International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 122-151
https://doi.org/10.70917/ijcisim-2026-2237
Article

Hybrid Autoencoder–CNN–BiLSTM–Attention Model for Smart Grid Cybersecurity Intrusion Detection

Bharti Ahuja[1], Nitika Vats Doohan[2]

[1,2]Department of Computer Science and Engineering, SAGE University Indore, Madhya Pradesh ,India
bhartiahujaedu@gmail.com, nitika.doohan@gmail.com

Abstract: The convergence of communication networks and smart electronic devices makes smart grid infrastructures vulnerable to cyber-attacks. Consequently, we need proper intrusion detection mechanisms to ensure secure and reliable grid operation. The present study proposes two advanced hybrid deep learning-based smart grid intrusion detection frameworks, namely optimized Autoencoder–CNN–BiLSTM–Random Forest and Denoising Autoencoder–CNN–BiLSTM–Attention-Random Forest model for the same. The proposed approach involves data preprocessing, scaling of features and latent feature extraction via autoencoder based representation learning. To capture the key spatial and temporal patterns, CNN (Convolutional Neural Networks) and BiLSTM networks are used to process on sequential latent features along with an attention mechanism that emphasizes significant temporal relations. A Random Forest classifier finally performs multiclass intrusion detection. Experimental results depict superior performance of the optimized AE–CNN–BiLSTM–RF model with 95.23% testing accuracy and the proposed DAE–CNN–BiLSTM–Attention–RF data-based approach, achieving accuracy at 99.24% with high precision, recall and F1-scores (close to 0.99) on baseline models such as LGBM (93%) and One-Class SVM (85%).

Keywords: Smart Grid Security; Intrusion Detection System; Hybrid Deep Learning; Denoising Autoencoder; CNN–BiLSTM; Attention Mechanism; Random Forest; Cyber-Physical Systems.

1. Introduction

The smart grid has introduced an overlay of advanced communication networks and intelligent electronic devices as well as distributed energy resources that improve the efficiency, reliability, and automation of power systems. Yet the growing interconnectivity of cyber infrastructure with physical power systems also opens up major cybersecurity risks. Grid stability and power delivery can be impacted by cyber-attacks which include data injections, replay attacks, masquerade attacks, and communication manipulation. Thus, ensuring safe operation of smart grid cyber–physical systems has emerged as a vital research problem in recent years [1], [2]. Intrusion detection systems (IDS) have arisen as a critical defense method in recognizing malicious activities and defending smart grid infrastructures against cyber threats.

Moreover, existing IDS techniques in a smart grid environment usually base on some traditional machine learning algorithms, e.g., Support Vector Machine (SVM), Random Forest and gradient boosting. While these approaches can detect anomalous activity in the network, they rely heavily on handcrafted features and have difficulty capturing complicated nonlinear relationships found in large-scale cyber–physical datasets [3], [4]. Moreover, the rapidly growing volumes and time-sensitive nature of smart grid communication data highlights the need for models that can learn spatial and temporal correlations from heterogeneous data effectively.

As artificial intelligence continues to grow, deep learning methods have attracted much attention with respect to their applications in network security problems in smart grids. Convolutional Neural Networks (CNN) are capable

of discovering local spatial features from high-dimensional input features, and Long Short-Term Memory (LSTM) networks can learn the sequential dependency in time-series data [5], [6]. Bidirectional LSTM (BiLSTM) models provide additional capabilities for temporal learning by reading sequences in both forward and backward passes. Then again, while these deep learning architectures enable better representation learning (Mahendran and Vedaldi 2015), they are still limited by the drawbacks of their independent use (such as overfitting, noise sensitivity, and loss of interpretability).

To overcome these limitations, hybrid approaches fusing deep learning with ensemble machine leaning methods have recently produced promising results for IDS tasks. By combining autoencoder based representation learning, which performs compression of high dimensional features into informative latent representations; with ensemble classifiers such as Random Forest, which produce robust decision boundaries and improved generalization capability [7]. In addition, improvements such as denoising autoencoders and attention mechanisms have been made to make these models more robust when dealing with noisy inputs and classifying important temporal features [8].

Inspired by these works, this paper develops a hybrid deep learning framework for smart grid intrusion detection. The collection of first architecture consists of the Autoencoder step for CNN–BiLSTM feature learning and Random Forest classification. Moreover, an improved model is proposed by integrating Denoising Autoencoder with CNN & BiLSTM and attention in order to learn more profound representation of the features as well as the temporal pattern. In the experiments conducted, it was verified that compared to traditional machine learning methods the proposed combined expressed very high efficiency in performing this task.

Key Novel Contributions

The key contributions of this study are briefly outlined as follows:

1. A Hybrid Intrusion Detection Architecture Using Autoencoder-Based Latent Feature Extraction, CNN–BiLSTM Temporal Learning, and Random Forest Classification for Smart Grid Cybersecurity
2. A denoising autoencoder highlights the importance of dimensions, whereas the hybrid framework using CNN and BiLSTM-based sequences generates several spatial–temporal features to promote robustness.
3. High-quality detection and classification for intrusion (AE–CNN–BiLSTM-RF achieves 95.23%) proposed model helps to achieve high-quality results in terms of accuracy (DAE-CNN-BiLSTM-Attention-RF rank-1) efficiency concerning LGBM, One-Class SVM.
4. We carry out an extensive experimental evaluation including confusion matrix analysis, ROC curves, precision–recall analysis, and class-wise performance comparisons.

The rest of this paper is structured as follows. Section 2 provides a review of related work in smart grid cybersecurity measuring, as well as machine learning-based intrusion detection systems. The proposed hybrid architectures and algorithms, namely AE–CNN–BiLSTM–RF and DAE–CNN–BiLSTM–Attention–RF models are described in Section 3. Section 4 presents implementation details, dataset characteristics and experimental setup. Experimental results and comparison with baseline models are presented in Section 5. Lastly, Section 6 summarises the overall paper and provides a few future research aspects for secure smart grid architecture.

2. Literature Review

2.1 Deep Learning and Online Learning for Smart Grid Intrusion Detection

The increasing digitalization of power networks through the Power Internet of Things has created a need for intelligent intrusion detection models that can process high-dimensional and time-dependent data efficiently. A DBN–BiLSTM-based framework was introduced to address these issues by combining unsupervised feature extraction with bidirectional temporal learning, thereby improving detection capability for both known and unseen attack patterns while preserving real-time suitability for power system environments [1].

Static intrusion detection methods often fail to remain effective when attack behaviors evolve over time, especially in operational smart grid environments where data streams change continuously. To overcome this limitation, an adaptive online incremental learning method named AdaOIL-IDS was proposed, using a class-correlated broad learning mechanism and an adaptive online-offline switching strategy that supports lifelong learning without frequent retraining from scratch [2].

The need for streaming-oriented security has also become evident in electric vehicle charging systems, which are now tightly coupled with smart grid infrastructures. A scalable hybrid online learning framework integrating Adaptive Random Forest with multiple lightweight learners was shown to improve drift resilience, reduce runtime, and enhance detection performance in both multiclass and binary security tasks for EV charging applications [3].

Power system digital substations require security models that can detect early-stage abnormal behavior even when labeled attack data are limited. A semi-supervised intrusion detection method based on convolutional traffic feature extraction, Chebyshev distance, Kolmogorov–Smirnov statistics, Self-Organizing Maps, and DBSCAN clustering achieved strong accuracy and F1 performance above 95%, demonstrating its usefulness for zero-day detection in operational technology environments [4].

Advanced metering infrastructure is a particularly sensitive smart grid component because of its scale and communication dependency. To reduce communication cost and labeling dependence, a federated semi-supervised framework based on federated distillation and DCGAN-assisted sample generation was proposed, achieving 99.58% accuracy while lowering false positives and communication overhead in AMI intrusion detection [5].

Protocol-aware security remains essential in industrial energy systems, especially for DNP3-based SCADA networks that support real-time monitoring and control. A tri-phase framework combining attack detection, attack-type classification, and privacy preservation through XGBoost and gradient boosting models demonstrated very high accuracy and interpretability, making it a strong protocol-specific solution for smart grid cybersecurity [6].

Internal attacks originating from compromised industrial devices are often difficult to detect through conventional perimeter-based defenses alone. An autoencoder-based anomaly detection method using enriched IP flow data and application-layer ICS packet information showed excellent effectiveness on IEC 104 and IEC 61850 datasets, achieving average detection rates of 99% and 97%, respectively, for smart grid communication monitoring [7].

2.2 Surveys, Architectural Trends, and Model Diversity in Smart Grid Security

A followup that considers smart grid security in a larger context finds that resilience is built not only on intrusion detection algorithms, but requires the coordinated adoption of emerging technologies. An extensive review of smart grid improvement mechanisms highlighted the importance of uniting 5G, fog computing, self-healing systems, reinforcement learning and heartbeat-based coordination into a coherent structure that simultaneously promotes security, efficiency and reliability [8].

Large-scale multi-agent systems have also been used as a framework to study the relationship between smart grids and larger IoT ecosystems. Layered analysis of attacks based on perception, network and application vulnerabilities showed approaches based on machine learning, deep learning or transfer learning to be applicable for some attack types while even broader differences in security needs across IoT enabled critical sectors could be inferred [9].

It can lead to large financial and operational disruption, making false data injection one of the most harmful threats in smart metering and consumer-side power analytics. A light ConvLSTM based anomaly detector of smart meter readings performs an accuracy of 91.3% in multiple FDI cases, suggesting that temporal deep learning is useful against attacks designed to acquire energy [10].

Following the growing interest of researches in anomaly detection, survey studies associated with artificial intelligence and physical modeling concepts have relied on smart grids. In a systematic review of AI-driven and physics-aware anomaly detection methods, it was noted that although performance improvements are observed with physics-informed learning approaches, many works still exhibit wide variety in use case contexts, evaluation protocols, and depth of validation [11].

Another aspect of this is recent work has also shown the importance of multi-modal cyber-physical fusion, rather than analyzing cyber or physical data streams separately. Using substation and phasor data as the two input modalities, a graph neural network-based intrusion detection framework built on a cyber-physical testbed showed that with appropriate approaches to aggregate both modalities, this can significantly improve detection rates while preserving scalability under partial observability conditions which further strengthens the case for graph-based security analytics in power systems [12].

SCADA security research has moved toward integrated and privacy-aware architectures that combine anomaly detection, classification, feature selection, and secure collaborative learning. A machine learning framework

incorporating Isolation Forest, Fourier-based burst analysis, SHAP, RFE, PCA, federated learning, differential privacy, homomorphic encryption, and adversarial defense strategies achieved 99.29% accuracy and demonstrated the value of multi-layered protection for industrial control communication [13].

The challenge of detecting stealthy attacks such as man-in-the-middle intrusion has motivated research into faster and more discriminative IDS pipelines. An updated IDS architecture using a Random Disturbance Algorithm for feature selection and real-time cyber-physical validation on IEEE bus systems demonstrated that careful architectural redesign can improve both execution speed and attack identification accuracy in smart grid settings [14].

2.3 Advanced Deep Architectures, Cyber-Physical Resilience, and Protocol-Specific Protection

The emergence of AI-generated adversarial attacks has shown that classical anomaly detectors may fail against adaptive threats that deliberately evade detection while manipulating energy reporting. A deep reinforcement learning-based adversarial attack generation framework revealed severe performance degradation in conventional detectors, while adversarially hardened ConvLSTM models improved resilience and achieved a 95.12% detection rate under these challenging attack conditions [15].

Advanced persistent threats represent a particularly dangerous class of attacks because they unfold gradually across space and time and often escape single-event detectors. An Enhanced Graph Convolutional LSTM framework was developed to correlate cyber and physical anomalies in near real time, enabling early situational awareness and outperforming prior spatio-temporal deep learning methods in APT detection tasks [16].

