# Enhanced intrusion detection in smart grids using extended long short-term memory variants

Saida Baalia [a,1], Djalila Boughareb [a,2,*], Zineddine Kouahla [a,3], Hamid Seridi [a,4]

[a] Department of computer science, University of 8 Mai 1945, BP 401, Guelma 24000, Guelma, Algeria
[1] baalia.saida@univ-guelma.dz; [2] boughareb.djalila@univ-guelma.dz; [3] kouahla.zinedine@univ-guelma.dz; [4] seridi.hamid@univ-guelma.dz
* corresponding author

ARTICLE INFO

**Article history**
Received July 28, 2025
Revised September 9, 2025
Accepted October 14, 2025
Available online November 30, 2025

**Keywords**
Deep learning
Intrusion Detection System
xLSTM
Cybersecurity
Smart Grid

ABSTRACT

Smart grid systems, which integrate traditional energy infrastructure with modern communication technologies, face significant cybersecurity challenges due to their dynamic architecture and continuous data exchange. The diversity and interconnection of devices increase vulnerability to malicious intrusions, highlighting the need for advanced and scalable detection methods. This study aims to develop an intrusion detection system (IDS) for smart grids by leveraging recent advances in deep learning, specifically enhanced variants of Long Short-Term Memory (LSTM): xLSTM, sLSTM, and mLSTM. These sequence modeling architectures were adapted and fine-tuned within our IDS framework to capture complex spatio-temporal patterns and handle heterogeneous, high-dimensional data effectively. A comprehensive evaluation on two benchmark datasets, NSL-KDD and DNP3, demonstrates the robustness of the proposed approach. On the NSL- KDD, xLSTM, sLSTM, and mLSTM achieved accuracies of 98.16%, 98.55%, and 98.54%. On the more modern, protocol-specific DNP3 dataset, which represents real-world SCADA-focused attacks, the models maintained their superior performance, achieving accuracies of 99.50%, 99.33%, and 99.42%, respectively. The high and consistent accuracy across both datasets demonstrates the models' dependability and adaptability for intrusion detection in smart grid infrastructures. The study's targeted enhancement of LSTM-based architectures contributes a novel and effective approach to protecting critical intelligent systems from emerging cyber threats.



## 1. Introduction

The integration of traditional power infrastructure with advanced communication technologies has led to the emergence of smart grid systems, which provide numerous advantages, including efficiency, resilience, and real-time monitoring [1]. However, the complexity and heterogeneity of devices, combined with large-scale interconnected networks, introduce significant cybersecurity challenges [2]. Cyberattacks on smart grids can cause severe consequences such as power outages, grid instability, economic losses, infrastructure damage, and threats to human safety—particularly when critical facilities like hospitals, electric vehicle charging stations, or smart homes are targeted. These attacks often exploit protocol vulnerabilities, overload communication layers, or manipulate control systems, highlighting the urgent need for robust cybersecurity mechanisms, especially advanced Intrusion Detection Systems (IDS) [3].

Supervisory Control and Data Acquisition (SCADA) systems play a vital role in monitoring and controlling processes across critical infrastructure sectors, including power distribution, telecommunications, transportation, manufacturing, and water treatment [4],[5],[6]. In smart grid SCADA systems, monitor electricity distribution using sensors that provide continuous measurements of electrical parameters, enabling real-time automated control for safe and efficient energy delivery [7][8]. However, the integration of digital communication within SCADA has also introduced new security vulnerabilities. Among these, the Distributed Network Protocol version 3 (DNP3), a widely adopted protocol in SCADA-based energy systems, is particularly vulnerable to cyberattacks [9].

Various types of cyber threats are increasingly targeting smart grid environments, including phishing attacks [10], Denial-of-Service (DoS) [11], malware [12], and eavesdropping or traffic analysis [13]. Traditional IDS approaches often struggle to handle these sophisticated attacks due to their limited ability to capture temporal dynamics and protocol-specific traffic patterns. However, existing models frequently fail to capture the sequential complexity and protocol-specific features of smart grid traffic, particularly within DNP3-based environments. This limitation highlights a critical research gap that motivates the present study [14].

Recent studies have demonstrated that machine learning [15], [16], [17], [18] and deep learning [14], [19], [20], [21] methods can improve IDS performance. Deep learning models, in particular, have shown strong capabilities in anomaly detection and pattern recognition. Nevertheless, there remains a need for models that can better exploit temporal dependencies and heterogeneous traffic patterns in smart grids. To address this gap, this study aims to develop and evaluate advanced variants of Long Short-Term Memory (LSTM) architectures, xLSTM, sLSTM, and mLSTM, to improve intrusion detection performance in smart grid networks.

The main contribution of this work is the novel application and comprehensive evaluation of Extended Long Short-Term Memory (xLSTM) architectures to design a highly effective Intrusion Detection System (IDS) for smart grid environments. The proposed xLSTM framework introduces two key innovations,exponential gating and new memory structures, which extend the capabilities of conventional LSTM models. These enhancements give rise to two specialized variants: scalar LSTM (sLSTM) and matrix LSTM (mLSTM). The sLSTM employs scalar memory, scalar updates, and controlled memory mixing to enhance information flow across multiple memory cells and heads, while the mLSTM uses matrix memory with a covariance-based update rule to enable full parallelization by removing hidden-to-hidden recurrent connections. Both variants utilize exponential gating to improve learning dynamics and gradient stability. When integrated with residual block modules, these variants form xLSTM blocks, which can be stacked to build deeper architectures capable of learning complex temporal dependencies while maintaining efficient gradient propagation.

Through extensive experiments on two benchmark datasets, the NSL-KDD and the DNP3 dataset (which simulates real-world SCADA-based cyberattacks), the study demonstrates that the proposed framework achieves high accuracy, robustness, and scalability, outperforming conventional IDS approaches. The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 details the proposed approach and presents the architecture; Section 4 discusses experimental results; and Section 5 concludes the paper and outlines future directions.

## 2. Related Works

### 2.1. Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) have emerged as critical technologies for monitoring network traffic and detecting potential cyberattacks across complex technological infrastructures [22]. Researchers have proposed a number of strategies for detecting and mitigating network threats, which can be broadly classified into three primary detection methodologies [23]: signature-based, anomaly-based, and specification-based IDS. Signature-based IDS compares network traffic against a database of known attack signatures. They are highly effective at detecting previously encountered threats and exhibit low false-positive rates. However, their primary limitation is their inability to detect novel or

zero-day attacks, as their performance is dependent on regular and timely signature database updates in order to remain effective against emerging threats [24].

Anomaly-based IDS detects intrusions by looking for deviations from established patterns of normal behavior. This approach is especially useful for detecting unknown or evolving attacks, providing greater adaptability to dynamic threat landscapes [25]. Despite this advantage, anomaly-based systems are more prone to false positives and must establish an accurate, comprehensive baseline of normal activity to function properly. Specification-based IDS identifies abnormal behavior by comparing system activities to a set of predefined specifications that describe acceptable system behavior. These specifications are usually extracted manually. Such systems are capable of detecting previously unknown attacks and frequently achieve low false-positive rates. However, their effectiveness is heavily reliant on the precision and breadth of the defined specifications. In highly complex environments, such as networks with diverse protocols and applications, scalability and accuracy may be challenging; moreover, maintaining up-to-date specifications requires continuous effort to align with evolving system architectures and usage patterns [26].

### 2.2. Techniques of Intrusion Detection in Smart Grids

Smart grids are a transformative paradigm in modern power systems, combining advanced communication technologies with traditional electrical infrastructure. This integration allows for intelligent, real-time management of electrical energy while also increasing the vulnerability of smart grids to sophisticated cyberattacks. To address this, a large body of research has focused on developing strong Intrusion Detection Systems (IDS) to improve the security and dependability of these critical infrastructures. This section provides a systematic overview of the most significant IDS developments for smart grids, categorizing them into four major methodological groups: (a) machine learning-based approaches [9], [27], [31] (b) deep learning-based techniques [22], [28], [29], [30], [32], [33], (c) hybrid systems combining multiple ML/DL models [14], [34], [37], [39], and (d) advanced architectures incorporating GNNs and other optimizations [19], [35], [36], [38], [40], [41].

Several studies have applied traditional machine learning (ML) methods to intrusion detection in smart grids. For instance, [27] proposed a False Data Injection (FDI) detection approach using Principal Component Analysis (PCA), projecting grid states onto a residual subspace and applying a detection threshold to identify anomalies. The method achieved high detection probabilities for attacks targeting voltage angles and magnitudes, as demonstrated in several case studies. However, its linear nature limits its effectiveness in detecting nonlinear or adaptive attack patterns, and its performance depends on the quality of historical data and the appropriate selection of model parameters such as subspace size and detection threshold. Similarly, [16] explored supervised ML classifiers such as SVM, KNN, and ANN in combination with heuristic feature selection algorithms (e.g., BCS, BPSO, and Genetic Algorithm). Among these combinations, the SVM with GA achieved the highest accuracy of 90.59% on the IEEE 118-bus system. However, despite these promising results, heuristic methods may overlook structural dependencies between network features, leading to biased models that struggle to detect complex and distributed attack patterns such as APTs.

In [31], various supervised models (SVM, KNN, Random Forest, ANN, Logistic Regression, and AdaBoost) were trained on SCADA MODBUS/TCP data. All models achieved perfect performance, with Accuracy, Precision, Recall, and F1-score equal to 100%, except for Naive Bayes, which obtained slightly lower values around 99.6%. However, the study relies on a limited dataset and lacks cross-validation or comparisons with more recent approaches, raising concerns about potential overfitting and limiting the generalizability of the results.

Deep learning (DL) methods have gained prominence for their ability to extract hierarchical representations automatically. In [28], a hierarchical multi-layer deep learning model was proposed. The method is based on a sequence of ordered binary classifiers, where each classifier is trained to distinguish a specific attack type from normal traffic, enabling step-by-step detection across multiple network layers. The model achieved an accuracy of 99.99% on the CICIDS2017 dataset. However, the proposed approach was evaluated only on the CICIDS2017 network traffic dataset, without validation on simulated

or real smart meter communications. In this approach [29], multiple deep autoencoders are first trained in an unsupervised manner to learn rich and representative features from the training data. These features are then automatically fused using the Multiple Kernel Learning (MKL) algorithm, which adaptively determines the optimal weights for their combination. The final detection model produced through the MKL process is subsequently used to classify test samples and identify potential DDoS attacks, achieving a high accuracy of 97%. However, the approach involves high computational complexity due to the training of multiple deep autoencoders and the MKL fusion process, which may hinder real-time deployment in large-scale network environments. Likewise, [30] combines deep autoencoder (DAE)-based feature reduction with semi-supervised anomaly detection algorithms such as OCSVM, KNNOD, and CBLOF, resulting in highly effective detection of previously unseen threats. However, its main limitation lies in a relatively high false positive rate (FPR), which may lead to alert saturation.

