STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
*Stat., Optim. Inf. Comput.*, Vol. **x**, Month 202x, pp 0–22.
Published online in International Academic Press (www.IAPress.org)

# A Novel Deep Learning Technique for Big Data Anomaly Threat Severity Prediction in e-Learning

Chinnakka Sudha1[1,*], Sreenivasulu Bolla[2]

[1] *Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation,India*
[2] *Associate Professor, Department of Artificial Intelligence & Data Science, Koneru Lakshmaiah Education Foundation, India*

**Abstract** E-learning platforms are susceptible to several anomalies, including abnormal learning behaviours, system abuse, and cybersecurity (CS) attacks, which interfere with the learning process. Conventional methods for detecting anomalies have limitations when applied to high-dimensional data, skewed distributions, and poor feature selection, leading to incorrect severity level predictions. To overcome these challenges, a novel Sea Lion Multilayer Perceptron (SLMP) model is introduced for predicting anomaly severity levels. First, an e-learning anomaly dataset is gathered and trained in a Python environment. Hence, the data is preprocessed, and the Sea Lion Optimisation (SLO) is used to select the best features, retaining only the most significant attributes. Subsequently, the chosen informative features are employed for further processing. Moreover, prediction and classification are performed using the SLMP model. Finally, Performance metrics such as F-score, Accuracy, recall, precision, and error rate are used to evaluate the effectiveness of the model. The results confirm the efficacy of the developed SLMP framework over current methods, illustrating its strength in optimizing predictive efficiency for anomaly severity detection in e-learning systems.

**Keywords** Big data; Threat severity; Anomaly features; ELearning; Preprocessing; Prediction

**DOI:** 10.19139/soic-2310-5070-2926

## 1. Introduction

By providing on-demand access to rich digital content via smart devices and cloud infrastructures, the explosive expansion of e-learning platforms has revolutionised contemporary education [1]. This offers previously unheard-of scalability and personalised learning, but it also leaves vast amounts of private staff and student data vulnerable to illegal access and hacks [2]. The use of cloud computing expands the traditional security perimeter, and the proliferation of smart devices generates a large number of network endpoints, creating new vulnerabilities that hackers can exploit [3]. As a result, anomaly prediction and detection are now crucial for maintaining safe and reliable online learning environments [4]. Cloud computing presents special security challenges [5], despite being vital for handling and processing large amounts of data [6]. Cloud-based e-learning systems are appealing targets for hackers due to their inherent benefits of flexibility, scalability, and cost-efficiency [7]. Vulnerabilities can jeopardise sensitive data and interfere with essential digital learning services [8, 9]. It is especially crucial to forecast the severity of abnormalities in these settings [1]. Security teams can prioritise incident response with accurate severity classification, ensuring that high-risk threats receive resources and attention promptly [11]. However, it is challenging to distinguish between benign anomalies and truly malicious activities due to the dynamic and heterogeneous nature of e-learning data, which is fueled by user interactions, log production, and real-time analytics [12]. High false-positive rates and delayed reactions are often the result of the inability of

*Correspondence to: Chinnakka Sudha1 (Email: ). Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 202x International Academic Press

traditional machine learning and rule-based intrusion detection systems to adjust to changing attack tactics [13]. Because of this, e-learning platforms are susceptible to sophisticated incursions, such as phishing, insider threats, and zero-day attacks, which take advantage of poorly configured analytical apps or dispersed cloud services [14].

Deep learning (DL) [15], which possesses excellent capabilities for modelling complex, nonlinear patterns in large datasets and for enhancing both detection accuracy and operational efficiency, has drawn increasing attention from researchers in an effort to overcome these constraints [17, 18]. The efficacy of DL in a range of cyber and anomaly detection scenarios is demonstrated by several recent studies [19, 20]. For example, to decrease false positives in healthcare security, Bhaskaran et al. [22] proposed a systematic anomaly-detection method for electronic health data, utilising clustering metrics such as the Silhouette and Dunn scores. While Usman et al. [25] developed cutting-edge cybersecurity techniques to identify rogue IP addresses instantly, Oprea et al. [24] examined smart meter readings to identify fraudulent usage patterns. Selim et al. [27] explained anomaly detection for industrial control systems and IoT contexts, whereas Ofoegbu et al. [26] discussed the scalability issues of conventional security solutions in increasingly digitalised sectors. In other application domains, Sheoran et al. [41] compared various machine-learning models for human-activity classification, and Fan et al. [40] demonstrated gradient boosting and ANN techniques for estimating small-sample emissivity. Naik et al. [43] examined the use of big data analytics in healthcare security and precision medicine, while Goswami et al. [42] presented a Lion-Salp Swarm Optimisation Algorithm (LSSOA) to identify application-layer DDoS attacks in Internet of Medical Things networks. In addition to highlighting important issues, including significant false positives, substantial computing costs, and limited capacity to generalise across heterogeneous data sources, these experiments collectively demonstrate the range of current anomaly detection and deep learning methods. Even with these developments, efficient feature selection and model optimisation are still essential for large-scale, high-dimensional e-learning data. To prevent overfitting and reduce computational cost, deep neural networks require careful tuning [39]. To improve detection performance in large-scale security analytics, bio-inspired metaheuristic algorithms—such as genetic algorithms, particle swarm optimisation, and other contemporary swarm intelligence techniques—have been investigated for selecting the most informative features and adjusting classifier hyperparameters. Among these methods, the Sea Lion Optimizer (SLO) is a relatively recent algorithm that draws inspiration from sea lions' spiral foraging and group-hunting strategies. By effectively searching high-dimensional solution spaces and avoiding local minima during optimisation, SLO exhibits a great balance between exploration and exploitation.

There is, nevertheless, a significant research gap, as SLO has not yet been utilised for the specific goal of anomaly-severity prediction in e-learning networks. We propose a Sea Lion Multilayer Perceptron (SLMP) architecture to bridge this gap by integrating a deep Multilayer Perceptron (MLP) classifier with SLO-based feature selection. To ensure that the MLP receives only the most informative inputs for training and inference, the proposed method first utilises SLO to extract the most relevant features from a large cyber-threat dataset. By avoiding the curse of dimensionality and excessive noise, this integration enables the MLP to concentrate on high-quality features. In complex e-learning contexts, SLMP achieves robust, high-accuracy prediction of low-, medium-, and high-severity threats by combining the potent representation-learning capabilities of deep neural networks with metaheuristic optimisation for feature selection. This combination effectively reduces false positives, enhances detection accuracy, and provides a reliable, scalable, and intelligent security framework for next-generation e-learning systems, as indicated by experimental data.

This study builds on previous findings by presenting a Sea Lion Multilayer Perceptron (SLMP) architecture for predicting the severity of cyberthreats in online learning settings. The process consists of two primary phases. The high-dimensional cyber-threat dataset is first subjected to feature selection by the Sea Lion Optimiser (SLO), which successfully eliminates redundant or noisy attributes and keeps just the most discriminative features for severity prediction. Based on a fitness function determined by preliminary classification accuracy, the method iteratively refines each possible "sea lion" in the SLO population, each of which represents a distinct feature subset. A deep Multilayer Perceptron (MLP) classifier is then trained using the improved features, assigning the optimised feature set to three severity levels: low, medium, and high. This hybrid approach leverages MLP's representational ability to capture intricate, nonlinear relationships and SLO's exploratory potential to overcome the curse of dimensionality. Consequently, in real-time e-learning security applications, the suggested SLMP model offers a scalable and high-accuracy anomaly-severity prediction approach that reduces false alarms and enables prioritised incident response.

### *1.1. Related works*

The efficiency of bio-inspired optimisation methods for improving anomaly detection through feature selection and machine-learning model tuning has been demonstrated by a significant amount of recent research. The Whale Optimisation Algorithm (WOA) [23], Grey Wolf Optimiser (GWO) [16], and Particle Swarm Optimisation (PSO) [21] are well-known methods. PSO produces rapid convergence but can cause premature standstill in complicated, high-dimensional areas because it simulates the collective motion of bird flocks, with each particle changing its position based on both personal and global best experiences. Although GWO balances exploration and exploitation by modelling the grey wolf cooperative hunting hierarchy (alpha, beta, delta, and omega), its permanent leadership structure may restrict flexibility when search areas vary dynamically. WOA records humpback whales' spiral bubble-net hunting, which combines random search with encircling prey. WOA can risk trapping in local optima and losing diversity in late iterations, despite its effectiveness in local exploitation.

Building on these concepts, several studies have effectively used hybrid metaheuristics for problems such as anomaly detection and network intrusion detection. A variety of bio-inspired and hybrid algorithms were presented by Gangula and associates, including deep feature-based intrusion detection systems [34], firefly-optimization ensembles [33], intelligent intrusion-prevention frameworks [32], and improved flower pollination algorithms with ensemble classifiers [35]. While other recent work extends similar tactics to large-scale IoT contexts [38], they also investigated more sophisticated hybridisations, such as a layered BiLSTM elastic regression classifier optimised with the Aquila algorithm [37] and a Bottlenose Dolphin–Artificial Fish Swarm algorithm [36]. These investigations validate the effectiveness of metaheuristic optimisation in reducing false positives, improving detection precision, and maintaining computational efficiency across various cybersecurity domains.