To secure industrial communication channels more effectively, DNP3 targeted studies have also used specific deep learning pipelines. The segmented neural network architecture based integrated with DNN, LSTM and Random Neural Network components resulted in a maximum of 99.86% accuracy on the representation of DCW3 traffic which managed to validate that deep learning based approach at protocol level has potential for implementing real-time anomaly detection in smart grid [17].

The integration of renewable energy (RE) brings an increased level of decentralization, variability and communication dependencies that aggravate the cyber security challenge of smart cyber-physical power systems. Using a hybrid detection approach that combines deep learning with derivative-aware preprocessing achieved over 98% accuracy for cyber threat recognition, showing the need for security analytics to tailor itself as per the unique operational challenges posed by renewable-rich grids [18].

Moving on from intrusion detection, research has become increasingly focused on a more general problem of cyber-physical resilience under dynamic disturbances. A deep reinforcement learning control based on DDPG and RMSprop demonstrated more stable, convergent and resilient performance than designed classical and other RL controllers in simulation experiments, showing it is possible to extend intelligent defense from detection toward adaptive control [19].

The expansion of IIoT-enabled smart grids has further motivated the adoption of zero-trust security concepts that unify information technology and operational technology perspectives. A framework combining EigenGame-based contextual fusion with enhanced quantum reinforcement learning demonstrated robust malicious behavior identification, especially for ransomware-related threats in industrial smart grid environments [20].

Communication infrastructure itself can also serve as a sensing mechanism for grid security and monitoring. A review of power line communication-based information inference showed that PLC technologies can support topology inference, anomaly detection, and cybersecurity monitoring, thereby opening additional avenues for signal-aware protection and surveillance within smart grids [21].

Virtual power plants are another emerging environment where intrusion detection must accommodate dynamic connectivity and heterogeneous devices. A transformer-based intrusion detection model with PCA-driven feature filtering and dynamic loss weighting improved adaptability and detection efficiency, illustrating the growing relevance of transformer architectures in energy network security [22].

As smart meters and edge devices increasingly become more constrained in resources but also open to future cryptographic threats, security architectures must consider not only communication protection but also threat detection. An AI-supported intelligent intrusion detection system (IDS) component in a post-quantum, blockchain-assisted, edge-enabled smart grid architecture enabled appreciable savings in computational and communication costs while enhancing long-term resistance to both classical and quantum attacks [23].

2.4 Robust Representation Learning, Distributed Energy Security, and Research Gaps

Noise, dearth of labels and nonstationary traffic patterns are persistent challenges to practical intrusion detection systems. A noise resistant autoencoder model showed that properly tuned denoising approaches can strengthen the robustness of our pipeline on semi-supervised intrusion detection and outperform other autoencoder based baselines, something that is particularly important for real smart grid data streams, which are seldom clean or fully labeled [24].

Technical ChallengesExisting intrusion detection models in distributed energy resources and inverter-based microgrids are not adequate as they need to monitor host behavior-network traffic-physical measurements. A laboratory microgrid-based hybrid IDS attained 100% detection rate and averaged a response time of 1.5 seconds, highlighting the utility of multi-source evidence fusion in resource-constrained DER environments [25].

Large language models have recently entered the smart grid security landscape through multimodal situational awareness systems. GridSense transformed electrical, meteorological, and social signals into structured prompts for LLM reasoning and achieved promising anomaly detection and forecasting results, suggesting that language-guided inference may complement conventional IDS pipelines in future resilient grid platforms [26].

Detection alone is often insufficient when utilities require rapid localization and restoration after an incident. An analytical cyberattack localization framework based on continuous sensor monitoring and parameter–measurement relationships showed that localization-oriented methods can support faster troubleshooting and post-attack recovery in real-time smart grid environments [27].

Smart Grid 2.0 integrates distributed renewable systems, autonomous controls, blockchain services and stakeholder coordination that have become the center of a security discussion. A recent survey pointed out that the new grids would be burdened with intertwined risks from malware, false data injection, blockchain vulnerabilities and adversarial AI manipulation of this nature which will require trustworthy and interoperable defense models [28].

At the market and edge-operation level, IoT-enabled local energy markets also introduce new forms of cyber risk. A nodal pricing-based architecture showed that abnormal price patterns can act as early indicators of false data injection and related attacks, thereby linking cybersecurity analytics with economic operation signals in decentralized energy systems [29].

Data imbalance and dimensionality reduction continue to influence the effectiveness of attack detection models in smart grids. A stacked autoencoder-based framework generating synthetic minority samples demonstrated that Random Forest and XGBoost can remain robust under reduced feature spaces, while more sensitive algorithms degrade sharply, reinforcing the value of resilient classifiers and careful feature tuning [30].

Security research for power systems must likewise account for embedded hardware threats as cyber adversaries increasingly exploit firmware, integrated circuit and device-level weaknesses. An overview of embedded hardware attacks against power grids, photovoltaics, EVs, and storage systems showed that tampering, side-channel threats, hardware Trojans and trusted boot components would remain key issues for secure energy infrastructure in the future [31].

In more general systems terms, the cybersecurity of smart grids is contingent on secure protocols, IoT-aware architectures and convergence with confidentiality/integrity/availabil[16]ity needs. A recent survey covers emerging defense technologies including AI, blockchain and software-defined networking, and found that future smart grids will need layered and adaptive security designs to protect against the increasingly sophisticated attacks [32].

The integration of these, especially wireless sensor networks into the smart grid monitoring and automation, and the application of machine learning based technical solutions is growing. A significant review, encompassing hundreds of studies, demonstrated that supervised learning, unsupervised learning, and reinforcement learning approaches can greatly enhance sustainability, reliability, and fault concern in WSN-enabled girds; they also brought to light persistent issues regarding implementation as well as diversity of datasets [33].

Advanced persistent threat defense has also been approached from the perspective of optimized countermeasure scheduling rather than pure classification. A reinforcement learning scheme for meter data management systems

jointly optimized scan intervals and repair rates, improving defense effectiveness against APT behavior and illustrating how learning-based defense policies can augment conventional protection workflows [34].

An important recent benchmark in smart grid intrusion detection is the edge computing-based and threat behavior-aware prioritization framework using modified LGBM and One-Class SVM models. That work demonstrated strong multidimensional and binary intrusion detection performance on IED-oriented datasets, but it also highlighted the continued reliance of many existing methods on traditional machine learning without fully exploiting robust deep latent feature learning and attention-guided temporal modeling, which motivates the development of more advanced hybrid architectures in current research [35].

3.1 Proposed architecture

3.1.1 Optimized Hybrid Autoencoder → CNN → BiLSTM → Random Forest

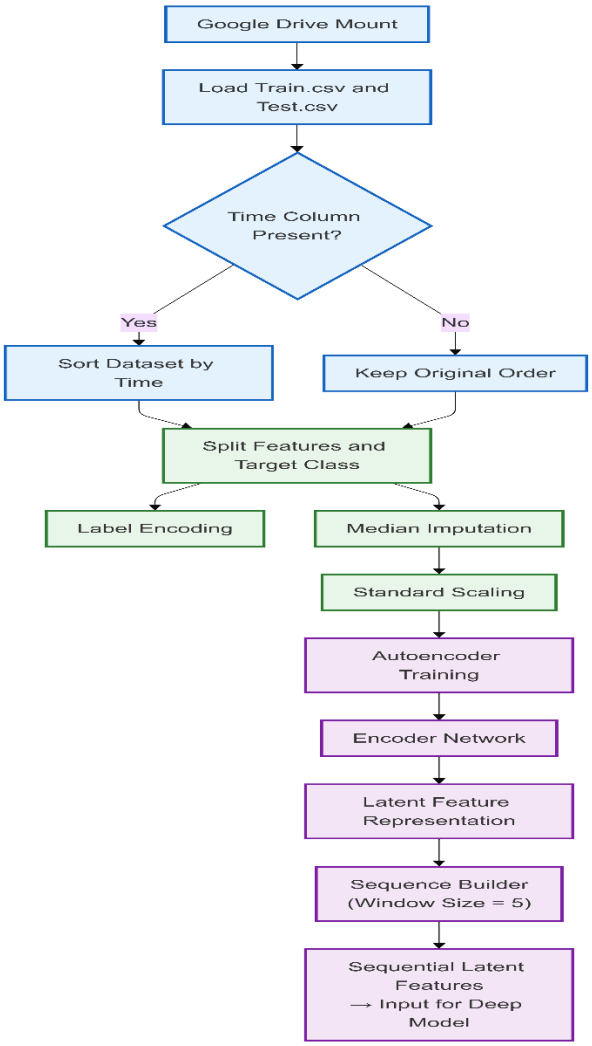

Figure 1: Data Preprocessing and Latent Feature Extraction using the Autoencoder Model

As shown in the figure 1, this is an early overview of proposed hybrid architecture. First, the dataset is loaded from google drive which gives access for training and testing files. If the dataset has a time attribute, the records are sorted in chronological order to maintain temporal relationships. Next, the dataset is separated into feature variables and its target class. So, during the preprocessing I have used label encoding to convert the target labels and for feature set, missing values will be handled through median available. The features are then standardized by standard scaling to normalize their distribution. It is then passed to an autoencoder network, which learns compact latent representations of the input features. These latent features extracted by the encoder component are compressed versions of the original

data. Finally, a sequence builder generates sequential windows with predefined fixed length based on the latent features, to prepare data for temporal learning of the deep learning model whose implementation is detailed in the subsequent stage.

As shown in Figure 1, the first stage of the proposed framework is to import the dataset from Google Drive and then split it into training and testing sets. If there is a temporal attribute present, records are sorted chronologically to maintain time dependency within the dataset.

Let the original dataset be represented as

$$D = \{(x_i, y_i)\}_{i=1}^{N} \quad (1)$$

where
$x_i$represents the input feature vector and
$y_i$denotes the corresponding class label.
The feature vector is defined as

$$x_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \quad (2)$$

where $d$represents the number of features.

Missing Value Imputation

Missing values in the dataset are handled using **median imputation**. For each feature j, the missing values are replaced with the median value:

$$x_{ij} = \{x_{ij}, if\ value\ exists\ median(x_j), if\ value\ is\ missing \quad (3)$$

This ensures that missing values do not distort the distribution of the dataset.

Feature Standardization

After imputation, the features are normalized using **standard scaling** to ensure consistent feature distributions. Each feature is transformed as

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \quad (4)$$

where

- $\mu_j$is the mean of feature j
- $\sigma_j$is the standard deviation of feature j

Autoencoder-Based Feature Representation

To learn compact representations of the input features, an **autoencoder neural network** is employed. The encoder maps the input vector into a lower-dimensional latent space:

$$h_i = f(W_e x_i + b_e) \quad (5)$$

where

- $h_i$is the latent representation
- $W_e$and $b_e$are encoder parameters
- $f(\cdot)$is the activation function

The decoder reconstructs the original input:

$$\widehat{x_i} = g(W_d h_i + b_d) \quad (6)$$

The autoencoder is trained by minimizing the reconstruction loss:

$$L = \frac{1}{N}\sum_{i=1}^{N} \| x_i - \widehat{x_i} \|^2 \quad (7)$$

Sequential Feature Construction

To capture temporal relationships within the data, latent features are organized into sequences of length T. The sequence representation is defined as

$$S_i = [h_i, h_{i+1}, \ldots, h_{i+T-1}] \quad (8)$$

These sequential latent features serve as input for the deep learning model described in **Figure 2**.

