
2: - Raw network packets with timestamp
3: - System metrics (CPU, Memory, Disk, Network)
4: Output:
5: - Attack classification (TVMwL, NTSwL, ..., TSK)
6: - Threat confidence score
7: Phase 1: Data Preprocessing
8: 1. Collect raw packets and system performance metrics
9: 2. Remove corrupted packets and validate data integrity
10: 3. Normalize all numerical features to [0, 1] range
11: 4. Reshape data into 30-second sequential windows
12: 5. One-hot encode the 9 attack class labels
13: 6. Prepare data for k-fold validation (shuffle and partition)
14: Phase 2: Model Architecture
15: 1. CNN Module:
16: a. Apply three 1D convolutional layers (64 filters)
17: b. Detect: packet anomalies → payload patterns → protocol-specific threats
18: c. Apply max-pooling after each layer
19: 2. LSTM Module:
20: a. Process CNN outputs with bidirectional LSTM (64 units)
21: b. Maintain memory gates for temporal dependencies
22: 3. Attention Mechanism:
23: a. Compute importance weights for each time step

24: b. Generate weighted threat representation
25: 4. Output Layer:
26: a. Softmax classification for 9 attack types
27: b. Confidence score generation
28: Phase 3: Evaluation with 10-Fold Cross-Validation
29: for k = 1 to 10 do
30: a. Set fold k as validation set
31: b. Use remaining 9 folds for training
32: c. Train model for 100 epochs with early stopping
33: d. Store validation metrics for fold k
34: end for
35: 2. Aggregate results across all folds:
36: a. Calculate mean accuracy ± standard deviation
37: b. Compute macro-average precision/recall
38: c. Generate confusion matrix from all fold predictions
39: 3. Final Model Training:
40: a. Train on complete training set (all 10 folds)
41: b. Evaluate on held-out test set (10 % of original data)
42: c. Calculate:
43: i. Test accuracy
44: ii. Class-wise F1-scores