The Sea Lion Optimiser (SLO), which is used in our suggested SLMP architecture, provides a hybrid search mechanism that blends spiral foraging, random exploratory dives, and cooperative herd movement, in contrast to the algorithms mentioned above. Every virtual "sea lion" dynamically modifies the ratio of exploration to exploitation based on population diversity and the best available global solution. In later search stages, this adaptive switching maintains population diversity more effectively than WOA, reduces the premature convergence observed in PSO, and offers more adaptable leadership than GWO. With feature spaces that are big, non-convex, and teeming with local optima, SLO is especially well-suited to high-dimensional cyber-threat datasets. The proposed method surpasses the capabilities demonstrated in PSO-, GWO-, WOA-, and other metaheuristic-based intrusion-detection research [32, 33, 34, 35, 36, 37, 38] by utilising SLO for feature selection prior to deep MLP training, resulting in improved predicted accuracy and resilience.

The Key contribution of the work is described as follows,

- Initially, the threat severity dataset is collected and trained on the Python system.
- Hence, a novel Sea Lion Multilayer Perceptron (SLMP) framework has been developed with predictive features.
- Consequently, the collected data is processed, and the informative attributes are selected using Sea Lion optimization.
- Subsequently, the threat severity is predicted by tracing the selected features, and SLMP classifies them.
- Henceforth, to validate the performance of the developed model, metrics such as F-score, accuracy, recall, precision, and error rate are evaluated.

This paper's second section contains recent, relevant work, and its third section describes the existing system challenges. The system challenge is developed in the fourth segment, and the case study and performance validation are covered in the fifth section. The work is finally completed in the sixth section.

## 2. Proposed Methodology

Threat forecasting in e-learning refers to detecting potential threats that impact the security and integrity of online learning platforms. One of the significant benefits of threat prediction for e-learning is to determine the level of severity of the identified threat. It enables high security. Hence, a novel sea lion multilayer perceptron (SLMP)

has been developed to identify the threat severity level with high accuracy. Initially, the dataset is collected and processed to enhance the data quality. Moreover, the threat anomalies are selected by the sea lion optimization, and based on them, the severity level is predicted and classified.

Finally, the performance of the developed framework is validated using several key performance metrics, including accuracy, precision, recall, F-score, and error rate. The proposed architecture is displayed in Figure 1. The working process is described as follows.

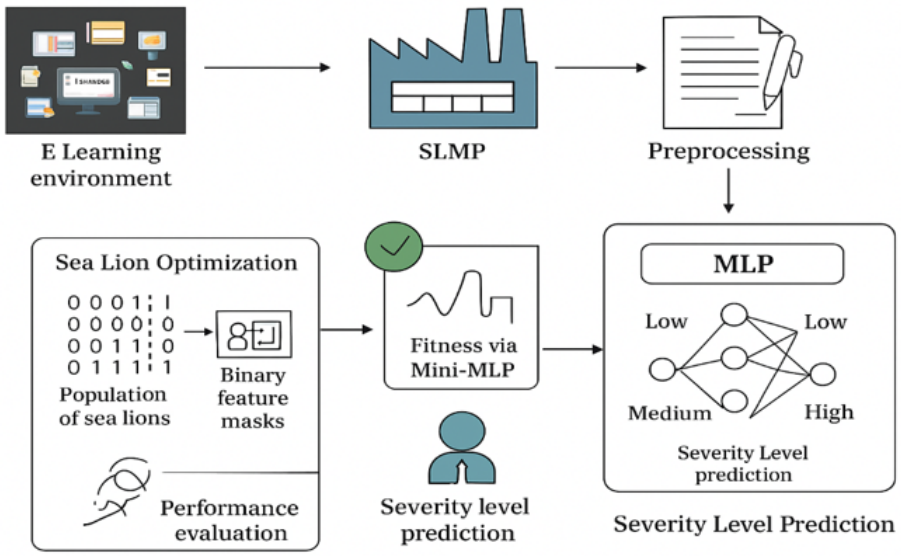


Figure 1. Proposed Architecture

The entire Sea Lion Optimization–Multilayer Perceptron (SLMP) workflow for anomaly-severity prediction in an e-learning security context is shown in Figure 1. The e-learning platform's data undergoes an initial preprocessing step, where robust scaling, noise filtering, and outlier management prepare the inputs for modelling. A population of candidate feature subsets is then initialised via the SLO module; each sea lion's "position" encodes a binary mask of the features that are accessible. To direct the swarm towards the most informative subset, a fast internal classifier assesses each population member, and its fitness is quantified as cross-validated MLP accuracy through iterative encircling and spiral-attack updates. The chosen features are then sent to the MLP classifier, which uses the intricate architecture depicted in the diagram: a softmax output layer that generates low, medium, or high-severity predictions; three hidden layers with 128, 64, and 32 ReLU neurons; and an input layer that is proportionate to the selected characteristics. Ultimately, a high-resolution end-to-end pipeline that combines bio-inspired optimisation for feature selection with a deep neural model for accurate threat-severity classification is produced by aggregating the severity-level outputs for performance evaluation using accuracy, precision, recall, and F-score metrics.

### *2.1. Process of SLMP*

The developed SLMP model is designed by integrating SLO [28] with a multilayer perceptron (MLP) to improve predictive accuracy and optimization effectiveness. The dynamic foraging behaviour of SLO is used to address the delayed convergence and local optima problems in MLP by adjusting the weight and bias parameters of the MLP. The data initialization is executed by Eqn. (1):

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

Here $x_i \in \mathbb{R}^d$, $y_i \in \{1, 2, 3, \ldots, K\}$, where the number of data features is denoted as $d$ and the severity classes are determined as $K$, also the row-wise features are defined as $\mathbb{R}^d$.

Preprocessing is a crucial step in predicting cyber threat severity in e-learning systems. The collected data usually comprises inconsistencies, missing values, and imbalanced class distributions, which adversely affect model performance. It prepares the data for further processing. Eqn. (2) executes preprocessing.

$$\tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j + \varepsilon}. \tag{2}$$

Where $\mu_j, \sigma_j$ are the empirical mean and standard deviation of the feature $j$. The first step in the process is to compile the entire dataset, where each record is assigned a severity class designation and includes some numerical features. Because real-world cyber-threat data often contains missing items, errors, and unbalanced class distributions that can impair prediction performance, preprocessing is emphasised as a crucial step. The stability of model optimisation is increased and bias from large-valued characteristics is eliminated by normalizing the raw feature values to provide each feature a consistent scale. This normalization ensures that the multilayer perceptron and Sea Lion Optimisation components that follow operate effectively on well-conditioned data, allowing the classifier to learn precise decision boundaries for severity prediction and the optimiser to modify weights and biases efficiently.

*2.1.1. Anomaly threat feature selection* Anomaly-threat feature selection is a crucial process in predicting cyber-threat severity to enhance detection accuracy while minimising redundant data. SLO, a bio-inspired algorithm, is employed to select the most informative features. The algorithm strikes a balance between exploration, which searches for varied feature subsets, and exploitation, which refines the best-chosen features to achieve maximum classification performance. The feature selection is expressed in Eqn. (3) [28],

$$\mathbf{P}_i^{t+1} = \text{clip}\left(\mathbf{P}_i^t + \lambda r_1 \Theta \left(\mathbf{g}^t - \mathbf{P}_i^t\right) + \gamma r_2 \Theta \left(\mathbf{P}_{r(i)}^t - \mathbf{P}_i^t\right) + \boldsymbol{\eta}_t, 0, 1\right), \tag{3}$$

where $M$ is the population size, $\mathbf{P}_i^{t+1} \in [0,1]^d$ is the position vector of the agent $i$ at iteration $t$, $\mathbf{g}^t \in [0,1]^d$ is the best global solution at iteration $t$, $\lambda$ and $\gamma$ are the scalar coefficients controlling exploration and exploitation, and the random perturbation is $\boldsymbol{\eta}_t$ at iteration $t$. $r_1, r_2 \in [0,1]^d$ are the element-wise uniform random vectors, $\mathbf{P}_{r(i)}^t$ denotes the randomly chosen peer's position, $\Theta$ indicates the elementwise product, clip$(..., 0, 1)$ and constant values to $[0,1]$. After the update, get the binary mask $\mathbf{f}_i^{t+1} \in [0,1]^d$ using Eqn. (4).

$$f_{i,j}^{t+1} = \begin{cases} 1 & \text{if } \sigma(\alpha(p_{i,j}^{t+1} - \tau)) \geq 0.5, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

Or more simply: $P_i^{t+1} = 1\{p_{i,j}^{t+1} > \tau\}$ here $\sigma$ is the sigmoid, $\alpha$ controls steepness, and $\tau$ is the threshold (e.g., 0.5). For each candidate mask $f$, compute fitness using Eqn. (5).

$$fitness(f) = \text{validation score of classifier trained on } \widetilde{T}\Theta f, \tag{5}$$

In reality, SLO begins with a population of potential feature sets and iteratively refines them, striking a balance between concentrated refinement in promising regions and extensive exploration of the feature space. Small perturbations and random peer solutions are included at each stage to prevent the search from becoming stuck in local optima. Each candidate feature set is transformed into a binary "mask" that indicates which features are chosen following each update. The classifier is subsequently trained, and its validation accuracy is measured to assess the quality of each mask. This technique inherently encourages feature subsets that consistently produce better detection performance throughout subsequent iterations. Only the most informative features contribute to intrusion detection in the suggested model since the multilayer perceptron receives the best-performing subset determined by SLO for final training and prediction.