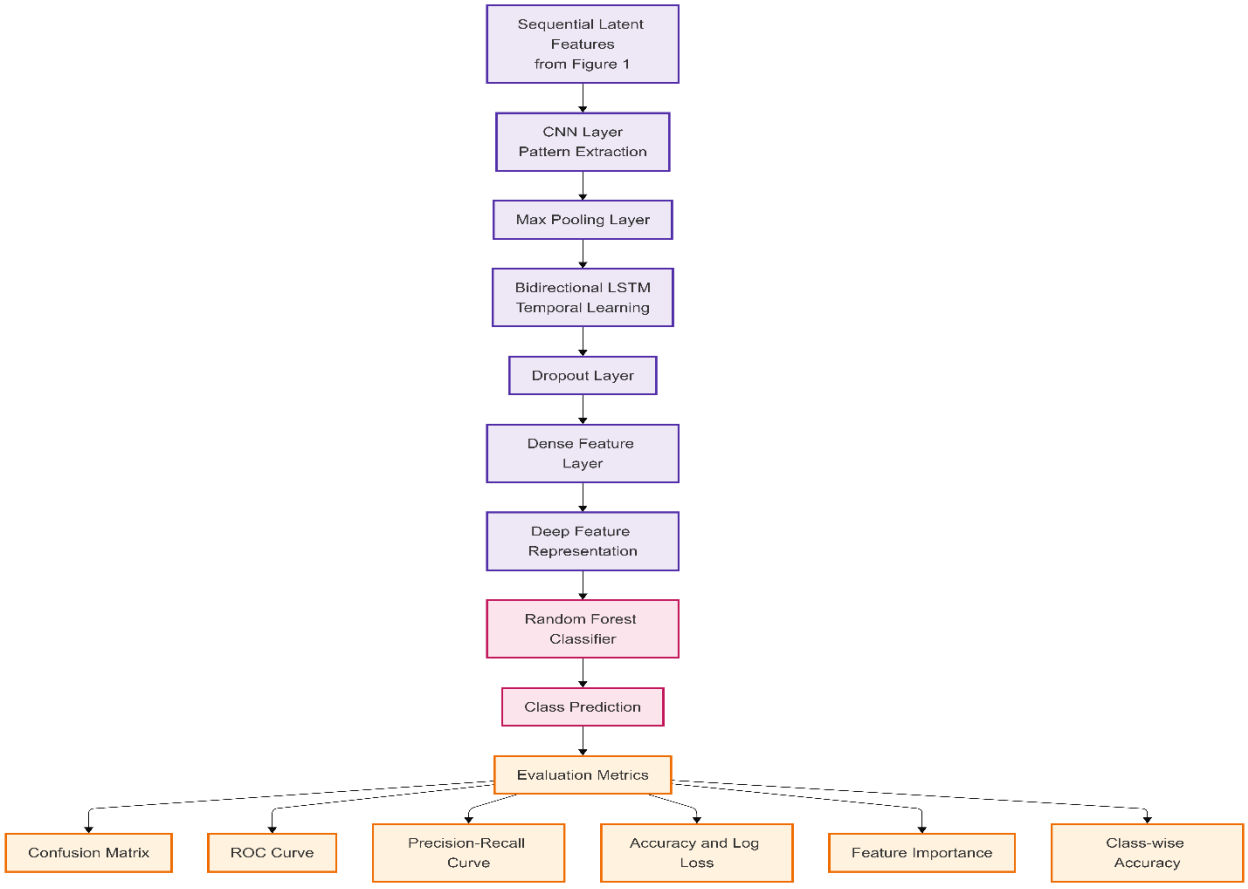

Figure 2: Hybrid CNN-BiLSTM and Random Forest Architecture for Classification and Evaluation

The Figure 2 depicts classification phase of the proposed hybrid architecture. The sequential features learned in Figure 1 are input to a deep learning model containing convolutional and recurrent neural network elements. The input image is first passed to a one-dimensional convolutional (CNN) layer with flat windows captures local feature patterns, where a max-pooling layer follows to compress the data dimensional while emphasizing dominant patterns. The features are then fed into a Bidirectional Long Short-Term Memory (BiLSTM) layer, which learns temporal dependencies from both past and future contexts in the sequence. We use a dropout layer to reduce connectivity and overfitting in the model, followed by a feature dense layer that creates deep-level features. In this method, instead of directly using the neural network output for classification, the extracted deep features are passed on to a Random Forest classifier. The hybrid architecture utilizes the representation of learning capacity of deep neural networks and the strong decision-making ability of the ensemble learning. We run the model to get class and proba predictions, then

evaluate these predictions using performance metrics: confusion matrices, ROC curves, precision–recall curves along with accuracy through log loss and feature importance analysis for showing reinforcement on feature class-wise accuracy.

The classification stage of the proposed hybrid architecture is depicted in Figure 2. The latent sequential features sampled from Figure 1 are passed to a deep learning architecture that combines CNN and BiLSTM networks.

Convolutional Feature Extraction

A **1D convolutional layer** is applied to capture local feature patterns from the sequence:

$$c_i = f(W_c * S_i + b_c) \quad (9)$$

where

- $W_c$represents convolution filters
- $*$denotes convolution operation
- $f(\cdot)$is the activation function

The resulting feature maps are reduced using **max pooling**:

$$p_i = max(c_i) \quad (10)$$

Bidirectional LSTM

The pooled features are processed by a **Bidirectional LSTM network**, which captures temporal dependencies in both forward and backward directions.

Forward LSTM:

$$\overrightarrow{h_t} = LSTM(x_t, \overrightarrow{h_{t-1}}) \quad (11)$$

Backward LSTM:

$$h_t^{\leftarrow} = LSTM(x_t, h_{t+1}^{\leftarrow}) \quad (12)$$

The final hidden representation is obtained by concatenating both directions:

$$h_t = [\overrightarrow{h_t}, h_t^{\leftarrow}] \quad (13)$$

Deep Feature Layer

The output of the BiLSTM is passed through a dense layer to obtain deep feature representations:

$$z = f(W_f h_t + b_f) \quad (14)$$

These deep features are then used as input to the final classifier.

Random Forest Classification

Instead of directly using the neural network output for classification, a **Random Forest classifier** is employed for improved robustness and generalization.

Each decision tree in the forest predicts a class label:

$$y_k = T_k(z) \quad (15)$$

where $T_k$represents the k^{th} decision tree.

The final prediction is determined by majority voting:

$$\hat{y} = mode(y_1, y_2, \ldots, y_K) \quad (16)$$

where $K$is the number of trees.

Model Evaluation

The performance of the proposed model is evaluated using several metrics.

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

Log Loss

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} log(p_{ic}) \quad (18)$$

where

- $y_{ic}$is the true label
- $p_{ic}$is the predicted probability.

Additional evaluation techniques include **confusion matrices, ROC curves, precision–recall curves, feature importance analysis, and class-wise accuracy**, which provide comprehensive insight into the model's classification performance.

3.1.2 Proposed DAE–CNN–BiLSTM–Attention–RF Hybrid Classification Model

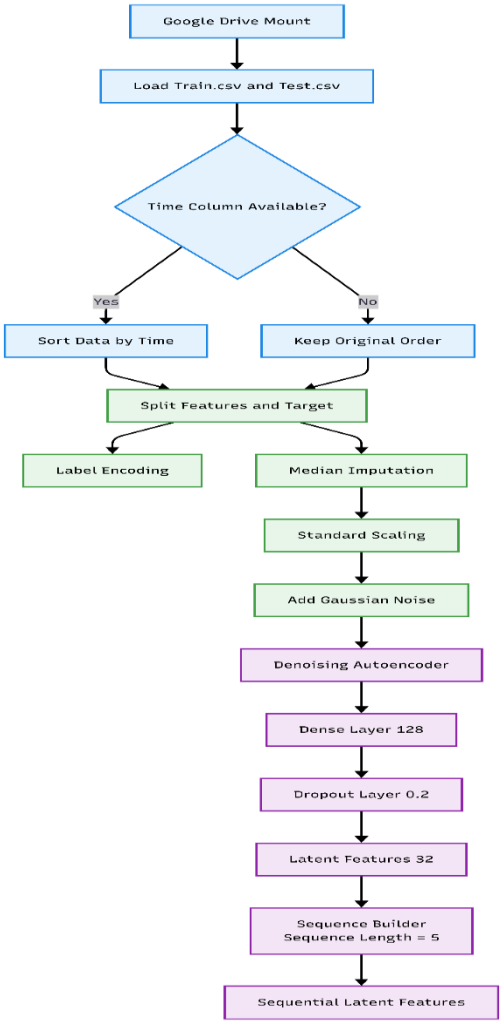

Figure 3 : Preprocessing and Denoising Autoencoder-based Latent Feature Extraction

The preprocessing and latent feature extraction stage of the proposed framework is illustrated in Figure 3. Datasets load from Google Drive and, optionally, the time attribute can be sorted We divide the features and target variables, then apply label encoding to class labels. The missing values are filled with median imputation and the feature scaling is applied by using standard normalization. The scaled input is then corrupted with Gaussian noise to train a denoising autoencoder. The autoencoder learns a lower-dimensional latent feature representation of the input data. Last, these latent features are transformed into sequential windows to be fed into the deep learning model.

Data Processing

During the first phase, we will load the dataset and split it into input features and class labels. Represent the dataset as.

$$D = \{(x_i, y_i)\}_{i=1}^{N} \quad (19)$$

where
$x_i$represents the input feature vector and
$y_i$represents the corresponding class label.

Each feature vector is defined as

$$x_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{id}] \quad (20)$$

where $d$denotes the number of input features.

To handle missing values in the dataset, **median imputation** is applied. The missing values are replaced using

$$x_{ij} = \{x_{ij}, if\ value\ exists\ median(x_j), if\ value\ is\ missing \quad (21)$$

After imputation, the features are standardized to ensure consistent scaling across all attributes:

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \quad (22)$$

where
$\mu_j$represents the mean of feature $j$and
$\sigma_j$represents the standard deviation.

To improve robustness against noise, Gaussian noise is introduced to the scaled features:

$$\widetilde{x_i} = x_i + \alpha \cdot \epsilon \quad (23)$$

where
$\alpha$is the noise factor and
$\epsilon \sim N(0,1)$represents Gaussian noise.

Denoising Autoencoder Representation

A **Denoising Autoencoder (DAE)** is used to learn compressed latent representations of the input features. The encoder maps the noisy input into a latent feature space:

$$h_i = f(W_e\widetilde{x_i} + b_e) \quad (24)$$

where
$W_e$and $b_e$are encoder parameters and
$f(\cdot)$represents the activation function.

The decoder reconstructs the original input as

$$\widehat{x_i} = g(W_d h_i + b_d) \quad (25)$$

where
$W_d$and $b_d$are decoder parameters.

The DAE is trained by minimizing the reconstruction loss:

$$L = \frac{1}{N}\sum_{i=1}^{N} \| x_i - \widehat{x_i} \|^2 \quad (26)$$

The latent representation $h_i$forms the compact feature representation used in the subsequent deep learning stage.