RESULTS

Confusion Matrix

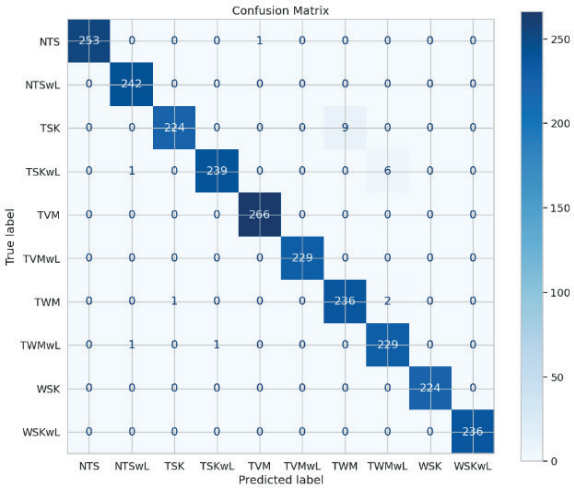

Figure 5. CNN+LSTM+Attention integrated PS Model Confusion Matrix

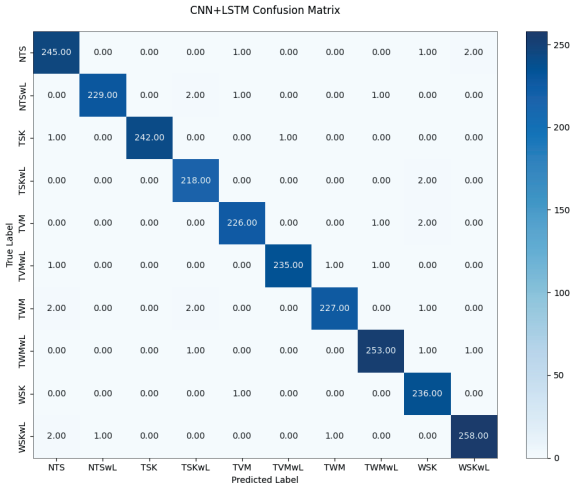

Figure 6. CNN+LSTM model Confusion Matrix

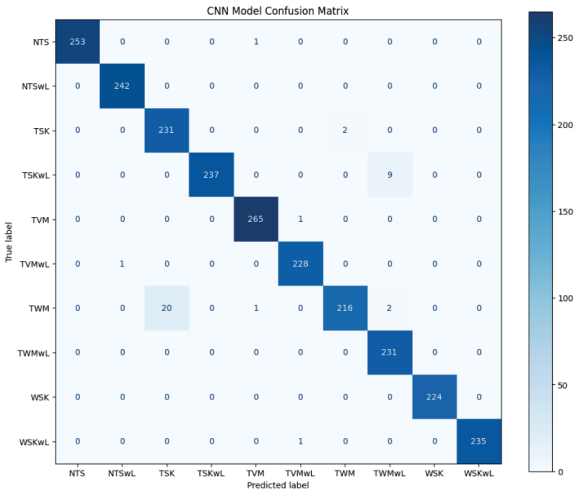

Figure 7. CNN model Confusion Matrix

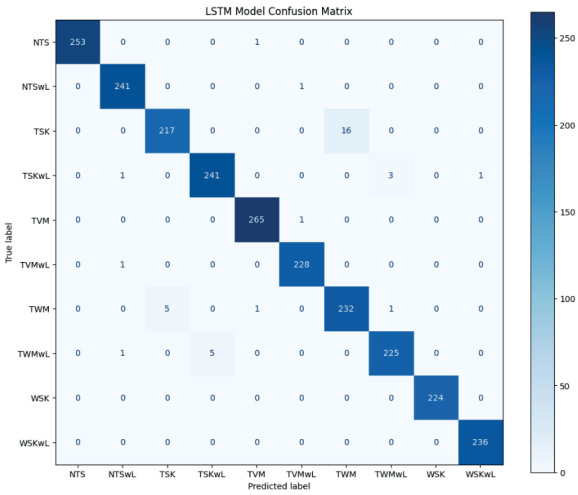

Figure 8. LSTM Model Confusion Matrix

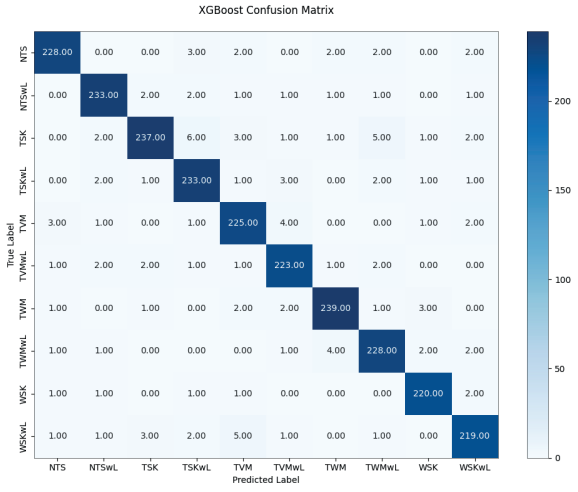

Figure 9. XGBoost model Confusion Matrix

A comprehensive examination of the classification results from comparing the different model.This displays a CNN-LSTM + Attention model with integrated PS(Packet Sniffing) that has an attention mechanism for use with both CNN and LSTM models independently. Accuracy, loss, and confusion matrices throughout training iterations We use recall, precision, and F1-score to evaluate each model. The results are organized and supported with visuals to explain how each model was performed. The confusion matrices provide classification accuracy and

misclassification rates, revealing each model's classification strengths and limitations. The accuracy and loss graphs compare training.