In [22], several Recurrent Neural Network (RNN) variants (LSTM, GRU, and Simple RNN) were combined with XGBoost for feature selection. In binary classification, XGBoost-LSTM achieved 88.13% Accuracy on NSL-KDD, while XGBoost-Simple RNN achieved 87.07% on UNSW-NB15. The multiclass classification results were slightly lower, with XGBoost-GRU scoring 78.40% on UNSW-NB15. A key limitation is that XGBoost-LSTM fails to detect attack class 2 (likely U2R) in the NSL-KDD dataset, likely because XGBoost-FST reduces the dataset from 41 to 22 attributes, thereby removing essential features for this rare attack type. [32] and [33] investigated hybrid deep models for DDoS detection. [32] combined Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) to achieve 99.86% accuracy on CIC-DDoS2019. However, the main limitation of the hybrid CNN-GRU model is its focus on DoS/DDoS attacks, necessitating further development to effectively address the wider range of threats present in Smart Grid systems. [33] proposed a CNN-GRU-FF framework for double-layer feature extraction that uses a modified focal loss to address class imbalance. The model scored 98.22% and 99.68% on NSL-KDD and UNSW-NB15, respectively. However, the hybrid model's complex feature fusion poses a practical limitation, potentially making it computationally heavy and slow for real-world deployment

Hybrid systems that combine multiple ML/DL techniques have gained popularity due to their ability to detect a wide range of attack characteristics. For example, [14] demonstrated a CNN-LSTM model that captures both spatial and temporal patterns. When tested on the DNP3 dataset, it achieved 99.50% Accuracy. However, the CNN-LSTM model is burdened by high computational cost and memory consumption, a challenge the authors explicitly acknowledge as an area for future enhancement. Furthermore, [34] developed a dual-hybrid system for FDIA attacks that combines Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO) for feature selection. Using CNNs and LSTMs, the system achieved an Accuracy of 97.5% on the ICS Cyber Attack dataset. Also, [37] proposed HAE-HRL, a three-stage hybrid model that includes an autoencoder for anomaly detection, a CNN for spatial feature learning, and a ResNet-LSTM module for sequential pattern recognition. It achieved 95.22% Accuracy on the NSL-KDD, WSN-DS, and simulated SCADA datasets, thereby improving precision and reducing false positives compared to classical methods. However, the residual false-positive rate of 15.25% remains a limitation that warrants further refinement. In [39], a cyber threat intelligence framework, DT-ML-SCADA-CTI, was created by combining Digital Twin models, knowledge graphs, and ML classifiers (such as Extra-Trees, XGBoost, and Random Forest). The system effectively detected FDIA, RTCI, and SRA with 98% Accuracy on the Power System Attack Dataset. The authors noted that the study was limited by the absence of DoS/DDoS scenarios and suggested further refinement of the digital twin integration.

Advanced techniques for addressing the spatiotemporal complexity of smart grid networks have been proposed, including graph-based models and optimization strategies. For example, [19] created a Probabilistic Recurrent Neural Network (PRNN) optimized using Improved Aquila Swarm Optimization (IASO) that achieved 99.91% Accuracy on smart grid data and 99.8% on the KDD99 dataset. In addition, [39] proposed ARDL-FDIAR, which uses a Modified Lemur Optimization Algorithm (MLOA) for feature selection and an Improved Deep Belief Network (IDBN) for classification. It achieved an accuracy of 98.58% on the IEEE 14bus and 97.73% on the IEEE 39bus. Also, [41] used the A-BiTG model, which combines a Bidirectional Temporal Convolutional Network

(BiTCN) and a Bidirectional GRU (BiGRU) with attention mechanisms. It achieved an accuracy of 96.23% on the IEEE 14-bus and 95.77% on the IEEE 118-bus systems. Specifically, several approaches have successfully leveraged graph structures for intrusion detection: [35] used an autoencoder for feature extraction and a Graph Convolutional Network (GCN) to model topological relationships. When tested on the DNP3 Intrusion Detection Dataset, it achieved 95% Accuracy. In [36], a Graph Attention Network (GAT) was combined with a Kolmogorov-Arnold Network (KAN) to detect structural and nonlinear patterns. The GraphKAN model was 97.63% accurate in binary classification, 98.66% in three-class, and 99.04% in 37-class tasks. Furthermore, [40] used Spatio-Temporal Graph Neural Networks (STGNNs), which combine Graph Convolutional Gated Recurrent Units (GCGRUs) with MLPs and a cross-domain learning strategy. On the IEEE 118bus system, it detected FDI and signal distortion with an accuracy of 89.2%. Despite these promising results, the efficacy of GN, GCN, and GAT-based approaches is frequently constrained by the high computational cost associated with the dynamic construction and updating of graphs from raw network traffic, thereby impeding scalability and real-time intrusion detection.

While previous contributions demonstrate promising levels of accuracy, many rely on benchmark datasets such as NSL-KDD and CICIDS, which, although widely adopted, do not fully reflect the unique characteristics of smart grid environments or industrial communication protocols like DNP3. Many existing methods target specific attacks and fail to generalize across diverse threat scenarios. Furthermore, simplified network assumptions limit their ability to represent the spatiotemporal dynamics and heterogeneous data flows typical of real-world smart grids. Moreover, several hybrid and graph-based methods increase architectural complexity without achieving proportional performance improvements, often requiring extensive graph construction or handcrafted feature engineering that limits scalability and real-time applicability.

In contrast, the proposed xLSTM solution focuses on a fundamental enhancement of the recurrent mechanism itself. The integration of exponential activation functions as replacements for traditional sigmoids in the input and forget gates provides the model with a novel dynamic control over memory flow. This modification eliminates saturation issues and mitigates gradient vanishing, enabling the model to analyze long, heterogeneous sequences in Smart Grids while preserving temporally distant information, which is crucial for identifying Advanced Persistent Threats (APT). In parallel, the introduction of a Matrix Memory in the architecture represents a decisive advancement for detecting modern threats. Unlike conventional one-dimensional vector memories that can represent only isolated states, this new structure stores information in matrix form, enabling the model to capture not only the values of individual events but also the relationships and structural correlations among them. Powered by a covariance-based update rule, this matrix memory excels at endogenously detecting complex and distributed dependencies within network traffic, thereby enhancing the model's capability to recognize subtle and multi-scale attack patterns.

To ensure comparability with prior research, a baseline is first established using the NSL-KDD dataset. a baseline is first established using the NSL-KDD dataset. The primary focus, however, is on the novel application and comparative analysis of advanced LSTM variants, specifically xLSTM, sLSTM, and mLSTM, for intrusion detection in DNP3 traffic, a protocol essential to SCADA systems in smart grid environments. A real-world DNP3 dataset is used to evaluate contemporary cyberattack scenarios. These architectures are evaluated based on their ability to model complex temporal dependencies and detect intrusions with high accuracy. The goal is to close gaps in current research by developing a protocol-aware IDS that is both robust and adaptable to the changing threat landscape in smart grid environments.

## 3. Method

### 3.1. Long Short-Term Memory (LSTM)

Traditional Long Short-Term Memory (LSTM) networks are effective for sequence modeling; however, several fundamental limitations constrain their performance on complex temporal tasks such

as cyber intrusion detection. One significant disadvantage stems from the use of the sigmoid activation function in gating mechanisms. When exposed to large input values, this function frequently saturates, resulting in near-zero gradients, also known as the vanishing gradient problem [42]. This severely limits the ability to update gate parameters effectively, particularly for long sequences that require sustained memory retention. Furthermore, the scalar representation of the internal cell state limits memory flexibility, reducing the network's ability to model complex and long-range temporal dependencies. The sequential computation pattern further restricts parallelism, limiting scalability and efficiency in high-throughput environments.

To address these challenges, the Extended Long Short-Term Memory (xLSTM) architecture was introduced, combining two advanced variants: the scalar LSTM (sLSTM) and the matrix LSTM (mLSTM). Both architectures share a common foundation based on exponential gating, which replaces the traditional sigmoid activation to improve learning dynamics and stability. This mechanism allows both models to better control memory updates and enhance gradient flow across long sequences. While sLSTM introduces scalar memory with new memory mixing across multiple cells and heads, mLSTM employs a matrix-based memory with a covariance update rule, enabling full parallelization and higher memory capacity. Together, they preserve LSTMs' sequential-reasoning strength while addressing scalability and computational efficiency. The xLSTM framework is thus particularly well-suited to security-critical applications in smart grids, where capturing long-term dependencies and subtle temporal variations is essential for detecting complex cyber threats.

### 3.2. Scalar LSTM (sLSTM)

The Scalar LSTM (sLSTM) significantly improves the traditional LSTM architecture by incorporating exponential gating functions and a memory normalization strategy. A normalizer state is included to regulate hidden-state transitions and maintain numerical stability during training. Furthermore, this architecture utilizes multi-head memory blocks to efficiently integrate temporal information across time steps while remaining computationally tractable. These structural changes allow for better modeling of long-term dependencies and provide increased resilience to gradient degradation over longer sequences. Eqs. (1-10) depict the forward-pass equations governing the sLSTM unit [43]:

$$z_t = \varphi(w_z x_t \dot{+} r_z h_{t-1} + b_z) \quad (1)$$

$$i_t = \exp(w_i \top x_t + r_i \top h_{t-1} + b_i) \quad (2)$$

$$f_t = exp\left(w_f \top x_t + r_f \top h_{t-1} + b_f\right) \quad (3)$$

$$o_t = \sigma(w_o x_t + r_o h_{t-1} + b_o) \quad (4)$$

$$m_t = max\left(log(f_t) + m_{t-1}, log(i_t)\right) \quad (5)$$

$$i_t' = exp(log(i_t) - m_t) = \exp(i_t - m_t) \quad (6)$$

$$f_t' = exp(log(f_t) + m_{t-1} - m_t) \quad (7)$$

$$c_t = f_t'.c_{t-1} + i_{t.}' z_t \quad (8)$$

$$n_t = f_t' n_{t-1} + i_t' \quad (9)$$

$$h_t = o_t \left(\frac{c_t}{n_t}\right) \quad (10)$$

where, $z_t$ represents the cell input, which is the raw information proposed to be added to the cell state; $i_t$ is the input gate, controlling how much of the new cell input is added to the cell state; $f_t$ is the forget gate, determining how much of the previous cell state is retained; $o_t$ is the output gate, regulating what information from the cell state is output to the hidden state; $n_t$ is a stabilizer state. It is applied to the input gate ($i_t$) and the forget gate ($f_t$) to prevent exploding gradients during training, resulting in the

stabilized gates $i'$ and $f'$; $c_t$ is the cell state, which stores the long-term memory of the sequence; $m_t$ is the normalizer state, used in the calculation of the hidden state; $h_t$ is the hidden state at the current time step, representing the output of the sLSTM unit; $exp$ represents the exponential function, used here instead of the sigmoid activation; $w_i$ and $w_f$ denote the weight matrices associated with the input gate ($i_t$) and the forget gate ($f_t$), respectively, when processing the input vector; $x_t$ is the input vector at the current time step, $t$; $r_i$ and $r_f$ are the recurrent weight matrices specifically for the input gate ($i_t$) and the forget gate ($f_t$), handling the influence of the previous hidden state; $h_{t-1}$ is the hidden state vector from the previous time step ($t-1$), encapsulating information learned from the sequence up to that point; and $b_i$ and $b_f$ are the bias vectors for the input gate ($i_t$) and the forget gate ($f_t$), respectively.