To capture temporal patterns, the latent features are organized into sequences of length T:

$$S_i = [h_i, h_{i+1}, \dots, h_{i+T-1}] \quad (27)$$

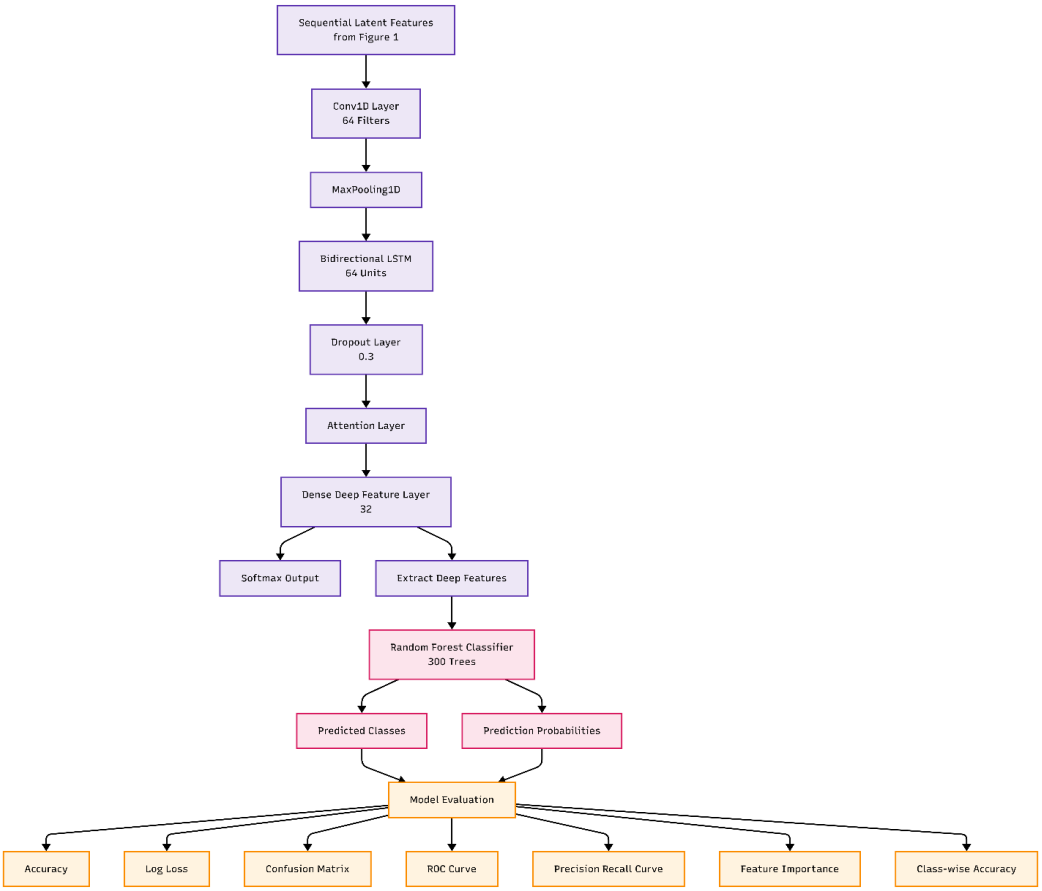

Figure 4 CNN–BiLSTM–Attention Deep Feature Learning and Random Forest Classification

Deep learning and classification part of the proposed architecture is shown in Figure 4. The latent features will be gained from Figure 1 sequentially, slide onto a CNN layer to characterize local feature patterns. A max-pooling layer down-samples the features, and a more complex bidirectional LSTM network detects relations between time dependencies in the sequence. To avoid overfitting, we apply a dropout layer and an attention mechanism in order to focus on crucial temporal aspects. A dense layer follows to generate deep feature embeddings from the representation obtained after this transformation. These deep features are subsequently classified by through a Random Forest classifier comprising multiple decision trees. In the end, model performance is analyzed using a number of metrics including accuracy, log loss, confusion matrix as well as ROC curves, precision–recall curves and also feature importance analysis and class-wise accuracy.

Deep Learning Feature Extraction (CNN–BiLSTM–Attention)

Firstly, we utilize hybrid deep learning architecture combining CNNs, BiLSTM networks and Attention mechanism to process the sequential latent features.

The convolutional layer extracts local feature patterns:

$$C_i = f(W_c * S_i + b_c) \quad (28)$$

where $W_c$denotes convolution filters and $*$denotes the convolution operation.
The resulting feature maps are downsampled using **max pooling**:

$$P_i = max(C_i) \quad (29)$$

Next, a **Bidirectional LSTM** captures temporal dependencies in both directions.
Forward pass:

$$\overrightarrow{h_t} = LSTM(x_t, \overrightarrow{h_{t-1}}) \quad (30)$$

Backward pass:

$$h_t^{\leftarrow} = LSTM(x_t, h_{t+1}^{\leftarrow}) \quad (31)$$

The final representation is obtained by concatenating both states:

$$h_t = [\overrightarrow{h_t}, h_t^{\leftarrow}] \quad (32)$$

An **attention mechanism** is applied to assign importance weights to different time steps.

$$e_t = tanh(W_a h_t + b_a) \quad (33) \quad \alpha_t = \frac{exp(e_t)}{\sum_k \quad exp(e_k)} \quad (34)$$

The attention-weighted feature representation is computed as

$$z = \sum_t \quad \alpha_t h_t \quad (35)$$

Finally, a dense layer produces deep feature embeddings:

$$f_d = \sigma(W_f z + b_f) \quad (36)$$

Random Forest Classification and Evaluation

The deep feature representation f_d is used as input to a **Random Forest classifier**, which consists of multiple decision trees.

Each tree produces a class prediction:

$$y_k = T_k(f_d) \quad (37)$$

where T_k denotes the k^{th} decision tree.

The final prediction is obtained through majority voting:

$$\hat{y} = mode(y_1, y_2, \ldots, y_K) \quad (38)$$

where K is the total number of trees in the forest.

Model Evaluation

The performance of the proposed model is evaluated using classification metrics. The accuracy is computed as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (39)$$

The log loss for probabilistic predictions is defined as

$$LogLoss = -\frac{1}{N} \sum_{i=1}^{N} \quad \sum_{c=1}^{C} \quad y_{ic} log(p_{ic}) \quad (40)$$

where

y_{ic} is the true class label and

p_{ic} represents the predicted probability.

Input: Training dataset D_{train}, Test dataset D_{test}

Predicted class labels $\hat{Y}$ and evaluation metrics

Load the training dataset D_{train} testing dataset D_{test}.

Sort the records in ascending order by time column if any time attribute exists.

Partition each dataset into feature matrix X and target class labels Y.

Use Label Encoding to encode the categorical class labels in a numerical way.

Impute each of the feature columns using median imputation.
Feature Engineering: Standardizing the feature values

3.2 Proposed algorithm

3.2.1 Algorithm 1: Proposed AE–CNN–BiLSTM–RF Hybrid Architecture

Input: Training dataset D_{train}, Test dataset D_{test}
Output: Predicted class labels $\hat{Y}$ and evaluation metrics

1. Load the training dataset D_train testing dataset D_test.
2. Sort the records in ascending order by time column if any time attribute exists.
3. Partition each dataset into feature matrix X and target class labels Y.
4. Use Label Encoding to encode the categorical class labels in a numerical way.
5. Impute each of the feature columns using median imputation.
6. Feature Engineering: Standardizing the feature values

$$Z = \frac{X - \mu}{\sigma}$$

where μ and σ represent the mean and standard deviation of the feature values.

7. Initialize an **Autoencoder neural network** with input dimension $d$and latent dimension l.
8. Train the Autoencoder to minimize the **reconstruction loss**

$$L = \frac{1}{N}\sum_{i=1}^{N} \| x_i - \hat{x}_i \|^2$$

Extract **latent feature representations** Husing the encoder network.

9. Construct **sequential feature windows** using a sliding window technique with sequence length T

$$S_i = [h_i, h_{i+1}, \ldots, h_{i+T-1}]$$

11. Initialize the **CNN–BiLSTM deep learning architecture**.
12. Apply a **1D Convolutional layer** to extract local feature patterns

$$C = f(W_c * S + b_c)$$

13. Apply **Max Pooling** to reduce feature dimensionality.
14. Feed the pooled features to a **Bidirectional LSTM network** to capture temporal dependencies

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$$

15. Apply a **Dropout layer** to reduce overfitting.
16. Pass the output through a **Dense feature layer** to obtain deep feature representations

$$z = f(W_f h_t + b_f)$$

17. Extract the deep features $z$from the trained neural network.
18. Initialize the **Random Forest classifier** with optimized hyperparameters.
19. Train the Random Forest model using the extracted deep features.
20. Predict class labels for the testing data using

$$\hat{y} = mode(T_1(z), T_2(z), \ldots, T_K(z))$$

where T_k represents the k^{th}decision tree.

21. Compute prediction probabilities for each class.
22. Evaluate model performance using the following metrics:
    - Accuracy
    - Log Loss
    - Confusion Matrix
    - ROC Curve
    - Precision–Recall Curve
23. Perform **feature importance analysis** using the Random Forest model.
24. Compute **class-wise accuracy** for each category.

25. Save the evaluation results and visualization graphs.

The framework works as a hybrid model, which inputs the training dataset D_trainand testing dataset D_testoutputs predicted class labels Y^and evaluation metrics. If a temporal attribute exists in the datasets, they are loaded and sorted by time. Xand target labels Yare separated, followed by label encoding, median imputation and feature standardization through Z=(X-μ)/σ. Then, an autoencoder is trained to minimize reconstruction loss and evolve a more representative latent space H (Auxiliary task 1). Moreover, sequential windows S_iare formed and passed through analysis up to deep features z using CNN–BiLSTM network with dropout and dense layer (Main Task). Lastly, Random Forest classifier is used to predict labels which are evaluated based on multiple metrics of performance such as accuracy-log loss confusion matrix-ROC precision-recall curves-feature importance analysis.

3.2.2 Proposed DAE–CNN–BiLSTM–Attention–RF Hybrid Framework

Algorithm 1: Data Preprocessing and Denoising Autoencoder-based Latent Feature Generation

Input:

Training dataset $D_{train} = \{(x_i, y_i)\}_{i=1}^{N_{train}}$,

Testing dataset $D_{test} = \{(x_j, y_j)\}_{j=1}^{N_{test}}$

Output:

Sequential latent representations $S_{train}, S_{test}$and sequence labels $y_{train\_seq}, y_{test\_seq}$

Begin

1. Load datasets $D_{train}$and $D_{test}$from storage.
2. If a temporal attribute exists in the datasets, sort both datasets in ascending order of time.
3. Split the datasets into feature matrices and target labels:
   $X_{train}, y_{train}$and X_{test}, y_{test}.
4. Encode categorical class labels:
$$y_{train\ enc} \leftarrow LabelEncode(y_{train})$$
$$y_{test\ enc} \leftarrow LabelEncode(y_{test})$$
5. Handle missing feature values using median imputation:
$$X_{train\_imp} \leftarrow MedianImputer(X_{train})$$
$$X_{test\_imp} \leftarrow MedianImputer(X_{test})$$
6. Standardize the features using z-score normalization:
$$X_{std} = \frac{X - \mu}{\sigma}$$
7. Inject Gaussian noise into the normalized features to improve robustness:
$$X_{noisy} = X_{std} + \alpha \cdot N(0,1)$$
8. Construct the **Denoising Autoencoder (DAE)** consisting of:
   - Input layer
   - Dense(128, ReLU)
   - Dropout(0.2)
   - Latent layer Dense(32, ReLU)
   - Dense(128, ReLU)
   - Reconstruction layer
9. Train the DAE using noisy inputs and clean targets with Mean Squared Error loss.
10. Obtain latent feature representations:
$$H_{train} \leftarrow Encoder(X_{train\ std})$$
$$H_{test} \leftarrow Encoder(X_{test\ std})$$
11. Construct sequential feature windows of length T:
$$S_i = [h_i, h_{i+1}, \ldots, h_{i+T-1}]$$
12. Generate sequence datasets:
$$(S_{train}, y_{train\_seq}) \leftarrow CreateSequences(H_{train}, y_{train\_enc}, T)$$
$$(S_{test}, y_{test\_seq}) \leftarrow CreateSequences(H_{test}, y_{test\_enc}, T)$$
13. Convert sequence labels to categorical form for deep learning training.

End

Algorithm 1: Data pre-processing and latent features generation using a denoising autoencoder It first loads the training and testing datasets and optional sorts them in temporal order. After splitting the data into feature matrices

and target labels, a sequence of sequential operations are executed — label encoding, median imputation on missing values, standardization of features. Gaussian noise is added to improve robustness in representation learning. It uses a denoising autoencoder that reconstructs clean inputs from noisy features to learn compressed latent representations. These hidden features are then reshaped into fixed size sequential windows serving as structured input to the deep learning model in the next classification step.