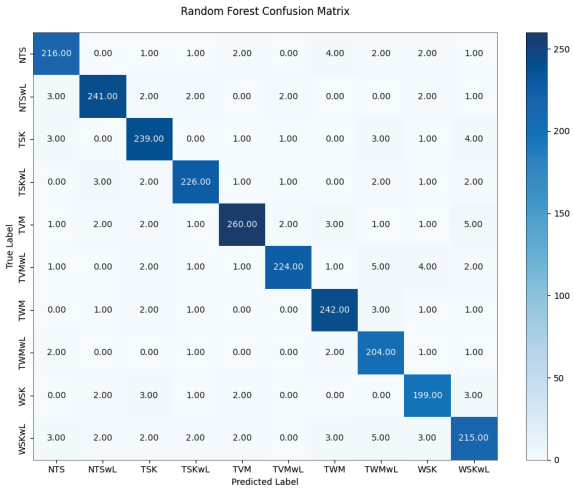

Figure 10. Random Forest model Confusion Matrix

The graphic illustrates three confusion matrices that depict the performance of several models: a Hybrid algorithm, an LSTM model and a convolutional neural network (CNN) model. The actual labels are shown in the rows of each matrix, while the projected classifications for different classes are shown in the Report: CSO Report was generated on Monday, Oct 21, 2024, 05:32 PM Page 20 of 34 columns. The colour intensity denotes the quantity of accurate or inaccurate forecasts, with deeper blue signifying more values. Figures 5–10 display confusion matrices that evaluate the classification efficacy of different deep learning and machine learning models across ten specific classes: NTS, NTSwL, TSK, TSKwL, TVM, TVMwL, TWM, TWMwL, WSK, and WSKwL. The suggested CNN+LSTM+Attention integrated PS(Packet Sniffing) model (Figure 5) exhibits exceptional performance, attaining the maximum correct classification counts for almost all classes with negligible misclassification. NTS is accurately classified 253 times, with a single instance misclassified as TSK; NTSwL achieves 242 correct classifications with no significant errors; TSK records 224 correct classifications, with nine misclassified as TWM; and notably, TVM attains a flawless score of 266 correct predictions without any mistakes. TWM attains 236 accurate predictions with merely two misclassified occurrences, while TWMwL registers 229 right predictions with only slight confusion involving TSK and TVM. These results demonstrate a commendable equilibrium between precision and recall, signifying the efficacy of incorporating the attention mechanism into CNN+LSTM.The CNN+LSTM model depicted in figure 6 demonstrates robust performance; nevertheless, it exhibits a marginally higher rate of misclassifications relative to the suggested model. NTS registers 245 accurate predictions with five misclassifications among other categories, while TSK attains 242 accurate predictions with four errors. TWMwL excels with 253 accurate predictions, whilst WSK achieves 236, demonstrating that this architecture continues to manage sequential and spatial patterns proficiently, albeit lacking the enhanced refinement provided by attention mechanisms.The CNN model (figure 7) yields robust findings; nonetheless, it is afflicted by extensive misclassification clusters. TWM demonstrates 216 accurate predictions, with 20 occurrences incorrectly labeled as TSK, indicating a substantial decline in performance for that category. Likewise, TSKwL attains 237 correct classifications with nine misclassifications into TWM, highlighting its limits in differentiating visually identical categories without sequence modeling.The LSTM model depicted in Figure 8 has similar overall accuracy to the CNN, albeit with distinct misclassification patterns. For instance, TSK registers 217 accurate predictions but misclassifies 16 as TVM, while TWM attains 232 correct predictions with just tiny, sporadic errors. This indicates that although LSTM effectively captures sequential relationships, it may have difficulties in distinguishing solely spatial features relative to CNN.Conventional models like XGBoost (figure 9) and Random Forest (figure 10) lag behind deep learning methodologies. XGBoost achieves reasonable accuracy, exemplified by NTS with 228 correct predictions and WSKwL with 219 correct, although demonstrates a broader distribution of mistakes across several classes. The Random Forest model has the poorest performance, achieving 216 correct classifications for NTS, 199 for WSK, and numerous classes encountering 4 to 5 misclassifications into other categories.The examination of the confusion matrix distinctly demonstrates that the suggested CNN+LSTM+Attention integrated PS model is the most effective, achieving the highest accurate classification counts and the lowest misclassification rates, thereby surpassing all other evaluated architectures.

Model Accuracy and Model Loss

To evaluate the learning behaviour and stability of each architecture, we monitored both accuracy and loss over 100 training epochs for the hybrid CNN–LSTM model, the standalone CNN, and the standalone LSTM. These metrics provide insight into each model's convergence speed, generalization ability, and risk of overfitting or underfitting. The following subsections detail the observed performance patterns, supported by graphical comparisons.