### 3.3. Matrix LSTM (mLSTM)

Matrix Long Short-Term Memory (mLSTM) is an evolution of the traditional LSTM framework that combines a matrix-based memory structure with advanced update dynamics. Rather than relying on scalar-valued memory cells, mLSTM encodes and stores information using matrices, significantly increasing memory capacity. This matrix formulation is beneficial for managing high-dimensional, complex input patterns, which are common in modern cyber-physical systems. An additional improvement is the implementation of a covariance-inspired update mechanism inspired by Bidirectional Associative Memories (BAM). This mechanism enables more efficient encoding and retrieval of key-value associations, thereby improving the model's ability to maintain coherent representations over time.

Crucially, mLSTM eliminates the strict sequential dependency that characterizes traditional LSTM computation. Because of its architectural design, memory integration is fully parallelizable, making the model highly compatible with modern hardware accelerators such as GPUs and TPUs.This enables rapid training and inference in high-throughput scenarios, which is critical for real-time applications such as intrusion detection in smart grids [44]. The computational operations defining the mLSTM forward pass are formalized using Eqs. (11-22) [43]:

$$q_t = W_q x_t + b_q \tag{11}$$

$$k_t = \frac{1}{\sqrt{d}} W_k x_t + b_k \tag{12}$$

$$v_t = W_v x_t + b_v \tag{13}$$

$$i_t = exp(w_i \top x_t + b_i) \tag{14}$$

$$f_t = exp\left(w_f \top x_t + b_f\right) \tag{15}$$

$$o_t = \sigma(w_o x_t + b_o) \tag{16}$$

$$m_t = max\left(log(f_t) + m_{t-1}, log(i_t)\right) \tag{17}$$

$$i'_t = exp(log(i_t) - m_t) = exp(i_t - m_t) \tag{18}$$

$$f'_t = exp(log(f_t) + m_{t-1} - m_t) \tag{19}$$

$$c_t = f'_t . c_{t-1} + i'_t . v_t k_t \tag{20}$$

$$n_t = f'_t n_{t-1} + i'_t k_t \tag{21}$$

$$h_t = o_t \odot (c_t . q_t / \max\{n_t \top q_t, 1\}) \tag{22}$$

where, $q_t$ (query input), $k_t$ (key input), and $v_t$ (value input) are derived from the input vector $x_t$ through linear transformations using weight matrices $W_q$, $W_k$, $W_v$, respectively. The gates control the flow of information: $i_t$ (input gate) determines how much of the new input $x_t$ contributes to the cell state, $f_t$ (forget gate) decides how much of the previous cell state $c_{t-1}$ is retained, $o_t$ (output gate)

regulates how much of the current cell state is exposed to the output (hidden state). These gates are typically activated using a sigmoid function ($\sigma$), $m_t$ (stabilizer state) is a mechanism applied to the input gate $i_t$and forget gate $f_t$ to mitigate the issue of exploding gradients during training. This results in modified gate values $i'$ and $f'$, $c_t$ (cell state) is the cell state, which incorporates the outer product of the value and key inputs, $n_t$ is the normalizer state; $h_t$ is the hidden state. The hidden state calculation involves element-wise multiplication (denoted by $\odot$) with the output gate and a normalized version of the cell state, using the query input and normalizer state.

### 3.4. Extended Long Short-Term Memory (xLSTM)

The xLSTM architecture is an advanced recurrent neural network framework that improves on the classical LSTM's capabilities with two key innovations: exponential gating mechanisms and enhanced memory representations. It includes two specialized memory variants: sLSTM, which uses scalar-valued memory enhanced with refined memory mixing strategies, and mLSTM, which uses a matrix-based memory architecture with a covariance-inspired update mechanism and supports complete parallel computation. These memory variants are contained within modular xLSTM blocks, which incorporate residual connections and dimensionality expansion via up-projection mechanisms specific to each variant.

Specifically, the sLSTM block employs a post-up-projection configuration, whereas the mLSTM block employs a pre-up-projection configuration. Stacking xLSTM blocks is implemented within a residual pre-LayerNorm backbone, enabling deeper network architectures while maintaining stable gradient propagation and facilitating efficient training. xLSTM models long-range dependencies more effectively and handles memory-intensive tasks better than many recent models, including Transformers and State Space Models. Fig. 1 illustrates the architectural differences and fundamental innovations that distinguish the classic LSTM architecture from its three extended variants: sLSTM, mLSTM, and xLSTM.

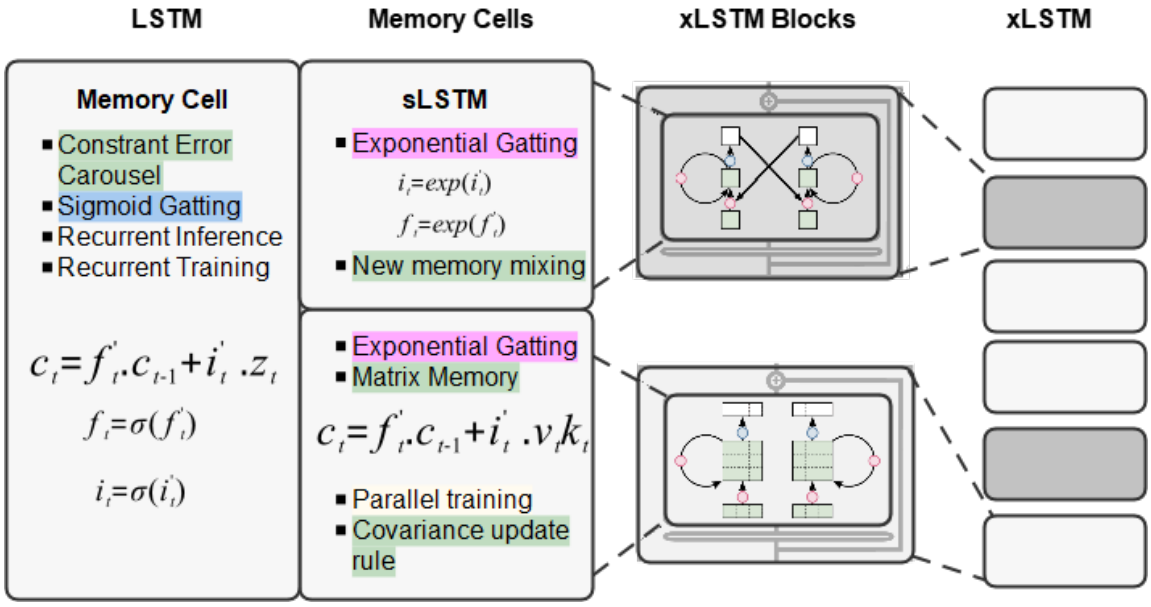


**Fig. 1.** The xLSTM (extended LSTM) family [43]

### 3.5. Proposed Architecture

The proposed intrusion detection system for smart grid environments is built on a systematic, modular pipeline. The process begins with a thorough data preprocessing step that converts raw network traffic into a format suitable for analysis (Fig. 2). Data cleansing, precise categorical encoding, and robust normalization techniques are all required during this phase to ensure feature consistency and scale invariance.

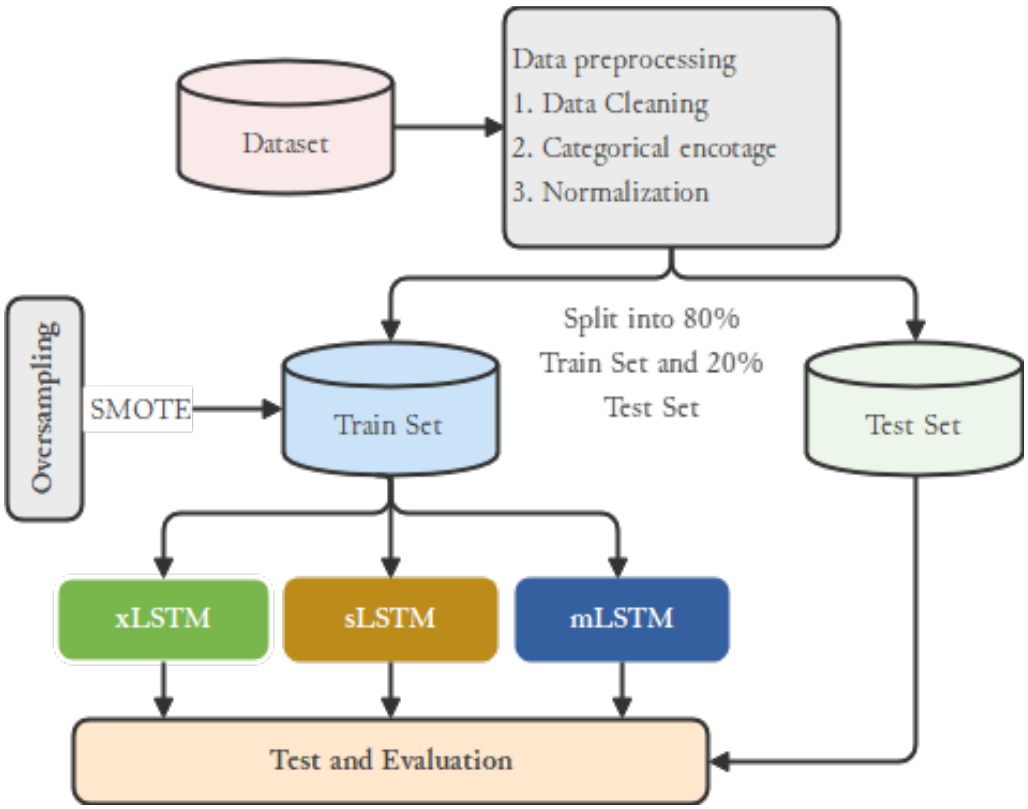


**Fig. 2.** Architecture of the proposed xLSTM-based model.

We divide the dataset into training and testing sets. Because cybersecurity datasets often contain few attack samples, we use targeted oversampling to balance the training set. Thus, the model construction phase is central to the system. Three advanced variants of Long Short-Term Memory (LSTM) networks, sLSTM, mLSTM, and xLSTM, are trained separately on the refined data. The sLSTM architecture is based on a single sLSTMBlock, which combines exponential gating functions with a novel memory-mixing mechanism to capture temporal patterns and long-range dependencies. In contrast, the mLSTM model employs an mLSTMBlock, which is distinguished by its matrix-based memory representation and a covariance-driven update mechanism that improves parallelization and memory richness.