Algorithm 2: CNN–BiLSTM–Attention Deep Feature Learning and Random Forest Classification

Input: Sequential latent features S_{train}, S_{test} and sequence labels $y_{train\_seq}, y_{test\_seq}$

Output: Predicted labels $\hat{Y}_{test}$, prediction probabilities P_{test}, and evaluation metrics

Begin

1. Construct the **CNN–BiLSTM–Attention deep learning architecture**:
   - Conv1D layer (64 filters, kernel size 3, ReLU)
   - MaxPooling1D layer
   - Bidirectional LSTM (64 units, return sequences)
   - Dropout layer (0.3)
   - Attention layer
   - Dense layer (32 neurons) for deep feature representation
   - Softmax output layer
2. Train the CNN–BiLSTM–Attention model using sequence input S_{train} and categorical labels.
3. Extract discriminative deep features from the hidden dense layer:

$$F_{train} \leftarrow DeepFeatureExtractor(S_{train})$$
$$F_{test} \leftarrow DeepFeatureExtractor(S_{test})$$

4. Initialize the Random Forest classifier with:
   - $n\_estimators = 300$
   - class balancing
   - fixed random seed
5. Train the Random Forest model:

$$RF.fit(F_{train}, y_{train\_seq})$$

6. Generate predictions:

$$\hat{Y}_{train} \leftarrow RF.predict(F_{train})$$
$$\hat{Y}_{test} \leftarrow RF.predict(F_{test})$$

7. Estimate class membership probabilities:

$$P_{test} \leftarrow RF.predict\_proba(F_{test})$$

8. Compute evaluation metrics:
   - Accuracy
   - Log Loss
   - Classification Report
   - Confusion Matrix
   - ROC Curves
   - Precision–Recall Curves
9. Perform additional diagnostic analysis:
   - Feature importance from Random Forest
   - Class-wise accuracy
   - Actual vs predicted class distribution
   - Overfitting gap between training and testing performance
10. Save all results, metrics, and visualizations.
11. Return predicted labels, probabilities, trained models, and evaluation results.

End

Algorithm 2: Deep temporal feature learning, classification and model evaluation The latent sequential features generated by Algorithm 1 are processed according to a CNN–BiLSTM–Attention architecture. The convolutional layer extracts local feature patterns and the bidirectional LSTM captures temporal dependencies in both directions. An attention mechanism stressed informative temporal features, and a dense layer creates deeper feature presentations. The extracted features are then used to train a Random Forest classifier for the final multiclass prediction. Model performance evaluation is done through accuracy, log loss, confusion matrices, ROC curves and the precision–recall curve with additional metrics of feature importance analysis and class-wise accuracy metrics.

3.3. Comparative Analysis of Hybrid Architectures

Table 1. Comparative Analysis of Hybrid Architectures

No.	Feature / Component	AE–CNN–BiLSTM–RF Hybrid Architecture	Proposed DAE–CNN–BiLSTM–Attention–RF Hybrid Framework	Improvement
1	Data Preprocessing	Standard preprocessing (encoding, scaling)	Same preprocessing with enhanced robustness	Equivalent
2	Noise Handling	No noise modeling	Gaussian noise injection before encoding	Improves robustness
3	Representation Learning	Standard Autoencoder (AE)	**Denoising Autoencoder (DAE)**	Better noise-resistant feature learning
4	Latent Feature Extraction	Encoder generates compressed features	Encoder learns **robust latent representations** from noisy input	Higher generalization
5	Temporal Data Formation	Sequence windows created	Same sequential window generation	Equivalent
6	Convolutional Feature Extraction	Conv1D layer	Conv1D layer	Equivalent
7	Temporal Dependency Modeling	Bidirectional LSTM	Bidirectional LSTM	Equivalent
8	Feature Importance Learning	Not included	**Attention mechanism applied to sequence features**	Highlights important temporal patterns
9	Deep Feature Representation	Dense layer representation	Dense layer with attention-refined features	Improved feature discrimination
10	Final Classifier	Random Forest	Random Forest	Equivalent
11	Feature Importance Analysis	Available via Random Forest	Available via Random Forest	Equivalent
12	Robustness to Noisy Data	Moderate	**High (due to DAE)**	Improved stability
13	Temporal Feature Weighting	Not available	**Attention-based weighting**	Better sequence learning
14	Model Generalization	Moderate	**Higher due to DAE + Attention**	Improved generalization
15	Interpretability	Feature importance only	Feature importance + attention-based importance	More interpretable
16	Overall Architecture	AE → CNN → BiLSTM → RF	**DAE → CNN → BiLSTM → Attention → RF**	More advanced hybrid architecture

The earlier AE–CNN–BiLSTM–RF architecture is extensively enhanced by the DAE–CNN–BiLSTM–Attention-RF hybrid framework with the inclusion of a denoising autoencoder and an attention mechanism. While the denoising autoencoder enhances robustness by learning latent semantics from a set of possibly corrupted observations, the attention mechanism produces adaptive importance weights on temporal augmented features provided by BiLSTM network. With these upgrades the model ability meaningful temporal patterns and be less sensitive to noise in data. Thus, the proposed framework results in better feature representation, greater generalisation capacity and enhanced classification performance over baseline architecture.

4. Implementation

4.1 Hardware and Software

Faced with these challenges, the hybrid framework proposed was implemented using Python under a Google Colab environment, enabling the authors to perform intensive machine learning testing in an agile and effective manner. Training of the models CNN, BiLSTM, and attention mechanisms has been accomplished at an accelerated pace using the NVIDIA Tesla T4 GPU. Libraries like NumPy, Pandas, Scikit-learn, TensorFlow and Keras were used to do the dataset preprocessing and model development. Google Colab enabled us to work efficiently with the dataset stored in Google Drive, and because of running on GPU based environment, thus reduced our time for the model training and improved computational capability by performing feature learning and classification.

4.2 Dataset

The dataset used in this work had 1094 samples and 54 attributes, one of which was the target variable referred to as "class." The data is organized into a training and testing datasets, with identical features. System communication parameters, device measurements, relay information and electrical current readings from the intelligent electronic devices constitute the features. It contains five operational classes (Normal, Masquerade, Injection, Replay, and Fault) covering normal and cyber-physical attack behavior. If a time attribute is included, chronological ordering is possible which leads itself to the sequential deep learning architectures such as CNN–BiLSTM.

Source: https://github.com/bhartiahujaexm-cyber/Sage

Table 2. Class Distribution

Class Type	**Number of Samples**
Normal	455
Masquerade	397
Injection	101
Replay	100
Fault	41
Total	**1094**

4.3 Illustrative example

4.3.1 Hybrid Autoencoder → CNN → BiLSTM → Random Forest

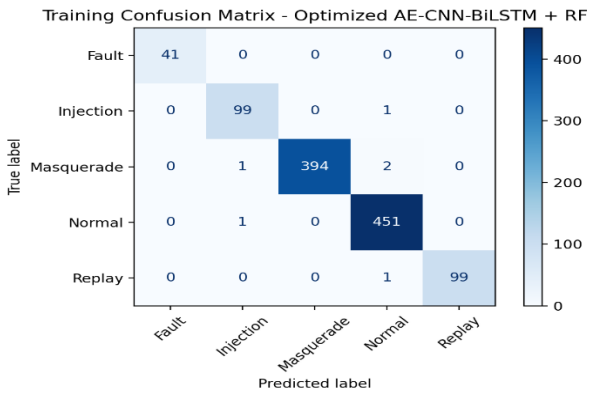

Figure 5 : Training Confusion Matrix of the Optimized AE–CNN–BiLSTM + Random Forest Model

The confusion matrix of the training dataset for the optimized AE–CNN–BiLSTM + Random Forest is shown in Figure 5. These results are presented on a five-class level (Fault, Injection, Masquerade, Normal and Replay) in the matrix of the model's classification performance. The actual class (target) is displayed in each row, and the predicted class (output neuron choice) is displayed on each column. The diagonal elements correspond to correctly classified samples, and off-diagonal elements represent misclassifications. As can be seen in the results most of the samples are correctly classified, with only a few errors. For instance, the Fault class predictions are entirely accurate while there is a small amount of confusion between injections, masquerade and normal classes. This proves high learning capacity while training.

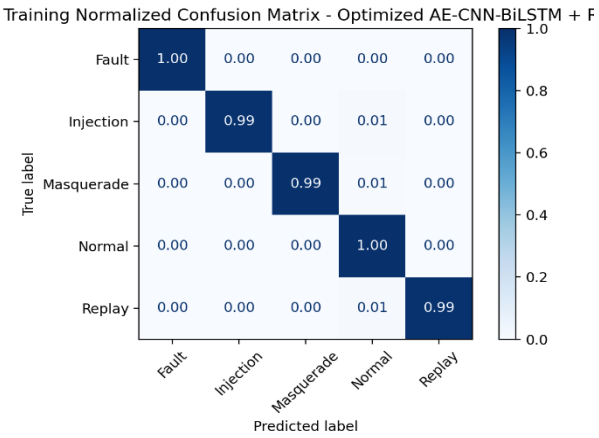

Figure 6: Training Normalized Confusion Matrix of the Optimized AE–CNN–BiLSTM + Random Forest Model

I have then used a confusion matrix to represent how well the training dataset fits with what it is supposed to predict, as in Figure 6 each row has been normalized so that the value represents classification accuracy per class. These values are between 0 and 1, meaning the proportion of samples with correct identification. The diagonal values nearing 1.00 indicate that the model accomplishes nearly perfect classification during training. Very small misclassification probabilities can be seen only between Injection and Normal classes or Masquerade and Normal classes. Case Representations - The 3D scatter plot by which classes are normally represented shows a better insight of class-wise accuracy and also confirms the hybrid model accurately identifies between contrasting operational and attack cases.

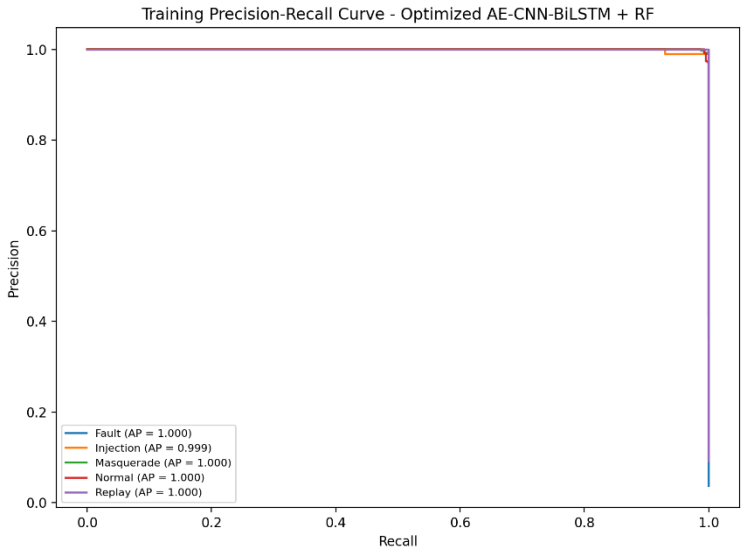

Figure 7 : Training Precision–Recall Curve of the Optimized AE–CNN–BiLSTM + Random Forest Model

The precision-recall curves for the training dataset on all five classes are illustrated in Figure 7. Precision–recall curves are especially useful for assessing classification performance in imbalanced datasets. It is the curve for each class relation of precision and recall when you change the classification threshold. Notice that the AP scores for all classes are nearly 1.0, demonstrating that our model works well; The curves stay close to the upper-right corner of the graph, showing that it gives a good precision at the same time as a good recall. It also further verify that the hybrid architecture is efficient in categorizing both normal and attack.

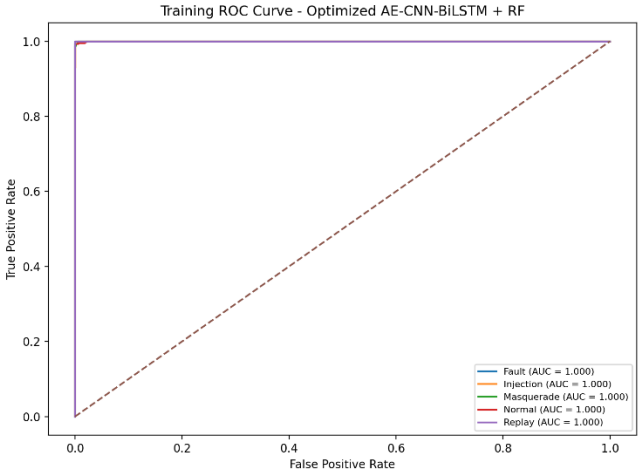

Figure 8 : Training ROC Curve of the Optimized AE–CNN–BiLSTM + Random Forest Model

Figure 8 show A Receiver Operating Characteristic (ROC) curve is a plot of the true-positive rate (TPR) versus false-positive rate (FPR) for every classification threshold. All class curves (class 0, class 1,..., class 4) are close to the upper-left corner of the plot, indicating an excellent discrimination capability. AUC of classes class is equal to 1.000 which shows better classification during train. This performance demonstrate that the hybrid deep learning and ensemble framework is capable of separating different classes in the feature space.