Hybrid Model Accuracy and Loss

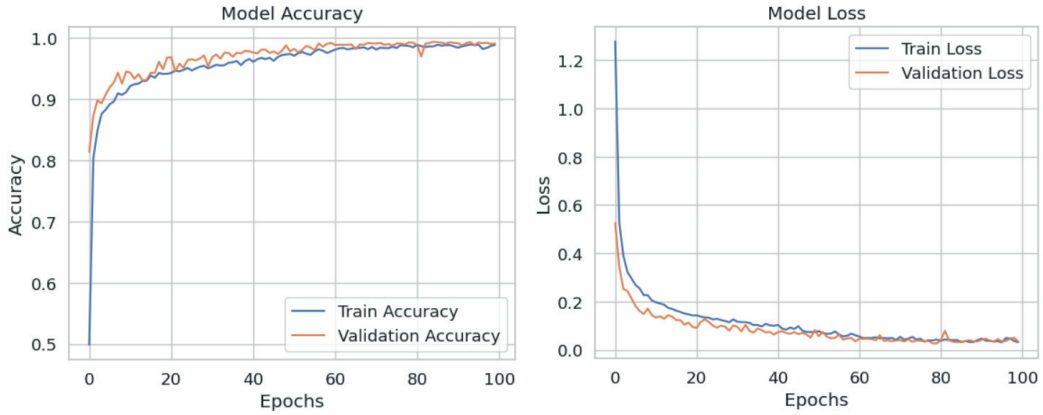

Figure 11. Hybrid Model Accuracy and Loss

Figure 11 displays how well the CNN+LSTM+Attention model with PS performed a hundred times during training. This depiction of training and validation accuracy is shown graphically, with the x-axis showing the total number of epochs and the y-axis showing the precise level. The model exhibits fast learning, attaining elevated accuracy within the initial 20 epochs. The training accuracy curve nears perfection at roughly 0,98 , while the validation accuracy displays a similar pattern, stabilizing somewhat below the training curve. The convergence of the two curves signifies little overfitting and robust generalization ability on both the datasets used for training and unseen validation, proving that the model is entirely accurate. The following graph shows the loss for the training set (blue curve) and the validation set (orange curve) across 100 repetitions. The x-axis represents the number of epochs, while the y-axis indicates the loss value. First, training and validation loss commences at comparatively elevated levels, with the training loss initiating at around 1,2 . Both losses significantly decrease during the initial epochs, signifying the model's accelerated learning process. The abrupt decline indicates that the model rapidly decreases errors and modifies its parameters to optimize the goal function. After around 20 epochs, both curves plateau, attaining a near-constant value, with training loss converging around 0,1 and validation loss stabilizing slightly lower. During training, there was a significant correlation between the loss curves for training and validation, which means that the model did not overfit and may thus generalize well to new data. The slight divergence between the two losses reinforces the model's resilience since it circumvents prevalent issues like overfitting to the training data or underfitting. This trend of loss reduction indicates excellent learning and optimization, with both losses attaining low levels, suggesting a well-trained and highly efficient model.

Performance Metrics

Cyber-Physical Systems (CPS) cybersecurity is greatly improved by integrating Packet Sniffing (PS) with deep learning architectures. This is evidenced by the outstanding performance of the hybrid CNN+LSTM+Attention model with PS as shown in figure 12, which achieved an accuracy of 99,08 % and nearly identical F1-score (0,9908), recall (0,9908), and precision (0,9909). This model processes real-time data from packet sniffing (PS) and uses convolutional neural networks (CNN) to extract spatial features from network packets, long short-term memory (LSTM) networks to identify temporal attack patterns, and an attention mechanism to prioritize important traffic segments. This method is very dependable for intrusion detection in CPS situations due to its excellent precision and recall values, which show little false positives and negatives.With 98,74 % accuracy (F1: 0,9868 , recall: 0,9864 , precision: 0,9874), the standalone CNN+LSTM model comes in second, proving that deep learning models perform better even in the absence of attention mechanisms. However, a ~0,34 % increase in accuracy is obtained by adding PS and attention techniques, underscoring the significance of adaptive feature weighting and real-time packet inspection. In contrast, the performance of the LSTM (98,41 % accuracy) and pure CNN (98,42 %) models is somewhat worse, indicating that hybrid architectures are crucial for identifying both temporal and spatial attack signs in CPS networks.On the other hand, deep learning techniques

outperform traditional machine learning models such as Random Forest (94,43 % accuracy, F1: 0,9438) and XGBoost (95,23 % accuracy, F1: 0,9519). This discrepancy (around 4-5 percent lower accuracy) highlights how typical machine learning is not up to par when it comes to processing high-dimensional, sequential network traffic data. For easier classification tasks, XGBoost and Random Forest are still helpful, but they have trouble with the dynamic, non-linear patterns found in CPS cyberattacks.