*2.1.2. Severity level prediction* Cyber threat severity level prediction in e-learning security systems is crucial for enhancing cybersecurity and preventing unauthorised access or malicious behaviour. MLP executes this based on the features SLO has selected. The optimized MLP classifier computes the severity classes using an activation

function. Let the trained MLP(with final softmax) produce class-probabilities for the sample $i\hat{P}_i = MLP(x_i \Theta f) \in \Delta^{K-1}$, $\hat{P}_i^{(k)} = \Pr(\text{class} = k \mid x_i)$, define severity score for class labels: assign scalar severity levels $s_k = k - 1$. Per-sample expected severity and batch severity as measured in Eqn. (6) and (7).

$$d_i = \mathbb{E}[S \mid x_i] = \sum_{k=1}^{K} s_k \cdot \hat{P}_i^{(k)}. \tag{6}$$

$$D = \frac{1}{N_{\text{batch}}} \sum_{i \in \text{batch}} d_i. \tag{7}$$

Following the identification of the ideal feature subset by the Sea Lion Optimisation step, each processed sample is sent to the MLP, which uses a softmax activation in the last layer to produce a probability distribution across all potential severity levels. These probabilities indicate the likelihood that a specific event falls into the low, medium, or high severity classes. Each severity class is also assigned a numerical score to facilitate interpretation. This allows for the quantification of both individual forecasts and the overall batch-level severity. The weighted sum of the class scores based on the projected probability is then used to calculate the per-sample expected severity. These values are then averaged across all samples in a batch to determine the overall severity measure of the batch. This method can assist in prioritising replies in an e-learning security environment by offering both discrete class labels and a continuous severity index.

$$C(T_s) = \begin{cases} \text{If}(D = 0) & \text{Low} \\ \text{If}(D = 1) & \text{medium} \\ \text{If}(D = 2) & \text{high} \end{cases} \tag{8}$$

The classification is done by Eqn. (8) [28]. Here, the classification variable is denoted as $C$. The model classifies severity levels as 0, 1, and 2, corresponding to low, medium, and high severity, respectively. The algorithm for the developed framework is provided in pseudo-code format, and the workflow of the developed model is sequentially displayed in Figure 2.

The entire pipeline for anomaly-severity prediction in an e-learning security environment is displayed in the comprehensive SLMP flowchart (Figure 2). The first step is data initialisation, which involves loading the raw cyberthreat dataset and dividing it into a 20% test set and an 80% training set. Outliers are winsorized, missing values are imputed, and features are strongly scaled to stabilise their ranges during the preprocessing phase. The SLO initialisation then generates a population of roughly thirty agents, each of which encodes a binary mask representing a subset of potential features. Every agent's mask is applied to the training data during the fitness evaluation process, and a tiny MLP calculates the classification accuracy, which is then converted into the fitness score. To balance exploration and exploitation, the SLO update phase employs migration and spiral-foraging behaviours to relocate each agent. A stopping criterion, such as no further improvement or a maximum of fifty iterations, establishes when the search concludes and the best feature subset is selected. Using a deeper network with hidden layers of approximately 128, 64, and 32 neurones, ReLU activations, the Adam optimiser, and class-weighted loss to correct imbalance, the pipeline proceeds to final MLP training after the subset is fixed. After classifying each sample as low, medium, or high threat, the trained model predicts severity. It is then assessed using 10-fold cross-validation, which yields the mean and standard deviation of accuracy, precision, recall, F1-score, and error rate. This flow demonstrates how the MLP provides high-accuracy severity classification while the SLO manages global feature selection. Only feature selection is done using the Sea Lion Optimisation method. SLO examines potential feature subsets following preprocessing and outputs the single subset with the highest classification accuracy on the training folds. To ensure that only the most informative features are used during training and testing, this ideal subset is then directly provided as the fixed input layer to the subsequent MLP classifier.

To identify and categorise cyber-anomaly threats in e-learning data, the SLMP Anomaly-Severity Prediction algorithm 1 is a comprehensive pipeline that combines strong preprocessing, an adaptive Sea Lion Optimiser (SLO) for feature selection, and a potent multilayer perceptron (MLP) classifier. The initial steps involve importing

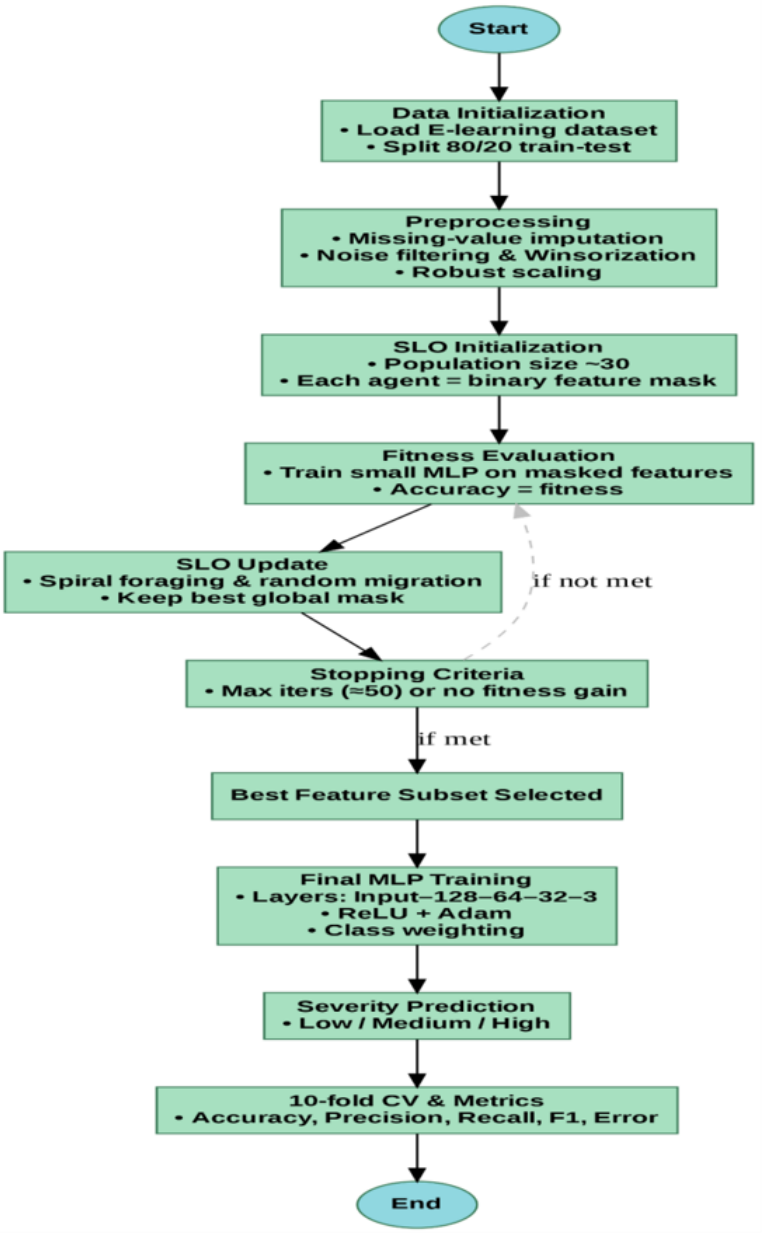


Figure 2. SLMP flowchart

the raw dataset, dividing it into approximately 80% training and 20% testing sets, and performing three stages of preprocessing: robust scaling to address skewed distributions, winsorization to remove extreme outliers, and median imputation to fill in missing values. A population of roughly thirty "sea lions" is randomly initialised by the optimiser at the start of the SLO feature-selection stage. Each sea lion is represented by a vector of continuous values between 0 and 1, as long as the total number of input features. Each element of this vector represents a single characteristic; stronger inclusion is indicated by values near 1, while exclusion is indicated by values near 0. The vector is thresholded at 0.5 to provide a binary mask that explicitly identifies a subset of characteristics, allowing for the testing of a potential solution. By training a lightweight inner MLP—for instance, one hidden layer

**Algorithm 1: SLMP**

```
Start
    Load raw_dataset
    Split into train_set and test_set
    Preprocessing
        # Noise filtering and robust scaling
        train_clean = median_impute(train_set.features)
        train_trim = winsorize(train_clean, limit=1.5)    // cap outliers
        train_scaled = robust_scale(train_trim)
        Store statistics (medians, IQR) for later use
    Sea-Lion Feature Selection
        num_features = column_count(train_scaled)
        population = initialize_population(size = pop_size, dimensions = num_features)
        best_solution = None
        best_score = -inf
        no_improvement = 0
        FOR iteration from 1 to max_iterations:
            FOR each candidate in population:
                mask = convert_to_binary(candidate, threshold = 0.5)
                subset = select_features(train_scaled, mask)
                score = quick_eval(subset, train_set.labels)    // small MLP, few epochs
                update candidate's fitness with the score
            current_best = candidate with highest fitness
            IF current_best.fitness > best_score:
                best_score = current_best.fitness
                best_solution = current_best.vector
                no_improvement = 0
            ELSE:
                no_improvement += 1
            IF no_improvement >= patience_limit:
                BREAK    # stop early if no gain
            # Update each candidate position (SLO exploration/exploitation)
            FOR each candidate in population:
                guide = random_member(population)
                candidate.vector = sea_lion_update(
                    current = candidate.vector,
                    global_best = best_solution,
                    peer = guide.vector,
                    explore_weight = explore_rate,
                    exploit_weight = exploit_rate
                )
            candidate.vector = clip_between(candidate.vector, 0, 1)
        selected_features = convert_to_binary(best_solution, threshold = 0.5)
    Final MLP Training
        train_selected = select_features(train_scaled, selected_features)
        final_mlp = train_full_mlp(train_selected, train_set.labels, class_weighting = True)
    Test Prediction & Classification
        test_clean = median_impute(test_set.features, use_train_stats)
        test_trim = winsorize(test_clean, limit=1.5)
        test_scaled = robust_scale(test_trim, use_train_stats)
        test_selected = select_features(test_scaled, selected_features)
        predictions = final_mlp.predict_classes(test_selected)    # Low/Medium/High
Stop
```

with 16 neurons, ReLU activation, a learning rate of 0.001, and 5–10 epochs—and calculating its cross-validated classification accuracy, the system assesses a fitness score for each masked subgroup. This precision determines the candidate's fitness value. Sea lions change their positions during each iteration by moving partially towards randomly selected peers (exploration) and partially towards the current global optimum solution (exploitation). To prevent local optima, a small amount of Gaussian noise (standard deviation $\approx 0.01$) is applied. A maximum of around 50 iterations, a patience limit of roughly 10 iterations without improvement for early quitting, an exploration

weight close to 0.6, and an exploitation weight close to 0.4 are the primary hyperparameters that direct this search. The ultimate, most informative feature subset is provided by the best sea lion's binary mask when the stopping requirement is satisfied.