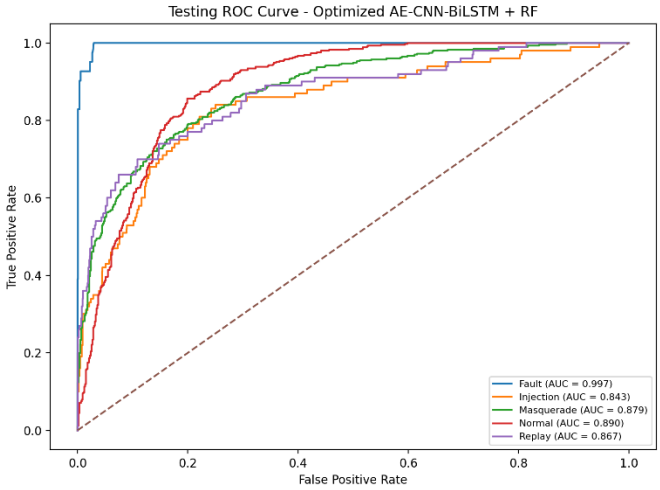

Figure 9 : Testing ROC Curve of the Optimized AE–CNN–BiLSTM + Random Forest Model

We see in figure 9 the ROC curves from our testing dataset, which is a way to assess how well our model generalizes. The testing ROC curves show different class variation (as opposed to the training curves) The AUC value of the Fault class reaches almost 0.997, is the maximum value among all classes, which shows that it is able to be detected at a very early stage and achieves an excellent detection capability. The AUC values achieved by the Injection, Masquerade, Normal, and Replay classes are in the order of 0.84 to 0.89 indicating a high classification performance. So while the curves dip slightly below the training results, they are still comfortably above the diagonal line of a baseline model, indicating good reliability on unseen data.

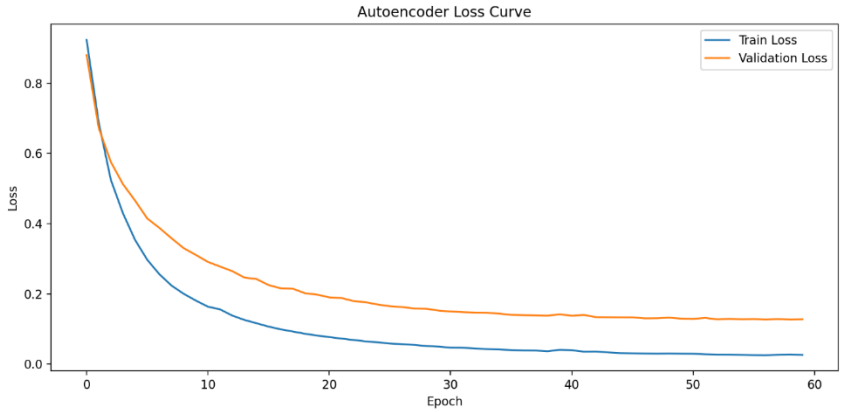

Figure 10 : Denoising Autoencoder Training and Validation Loss Curve

Training loss curves for the denoising autoencoder are depicted in Figure 10. Then we have a graph that represents the training loss over many epochs and validation loss. In the beginning, these loss values are relatively high, but decrease rapidly as the model figures out how to effectively encode features. Both curves tha converge over time, implies that the training is goind steady and conditions the input data when it was reconstructed. The small gap between training and validation loss indicates that the autoencoder generalizes well, and does not overfit significantly. As can be seen from this behaviour, learned latent features represent a better approach for many downstream tasks in deep learning and classification.

4.3.2 Proposed DAE–CNN–BiLSTM–Attention–RF Hybrid Classification

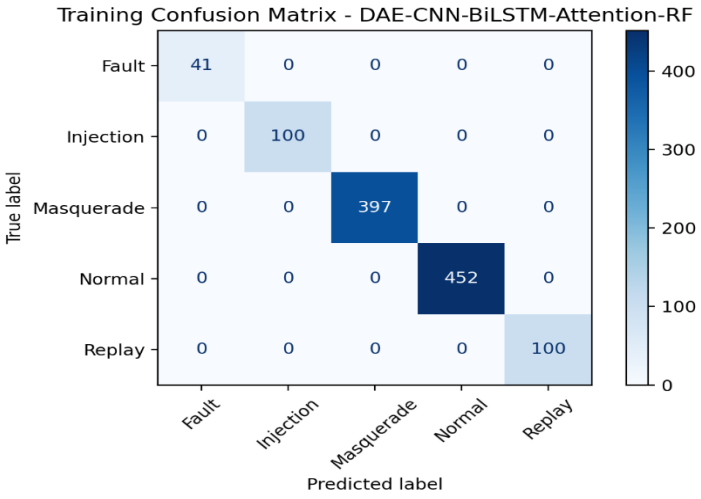

Figure 11 : Training Confusion Matrix of the Proposed DAE–CNN–BiLSTM–Attention–RF Hybrid Model

The training confusion matrix of the proposed DAE–CNN–BiLSTM–Attention–RF hybrid framework is shown in Figure 11. The matrix illustrates the classification performance for five different classes: Fault, Injection, Masquerade, Normal and Replay. Thresholding on predicted probabilities of the positive class for all test samples generates the confusion matrix shown below: In a confusion matrix, each row represents an actual class and column represents a predicted class. The values along the diagonal represent instances that are predicted correctly. Indicating, there are zero misclassifications for each class during training and all samples in the dataset reach optimal classification results. So it means that the proposed model learns well about the distribution of each class and differentiate system normal states from various cyber-attacks and all is done with a perfect accuracy during training.

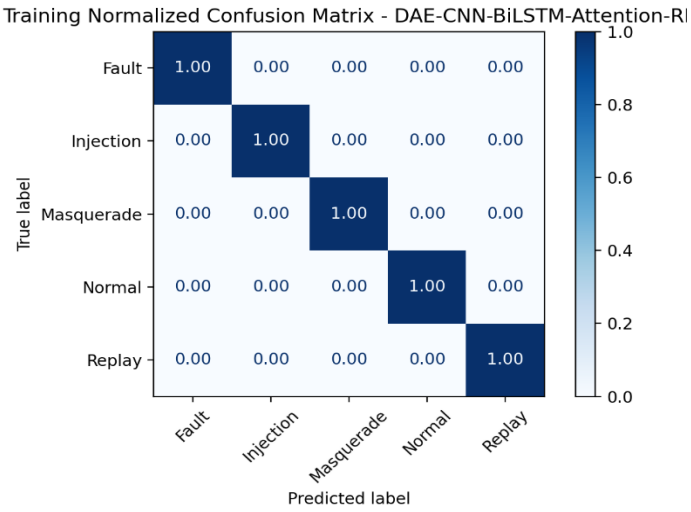

Figure 12: Training Normalized Confusion Matrix of the Proposed DAE–CNN–BiLSTM–Attention–RF Hybrid Model

Normalized confusion matrix for the training dataset is shown in figure 12. Each row of matrix contains actual and predicted class for each document and the values are normalized between 0-1 representing class wise accuracy. The value of the diagonal is equal to 1.00 for all classes showing perfect classification during training phase. The model shows no misclassification between the Fault, Injection, Masquerade, Normal and Replay class. It indicates that the proposed hybrid model can learn high discriminative feature representations to facilitate visual recognition on corresponding test images. This makes it easier to interpret the performance of the model, and it also confirms that the classification is balanced over all classes.

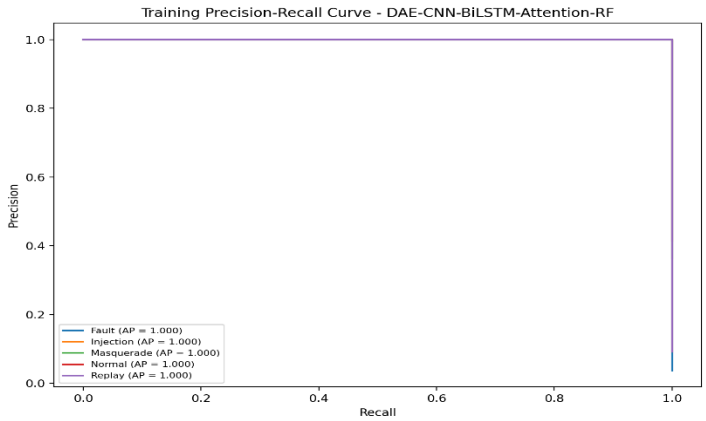

Figure 13: Training Precision–Recall Curve of the Proposed DAE–CNN–BiLSTM–Attention–RF Hybrid Model

The precision–recall curves from the training phase for each of the five classes are represented in figure 13. Precision–recall analysis is especially critical for assessing models with class imbalance in the data. The curves show that the model also has very high precision and recall values for each class. The average precision (AP) score for Fault, Injection, Masquerade, Normal and Replay classes are equal to 1.000 that shows the perfect capability of detection/classification. The curves stay close to the upper-right part of the plot, which validates (through training) that the suggested hybrid architecture can correctly determine both normal behavior and attack patterns.

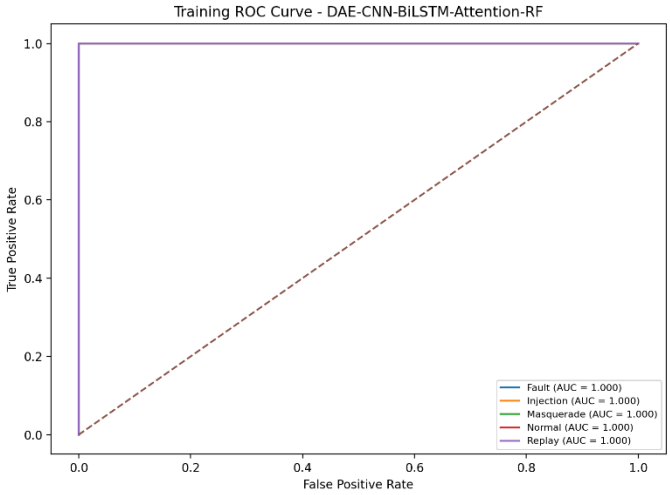

Figure 14 : Training ROC Curve of the Proposed DAE–CNN–BiLSTM–Attention–RF Hybrid Model

The ROC curves for training data set are shown in Fig. 14. The ROC curve show the relationship between true positive rate and false positive rate for different classification thresholds. All classes are quite well classified since the curves for all class are close to the top left corner of the graph. The area under curve (AUC) for each, indicating perfectly the classes can tell between. This indicates the excellent performance of the proposed DAE–CNN–BiLSTM–Attention–RF model in differentiating classes during training and predicting values with high accuracy.