Figure 12. Performance Metrics

Table 2. 10-Fold Cross-Validation Performance of the Proposed CNN-LSTM-Attention with PS Model

Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	False Positive Rate (%)
1	98,42	98,10	98,25	98,17	2,3
2	98,36	98,05	98,18	98,11	2,4
3	98,51	98,22	98,30	98,26	2,1
4	98,39	98,09	98,22	98,15	2,5
5	98,47	98,16	98,28	98,22	2,2
6	98,50	98,18	98,32	98,25	2,3
7	98,44	98,12	98,27	98,19	2,3
8	98,53	98,23	98,33	98,28	2,1
9	98,48	98,20	98,29	98,24	2,2
10	98,45	98,15	98,26	98,20	2,3
Mean ± SD	98,45 ± 0,05	98,15 ± 0,06	98,27 ± 0,5	98,21 ± 0,05	0,11

The table -2 shows the 10-fold cross-validation results demonstrate the stability and high performance of the proposed CNN-LSTM-Attention model with PS across multiple data splits. Accuracy remained consistently above 98,3 % in all folds, with minimal standard deviation (±0,05 %), indicating strong generalization. Precision, recall, and F1-score also showed very small variations, confirming balanced detection capability across attack types. Importantly, the false positive rate (FPR) stayed low, averaging 2,27 %, ensuring that benign traffic was rarely misclassified as malicious — a critical factor for practical deployment in cybersecurity environments.

DISCUSSION

The thorough assessment of the proposed CNN-LSTM-Attention model, incorporating Packet Sniffing (PS), reveals its exceptional efficacy in classifying network traffic into ten unique categories (NTS, NTSwL, TSK, TSKwL, TVM, TVMwL, TWM, TWMwL, WSK, WSKwL). The confusion matrices (figures 5–10) demonstrate that the hybrid model attains nearly flawless classification, with 253 accurate predictions for NTS, 266 for TVM (impeccable), and 236 for TWM, while displaying negligible misclassifications (e.g., with 2 errors for TWM). Conversely, independent models such as CNN and LSTM exhibit elevated misclassification rates (e.g., 20 misclassifications for TWM in CNN, 16 for TSK in LSTM), underscoring their deficiencies in managing spatial and sequential connections absent attention processes. Conventional models (XGBoost, Random Forest) exhibit markedly inferior performance, with Random Forest attaining merely 199 accurate classifications for WSK,

underscoring the imperative for deep learning in intricate cyber-physical system (CPS) security endeavors. The training dynamics (figure 11) further corroborate the hybrid model's durability, achieving a training accuracy of approximately 98 % within 20 epochs and a validation loss stabilizing at 0,1, signifying negligible overfitting. The performance measures (figure 12) confirm its superiority, attaining 99,08 % accuracy, 0,9909 precision, 0,9908 recall, and a 0,9908 F1-score, surpassing CNN-LSTM (98,74 % accuracy), standalone LSTM (98,41 %), and CNN (98,42 %). The 10-fold cross-validation (table 2) demonstrates consistency, yielding a mean accuracy of 98,45 % (±0,05 %) and a low false positive rate (FPR) of 2,27 %, which is essential for practical implementation. The amalgamation of PS with deep learning is crucial, as real-time packet analysis improves feature extraction, while the attention mechanism sharpens focus on essential traffic segments. The approximately 0,34 % improvement in accuracy over CNN-LSTM and the 4–5 % disparity compared to XGBoost/Random Forest highlight the constraints of conventional techniques in high-dimensional, dynamic Cyber-Physical Systems contexts. The results establish the suggested model as a cutting-edge method for intrusion detection, achieving a high recall rate while maintaining low precision, which is crucial for protecting critical infrastructure. Future research may investigate real-time implementation and adversarial resilience to enhance practical applicability validation.