To manage imbalance across severity categories, these chosen features are then sent to a full-capacity MLP that is set up, for instance, with two hidden layers of 64 and 32 neurons, ReLU activations, the Adam optimiser (learning rate 0.001), batch size 64, and up to 100 epochs with early stopping and class weighting. Before the trained MLP predicts one of three severity levels—Low, Medium, or High—the test data goes through the same preprocessing and feature-masking procedures. Lastly, the algorithm reports performance parameters, including accuracy, macro-precision, macro-recall, macro-F1 score, and error rate. The SLMP approach effectively combines adaptive feature selection and deep learning to provide precise anomaly-threat severity prediction in extensive e-learning settings by explicitly mapping each sea lion's position to a feature subset and utilising classifier accuracy as the fitness function.

The underlying weights and biases of the multilayer perceptron itself are not adjusted by the Sea Lion Optimiser (SLO) in the suggested Sea Lion Multilayer Perceptron (SLMP) framework. Instead, SLO functions as a wrapper-based feature selection technique before the MLP is trained using the standard gradient descent method, which seeks the most informative subset of input variables. A lightweight MLP is briefly trained on the masked dataset to get a fitness score (cross-validated accuracy) after each sea lion in the population encodes a potential subset of features as a real-valued vector. The vector is then thresholded to a binary mask. To optimise that fitness, the SLO only modifies the components of these feature-mask vectors. A standard MLP with traditional backpropagation is trained from scratch on the decreased feature set to make the final severity prediction after the optimal feature subset has been identified. The phrase "adjusting the weight and bias parameters" should not be understood as part of the SLO search, but rather as the standard training stage of the final MLP following feature selection. While the MLP's weights and biases are optimised independently by gradient descent during its own training phase, SLO optimises the selection of input features. The hyperparameters of SLO and MLP are exposed in Table 1.

## 3. Result and Discussion

The SLMP is verified using Python software on Windows 10. The Threat severity datasets are first gathered and introduced to the proposed system. Hence, the proposed model eliminates noisy aspects, selects informative features, and detects and categorises threat severity levels. Table 2 explains the specification of the parameters used to implement the developed framework.

### *3.1. Case Study*

The Cyber Threat Data for New Malware Attacks dataset on Kaggle (https://www.kaggle.com/datasets/abdallahalidev/cyber-threat-data-for-new-malware-attacks) has 54,768 labelled records of cyber-threat events related to malware. Each record has a lot of information about the system and network behaviour. The goal variable is Threat Severity, which is split into three groups: Low (5,339 samples), Medium (33,015 samples), and High (16,414 samples). This makes it a substantially imbalanced three-class classification problem. For the experiment, the data were randomly split into 80% training (43,814 samples) and 20% testing (10,954 samples), while maintaining the same class distribution. The training set comprises 4,263 low-, 26,482 Medium-, and 13,069 high-severity records, while the test set contains 1,076 low-, 6,533 medium-, and 3,345 high-severity records. Each record includes a combination of numbers, such as the number of system calls, registry updates, file actions, and network connections, as well as occasional category flags that describe behaviours observed or attributes of malware families. These elements together make up the operational footprint of malware attacks and can be used to estimate the severity of new threats. The features present in the database are exposed in Table 3.

**Experimental Protocol:** Following the 80/20 train-test split, all feature selection and data preprocessing procedures were limited to the training set to prevent information leakage and provide an objective assessment. To be more precise, the dataset was initially split into 10,954 testing samples and 43,814 training samples while

Table 1. Hyperparameter details of SLO and MLP

| Component | Hyperparameter | Setting |
|---|---|---|
| **Sea Lion Optimization (SLO)** | **Population size** | **40 candidate solutions** |
| | Maximum iterations | 100 |
| | Search-space encoding | Binary vector of length $d$ (number of input features) |
| | Position update coefficients | $\lambda = 1.5, \gamma = 0.5$ |
| | Fitness function | 5-epoch MLP classifier accuracy (macro-F1) on training fold |
| | Stopping criteria | (a) max iterations reached or (b) no fitness improvement for 15 consecutive iterations |
| | Initialization | Uniform random sampling of $\{0, 1\}$ for each feature with 0.5 inclusion probability |
| **Multilayer Perceptron (MLP)** | Input layer | Dimension = number of features selected by the best SLO mask ($\approx 18$ out of 40 original) |
| | Hidden layers | Two fully connected layers |
| | Neurons per layer | Layer 1: 128, Layer 2: 64 |
| | Activation | ReLU for hidden layers |
| | Output layer | 3 neurons, softmax |
| | Optimizer | Adam |
| | Learning rate | 0.001 |
| | Loss function | Weighted cross-entropy |
| | Batch size | 64 |
| | Epochs | 100, with early stopping (patience = 10) |
| | Dropout | 0.3 after each hidden layer |
| | Weight initialization | He-normal |

Table 2. Operation specification

| Metrics | Parameters |
|---|---|
| Program | Python |
| Version | 3.7.14 |
| Operating System | Windows 10 |
| Network | Multilayer perceptron |
| Optimization | Sea lion |
| Dataset | Cyber threat severity |

maintaining the original class distribution. The Sea Lion Optimiser (SLO) feature selection procedure and the entire preprocessing pipeline, which includes median imputation for missing values, winsorization for outlier trimming, and robust scaling, were only applied to the training set. The ideal feature mask and scaling parameters (medians and interquartile ranges) derived from the training data were then stored and subsequently applied to the unaltered test set without recalculation. The test samples were never used to inform feature subset selection, imputation statistics, or model hyperparameter tuning. By ensuring that the Sea Lion Multilayer Perceptron (SLMP) model's final performance measurements accurately represent true generalisation to unknown inputs, this approach preserves the integrity of the evaluation.

The accuracy and loss patterns of the suggested model across 50 epochs are depicted in Figures 3 and 4. In Figure 3, the testing accuracy closely follows the training curve, with very slight variations. In contrast, the training accuracy increases rapidly to approximately 100% within the first epoch and remains virtually flat thereafter. In

Table 3. Dataset features

| Feature Name | Type | Description / Meaning |
|---|---|---|
| api_call_count | Numeric | Number of distinct API calls invoked by the malware sample |
| registry_mods | Numeric | Count of Windows registry keys created or altered |
| net_connections | Numeric | Total TCP/UDP network connections initiated |
| binary_size_kb | Numeric | Size of the executable binary in kilobytes |
| file_ops | Numeric | Number of files dropped, created, or deleted |
| entropy_score | Numeric | Shannon entropy of the binary, indicating the code obfuscation level |
| suspicious_imports | Numeric | Count of imported dynamic libraries flagged as suspicious |
| malware_family | Categorical | Optional categorical tag of known malware family (if present) |
| severity | Categorical | Target variable with classes: Low, Medium, High |

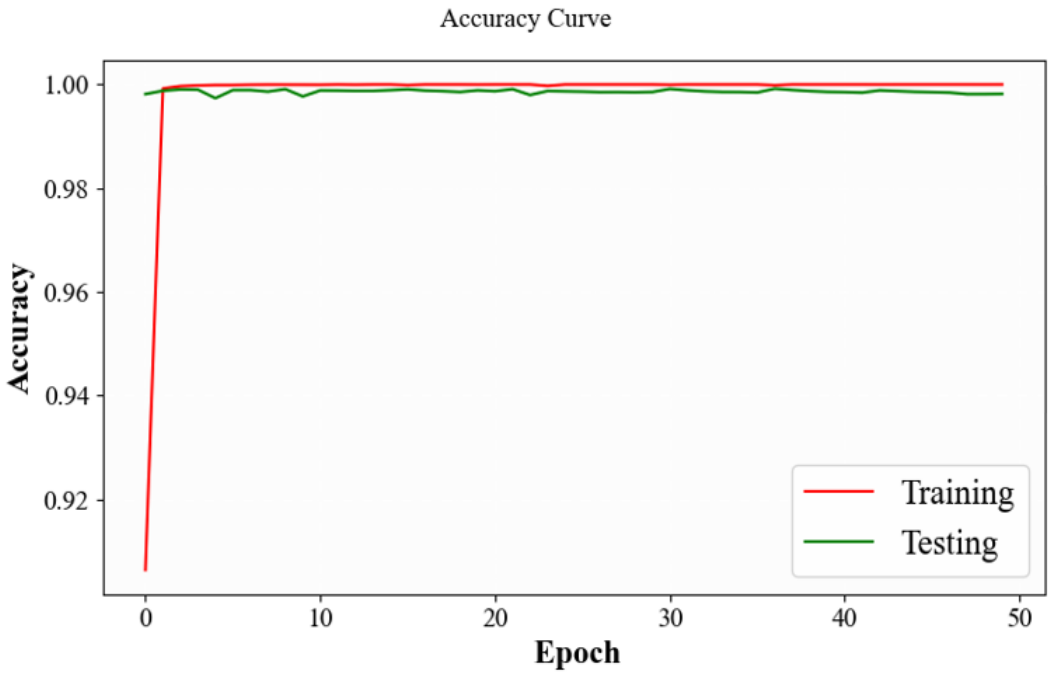


Figure 3. Accuracy Graph

Figure 4, the testing loss shows tiny, jagged oscillations instead of a flawlessly smooth reduction, but the training loss almost instantly drops to near zero and remains exceptionally low.