The xLSTM architecture serves as a hybrid framework that combines the distinct strengths of sLSTM and mLSTM. It is implemented by stacking an sLSTMBlock followed by an mLSTMBlock, allowing the model to use both exponential gating dynamics and matrix-level memory updates. This hybrid stacking is integrated into a residual backbone to ensure efficient gradient flow and increased network capacity. Finally, the evaluation stage compares the trained models to the reserved test set, using performance metrics to determine each model's effectiveness in detecting cyber intrusions in the smart grid context.

Table 1 presents detailed specifications of the sLSTMBlock and mLSTMBlock architectures, and compares the architectural characteristics and memory mechanisms of the sLSTM, mLSTM, and xLSTM models across different experimental scenarios. The sLSTM employs a scalar-based memory structure with exponential gating and memory mixing, enabling stable temporal modeling while mitigating vanishing and exploding gradient issues. In contrast, the mLSTM introduces a matrix-based memory representation via outer-product operations, enhancing its ability to capture high-dimensional relationships through covariance-based updates and fully parallel execution. The xLSTM integrates the strengths of both sLSTM and mLSTM by sequentially stacking their blocks and incorporating residual connections, allowing for deeper architectures and improved long-context modeling. Overall, the results indicate that the hybrid xLSTM design provides a more expressive and flexible framework for memory-intensive, long-sequence learning tasks.

**Table 1.** Architectural components of sLSTM, mLSTM, and xLSTM blocks.

| Aspect | sLSTM | mLSTM | xLSTM |
|---|---|---|---|
| Layer type | Single block layer | Single block layer | Sequential stacking of sLSTM and mLSTM blocks with optional residuals |
| Gating mechanisms | Input (i), forget (f), output (o), and cell input (z) gates, with exponential (exp) activations for i and f. | Input (i), forget (f), and output (o) gates, with exponential (exp) activations for $i$ and $f$. The output gate (o) uses a sigmoid function. | Combines sLSTM and mLSTM gate operations |
| Memory structure | Scalar cell state ($c_t$), normalizer state ($n_t$), and memory mixing state ($m_t$). | Matrix-based memory (outer product of value and key: v⊗k), with a matrix normalizer state. | Combines the scalar memory of the sLSTMBlock with the matrix memory of the mLSTMBlock. |
| Memory update | Updates the cell state using the memory mixing state ($m_t$) and exponential gates. | Updates the cell state using a covariance-based approach (v⊗k for $c_t$ and k for $n_t$) and exponential gates. | Updates cascade sequentially through the sLSTMBlock and then the mLSTMBlock. |
| Key components | - CausalConv1D for contextual feature extraction.<br>-BlockDiagonal for linear transformations ($W_i$, $W_f$, $W_z$, $W_o$, $R_i$, $R_f$, $R_z$, $R_o$).<br>- LayerNorm, GroupNorm.<br>- Up/down projection (up_proj_left, up_proj_right, down_proj) for dimension management. | - CausalConv1D for contextual feature extraction.<br>- BlockDiagonal for Query ($W_q$), Key ($W_k$), Value ($W_v$) transformations.<br>- Linear transformations for gates ($W_i$, $W_f$, $W_o$).<br>- LayerNorm, GroupNorm.<br>- Up/down projection (up_proj_left, up_proj_right, down_proj) and skip connection. | Stacks the components of both sLSTMBlock and mLSTMBlock. Utilizes residual connections (output + x in individual blocks) to facilitate the training of deep networks. |
| Specific mechanisms | - Exponential Gating to mitigate vanishing/exploding gradient problems.<br>- Memory Mixing (implied via $m_t$, $m_{prev}$) for complex temporal dependencies.<br>- Multi-Head Blocks with block-diagonal weight matrices. | - Matrix Memory for handling high-dimensional inputs.<br>- Fully parallel execution(absence of recurrent memory mixing between cells).<br>- Key normalization | Integrates innovations from both sLSTM and mLSTM. Designed for long-context sequence modeling and memory-intensive tasks. |

### 3.6. SCADA Integration of the Proposed Model

In the proposed SCADA architecture, continuous data streams are acquired from field devices such as sensors, actuators, Remote Terminal Units (RTUs), and Programmable Logic Controllers (PLCs), which directly interface with physical processes involving voltage, current, and pressure regulation. Network-level communications are passively captured using probes or mirrored ports strategically positioned between control devices and supervisory servers.

The collected data are subsequently transmitted to the Edge layer, where frames extracted from industrial communication protocols, including DNP3/TCP, Modbus/TCP, and IEC-104, are intercepted and enriched with contextual metadata such as IP addresses, port numbers, function codes, and response times. This enriched information characterizes the temporal and behavioral dynamics of industrial communications and serves as the foundation for traffic analysis and anomaly detection. At this stage, the data also undergoes a series of local preprocessing operations to minimize latency and reduce computational load on higher architectural tiers. These preprocessing steps include data cleaning, categorical attribute encoding, normalization, and variable aggregation. The resulting transformation reorganizes the raw data into a format compatible with model training datasets. Thus, the Edge layer functions as an intelligent intermediary, generating homogeneous, lightweight, and normalized data streams that optimize resource utilization and minimize network bandwidth consumption.

The preprocessed data are then forwarded to the Fog layer, where the xLSTM-based detection model is deployed. This model is designed explicitly for near-real-time identification of anomalies and suspicious behavior in industrial network traffic. It integrates two complementary submodules: sLSTM, which performs fine-grained sequential detection of attack patterns by modeling long-term temporal dependencies, and mLSTM, which leverages matrix-based memory and parallel computation to capture multidimensional correlations and improve generalization. The combined model outputs are processed through a sigmoid activation layer to classify each communication sequence as either normal or malicious. In the event of an anomaly, an automated alert is generated and transmitted to the SCADA supervisory module, such as a Human–Machine Interface (HMI) or a Security Information and Event Management (SIEM) platform, facilitating rapid operator intervention.

At the upper level, the Cloud layer ensures continuous system adaptation and improvement. It archives historical communication data and event logs collected from the Fog layer, thereby constructing a knowledge base that supports periodic retraining of the xLSTM model. This retraining mechanism enhances the system's scalability and resilience against evolving cyber threats and changing network behaviors. The updated model weights and parameters are subsequently propagated back to the Fog layer, ensuring consistent and synchronized deployment of the latest detection capabilities.

Overall, this distributed Edge–Fog–Cloud architecture establishes a hierarchical supervisory framework that seamlessly integrates real-time local analysis, embedded intelligence, and adaptive learning. It achieves an optimal balance between responsiveness, security, and scalability, effectively addressing the performance, resilience, and reliability requirements of modern SCADA and Smart Grid environments.

### 3.7. Dataset

To thoroughly evaluate the proposed intrusion detection system designed for smart grid environments, two well-established benchmark datasets were used: NSL-KDD and DNP3. These datasets were selected for their complementarity: NSL-KDD serves as a foundational benchmark for traditional network intrusion detection, whereas DNP3 captures traffic patterns unique to SCADA protocols, which are critical in industrial control and smart grid systems. Together, they provide a balanced framework for evaluating the system's ability to detect generic and domain-specific cyber threats. The NSL-KDD dataset, created as an improved version of the previous KDD'99 dataset, addresses several known issues, including redundant records and class imbalance, which jeopardized the reliability of its predecessor. It provides a more balanced and cleaner dataset suitable for training and testing machine learning-based intrusion detection systems.

This dataset contains 148,269 labeled instances, each with 41 features representing different network-level, content-based, and temporal characteristics. Among these instances, 77,054 correspond to benign traffic, whereas 71,215 correspond to malicious activity, including Denial-of-Service (DoS), Probe, User-to-Root (U2R), and Remote-to-Local (R2L). The distribution is nearly equal, with benign traffic accounting for roughly 52% of the dataset and attack traffic accounting for 48%. This proportionality enables fair model training and testing without requiring extensive class-balancing techniques.

In addition to NSL-KDD, the DNP3 dataset was used to assess the system's performance with SCADA-specific communication protocols. DNP3 (Distributed Network Protocol 3) is widely used in critical infrastructure, particularly smart grid systems, to enable communication between control centers and field devices. Given its critical role, the protocol has become a target for a variety of cyberattacks, emphasizing the importance of robust detection mechanisms. The ITHACA Research Group at the University of Western Macedonia made the DNP3 dataset publicly available. It consists of 5,994 labeled network flow records, each with 99 features extracted using CICFlowMeter. These features include temporal information, protocol behavior, source and destination IP addresses and ports, as well as DNP3-specific application-layer characteristics.

The dataset contains 666 benign traffic instances and 5,328 attack instances, resulting in a highly skewed distribution, with malicious flows accounting for roughly 89% of the total. The attack types

depicted are varied and real, including Denial-of-Service (DoS), command injection via malformed input, and unauthorized DNP3 control actions. These attack patterns closely resemble threats found in real-world SCADA deployments. Fig. 3 depicts pie charts showing the distributions of attacks and benign traffic in the NSL-KDD and DNP3 datasets, respectively.