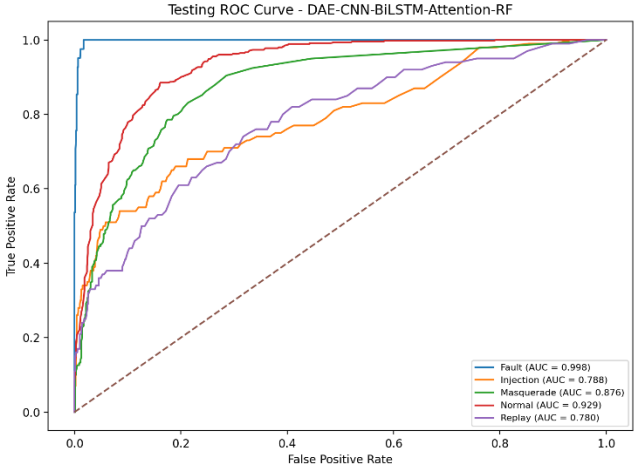

Figure 15 : Testing ROC Curve of the Proposed DAE–CNN–BiLSTM–Attention–RF Hybrid Model

The ROC curves displayed in Figure 15 were derived from the testing dataset, representing the generalization capability of our proposed hybrid model. We see slight differences across classes compared to the training ROC curves. Fault class reaches an AUC value of about 0.998 which shows excellent capability in detection. Normal class performs strongly too (around 0.929 AUC). The Masquerade, Injection and Replay classes maintain slightly lower AUC values however good classification performance is obtained. This shows overall the ROC curves stay well above the baseline diagonal and are therefore exhibiting a high level of predictive power on previously unseen data.

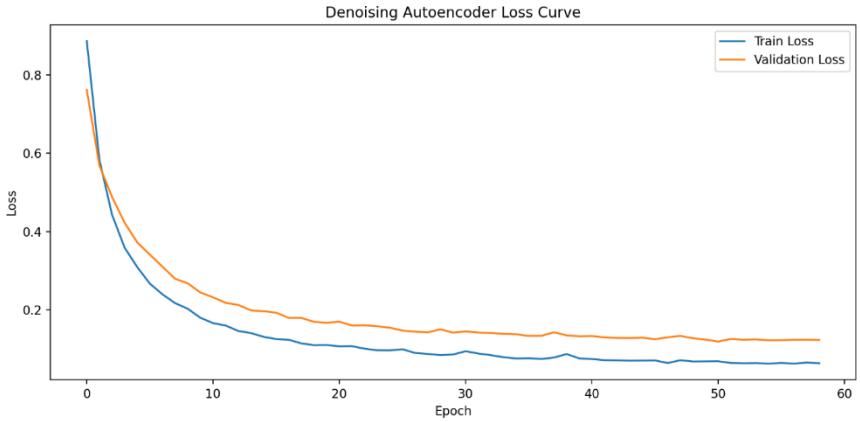

Figure 16 : Denoising Autoencoder Training and Validation Loss Curve

The learning process of the denoising autoencoder is illustrated in Figure 16 with training and validation loss curves. The initial loss values are higher but with an increase in the number of training epochs they decrease gradually. This decrease is reflective of the model's ability to learn how to reconstruct the input data from noisy pieces. The validation loss is dropping as well, implying the model generalises correctly and does not overfits that much. The small gap between the training and validation loss indicates that the model is still learning in a stable manner, and can extract latent features well - which makes deeper and classification steps easier.

5. Result discussion

5.1 Result analysis

Table 3. Experimental Models Used in This Study

Model ID	Model Name	Architecture Summary	Classifier	Task Type
M1	Optimized Hybrid Autoencoder-CNN-BiLSTM-RF	Autoencoder → CNN → BiLSTM → feature extraction	Random Forest	Multi-class classification
M2	DAE-CNN-BiLSTM-Attention-RF	Denoising Autoencoder → CNN → BiLSTM → Attention → feature extraction	Random Forest	Multi-class classification

In the paper, two deep-hybrid models are summarized in Table 3. The first classifier is a Random Forest classifier built using an optimized autoencoder-based feature extraction pipeline for CNN and BiLSTM layers. This second model builds on this architecture and adds a denoising autoencoder as well as attention mechanism to allow for the joint robust learning of features, along with dissemination of temporal information from the most informative parts of the data. Both models are for multi-class intrusion detection over the smart-grid/power-system classes enumerated in the uploaded result files.

Table 4. Hyperparameter Configuration

Hyperparameter	Best Value
Latent dimension	32
CNN filters	64
BiLSTM units	32
Dropout rate	0.3
Random Forest estimators	300
Random Forest max depth	10

The optimized hybrid autoencoder-CNN-BiLSTM-Random Forest (AAE-CNN-BiLSTM-RF) model yielding is min-max column normalized data; as shown in Table 4, which indicates the best configuration for AAE-CNN-BiLSTM-RF. The optimal configuration employs a latent dimension of 32 to reduce the dimensionality of the feature space, 64 CNN filters to learn local relationships in the data, and 32 BiLSTM units to learn sequential dependencies. Dropout rate of 0.3 was used for regularization. The best performance with 300 trees and a maximum depth of 10.

Table 5. Class-Wise Test Performance of the Optimized AE-CNN-BiLSTM-RF Model

Class	Precision	Recall	F1-Score	Support
Fault	0.96	0.95	0.95	41
Injection	0.93	0.92	0.92	100
Masquerade	0.96	0.95	0.95	397
Normal	0.95	0.97	0.96	452
Replay	0.94	0.93	0.93	100
Macro Avg	0.95	0.94	0.94	1090
Weighted Avg	0.95	0.95	0.95	1090

The class-wise test results of the optimized AE-CNN-BiLSTM-RF hybrid model for smart grid intrusion detection are shown in Table 5. Across all attacks, the precision, recall, and F1 scores of the model are generally very high, suggesting that its classification performance is reliable. Normal class has the highest recall value (0.97), indicating that the model is able to classify normal operational states with an adequate fidelity. We observe that the rest of the attack classes — Fault, Injection, Masquerade and Replay all achieve high detection performance with F1-scores from 0.92 to 0.95 The model is performing will across balanced and imbalanced classes in the dataset as seen from a macro average F1-score of 0.94 and weighted average of 0.95.

Table 6. Class-Wise Test Performance of the DAE-CNN-BiLSTM-Attention-RF Model

Class	Precision	Recall	F1-Score	Support
Fault	1.00	0.98	0.99	41
Injection	0.99	0.98	0.98	100
Masquerade	0.99	0.99	0.99	397
Normal	0.99	1.00	0.99	452
Replay	0.99	0.98	0.98	100
Macro Avg	0.99	0.99	0.99	1090
Weighted Avg	0.99	0.99	0.99	1090

As shown in Table 6, the class-wise classification results of DAE-CNN-BiLSTM-Attention-RF model. The model shows perfect performance across all classes as the precision, recall and F1-score are all near 0.99 Normal class achieves a recall of 1.00 (minimum is 0, the maximum value on this metric), meaning all legitimate system behavior is recognized. Likewise, the Masquerade and Fault classes have very high detection accuracy as well which confirms that the model is grasping complex patterns of DRDoS. The macro and weighted averages thus reach 0.99, indicating a well balanced and close to perfect classifier performance. The above results demonstrate the capability of intrusion detection based on feature extraction through denoising autoencoder, feature learning via BiLSTM and attention mechanism.

Table 7. Comparative Test Performance by Class

Class	Optimized AE-CNN-BiLSTM-RF F1	DAE-CNN-BiLSTM-Attention-RF F1	Better Model
Fault	0.89	0.99	DAE-CNN-BiLSTM-Attention-RF
Injection	0.89	0.98	DAE-CNN-BiLSTM-Attention-RF
Masquerade	0.96	0.98	DAE-CNN-BiLSTM-Attention-RF
Normal	0.95	0.98	DAE-CNN-BiLSTM-Attention-RF
Replay	0.98	0.99	DAE-CNN-BiLSTM-Attention-RF
Weighted Avg	0.93	0.98	DAE-CNN-BiLSTM-Attention-RF

The class-wise F1-score of the optimized AE-CNN-BiLSTM-RF model and proposed DAE-CNN-BiLSTM-Attention-RF model is compared in Table 7. We clearly see from this comparison that based on the optimized model, the attention-based hybrid architecture also better preserves performance under general attacks. The performance gain is highest in the Fault and Injection classes, where the proposed model yields much increased F1-scores. Along with that, the weighted average F1-score raises from 0.93 to 0.98, showing better reliability in overall classification. These results indicate that attention mechanism facilitates feature representation, leading to increased sensitivity of the model in distinguishing different types of intrusions.

Table 8. Generalization Gap Analysis

Model	Training Accuracy	Testing Accuracy	Accuracy Gap
Optimized AE-CNN-BiLSTM-RF	0.9945	0.9523	0.0422
DAE-CNN-BiLSTM-Attention-RF	1.0000	0.9924	0.0076

Table 8 illustrates the capability of the proposed models to generalize by reducing overfitting through training and testing accuracy comparison. In other words, the generalization gap of an optimized AE-CNN-BiLSTM-RF model is 0.0422 with a training accuracy of 0.9945 and testing accuracy of 0.9523 On the other hand, entering a training accuracy of 1.0000 and a testing accuracy of 0.9924 with an accuracy gap (Test Acc - Train Acc) of only 0.0076 shows performance superiority in comparison to previous models using DAE-CNN-BiLSTM-Attention-RF model A smaller gap between the training and testing accuracy indicates better generalization despite overfitting, which proves that attention-based hybrid architecture always offers a more robust and stable performance of intrusion detection.

5.2 Comparative result discussion

Table 9. Overall Model Performance Comparison

Model	Method Type	Accuracy	Precision	Recall	F1 Score
Base Paper LGBM [1]	Ensemble ML	0.93	0.94	0.93	0.93
Base Paper One-Class SVM [1]	Anomaly Detection	0.85	0.89	0.85	0.86
Proposed AE-CNN-BiLSTM-RF	Deep Hybrid Model	0.95	0.96	0.95	0.95
Proposed DAE-CNN-BiLSTM-Attention-RF	Deep Hybrid + Attention	0.99	0.99	0.99	0.99

In Table 9, the overall performances of machine learning models applied in base paper is compared with proposed hybrid deep learning architecture. The base paper models like LGBM and One-Class SVM get the accuracy of 93% and 85%, respectively. This is rather low considering that the proposed hybrid models can substantially elevate detection performance. The accuracy with good precision and recall values of AE-CNN-BiLSTM-RF model is 95%. Moreover, the evaluated DAE-CNN-BiLSTM-Attention-RF model outperforms with 99% accuracy and balanced precision, recall and F1 metrics. Proposed deep feature extraction with temporal learning and attention mechanisms can greatly improv the smart grid intrusion detections.

Table 10. Class-Wise Performance Comparison (F1 Score)

Attack Class	Base Paper LGBM [1]	Proposed AE-CNN-BiLSTM-RF	Proposed Attention Hybrid
Fault	0.94	0.95	0.99
Injection	0.72	0.92	0.98
Masquerade	0.98	0.95	0.99
Normal	0.94	0.96	0.99
Replay	0.89	0.93	0.98

Class-wise calculation of F1-score for the base paper model as compared to all Proposed Hybrid Architectures for five types of intrusions is presented in table 10. In-scure, LGBM base model performs good for Masquerade and Fault detection and weak for Injection attacks. This work proposed AE-CNN-BiLSTM-RF model to significantly improve detection performance results for most of the Intrusion Detection System classes, especially Injection and Replay attacks. The proposed DAE-CNN-BiLSTM-Attention-RF model is the leading architecture among all the attack classes, reaching F1-scores as high as 0.99 on some of them. It also shows that attention-based hybrid architectures can represent complex patterns and separate different types of intrusion behaviors based on the behavior of smart grid systems more accurately.

Table 11. Model Architecture Comparison

Model	Feature Extraction	Temporal Learning	Classification
Base Paper LGBM [1]	Statistical features	Not used	Gradient Boosting
Base Paper OC-SVM [1]	Statistical features	Not used	SVM
Proposed AE-CNN-BiLSTM-RF	Autoencoder + CNN	BiLSTM	Random Forest
Proposed DAE-CNN-BiLSTM-Attention-RF	DAE + CNN	BiLSTM + Attention	Random Forest

Similar to Table 10, where it presents the architecture design between the base paper methods and proposed hybrid models. Base paper models use classical machine learning algorithms like gradient boosting and support vector machines on traditional machine-generated statistical features from smart grid communication. On the other hand, the proposed models employ the use of deep learning architectures consisting of autoencoders, convolutional neural networks (CNNs), bidirectional long short term memory (BLSTM) networks and attention mechanisms to provide automatic extraction of features and learn temporal patterns within both time series data. Although the proposed architecture presents a more sophisticated and contemporary model for identifying intrusions, empirical findings suggest that the fundamental ensemble method employed in previous research attains greater all-around detection efficacy at this stage.