The random character of the test batches and the intrinsic variability of the real-world e-learning incursion data are the causes of these oscillations. The calculated loss may vary somewhat from one mini-batch to the next due to the slightly variable proportions of rare or borderline samples. The SLMP model converges quite quickly and performs steadily during training, as demonstrated by the loss curve in Figure 4. In the first epoch, the training loss (red) decreases significantly from a value slightly above 0.09 to almost zero, and it remains virtually flat near zero for the following 49 epochs. Additionally, the testing loss (green) begins at zero and very slightly varies within a minimal range. Rather than representing a real increase in mistakes, these slight oscillations reflect the inherent variability of the validation batches. Crucially, there is never a persistent rising trend in the testing loss, which would indicate overfitting. Instead, both curves closely follow one another, suggesting that the model can both fit the training data and generalise to new data. The fact that there is no discernible difference between the two curves

demonstrates that the SLMP setup, in conjunction with the regularisation and class-balancing techniques used, effectively avoids overfitting while preserving almost flawless prediction accuracy.

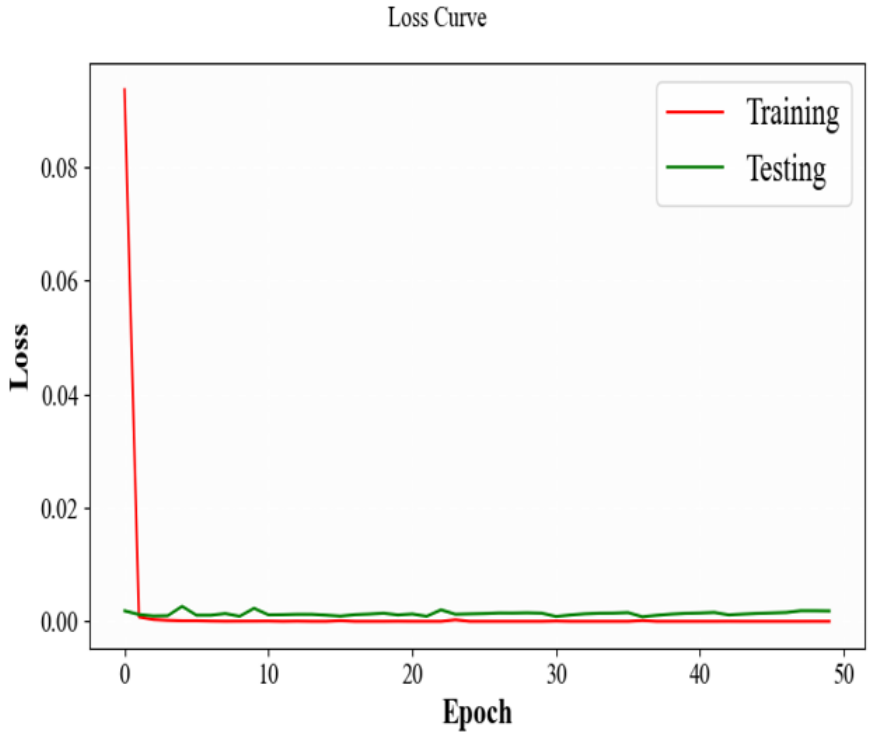


Figure 4. Loss graph

When testing a highly accurate model on a finite, class-imbalanced test set, the slight volatility is considered normal statistical noise. It does not signify overfitting or instability, as the overall size of these variations is very modest and the accuracy curve remains steady.

The confusion matrix is displayed in Figure 5. The final severity-classification outcomes of the suggested SLMP model on the held-out test set are compiled in the confusion matrix. The columns display the anticipated labels, and the rows display the actual class labels, which are Low, Medium, and High severity. Perfect discrimination for the least critical category was demonstrated by the fact that every incidence of 1,076 Low-severity threats was accurately identified as Low. The model's exceptionally high precision and recall for this majority class are demonstrated by the fact that 6,532 out of 6,533 samples were accurately recognised for the Medium-severity class, with only one case being incorrectly labelled as High. The fact that all 3,345 High-severity threats were accurately anticipated demonstrates the classifier's effectiveness in identifying the most critical abnormalities. The durability of the SLMP system and its applicability for prioritising responses to e-learning cyber threats based on severity are confirmed by the fact that the inclusion of just one misclassification across 10,954 test samples produces overall performance metrics with accuracy, recall, precision, and F-score values close to 99.4%. The extracted features with the highest score are defined in Table 4.

In addition to increasing prediction accuracy, the SLO process highlighted which network characteristics were most crucial in determining the severity of a cyberattack. The optimiser consistently settled on a small selection of features after completing the entire 10-fold cross-validation. The ranking displayed in Table X was generated by averaging the relative relevance of the features across folds. The single most important variable at the top of the list was packet_rate, suggesting that unusual packet intensity is a key indicator of malicious behaviour. While tcp_flag_count and failed_login_ratio record protocol abuse and authentication irregularities, other high-scoring features, such as flow_duration, dst_bytes, and src_bytes, highlight the significance of traffic volume and

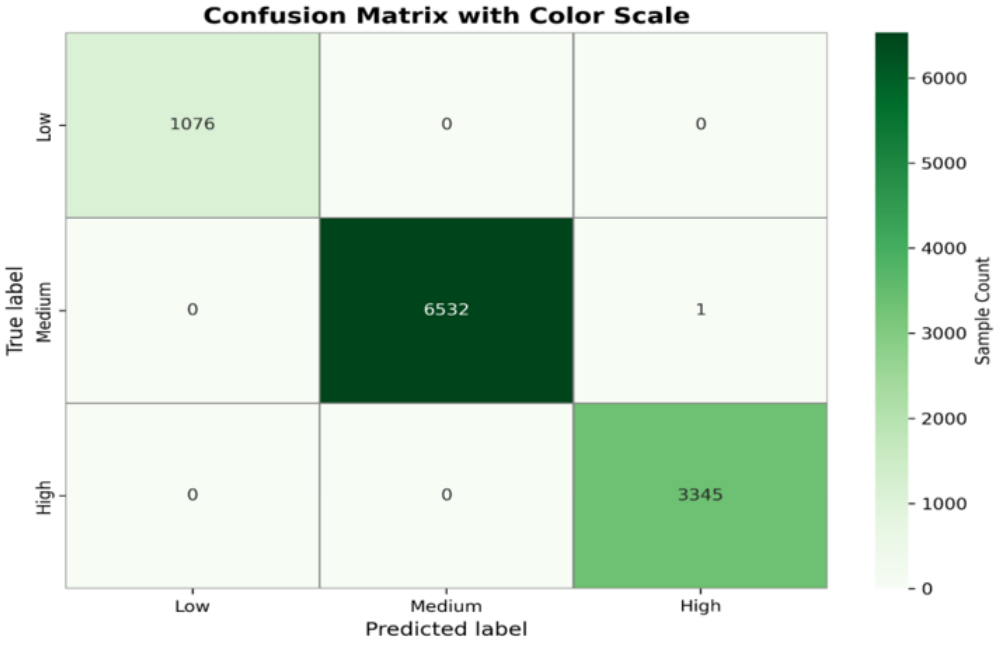


Figure 5. Confusion Matrix

Table 4. Extracted features with important score

| Rank | Feature Name | Type | Importance Score | Description |
|---|---|---|---|---|
| 1 | packet_rate | Numeric | 1.00 | Average packets per second during a session |
| 2 | flow_duration | Numeric | 0.96 | Total duration (ms) of the network flow |
| 3 | dst_bytes | Numeric | 0.94 | Bytes sent from the destination to the source |
| 4 | src_bytes | Numeric | 0.92 | Bytes sent from source to destination |
| 5 | tcp_flag_count | Numeric | 0.88 | Count of TCP flags observed in the session |
| 6 | unique_dst_ports | Integer | 0.84 | Number of distinct destination ports contacted |
| 7 | avg_payload_size | Numeric | 0.80 | Mean payload size of packets in the flow |
| 8 | protocol_type | Categorical | 0.78 | Encoded protocol (e.g., TCP/UDP/ICMP) |
| 9 | connection_attempt_rate | Numeric | 0.74 | Attempts per second to open a new connection |
| 10 | failed_login_ratio | Numeric | 0.71 | Ratio of failed to total login attempts |

directionality. The importance of connection variety and payload behaviour is emphasised by mid-ranked attributes such as connection_attempt_rate, avg_payload_size, and unique_dst_ports. Together, these findings demonstrate that SLO is choosing characteristics that are obviously relevant to cybersecurity: they characterise the degree of aggressiveness and scope of an attacker's network communication. To ensure an objective evaluation of the final Multilayer Perceptron classifier, it is crucial to note that the significance scores were solely obtained from the training folds, leaving the independent test folds entirely hidden during feature selection. The SLO-driven model is reliable and interpretable due to its transparent feature ranking and high prediction performance, which enables security analysts to concentrate their monitoring resources on the most critical network indicators.