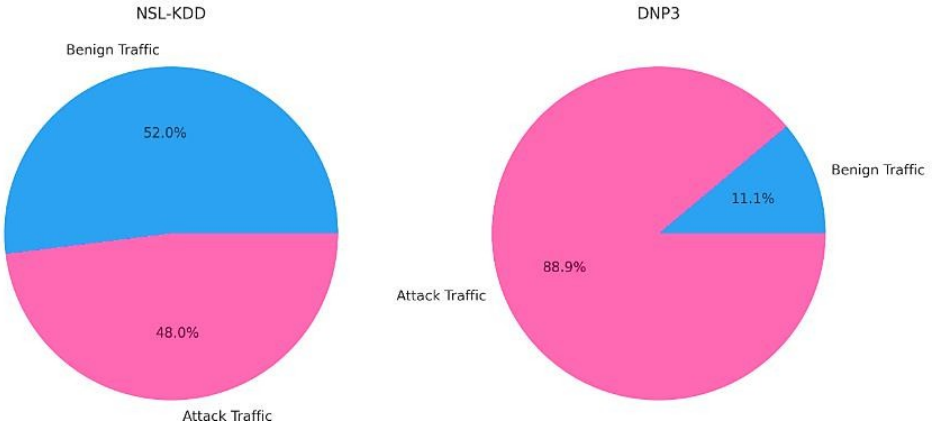


**Fig. 3.** Distribution of benign and attack traffic in the NSL-KDD and DNP3 datasets.

The selection of these two datasets is based on both technical relevance and research reproducibility. The NSL-KDD dataset has been widely used in the literature, enabling direct comparisons with other intrusion detection methods. On the other hand, the DNP3 dataset introduces a domain-specific challenge that assesses the system's ability to detect attacks within operational technologies. This dual-dataset evaluation enables benchmarking of the proposed system against both general-purpose and specialized intrusion-detection tasks, ensuring a comprehensive performance assessment and broader applicability in smart grid cybersecurity scenarios.

### 3.8. Pre-processing

The present study aims to perform binary classification on network traffic data. To prepare the dataset, data cleaning procedures were first applied, and each attack category was combined into a single "attack" label. These labels were converted to numeric values via label encoding, with 0 denoting benign traffic and 1 denoting attack traffic. The categorical features, such as protocol type, service, and flags, were then converted to numeric representations via one-hot encoding. Finally, the resulting numerical features were scaled to a [0, 1] range using Min-Max normalization, as defined by Eq.(23).

$$z_i = y_i \frac{y_i - \min(y)}{\max(y) - \min(y)} \tag{23}$$

where $y_i$ represents the original value of a particular data point for a specific feature; $min(y)$ represents the minimum value observed for the entire set of values (y) of that specific feature in the dataset; and $max(y)$ represents the maximum value observed for the entire set of values (y) of that specific feature in the dataset;

This normalization step converts all features to a standard numerical range, which improves the stability and convergence of training algorithms. After normalization, all available features were retained for model training, as no feature selection was employed. To evaluate model performance, the dataset was split into 80% for training and 20% for testing.

### 3.9. Dataset Balancing

A class balancing strategy was required due to the pronounced class imbalance observed, particularly in the DNP3 dataset, where benign traffic instances outnumbered attack samples by a significant margin. To address this issue, the Synthetic Minority Oversampling Technique (SMOTE) [45] was used. SMOTE creates synthetic instances of the minority class by interpolating between existing samples, enriching the training dataset with more representative examples of underrepresented attack patterns. This approach

exposes the learning algorithm to a more balanced class distribution, which improves its ability to distinguish between normal and malicious traffic. Following the implementation of SMOTE, the number of attack samples in the NSL-KDD dataset increased from 57,170 to 61,643, in line with the number of benign instances. Similarly, in the DNP3 dataset, the normal class was up-sampled from 533 to 4,262 to match the attack class.

### 3.10. Models Training Parameters

To enhance the performance and generalization capability of intrusion detection models, a structured training and regularization framework was used. Each model, sLSTM and mLSTM, had a single hidden layer, whereas the xLSTM model combined the best of both by stacking an sLSTM layer followed by an mLSTM layer. Training configurations were selected to maximize learning efficiency while remaining compatible with both GPU and CPU execution environments. Table 2 summarizes the key training parameters.

**Table 2.** Model training configuration.

| Parameter | Value |
|---|---|
| Head size | 16 |
| Number of heads | 8 |
| Batch size | 128 |
| Number of Epochs | 10 |
| Learning rate (LR) | 0.001 |
| Optimizer | Adam |
| Loss function | BCEWithLogitsLoss |
| Device | GPU (CUDA) / CPU |

Several regularization strategies were used to improve generalization while reducing overfitting. These techniques improve model stability and prevent performance degradation on unseen data. Table 3 displays the regularization settings.

**Table 3.** Regularization techniques and settings.

| Regularization strategy | Parameter/Value |
|---|---|
| Dropout | 0.5 |
| L2 Regularization | 1e-3 (0.001) |
| Early stopping | Patience: 7 epochs |
| Learning rate scheduler | ReduceLROnPlateau |
| Gradient clipping | Norm: 1.0 |
| Cross-validation | 5-fold |

### 3.11. Evaluation metrics

Standard performance metrics were used to evaluate the proposed model's ability to detect malicious connections. The metrics used were Accuracy, Precision, Recall, F1-score, and the confusion matrix. These metrics provide a comprehensive evaluation of the model, considering both its overall performance and its ability to reduce critical errors. Accuracy is the proportion of correct predictions among all predictions, as defined in Eq. (24).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{24}$$

Precision given by Eq. (25) is the ratio of correctly predicted positive cases (true positives) to all predicted positives. This is especially important when the cost of false positives (false alarms) is high, as an excessive number of false positives can overwhelm system administrators with unnecessary alerts.

$$Precisions = \frac{TP}{TP+FP} \tag{25}$$

Recall given by Eq. (26), also known as sensitivity, quantifies the model's ability to identify malicious attacks correctly. This metric is critical in an IDS context because a low Recall, which results in many false negatives, may allow undetected attacks to compromise the system.

$$Recall = \frac{TP}{TP+FN} \quad (26)$$

F1-score given by Eq. (27) strikes a balance between Precision and Recall, making it particularly useful when both false positives and false negatives must be reduced. This metric is commonly used in situations where neither Precision nor Recall should be overemphasized.

$$F1 = 2 * \frac{Precision.Recall}{Precision+Recall} \quad (27)$$

## 4. Results and Discussion

To ensure the reliability of the results, the performance of the different memory architectures (SLSTM, MLSTM, and xLSTM) was evaluated using 5-fold cross-validation. Table 4 and Table 5 synthesize the mean Accuracy and statistical robustness indicators for the three models on the NSL-KDD and DNP3 datasets, respectively, thereby enabling an in-depth comparative assessment.

**Table 4.** Cross-Validation (K=5) Results for sLSTM, mLSTM, and xLSTM Models on DNP3

| Performance Metric | sLSTM | mLSTM | xLSTM |
|---|---|---|---|
| Mean Accuracy | 99.16% | 98.45% | 99.08% |
| Standard Deviation (STD) | 0.34 | 0.63 | 0.25 |
| Variance | 0.001127 | 0.003953 | 0.000623 |
| 95% Confidence Interval (CI) | [98.75%–99.58%] | [97.67%–99.23%] | [98.77%–99.39%] |
| Statistical Uncertainty (±) | ±0.42% | ±0.78% | ±0.31% |

**Table 5.** Cross-Validation (K=5) Results for sLSTM, mLSTM, and xLSTM Models on NSL-KDD

| Performance Metric | sLSTM | Mlstm | Xlstm |
|---|---|---|---|
| Mean Accuracy | 98.78% | 98.64% | 97.96% |
| Standard Deviation (STD) | 0.15 | 0.25 | 0.16 |
| Variance | 0.000228 | 0.000615 | 0.000250 |
| 95% Confidence Interval (CI) | [98.59%–98.97%] | [98.33%–98.95%] | [97.77%–98.16%] |
| Statistical Uncertainty (±) | ±0.19% | ±0.31% | ±0.20% |

The analysis of the 5-fold Cross-Validation (CV) results reveals that all three models exhibit excellent overall performance, with distinct trade-offs between efficacy (accuracy potential) and stability (consistency). On the DNP3 dataset, the sLSTM model achieves the highest Mean Accuracy (99.16%), setting the benchmark for detection performance. However, its Standard Deviation (0.34%) indicates moderate variability across folds.

In contrast, the xLSTM model demonstrates superior robustness, with the lowest Standard Deviation (0.25%) and the tightest Confidence Interval ([98.77%–99.39%]), ensuring consistent performance across data splits. Although its Mean Accuracy (99.08%) is slightly lower than that of sLSTM, this marginal difference is offset by its greater stability, making it a highly reliable candidate for deployment. The mLSTM shows a slightly lower Mean Accuracy (98.45%) and the highest variability (STD 0.63%) in cross-validation, suggesting greater sensitivity to data partitions rather than poor generalization. Its final test performance, discussed later, provides further insight into its true generalization capability.

On the NSL-KDD dataset, a comparative evaluation confirms that sLSTM achieves the highest Mean Accuracy (98.78%), the lowest Standard Deviation (0.15%), and the tightest 95% Confidence Interval ([98.59%–98.97%]). These results highlight its excellent balance between predictive accuracy and stability within cross-validation, outperforming both mLSTM and xLSTM under these conditions.

The confusion matrix contributes to this analysis by providing a detailed breakdown of correct predictions, true positives, true negatives, and errors, including false positives and false negatives. This representation clarifies where the model makes errors and highlights areas for improvement. The

confusion matrices for the three models on the NSL-KDD and DNP3 datasets are shown in Fig. 4 and Fig. 5, respectively.

The confusion matrix consists of four fundamental components that describe classification performance. True Negatives (TN) represent the number of Normal activity instances that are correctly classified as Normal. False Positives (FP) denote Normal activity instances that are incorrectly classified as an Attack, commonly referred to as false alarms. False Negatives (FN) correspond to Attack instances that are misclassified as Normal, indicating missed intrusions. Finally, True Positives (TP) indicate the number of Attack instances that are correctly identified as Attacks.

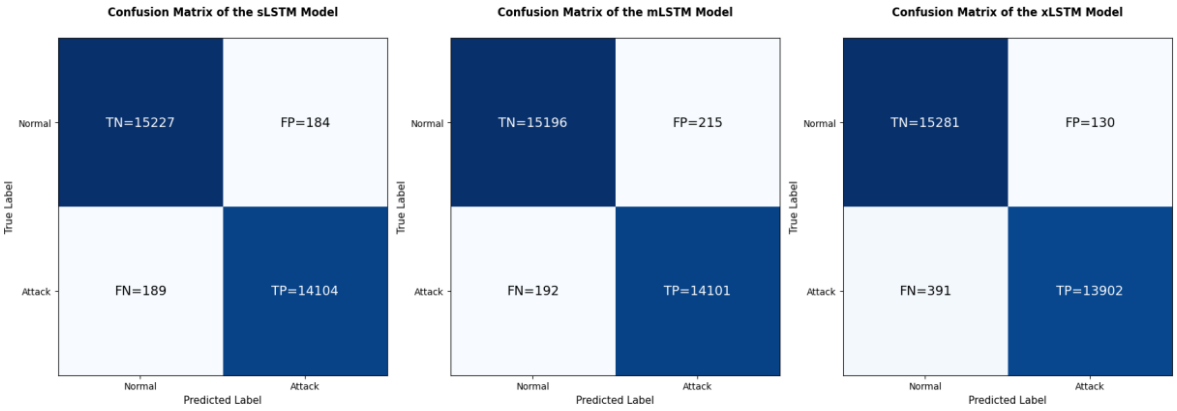


**Fig. 4.** Confusion matrices illustrating the classification performance of the sLSTM, mLSTM, and xLSTM models on the NSL-KDD dataset.