CONCLUSIONS

The study successfully achieved its objective of developing an advanced and operationally viable intrusion-detection framework for cyber-physical systems by integrating hybrid deep-learning components with real-time packet sniffing. The proposed CNN-LSTM architecture enhanced with an attention mechanism allowed for a coherent solution that could capture both spatial and temporal characteristics of network behavior while keeping interpretability and adaptive focus on critical traffic patterns. Its superior qualitative performance across models thus proved that the integration of packet-level data acquisition and deep spatiotemporal learning provided a more reliable and context-aware defense compared to any standalone or traditional approaches. This framework also put forth a clear methodological pathway-from live data collection to dynamic feature weighting-that might lead future CPS security designs. At the same time, the work recognized limitations concerning computational demands, dependence on high-quality labeled data, and reduced visibility in encrypted traffic that highlighted areas of further refinement. Generally, the research ensured a concrete, systematic, and practically relevant contribution to CPS cybersecurity and initiated a solid foundation for future improvements in real-time deployment, scalability, and resilience against evolving cyber threats.

BIBLIOGRAPHIC REFERENCES

1. Czekster RM, Metere R, Morisset C. Incorporating Cyber Threat Intelligence into Complex Cyber-Physical Systems: A STIX Model for Active Buildings. Appl Sci. 2022;12(10).

2. Smadi AA, Ajao BT, Johnson BK, Lei H, Chakhchoukh Y, Al-Haija QA. A comprehensive survey on cyber-physical smart grid testbed architectures: Requirements and challenges. Electron. 2021;10(9):1–25.

3. Thakur P, Kansal V, Rishiwal V. Hybrid Deep Learning Approach Based on LSTM and CNN for Malware Detection. Wirel Pers Commun. 2024;136(3):1879–901.

4. Zhang J, Pan L, Han QL, Chen C, Wen S, Xiang Y. Deep Learning Based Attack Detection for Cyber-Physical System Cybersecurity: A Survey. IEEE/CAA J Autom Sin. 2022;9(3):377–91.

5. Markevych M, Dawson M. A Review of Enhancing Intrusion Detection Systems for Cybersecurity Using Artificial Intelligence (AI). Int Conf KNOWLEDGE-BASED Organ. 2023;29(3):30–7.

6. Medjek F, Tandjaoui D, Djedjig N, Romdhani I. Fault-tolerant AI-driven Intrusion Detection System for the Internet of Things. Int J Crit Infrastruct Prot. 2021;34:100436.

7. Alsuwian T, Shahid Butt A, Amin AA. Smart Grid Cyber Security Enhancement: Challenges and Solutions—A Review. Sustain. 2022;14(21):1–21.

8. Kim S, Park KJ. A survey on machine-learning based security design for cyber-physical systems. Appl Sci. 2021;11(12).

9. Pimple J, Sharma A. Enhancing Cybersecurity in Medical Cyber-Physical Systems Using Blockchain and Deep Learning. In: 2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS). 2025. p. 1–6.

10. Kabir S, Hannan N, Shufian A, Rahman Zishan MS. Proactive detection of cyber-physical grid attacks: A pre-

attack phase identification and analysis using anomaly-based machine learning models. Array. 2025;27:100441.

11. Li J, Yang Y, Sun JS, Tomsovic K, Qi H. ConAML: Constrained Adversarial Machine Learning for Cyber-Physical Systems. In: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. New York, NY, USA: Association for Computing Machinery; 2021. p. 52–66. (ASIA CCS '21).

12. Zhang D, Wang QG, Feng G, Shi Y, Vasilakos A V. A survey on attack detection, estimation and control of industrial cyber–physical systems. ISA Trans. 2021;116:1–16.

13. Bhutani M, Dalal S, Alhussein M, Lilhore UK, Aurangzeb K, Hussain A. SAD-GAN: A Novel Secure Anomaly Detection Framework for Enhancing the Resilience of Cyber-Physical Systems. Cognit Comput. 2025;17(4):127.

14. Kamble D, Rathod S, Bhelande M, Shah A, Sapkal P. Original Research Article Correlating forensic data for enhanced network crime investigations: Techniques for packet sniffing, network forensics, and attack detection. J Auton Intell. 2024;7(4).

15. Grossi M, Alfonsi F, Prandini M, Gabrielli A. A Highly Configurable Packet Sniffer Based on Field-Programmable Gate Arrays for Network Security Applications. Vol. 12, Electronics. 2023.