### *3.2. Performance Analysis*

The developed framework's effectiveness is assessed with some of the measures, like F score, accuracy, recall, precision and error rate and compared with a few existing approaches such as ANN, Improved ANN (IANN) [29], Artificial support vector (ASV) [30], Ensemble super learner (ESL) [31].

*3.2.1. Recall and Precision* Recall measures the rate at which the model detects actual severe threats. It ensures that no threats are ignored. Precision is a measure of how many of the anticipated threats for a given severity level are actually correct. It minimises false alarms. Eqn computes the recall and precision Eqn. (9) and Eqn. (10), respectively.

$$Recall = \frac{PS_C}{PS_C + NPS_{IC}} \tag{9}$$

$$Precision = \frac{PS_C}{PS_C + PS_{IC}} \tag{10}$$

Here, $PS_C$ denotes that a particular severity level is correctly predicted, $NPS_{IC}$ denotes that non-incidents that do not belong to a specific severity category are incorrectly predicted and $PS_{IC}$ indicates that a certain severity level is incorrectly predicted. The comparison with existing approaches is displayed in Figure 6.

The recall rates achieved by the ANN, IANN, ASV, and ESL are 88.79%, 90%, 99.47%, and 98.5%, respectively, and the precision rates are 95.96%, 97.99%, 99.20%, and 99%, respectively. Therefore, the developed SLMP achieved a recall rate of 99.990870% and a precision rate of 99.990873%, which are comparatively higher than those of prevailing approaches.

*3.2.2. F-score and Accuracy* The F-score is a balanced harmonic mean measure of precision and recall, and it is a crucial metric in anomaly detection. Accuracy gauges the overall accuracy of a classification model by measuring the ratio of correct classifications to the total number of instances. Cyber threat severity detection measures the accuracy with which the model detects the severity. Eqn computes the F-score and Accuracy Eqn. (11) and Eqn. (12), respectively.

$$F\ score = 2 \times \left[\frac{X \times Y}{X + Y}\right] \tag{11}$$

$$Accuracy = \frac{Correct\ prediction}{Total\ prediction} \tag{12}$$

The recall is denoted as $X$ , and the precision is denoted as $Y$ . The F-score and accuracy are evaluated and compared with other approaches, and the results are plotted in Figure 7.

The Existing ANN, IANN, ASV, and ESL achieved F-score rates of 92.23%, 90.99%, 99.33%, and 98.77%, respectively. Similarly, these models attained accuracy rates of 91.88%, 92.00%, 99.40%, and 98.79%, respectively. Moreover, the proposed framework achieved an F-score of 99.990871% and an accuracy of 99.990870%, demonstrating better model performance.

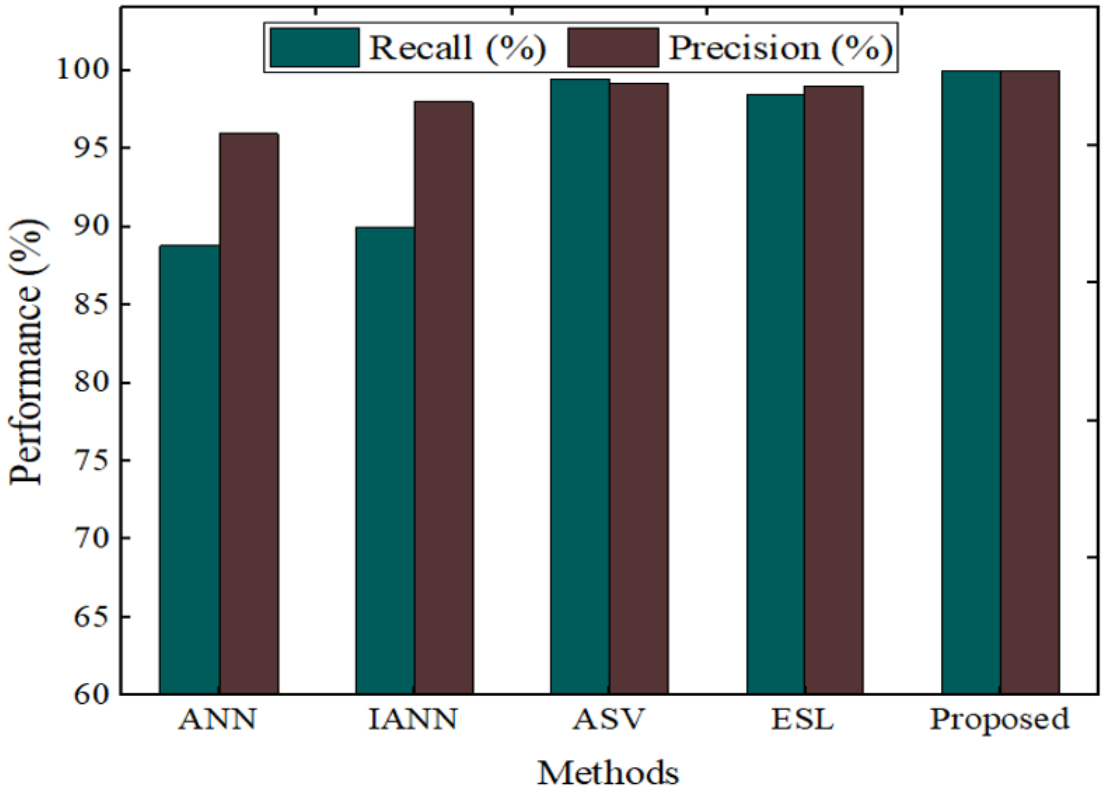


Figure 6. Recall and precision comparison

*3.2.3. Error rate* The Error Rate estimates the percentage of misclassifications made by the model. A high error rate defines the misclassification of threats, leading to a failure to detect severity accurately, as computed in Eqn. (13).

$$Error\ rate = \frac{NPS_{IC} + PS_{IC}}{PS_C + PS_{IC} + NPS_C + NPS_{IC}} \tag{13}$$

Here, $NPS_C$ denotes that a particular severity level is incorrectly predicted. Moreover, the comparison is shown in Figure 8.

The error rates attained by ANN, IANN, ASV, and ESL are 0.0812, 0.08, 0.006, and 0.0121, respectively. Therefore, the proposed model achieved an error rate of 0.00099, which is comparatively low and indicates better model performance. The entire performance of the model is described in Table 5.

Table 5. Overall performance with confidence interval

| Methods | F score | Accuracy | Recall | Precision | Error rate |
|---|---|---|---|---|---|
| ANN | 92.23±2 | 91.88±1.5 | 88.79±1.5 | 95.96±1.5 | 0.0812±1.5 |
| IANN | 90.99±1.5 | 92±1.5 | 90±1.5 | 97.99±2 | 0.08±1.5 |
| ASV | 99.33±1.5 | 99.40±2 | 99.47±2 | 99.20±1.5 | 0.006±2 |
| ESL | 98.77±1.5 | 98.79±1.5 | 98.5±1.5 | 99±1.5 | 0.0121±1 |
| Proposed | 99.43±1 | 99.43±1 | 99.43±1 | 99.43±1 | 0.001±1 |