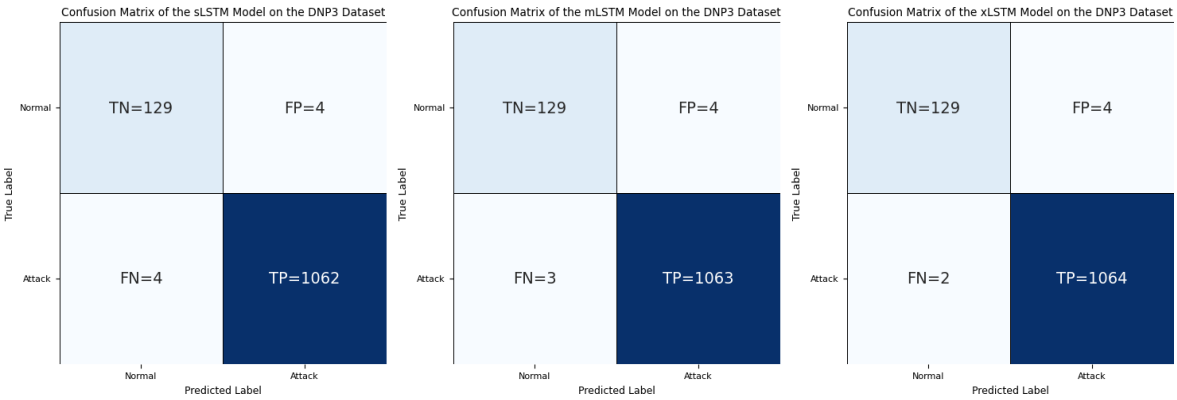


**Fig. 5.** Confusion matrices illustrating the classification performance of the sLSTM, mLSTM, and xLSTM models on the DNP3 dataset.

To evaluate model performance beyond a single classification threshold, we examine the Receiver Operating Characteristic (ROC) Curves. This graphical and quantitative approach analyzes the discriminatory capability across all possible thresholds, and its key indicator, the Area Under the Curve (AUC), confirms the robustness and generalization capacity of our architectures. Fig. 6 and Fig. 7 present the ROC curves and corresponding AUC values for evaluating our models on the DNP3 and NSL-KDD datasets, respectively. The Receiver Operating Characteristic (ROC) curve analysis further reinforces the findings from the confusion matrices, confirming the high robustness and discriminative capability of the sLSTM, mLSTM, and xLSTM architectures for intrusion detection. The computed Area Under the Curve (AUC) values are remarkably close to unity, with the xLSTM achieving 0.9985, and comparable results were observed for the other variants. These results indicate an almost perfect ability to distinguish between regular traffic and attack events across all classification thresholds.

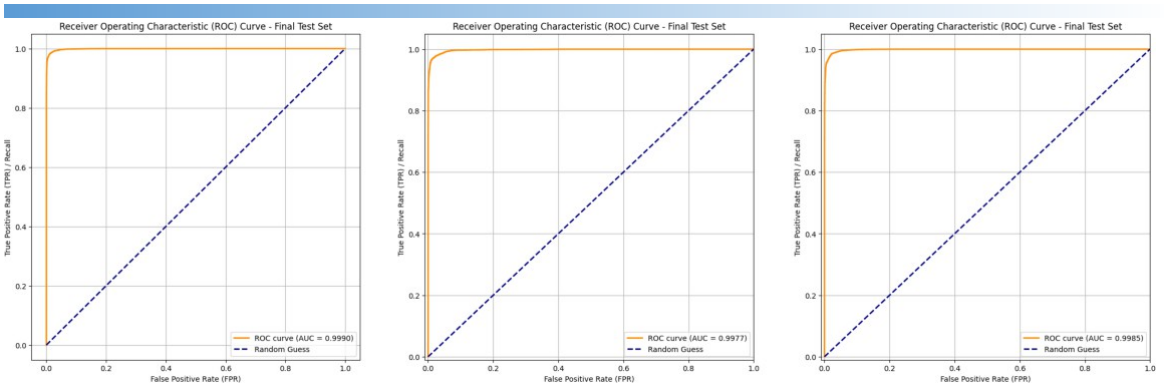


**Fig. 6.** ROC-AUC Evaluation of Recurrent Models (sLSTM, mLSTM, xLSTM) on the NSL-KDD Dataset

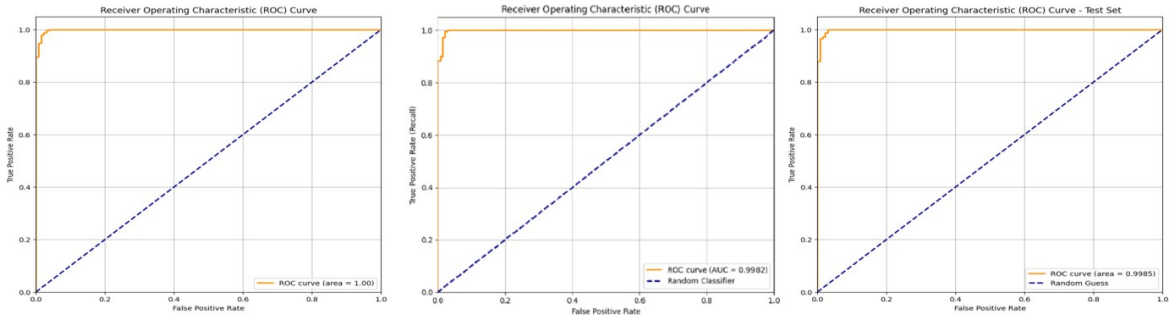


**Fig. 7.** ROC-AUC Evaluation of Recurrent Models (sLSTM, mLSTM, xLSTM) on the DNP3 Dataset

Visually, the ROC curves are positioned near the upper-left corner of the plot, indicating that the models maintain a high True Positive Rate (TPR) while minimizing the False Positive Rate (FPR). This outstanding performance arises from two key factors: (1) the strong discriminative features present in both DNP3 and NSL-KDD datasets, and (2) the advanced temporal modeling capabilities of the LSTM-based architectures, which effectively capture long-term dependencies and subtle sequential patterns in smart grid traffic. Consequently, the high AUC values provide compelling statistical evidence that the models have learned a robust and generalizable decision boundary rather than relying on random guessing. The final results, which confirm the models' performance on both the DNP3 and NSL-KDD datasets, are presented in Table 6.

**Table 6.** Performance comparison of LSTM variants on NSL-KDD and DNP3 datasets.

| Model name | Dataset | Precision(%) | Recall(%) | F1-score(%) | Accuracy(%) | Training time(seconds) | Test time(seconds) |
|---|---|---|---|---|---|---|---|
| Slstm | | 98.55 | 98.55 | 98.55 | 98.55 | 147.35 | 1.02 |
| Mlstm | NSLKDD | 98.55 | 98.52 | 98.53 | 98.54 | 96.25 | 0.71 |
| xLSTM | | 98.15 | 98.17 | 98.16 | 98.16 | 203.38 | 1.31 |
| sLSTM | | 98.31 | 98.31 | 98.31 | 99.33 | 11.19 | 0.2736 |
| mLSTM | DNP3 | 98.68 | 98.36 | 98.52 | 99.42 | 9.70 | 0.0629 |
| xLSTM | | 99.05 | 98.40 | 98.72 | 99.50 | 16.39 | 0.28 |

On the NSL-KDD dataset, a widely recognized benchmark with well-defined, structured attack patterns, all three LSTM variants achieved consistently high scores across key evaluation metrics, including Precision, Recall, F1-score, and Accuracy. This evaluation confirms the models' overall efficacy in identifying known intrusion types. The sLSTM model outperformed mLSTM and xLSTM in accuracy and F1-score, but the differences were minimal, indicating that all models performed similarly on this dataset. In contrast, performance on the DNP3 dataset is awe-inspiring. DNP3 is a foundational communication standard for Supervisory Control and Data Acquisition (SCADA) systems, which are critical to Smart Grid infrastructures. This study uses a DNP3 dataset built from DNP3-specific traffic. It's well-suited for testing intrusion detection in ICS and critical infrastructure environments. Across this dataset, all models performed exceptionally well, with Precision, Recall, F1-score, and Accuracy exceeding 98%, demonstrating their ability to distinguish benign traffic from malicious activity targeting

DNP3 operations. A detailed comparative analysis based on each model's architectural characteristics provides additional insight into these findings:

sLSTM performance: the sLSTM model builds on the standard LSTM by incorporating exponential gating, a memory mixing mechanism, and multi-head blocks with block-diagonal weight matrices. These innovations enable stable hidden state updates and capture complex temporal dependencies, resulting in high performance (98.31% Precision and 98.31% F1-score). mLSTM performance: the mLSTM model demonstrated strong performance on the DNP3 dataset. It replaces scalar memory cells with matrix-based memory, a design choice that enables it to generalize more effectively across heterogeneous traffic patterns in DNP3. Although the model's advantage over the sLSTM is modest (+0.21% higher F1-score; +0.37 % higher Precision; +0.09 % higher accuracy), this improvement suggests a better capacity to model the intricate dependencies and structural complexity inherent in DNP3-based attacks. xLSTM as the top performer: The xLSTM model clearly outperforms the two previous variants, sLSTM and mLSTM, across all evaluated metrics. In terms of precision, it achieves a gain of +0.74% over sLSTM and +0.37% over mLSTM, reaching 99.05%. The recall also shows a slight improvement, with increases of +0.09% and +0.04%, respectively. The F1-score rises by +0.41% compared to sLSTM and +0.20% compared to mLSTM, indicating better consistency between precision and sensitivity. Finally, the accuracy of xLSTM reaches 99.50%, representing a +0.17% gain over both architectures. Its architecture combines the innovations of sLSTM and mLSTM via sequential xLSTM blocks, each with residual connections and up-projection mechanisms, despite being designed for long-context sequence modeling and memory-intensive tasks.