16. Kibriya H, Masood M, Nawaz M, Rafique R, Rehman S. Multiclass brain tumor classification using convolutional neural network and support vector machine. In: 2021 Mohammad Ali Jinnah University international conference on computing (MAJICC). IEEE; 2021. p. 1–4.

17. Nguyen TT, Nguyen QVH, Nguyen DT, Nguyen DT, Huynh-The T, Nahavandi S, et al. Deep learning for deepfakes creation and detection: A survey. Comput Vis Image Underst. 2022;223:103525.

18. Alnajjar M. Image-based detection using deep learning and Google Colab. 2021;

19. Koonce B, Koonce BE. Convolutional neural networks with swift for tensorflow: Image recognition and dataset categorization. Springer; 2021.

20. Hua Y, Zhao Z, Li R, Chen X, Liu Z, Zhang H. Deep Learning with Long Short-Term Memory for Time Series Prediction. IEEE Commun Mag. 2019;57(6):114–9.

21. Smagulova K, James AP. A survey on LSTM memristive neural network architectures and applications. Eur Phys J Spec Top. 2019;228(10):2313–24.

22. Fang Z, Crimier N, Scanu L, Midelet A, Alyafi A, Delinchant B. Multi-zone indoor temperature prediction with LSTM-based sequence to sequence model. Energy Build. 2021;245:111053.

23. Nayak SS. Packet Sniffing. Int J Eng Manag Res Peer Rev Ref J e. 2024;14(1):71–6.

24. Sabeel A, Rajeev S., H.S C. with tips on how to get the Network Interface. Internetworking Res Exp. 2003;(December 2002):17–9.

25. Asrodia P, Sharma V. Network Monitoring and Analysis by Packet Sniffing Method. 2013;4(May):2133–5.

26. Ibrahim Diyeb IA, Saif A, Al-Shaibany NA. Ethical Network Surveillance using Packet Sniffing Tools: A Comparative Study. Int J Comput Netw Inf Secur. 2018;10(7):12–22.

27. Mohiuddin K, Welke P, Alam MA, Martin M, Alam MM, Lehmann J, et al. Retention Is All You Need. Int Conf Inf Knowl Manag Proc. 2023;(Nips):4752–8.

28. Niño-Adan I, Landa-Torres I, Portillo E, Manjarres D. Influence of statistical feature normalisation methods on K-Nearest Neighbours and K-Means in the context of industry 4.0. Eng Appl Artif Intell. 2022;111:104807.

29. Rodríguez P, Bautista MA, Gonzàlez J, Escalera S. Beyond one-hot encoding: Lower dimensional target embedding. Image Vis Comput. 2018;75:21–31.

30. Yang GW, Jing HF. Multiple Convolutional Neural Network for Feature Extraction BT - Intelligent Computing Theories and Methodologies. In: Huang DS, Jo KH, Hussain A, editors. Cham: Springer International Publishing; 2015. p. 104–14.

31. Hefron RG, Borghetti BJ, Christensen JC, Kabban CMS. Deep long short-term memory structures model temporal dependencies improving cognitive workload estimation. Pattern Recognit Lett. 2017;94:96–104.

32. Soydaner D. Attention mechanism in neural networks: where it comes and where it goes. Neural Comput Appl. 2022;34(16):13371–85.

33. Wardhani NWS, Rochayani MY, Iriany A, Sulistyono AD, Lestantyo P. Cross-validation Metrics for Evaluating Classification Performance on Imbalanced Data. 2019 Int Conf Comput Control Informatics its Appl Emerg Trends Big Data Artif Intell IC3INA 2019. 2019;14–8.

ETHICAL APPROVAL

In this research, there is no involvement of humans and animals.

AVAILABILITY OF DATA AND MATERIALS

We have used the publicly available dataset, and it is already cited in the reference section.

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Deepa Singh Sisodiya, Ritu Tiwari, Priyank Jain.
Data curation: Deepa Singh Sisodiya, Ritu Tiwari, Priyank Jain.
Formal analysis: Deepa Singh Sisodiya, Ritu Tiwari, Priyank Jain.
Drafting - original draft: Deepa Singh Sisodiya, Ritu Tiwari, Priyank Jain.
Writing - proofreading and editing: Deepa Singh Sisodiya, Ritu Tiwari, Priyank Jain.