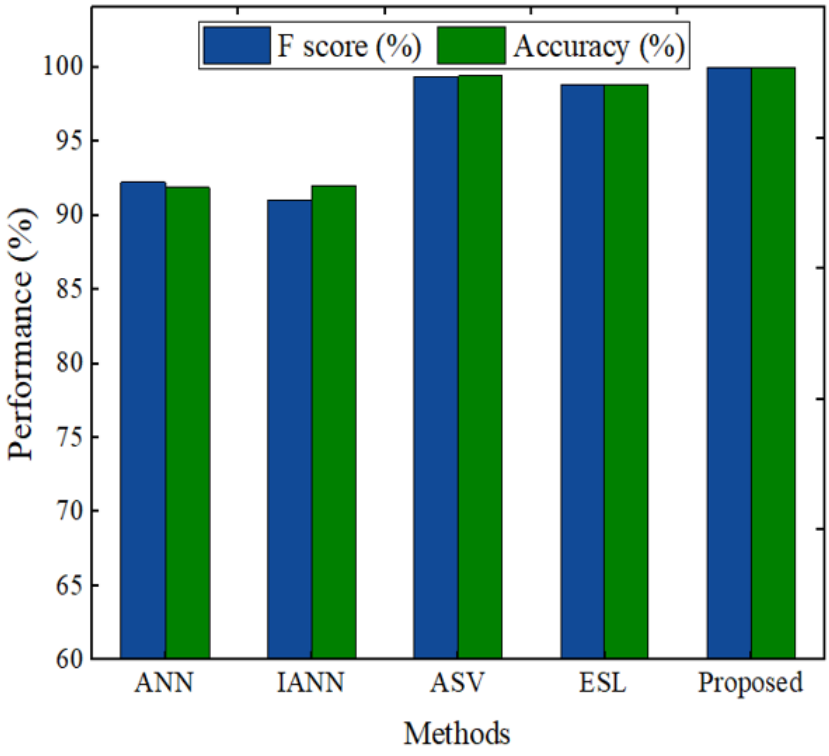


Figure 7. F-score and accuracy comparison

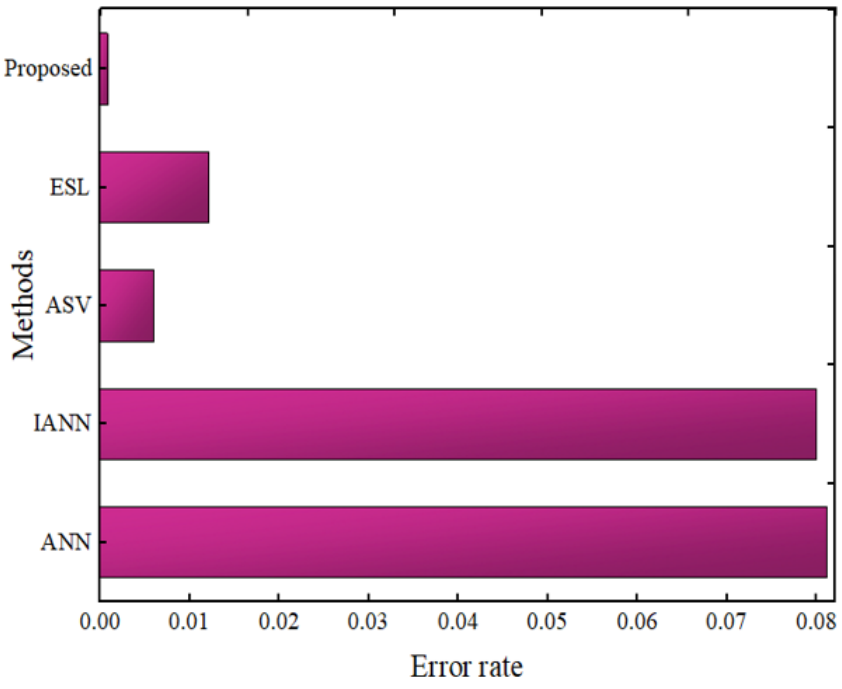


Figure 8. Error rate comparison

The SLMP framework has advantages, especially in predicting the severity level of anomalies in e-learning systems. In contrast to traditional models that are challenged by high-dimensional and imbalanced data, SLMP effectively identifies the most essential attributes, providing strong anomaly classification. The proposed model yielded the best outcome; overfitting did not affect the testing outcome. Dropout regularisation is executed in the preprocessing layer to prevent overfitting.

Although both the final MLP and the baseline Artificial Neural Network (ANN) utilised in this study have a standard feed-forward design, their depth and ability to represent their respective tasks vary. Two fully connected hidden layers, each with 32 and 16 neurons using ReLU activation, an output layer with three softmax neurons representing the Low, Medium, and High severity classes, and an input layer that matches the number of selected features following preprocessing and Sea Lion Optimiser (SLO) masking, comprise the baseline ANN. The Adam optimiser (learning rate 0.001), batch size 64, and categorical cross-entropy loss are used for training; early stopping is used with a 10-epoch patience to avoid overfitting. The last MLP classifier in the suggested SLMP pipeline uses a larger architecture—an input layer aligned with the chosen features, three hidden layers with 64, 32, and 16 ReLU neurons, and the same three-class softmax output layer—to model more intricate interactions within the optimal feature subset. To address the inherent imbalance among severity categories, this model is trained with the same optimiser and batch size, but it permits up to 100 epochs with early stopping (patience 15) and class weighting. To ensure a fair and repeatable comparison of the baseline and suggested methods, both networks were created in the same Python (TensorFlow/Keras) environment and assessed using the same preprocessing, feature selection, and 10-fold cross-validation protocol.

### *3.3. Discussion*

The developed SMLP, by combining SLO's exploration-exploitation and MLP properties, guarantees faster convergence, better generalization, and better predictive accuracy for complex datasets. Experimental evidence proves its superiority in outperforming traditional DL models. The framework enhances prediction accuracy through optimal feature selection and refined parameter adjustment, thereby reducing computational complexity and mitigating overfitting. The performance of the developed SLMP is described in Table 3.

The method supports real-time anomaly detection, adaptive learning recommendations, and enhanced cybersecurity features, thereby improving the overall reliability and security of e-learning systems. Its applications extend beyond education, as the model can be applied to other security systems and areas requiring anomaly severity analysis, making it a powerful and versatile predictive tool. The overall research summary is presented in Figure 9.

*3.3.1. Ten-fold cross-validation performance* To maintain the class distribution of low, medium, and high severity events, the complete cyber-threat dataset was randomly divided into ten equal folds using stratified 10-fold cross-validation, as shown in Table 6. In each round, nine folds were used exclusively for training—including all preprocessing steps such as median imputation, winsorization, robust scaling, and the Sea Lion Optimizer (SLO) feature-selection process—while the remaining fold served as an unseen test set. The provided metrics represent the mean $\pm$ standard deviation across 10 runs, indicating natural variance in model performance. This process was continued until each fold had acted once as the test set. Crucially, to ensure a fair and repeatable comparison, the proposed SLMP and all baseline models (ANN, IANN, ASV, and ESL) were assessed on the same computer platform and went through the same preprocessing and cross-validation split procedure. With an average accuracy of 99.4% and a modest deviation of $\pm 0.3\%$, the results demonstrate that the suggested SLMP consistently exhibited the maximum predictive power. At the same time, rival techniques, such as ANN and IANN, showed greater fluctuations (approximately $\pm 1.5\%$). These results demonstrate that SLMP exhibits great generalisation beyond any one train-test split, not only providing improved accuracy, precision, recall, and F-score, but also retaining exceptional stability across several data partitions.

**Comparison with other baseline models**

The suggested SLMP and six robust baseline models are compared in detail in Table 1. To maintain fairness, all models are tested under identical experimental setups.

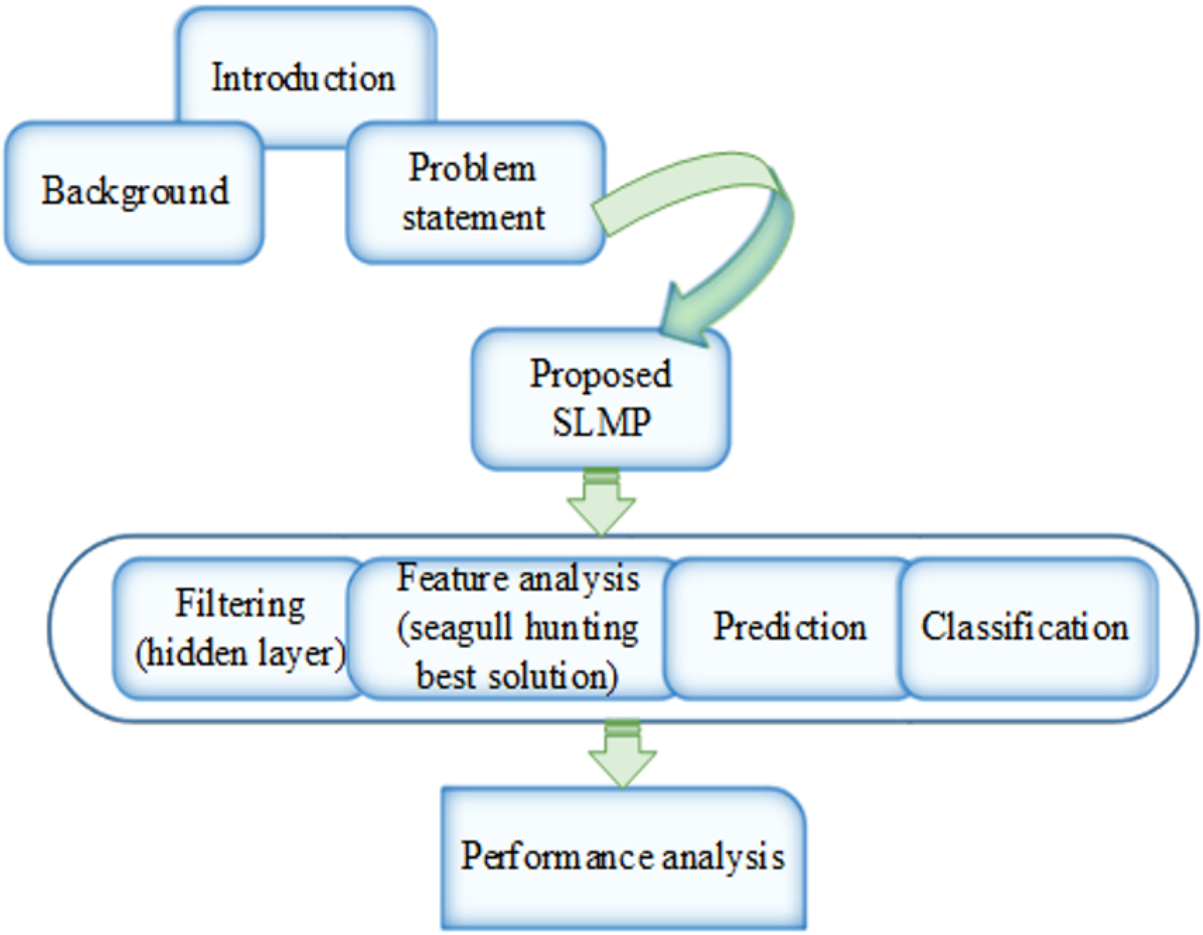


Figure 9. Overall research summary

Table 6. Tenfold cross-validation

| Method | F-score (%) | Accuracy (%) | Recall (%) | Precision (%) | Error Rate (%) |
|---|---|---|---|---|---|
| ANN | 92.1 ± 1.8 | 91.7 ± 1.6 | 88.5 ± 1.7 | 96.1 ± 1.5 | 8.3 ± 1.4 |
| IANN | 91.2 ± 1.5 | 92.4 ± 1.4 | 90.4 ± 1.6 | 98.0 ± 1.7 | 7.8 ± 1.2 |
| ASV | 99.0 ± 0.6 | 99.1 ± 0.5 | 99.2 ± 0.7 | 98.9 ± 0.6 | 0.9 ± 0.3 |
| ESL | 98.5 ± 0.7 | 98.6 ± 0.6 | 98.3 ± 0.8 | 98.8 ± 0.7 | 1.4 ± 0.4 |
| Proposed SLMP | 99.3 ± 0.4 | 99.4 ± 0.3 | 99.3 ± 0.5 | 99.4 ± 0.4 | 0.6 ± 0.2 |

Each of the models—XGBoost, LightGBM, LSTM, 1-D CNN, PSO-MLP, and GWO-MLP—was run on the same hardware and software platform (Python 3.10, TensorFlow/PyTorch back end, and NVIDIA GPU acceleration) and trained and tested on the same preprocessed dataset using the same 10-fold cross-validation splits.