In terms of computational efficiency, the mLSTM achieves the highest training and inference speeds among the architectures evaluated. Specifically, it achieves training and testing times of 9.70 s and 0.0629 s, respectively, on the DNP3 dataset. This superior efficiency primarily stems from its matrix-based memory structure, which facilitates parallelization and thereby accelerates computation. The sLSTM model, characterized by reduced architectural complexity, has a slightly longer training duration of 11,19 s and a testing time of 0,2736 s. These results confirm the model's lightweight design and its suitability for rapid execution in resource-constrained environments. Conversely, the xLSTM, which synergistically combines sLSTM and mLSTM blocks while incorporating exponential gating and enriched memory mechanisms, exhibits higher computational demands, with training and testing times of 16,39 s and 0,28 s, respectively. Despite this additional cost, the xLSTM achieves an optimal balance between computational complexity and detection performance, substantially improving temporal stability and classification accuracy.

These findings indicate that the internal optimization strategies of xLSTM, although more computationally intensive during training, yield superior temporal- and protocol-level modeling of DNP3 traffic. Such capabilities are crucial for advanced intrusion detection and enable effective real-time deployment in SCADA-based smart grid environments. The superior performance of the xLSTM model on the DNP3 dataset, compared with NSL-KDD, can be directly attributed to the data's intrinsic structural complexity. DNP3 traffic, which characterizes SCADA communications, exhibits a richer internal relational structure and stronger contextual dependencies at the attribute level (99 attributes versus 41 for NSL-KDD). The xLSTM architecture is specifically designed to capture such sophisticated multidimensional associations, effectively modeling both intra-step correlations and long-range temporal dependencies. In contrast, the NSL-KDD dataset, developed as a general-purpose benchmark for conventional network traffic, presents a simpler feature space with lower relational complexity. Under these conditions, the advanced capabilities of matrix-based memory become largely redundant, making a lighter and more stable architecture, such as sLSTM, not only sufficient but also slightly more effective.

Across all models, high recall ensures that a small number of attacks escape detection, which is critical for preventing security breaches in smart grid systems. Concurrently, high precision reduces false positives, which is critical for maintaining operational trust and avoiding alarm fatigue. These findings support the incorporation of these architectures into Intrusion Detection Systems (IDS) for smart grids, providing an effective and reliable solution that can adapt to evolving threats in dynamic critical-infrastructure environments.

Table 7 compares these advanced LSTM variants with state-of-the-art methods and demonstrates significant improvements in intrusion detection performance. These models, which incorporate enhanced memory mechanisms and architectural innovations, have demonstrated a superior ability to capture long-term temporal dependencies in sequential network traffic. This capability enables the detection of nuanced and time-dependent malicious behaviors that would otherwise go undetected by models with limited temporal context.

**Table 7.** Comparison with state-of-the-art methods

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | Dataset | Year | Reference |
|---|---|---|---|---|---|---|---|
| CNN–GRU– FL | 78.79 | 97.33 | 64.15 | 76.90 | NSL-KDD | 2023 | [46] |
| CNN-GRU-FF | 99.54 | 98.40 | 98.22 | 98.28 | UNSW-NB15 | 2024 | [33] |
| A-BiTG | 96.23 | 95.47 | 97.27 | 96. 37 | IEEE 14 | 2024 | [41] |
| A-BiTG | 95.77 | 95.77 | 95.14 | 96.96 | IEEE 118 | 2024 | [41] |
| CNN | 98.9 | 99 | 99 | 99 | DNP3 | 2025 | [47] |
| **sLSTM** | **98.55**<br>**99.33** | **98.55**<br>**98.31** | **98.55**<br>**98.31** | **98.55**<br>**98.31** | **NSL-Kdd**<br>**DNP3** | **2025** | **This paper** |
| **mLSTM** | **98.54**<br>**99.42** | **98.55**<br>**98.68** | **98.52**<br>**98.36** | **98.53**<br>**98.52** | **NSL-Kdd**<br>**DNP3** | **2025** | **This paper** |
| **xLSTM** | **98.16**<br>**99.50** | **98.15**<br>**99.05** | **98.17**<br>**98.40** | **98.16**<br>**98 .72** | **NSL-Kdd**<br>**DNP3** | **2025** | **This paper** |

Compared with general deep learning approaches such as CNN-GRU-FL [46] and CNN-GRU-FF [33], the proposed models perform comparably or better across both datasets. Furthermore, compared with hybrid and specialized architectures such as A-BiTG [41] and CNN-based models [47], the sLSTM, mLSTM, and xLSTM consistently produce marginally higher or comparable F1 Scores, a key indicator of detection efficacy at controlled false-positive rates. The mLSTM model stood out as the best performer, particularly on the DNP3 dataset, thanks to its matrix-based memory structure and covariance-aware update mechanisms. The robustness and adaptability of these models across various intrusion detection scenarios demonstrate their usefulness in real-world applications, particularly in dynamic environments such as Smart Grids and ICS networks. These findings confirm that deep memory architectures are a promising avenue for improving the precision and reliability of next-generation intrusion detection systems.

## 5. Conclusion

This study examined the use of extended Long Short-Term Memory (LSTM) architectures, specifically xLSTM, sLSTM, and mLSTM, to improve intrusion detection in smart grid environments. The fundamental novelty of xLSTM lies in introducing exponential gating and novel memory structures (sLSTM and mLSTM), which enable the model to revise storage decisions and explicitly increase its memory capacity. This innovation positions the architecture as a powerful alternative, maintaining the linear time complexity of LSTMs while competing with Attention-based models in computational efficiency and scalability through parallelization of mLSTM. The evaluation was conducted using two widely recognized benchmark datasets: NSL-KDD and DNP3. The proposed models demonstrated consistently high performance, robustness, and generalization capabilities across both datasets. The models (xLSTM, sLSTM, and mLSTM) demonstrated superior performance on the NSL-KDD dataset, with accuracy exceeding 98.1%. When tested on the DNP3 dataset, which simulates real-world cyberattack scenarios targeting the DNP3 protocol commonly used in SCADA systems, the models demonstrated exceptional performance, with Accuracy values exceeding 99.3%. These findings support the models' suitability for intrusion detection in critical infrastructure, particularly Smart Grid networks, where resilience to sophisticated attacks is critical. Despite demonstrating consistently high performance on critical infrastructure benchmarks, this study faces several important limitations that must be addressed to ensure practical deployment in real-world SCADA environments. These limitations include: (1) the absence of validation regarding real-time end-to-end latency in live operational settings; (2) the lack of evaluation of adversarial robustness against malicious or manipulated data; (3) the untested

scalability of the mLSTM and xLSTM architectures, particularly concerning matrix memory overhead, on large- scale streaming datasets; and (4) the unassessed impact of network noise, data corruption, and zero- day attacks on long-term model stability. Future research will extend this framework to additional industrial communication protocols, such as Modbus/TCP and IEC 61850, to ensure broader interoperability across the diverse protocols used in SCADA systems for Smart Grids. A key future direction involves the real-time integration and validation of the proposed model within operational SCADA environments, enabling direct interaction with live DNP3 traffic. These future directions aim to enhance cybersecurity in intelligent energy systems and to accelerate the development of adaptive, scalable detection mechanisms for evolving cyber threats.

## Declarations

**Author contribution.** S. Baalia executed the implementation, data preprocessing, model Development, and experimental assessment. D. Boughareb and Z. Kouahla contributed to the methodology design and oversaw the entire research process. H. Seridi and all co-authors participated in the review of the work and engaged in technical discussions to enhance and refine the manuscript

**Funding statement.** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Conflict of interest.** The authors declare that there is no conflict of interest regarding the publication of this paper.

**Additional information.** No additional information is available for this paper.

## References

[1] F. M. Almasoudi, "Enhancing Power Grid Resilience through Real-Time Fault Detection and Remediation Using Advanced Hybrid Machine Learning Models," *Sustainability*, vol. 15, no. 10, p. 8348, May 2023, doi: 10.3390/su15108348.

[2] S. Amanlou *et al.*, "Cybersecurity Challenges in Smart Grid Systems: Current and Emerging Attacks, Opportunities, and Recommendations," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 1965–1997, 2025, doi: 10.1109/OJCOMS.2025.3545153.

[3] I. Fursov, K. Yamkovyi, and O. Shmatko, "Smart Grid and wind generators: an overview of cyber threats and vulnerabilities of power supply networks," *Radioelectron. Comput. Syst.*, vol. 0, no. 4, pp. 50–63, Nov. 2022, doi: 10.32620/reks.2022.4.04.

[4] A. Yahia, A. Tag Eldien, and N. M. Abdel-Rahim, "Deep Learning based Attacks Detection of DNP3 Protocol," *Aswan Univ. J. Sci. Technol.*, vol. 2, no. 2, pp. 37–47, Dec. 2022, doi: 10.21608/aujst.2022.174148.1003.

[5] B. Al-Muntaser, M. A. Mohamed, A. Y. Tuama, and I. A. Rana, "Cybersecurity Advances in SCADA Systems," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 8, pp. 318–328, Aug. 2023, doi: 10.14569/IJACSA.2023.0140835.

[6] M. Zaman, D. Upadhyay, and C.-H. Lung, "Validation of a Machine Learning-Based IDS Design Framework Using ORNL Datasets for Power System With SCADA," *IEEE Access*, vol. 11, pp. 118414–118426, 2023, doi: 10.1109/ACCESS.2023.3326751.

[7] Sangeetha K., Shitharth S., and G. B. Mohammed, "Enhanced SCADA IDS Security by Using MSOM Hybrid Unsupervised Algorithm," *Int. J. Web-Based Learn. Teach. Technol.*, vol. 17, no. 2, pp. 1–9, Mar. 2022, doi: 10.4018/IJWLTT.20220301.oa2.

[8] A. Balla, M. H. Habaebi, E. A. A. Elsheikh, M. R. Islam, and F. M. Suliman, "The Effect of Dataset Imbalance on the Performance of SCADA Intrusion Detection Systems," *Sensors*, vol. 23, no. 2, p. 758, Jan. 2023, doi: 10.3390/s23020758.

[9] M. Altaha and S. Hong, "Anomaly Detection for SCADA System Security Based on Unsupervised Learning and Function Codes Analysis in the DNP3 Protocol," *Electronics*, vol. 11, no. 14, p. 2184, Jul. 2022, doi: 10.3390/electronics11142184.

[10] D. Faquir *et al.*, "Cybersecurity in smart grids, challenges and solutions," *AIMS Electron. Electr. Eng. 2021 124*, vol. 5, no. 1, pp. 24–37, 2021, doi: 10.3934/ELECTRENG.2021002.