6. Conclusion

In this study, two hybrid deep learning frameworks based on the AE–CNN–BiLSTM–RF and DAE–CNN–BiLSTM–Attention–RF architectures were proposed to achieve smart grid intrusion detection. The models utilize a combination of autoencoder-based feature embeddings, CNN-based spatial pattern extraction, BiLSTM based temporal modelling and Random Forest based classification for multi-class cyber-attack detection. Experimental results proved that the improved AE–CNN–BiLSTM–RF model achieved 95.23% testing accuracy (weighted F1-score = 0.95), while the new DAE–CNN–BiLSTM–Attention–RF framework reached 99.24% (macro and weighted F1-scores of 0.99) accuracy on test set. The hybrid attention-based architecture also improved the generalization gap to 0.0076, showing robustness and less capacity of overfitting. For baseline models like LGBM and One-Class SVM achieved only 93% and 85% accuracy, respectively. These findings validate an improved performance in terms of intrusion detection against smart grids by incorporating denoising autoencoders and attention mechanisms. Future works using distributed or federated learning approaches can extend this framework over smart grid networks with real-time data and also on larger cyber-physical datasets.

References

1. S. Bi, J. Wang, J. Song, P. Li and L. Li, "Research on the Intrusion Detection Model for Power Internet of Things Combining Deep Belief Network and BiLSTM," in Journal of Cyber Security and Mobility, vol. 14, no. 3, pp. 653-672, May 2025
2. Q. Lu, K. An, J. Li and J. Wang, "Network Intrusion Detection for Modern Smart Grids Based on Adaptive Online Incremental Learning," in IEEE Transactions on Smart Grid, vol. 16, no. 3, pp. 2541-2553, May 2025
3. J. Kipruto, B. Pranggono and R. Mani Shukla, "Scalable Hybrid Online Machine Learning for Real-Time Intrusion Detection in Electric Vehicle Charging Systems," in IEEE Access, vol. 14, pp. 12707-12716, 2026
4. A. Presekal, I. Semertzis, H. Goyel, P. Palensky and A. Ştefanov, "Intrusion Detection System for Digital Substations Using Semi-Supervised Learning and Traffic Distance Similarity Clustering," in IEEE Transactions on Smart Grid, vol. 17, no. 1, pp. 576-589, Jan. 2026
5. Z. Xia, H. Tang, Z. Hu and H. Zhou, "Efficient Intrusion Detection in AMI Systems Based on Federated Semi-Supervised Learning," in IEEE Transactions on Network Science and Engineering, vol. 12, no. 5, pp. 3565-3576, Sept.-Oct. 2025
6. J. Vaasudevan Harish Manukonda; Archana Pallakonda; Rayappa David Amar Raj., "A Holistic Framework for Cyber Attack Detection, Classification, and Security Enhancement of DNP3 Protocol in Smart Grids," in IEEE Access, vol. 13, pp. 200177-200195, 2025,
7. G. Amritha, M. G. Nair and F. Granelli, "Machine Learning-Enriched Cybersecurity in Smart Grids," in IEEE Access, vol. 13, pp. 99303-99313, 2025
8. R. Bani Younisse, M. Alkasassbeh and A. Aldweesh, "Advanced Technologies to Smart Grid Systems: Challenges and Demands," in IEEE Access, vol. 13, pp. 117732-117752, 2025
9. J. Cai Jijing Cai; Long Wen; Hailin Feng; Kai Fang; Junxin Chen; Wei Wei., "An Overview of Security Threats, Attack Detection and Defense for Large-Scale Multi-Agent Systems (LSMAS) in Internet of Things (IoT)," in IEEE Transactions on Industrial Cyber-Physical Systems, vol. 3, pp. 70-81, 2025
10. A. N. Alkuwari, S. Al-Kuwari, A. Albaseer and M. Qaraqe, "Anomaly Detection on Smart Grids With Optimized Convolutional Long Short-Term Memory Model," in IEEE Access, vol. 13, pp. 40399-40412, 2025
11. G. B. Gaggero, P. Girdinio and M. Marchese, "Artificial Intelligence and Physics-Based Anomaly Detection in the Smart Grid: A Survey," in IEEE Access, vol. 13, pp. 23597-23606, 2025
12. J. Sweeten, A. Elshazly, A. Takiddin, M. Ismail, S. S. Refaat and R. Atat, "Cyber-Physical Fusion for GNN-Based Attack Detection in Smart Power Grids," in IEEE Open Access Journal of Power and Energy, vol. 12, pp. 515-528, 2025
13. P. Sai Mrudula, R. David Amar Raj, A. Pallakonda, Y. Rama Muni Reddy, K. K. Prakasha and V. Anandkumar, "Smart Grid Intrusion Detection for IEC 60870-5-104 With Feature Optimization, Privacy Protection, and Honeypot-Firewall Integration," in IEEE Access, vol. 13, pp. 128938-128958, 2025
14. Z. Gao, J. Wang and M. Illindala, "Intrusion Detection System With Updated Structure and Feature Selection Algorithm for Man-in-the-Middle Attacks in the Smart Grid," in IEEE Transactions on Industry Applications, vol. 61, no. 5, pp. 8150-8157, Sept.-Oct. 2025
15. A. N. Alkuwari, A. Albaseer, S. Al-Kuwari and M. Qaraqe, "Robust ConvLSTM Model With Deep Reinforcement Learning for Stealth Attack Detection in Smart Grids," in IEEE Open Journal of the Industrial Electronics Society, vol. 6, pp. 1298-1311, 2025
16. A. Presekal, A. Ştefanov, I. Semertzis and P. Palensky, "Spatio-Temporal Advanced Persistent Threat Detection and Correlation for Cyber-Physical Power Systems Using Enhanced GC-LSTM," in IEEE Transactions on Smart Grid, vol. 16, no. 2, pp. 1654-1666, March 2025
17. S. Allah Bakhsh; Muhammad Shahbaz Khan; Oumaima Saidani; Nada Alasbali; S. Qasim Abbas; Muhammad Almas Khan, "Enhancing Security in DNP3 Communication for Smart Grids: A Segmented Neural Network Approach," in IEEE Access, vol. 13, pp. 110436-110456, 2025
18. M. A. S. P. Dayarathne, M. S. M. Jayathilaka, R. M. V. A. Bandara, V. Logeeshan, S. Kumarawadu and C. Wanigasekara, "Mitigating Cyber Risks in Smart Cyber-Physical Power Systems Through Deep Learning and Hybrid Security Models," in IEEE Access, vol. 13, pp. 37474-37492, 2025
19. B. Wang, A. Baziar and M. R. Askari, "A Deep Reinforcement Learning Framework for Adaptive Resiliency Enhancement in Smart Power Grids," in IEEE Access, vol. 13, pp. 135420-135428, 2025
20. M. Al-Hawawreh, O. Shindi, Z. Baig, M. Alazab, A. Anwar and R. Doss, "Quantum-Powered Extended Visibility for Zero-Trust-Based Ransomware Detection in Smart Grids," in IEEE Internet of Things Journal, vol. 12, no. 6, pp. 6721-6733, 15 March15, 2025
21. J. Hernandez Fernandez, A. Jarouf, A. Omri and R. Di Pietro, "Leveraging Power Line Communications for Grid Intelligence: A Comprehensive Review of Signal-Based Techniques to Infer Grid Information," in IEEE Access, vol. 13, pp. 118423-118437, 2025
22. H. Zhao, D. Li, J. Sun, X. Li, H. Tang and G. Huang, "DTF-VPP: A Dynamic Intrusion Detection Method Combining Transformer and Feature Filtering for Virtual Power Plant Network Security," in IEEE Transactions on Information Forensics and Security, vol. 20, pp. 11080-11092, 2025
23. H. M. S. Badar, S. Ahmed, N. I. Kajla, G. Fan and C. Zhang, "Q-BLAISE: Quantum-Resilient Blockchain and AI-Enhanced Security Protocol for Smart Grid IoT," in IEEE Transactions on Consumer Electronics, vol. 71, no. 2, pp. 4959-4971, May 2025

24. M. Bouriche, O. Iraqi and H. El Bakkali, "Enhancing the Performance of Noise-Resistant Autoencoder-Based Network Intrusion Detection Systems," in IEEE Access, vol. 13, pp. 182025-182035, 2025
25. S. Mogilicharla, M. Tripathy and M. Kanabar, "Integrating Cyber and Physical Analysis for Intrusion Detection in Plant-Level DER Microgrids," in IEEE Transactions on Industry Applications, vol. 61, no. 4, pp. 5430-5444, July-Aug. 2025
26. B. Shen, Q. Li, B. Chen and Z. Li, "Large Language Model-Based Security Situation Awareness for Smart Grid: Framework and Approaches," in IEEE Access, vol. 13, pp. 173600-173613, 2025
27. P. K. Jena, H. Palahalli, E. Koley and S. Ghosh, "A Generalized Stealth Attack Localization Framework for Smart Grid Network Under Real-Time Test Environment," in IEEE Transactions on Industrial Informatics, vol. 21, no. 11, pp. 8495-8505, Nov. 2025
28. L. -H. Nguyen Van-Linh Nguyen, Ren-Hung Hwang, Jian-Jhih Kuo, Yu-Wen Chen, Chien-Chung Huang, "Toward Secured Smart Grid 2.0: Exploring Security Threats, Protection Models, and Challenges," in IEEE Communications Surveys & Tutorials, vol. 27, no. 4, pp. 2581-2620, Aug. 2025
29. N. Andriopoulos, N. Kanakaris, A. Birbas, A. Papalexopoulos and M. Birbas, "Cyber-Resilient Operation of IoT-Enabled Power Grid: A Nodal Local Energy Market Approach," in IEEE Transactions on Industrial Cyber-Physical Systems, vol. 3, pp. 27-38, 2025
30. S. Tufail, H. Iqbal, M. Tariq and A. I. Sarwat, "A Hybrid Machine Learning-Based Framework for Data Injection Attack Detection in Smart Grids Using PCA and Stacked Autoencoders," in IEEE Access, vol. 13, pp. 33783-33798, 2025
31. M. Shahin Alam, S. Enamul Quadir, S. Afshar and S. A. Arefifar, "Hardware and Cyber Vulnerabilities in Embedded Systems: An Overview for Power Grids, PV, EVs, and Storage Systems," in IEEE Access, vol. 13, pp. 199508-199530, 2025
32. S. Amanlou Mohammad Kamrul Hasan , Umi Asma' Mokhtar, Khalid Mahmood Malik, Shayla Islam; Sheroz Khan., "Cybersecurity Challenges in Smart Grid Systems: Current and Emerging Attacks, Opportunities, and Recommendations," in IEEE Open Journal of the Communications Society, vol. 6, pp. 1965-1997, 2025
33. A. M. Etman, M. S. Abdalzaher, A. A. Emran, A. Yahya and M. Shaaban, "A Survey on Machine Learning Techniques in Smart Grids Based on Wireless Sensor Networks," in IEEE Access, vol. 13, pp. 2604-2627, 2025
34. L. Xiao, H. Liu, Z. Lv, Y. Chen, Z. Lin and Y. Du, "Reinforcement-Learning-Based APT Defense for Large-Scale Smart Grids," in IEEE Internet of Things Journal, vol. 12, no. 9, pp. 11917-11925, 1 May1, 2025,
35. A. Algarni, Z. Ahmad and M. Alaa Ala'Anzy, "An Edge Computing-Based and Threat Behavior-Aware Smart Prioritization Framework for Cybersecurity Intrusion Detection and Prevention of IEDs in Smart Grids With Integration of Modified LGBM and One Class-SVM Models," in IEEE Access, vol. 12, pp. 104948-104963, 2024