[11] S. Tufail, I. Parvez, S. Batool, and A. Sarwat, "A Survey on Cybersecurity Challenges, Detection, and Mitigation Techniques for the Smart Grid," *Energies*, vol. 14, no. 18, p. 5894, Sep. 2021, doi: 10.3390/en14185894.

[12] A. A. Bouramdane, "Cyberattacks in Smart Grids: Challenges and Solving the Multi-Criteria Decision-Making for Cybersecurity Options, Including Ones That Incorporate Artificial Intelligence, Using an Analytical Hierarchy Process," *J. Cybersecurity Priv.*, vol. 3, no. 4, pp. 662–705, Sep. 2023, doi: 10.3390/jcp3040031.

[13] P. Haji Mirzaee, M. Shojafar, H. Cruickshank, and R. Tafazolli, "Smart Grid Security and Privacy: From Conventional to Machine Learning Issues (Threats and Countermeasures)," *IEEE Access*, vol. 10, pp. 52922–52954, 2022, doi: 10.1109/ACCESS.2022.3174259.

[14] A. Alsaiari and M. Ilyas, "Deep Learning for Smart Grid Intrusion Detection: A Hybrid CNN-LSTM-Based Model," *Int. J. Artif. Intell. Appl.*, vol. 15, no. 3, pp. 01–16, May 2024, doi: 10.5121/ijaia.2024.15301.

[15] A. Subasi *et al.*, "Intrusion Detection in Smart Grid Using Data Mining Techniques," in *2018 21st Saudi Computer Society National Computer Conference (NCC)*, IEEE, Apr. 2018, pp. 1–6. doi: 10.1109/NCG.2018.8593124.

[16] J. Sakhnini, H. Karimipour, and A. Dehghantanha, "Smart Grid Cyber Attacks Detection Using Supervised Learning and Heuristic Feature Selection," in *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*, IEEE, Aug. 2019, pp. 108–112. doi: 10.1109/SEGE.2019.8859946.

[17] S. Khan, K. Kifayat, A. Kashif Bashir, A. Gurtov, and M. Hassan, "Intelligent intrusion detection system in smart grid using computational intelligence and machine learning," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 6, p. e4062, Jun. 2020, doi: 10.1002/ett.4062.

[18] T. Talaei Khoei and N. Kaabouch, "A Comparative Analysis of Supervised and Unsupervised Models for Detecting Attacks on the Intrusion Detection Systems," *Information*, vol. 14, no. 2, p. 103, Feb. 2023, doi: 10.3390/info14020103.

[19] P. Ganesan and S. Arockia Edwin Xavier, "An Intelligent Intrusion Detection System in Smart Grid Using PRNN Classifier," *Intell. Autom. Soft Comput.*, vol. 35, no. 3, pp. 2979–2996, Aug. 2023, doi: 10.32604/iasc.2023.029264.

[20] S. Stryczek and M. Natkaniec, "Internet Threat Detection in Smart Grids Based on Network Traffic Analysis Using LSTM, IF, and SVM," *Energies*, vol. 16, no. 1, p. 329, Dec. 2022, doi: 10.3390/en16010329.

[21] P. Kumar, R. Kumar, A. Aljuhani, D. Javeed, A. Jolfaei, and A. K. M. N. Islam, "Digital twin-driven SDN for smart grid: A deep learning integrated blockchain for cybersecurity," *Sol. Energy*, vol. 263, no. October, p. 111921, Oct. 2023, doi: 10.1016/j.solener.2023.111921.

[22] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Comput. Commun.*, vol. 199, no. February, pp. 113–125, Feb. 2023, doi: 10.1016/j.comcom.2022.12.010.

[23] G. Efstathopoulos *et al.*, "Operational Data Based Intrusion Detection System for Smart Grid," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, IEEE, Sep. 2019, pp. 1–6. doi: 10.1109/CAMAD.2019.8858503.

[24] B. Hu, J. Wang, Y. Zhu, and T. Yang, "Dynamic Deep Forest: An Ensemble Classification Method for Network Intrusion Detection," *Electronics*, vol. 8, no. 9, p. 968, Aug. 2019, doi: 10.3390/electronics8090968.

[25] S. Jose, D. Malathi, B. Reddy, and D. Jayaseeli, "A Survey on Anomaly Based Host Intrusion Detection System," *J. Phys. Conf. Ser.*, vol. 1000, no. 1, p. 012049, Apr. 2018, doi: 10.1088/1742-6596/1000/1/012049.

[26] P. Jokar, H. Nicanfar, and V. C. M. Leung, "Specification-based Intrusion Detection for home area networks in smart grids," in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, Oct. 2011, pp. 208–213. doi: 10.1109/SmartGridComm.2011.6102320.

[27] E. Drayer and T. Routtenberg, "Intrusion Detection in Smart Grid Measurement Infrastructures Based on Principal Component Analysis," in *2019 IEEE Milan PowerTech*, IEEE, Jun. 2019, pp. 1–6. doi: 10.1109/PTC.2019.8810858.

[28] R. Vijayanand, D. Devaraj, and B. Kannapiran, "A Novel Deep Learning Based Intrusion Detection System for Smart Meter Communication Network," in *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, IEEE, Apr. 2019, pp. 1–3. doi: 10.1109/INCOS45849.2019.8951344.

[29] S. Ali and Y. Li, "Learning Multilevel Auto-Encoders for DDoS Attack Detection in Smart Grid Network," *IEEE Access*, vol. 7, pp. 108647–108659, 2019, doi: 10.1109/ACCESS.2019.2933304.

[30] R. Qi, C. Rasband, J. Zheng, and R. Longoria, "Detecting Cyber Attacks in Smart Grids Using Semi-Supervised Anomaly Detection and Deep Representation Learning," *Information*, vol. 12, no. 8, p. 328, Aug. 2021, doi: 10.3390/info12080328.

[31] F. Martinelli, F. Mercaldo, and A. Santone, "A Method for Intrusion Detection in Smart Grid," *Procedia Comput. Sci.*, vol. 207, pp. 327–334, Jan. 2022, doi: 10.1016/J.PROCS.2022.09.066.

[32] U. AlHaddad, A. Basuhail, M. Khemakhem, F. E. Eassa, and K. Jambi, "Ensemble Model Based on Hybrid Deep Learning for Intrusion Detection in Smart Grid Networks," *Sensors*, vol. 23, no. 17, p. 7464, Aug. 2023, doi: 10.3390/s23177464.

[33] Y. Imrana, Y. Xiang, L. Ali, A. Noor, K. Sarpong, and M. A. Abdullah, "CNN-GRU-FF: a double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units," *Complex Intell. Syst.*, vol. 10, no. 3, pp. 3353–3370, Jun. 2024, doi: 10.1007/S40747-023-01313-Y/TABLES/14.

[34] S. H. Mohammed *et al.*, "Dual-hybrid intrusion detection system to detect False Data Injection in smart grids," *PLoS One*, vol. 20, no. 1, p. e0316536, Jan. 2025, doi: 10.1371/journal.pone.0316536.

[35] L. Basheer and R. P, "A deep learning framework for intrusion detection system in smart grids using graph convolutional network," *Eng. Res. Express*, vol. 7, no. 1, p. 015257, Mar. 2025, doi: 10.1088/2631-8695/adb3f4.

[36] Y. Wu *et al.*, "Graph attention and Kolmogorov–Arnold network based smart grids intrusion detection," *Sci. Rep.*, vol. 15, no. 1, p. 8648, Mar. 2025, doi: 10.1038/s41598-025-88054-9.

[37] A. Almalawi, S. Hassan, A. Fahad, A. Iqbal, and A. I. Khan, "Hybrid Cybersecurity for Asymmetric Threats: Intrusion Detection and SCADA System Protection Innovations," *Symmetry (Basel).*, vol. 17, no. 4, p. 616, Apr. 2025, doi: 10.3390/sym17040616.

[38] F. A. F. Alrslani *et al.*, "Enhancing cybersecurity via attribute reduction with deep learning model for false data injection attack recognition," *Sci. Rep.*, vol. 15, no. 1, p. 3944, Jan. 2025, doi: 10.1038/s41598-024-82566-6.

[39] N. Al-Qirim, M. Majdalawieh, A. Bani-hani, and H. Al Hamadi, "Cyber threat intelligence for smart grids using knowledge graphs, digital twins, and hybrid machine learning in SCADA networks," *Int. J. Eng. Bus. Manag.*, vol. 17, pp. 1–15, Jan. 2025, doi: 10.1177/18479790251328183.

[40] J. Qiu, X. Zhang, T. Wang, H. Hou, S. Wang, and T. Yang, "A GNN-Based False Data Detection Scheme for Smart Grids," *Algorithms*, vol. 18, no. 3, p. 166, Mar. 2025, doi: 10.3390/a18030166.

[41] W. He, W. Liu, C. Wen, and Q. Yang, "Detection of False Data Injection Attacks on Smart Grids Based on A-BiTG Approach," *Electronics*, vol. 13, no. 10, p. 1938, May 2024, doi: 10.3390/electronics13101938.

[42] K. Ohno, S. Kanai, and Y. Ida, "Fast Saturating Gate for Learning Long Time Scales with Recurrent Neural Networks," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 8, pp. 9319–9326, Jun. 2023, doi: 10.1609/aaai.v37i8.26117.

[43] A. Auer *et al.*, "xLSTM: Extended Long Short-Term Memory," in *Advances in Neural Information Processing Systems 37*, San Diego, California, USA: Neural Information Processing Systems Foundation, Inc. (NeurIPS), May 2024, pp. 107547–107603. doi: 10.52202/079017-3417.

[44] P. Baghdadi, S. Korukoglu, M. A. Bilici, and A. Onan, "The Potential of Energy-Based RBM and xLSTM for Real-Time Predictive Analytics in Credit Card Fraud Detection," *J. Data Anal. Inf. Process.*, vol. 13, no. 01, pp. 79–100, Feb. 2025, doi: 10.4236/jdaip.2025.131005.

[45] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[46] F. Zhai, T. Yang, H. Chen, B. He, and S. Li, "Intrusion Detection Method Based on CNN–GRU–FL in a Smart Grid Environment," *Electronics*, vol. 12, no. 5, p. 1164, Feb. 2023, doi: 10.3390/electronics12051164.

[47] K. Khattab and K. M. A. Alheeti, "Enhancing DNP3 Security Using CNN Deep Learning Techniques," *J. Cybersecurity Inf. Manag.*, vol. 15, no. 2, pp. 225–232, 2025, doi: 10.54216/JCIM.150217.