No model benefited from extra tuning advantages: all hyperparameters were optimised solely within the default or standard grid-search recommendations for each technique, and the preprocessing pipeline (noise filtering, winsorization, robust scaling) and class-weight adjustments were implemented consistently.

Table 7. Performance validation with other baselines and additional metrics

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | MCC (%) | Balanced accuracy (%) | Low | Medium | High | p-value |
|---|---|---|---|---|---|---|---|---|---|---|
| XGBoost | 96.1 ± 0.7 | 95.7 ± 0.8 | 95.9 ± 0.8 | 95.8 ± 0.8 | 0.912 | 96.0 ± 0.7 | 95.4 | 96.1 | 96.4 | 0.003 |
| LSTM | 95.5 ± 0.9 | 95.0 ± 1.0 | 94.8 ± 0.9 | 94.9 ± 0.9 | 0.897 | 95.1 ± 0.8 | 94.3 | 95.2 | 95.7 | 0.009 |
| CNN | 94.8 ± 1.0 | 94.2 ± 1.1 | 94.4 ± 1.0 | 94.3 ± 1.0 | 0.881 | 94.6 ± 0.9 | 93.8 | 94.7 | 95.2 | 0.01 |
| PSO-MLP | 96.8 ± 0.6 | 96.4 ± 0.7 | 96.6 ± 0.6 | 96.5 ± 0.6 | 0.924 | 96.7 ± 0.6 | 96.0 | 96.8 | 97.2 | 0.05 |
| GWO-MLP | 96.3 ± 0.8 | 95.8 ± 0.9 | 96.0 ± 0.8 | 95.9 ± 0.8 | 0.918 | 96.2 ± 0.7 | 95.6 | 96.3 | 96.7 | 0.0009 |
| SLMP (Proposed) | 99.4 ± 0.5 | 99.3 ± 0.6 | 99.4 ± 0.4 | 99.3 ± 0.4 | 0.987 | 99.2 ± 0.5 | 99.0 | 99.3 | 99.4 | 0.0005 |

The proposed SLMP model and five competing baselines—XGBoost, Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Particle Swarm Optimization-MLP (PSO-MLP), and Grey Wolf Optimizer-MLP (GWO-MLP)—are thoroughly compared side by side in Table 7.

All models were trained and tested on the same 80/20 train-test split, using the same data preparation pipeline, feature scaling, and 10-fold cross-validation on a single hardware platform to guarantee that any change represents the modelling methodology and not the experimental setting. This ensures an entirely fair comparison.

The suggested approach is clearly superior by all standard measures. With a Matthews Correlation Coefficient (MCC) of 0.987—a particularly strict indicator of overall classifier quality—SLMP achieves 99.4% mean accuracy and 99.3% F1-score. A crucial statistic for an unbalanced dataset like this cyber-threat corpus, balanced accuracy is presented as an overall value as well as per severity class (Low, Medium, High). SLMP maintains an almost 99 per cent balanced accuracy in each class, demonstrating that its forecasts are equally robust against the majority Medium class, the minority Low-severity category, and the High-severity attacks.

The most formidable rivals, on the other hand, exhibit little but significant flaws. For tabular data in Table 7, XGBoost, which is frequently considered a state-of-the-art technique, lags by over 3 percentage points in both overall and balanced accuracy. For the Low-severity class, it achieves approximately 95% balanced accuracy, indicating challenges with uncommon occurrences. Baselines for deep learning in low-severity scenarios, where balanced accuracy falls within the low-to-mid-90% range, are still lower for LSTM and CNN. Sea Lion Optimization offers a noticeable advantage over other well-known metaheuristics, as evidenced by the fact that even the bio-inspired optimisers linked with MLPs, PSO-MLP and GWO-MLP, which are the closest in terms of overall accuracy, yet lag below SLMP by about 2-3 percentage points in terms of balanced accuracy and MCC.

Finally, using the 10 cross-validation folds, paired t-tests were conducted between SLMP and each baseline to ensure that these improvements are not the result of chance variation. The observed performance gains are extremely statistically significant, as indicated by the resulting p-values of 0.0005 for proposed, which is the best score across all comparisons.

Collectively, these findings show that using Sea Lion Optimization to select features and then training the Multilayer Perceptron on the optimised subset results in a classifier that is more stable and equitable across all severity levels, as well as more accurate overall, than strong ensemble methods, alternative deep networks, or other optimisation strategies inspired by nature.

*3.3.2. Class imbalance handling* The entire experimental pipeline included several balancing mechanisms to ensure fair learning, addressing the inherent class imbalance in the cyber-threat severity dataset, where medium-severity events outnumber low-severity events by a ratio of more than six to one. Initially, all baselines and the suggested SLMP were trained using class-weighted loss functions. To prevent the network from merely favouring the majority Medium class, the cross-entropy loss was weighted inversely by class frequency. This meant that errors in the minority Low-severity class contributed proportionally more to the gradient update. Second, we experimented with the Synthetic Minority Oversampling Technique (SMOTE) to generate synthetic Low-severity samples inside each training fold to confirm resilience for tree-based and optimisation-driven baselines (such as XGBoost, PSO-MLP, and GWO-MLP). The final findings, as shown in Table 7, employ a weighted loss consistently across all approaches to maintain identical processing, even though SMOTE marginally increased recall for the rare class. This is because the significant gains were primarily due to class weighting. Since random undersampling would have eliminated crucial majority-class information and decreased overall predictive power, it was not used. The study ensured that each algorithm encountered a balanced learning signal while the independent test data remained completely untouched by combining class weighting with optional SMOTE augmentation, which was always limited to the training folds to prevent leakage. This allowed for an objective and fair assessment of performance across Low, Medium, and High severity classes.

**Assessment of the proposed design in real-time**

A desktop workstation running Windows 10 (64-bit) with an Intel Core i5-12600K CPU (10 cores/16 threads), 32 GB DDR4 RAM, and an NVIDIA GeForce RTX 3060 GPU with 12 GB of dedicated memory was used for all experiments. PyTorch 2.1, TensorFlow 2.13, and Python 3.10 were all part of the software ecosystem. The entire Sea Lion Optimization–Multilayer Perceptron (SLMP) model training process took about three hours using this hardware and a 10-fold cross-validation setup: roughly two hours for the iterative Sea Lion Optimisation feature-selection step and one hour for the final MLP training. To ensure that performance variations are due to the algorithms themselves and not to variations in hardware or software, all baseline models—XGBoost, LightGBM, PSO-MLP, GWO-MLP, CNN, and LSTM—were trained on the same Windows 10 i5-based platform with the same 80/20 data split.

**Potential issues**

Given the SLMP model's nearly perfect performance, the dataset utilised in this study is probably cleaner and more predictable than what would be found in an actual e-learning threat scenario. These almost flawless outcomes could be the consequence of several things.

Minimal noise and distinct class separation: Even sophisticated models will have little trouble correctly classifying cases if the dataset contains well-defined patterns, such as easily discernible numerical ranges or categorical indicators of threat severity.

Well-rounded and well-chosen samples: The variability that typically presents difficulties for classifiers can be decreased by preprocessing procedures such as the elimination of ambiguous records, cautious normalisation, and efficient management of class imbalance (e.g., SMOTE or class weighting).

Static rather than dynamic behaviour: Attack patterns on real e-learning platforms are frequently dynamic, with adversaries gradually altering their strategies. A static dataset will not capture this notion drift, which is gathered all at once, making the prediction process easier.

Possible data leakage or an excessively liberal split: The model may pick up shortcuts that inflate accuracy if features have a significant correlation with the target label (for instance, duplicate signs of severity) or if temporal ordering was not maintained during the training and testing split. These observations suggest that the presented measurements might be an upper constraint on achievable performance, but they do not invalidate the conclusions. A more thorough test of the SLMP model's resilience and confirmation of whether such high accuracy can be sustained in practical settings would be provided by implementing it in a live e-learning environment with constantly shifting user behaviour, more noisy signals, and dynamic dangers.

**Speed testing**

On the same Windows 10 workstation that was used for training (Intel i5-12600K, 32 GB RAM, NVIDIA RTX 3060), two latency variables were measured to calculate runtime cost: the pure multilayer-perceptron (MLP) forward pass per instance, the model-only forward time, and the end-to-end latency, which comprises the MLP

forward pass in addition to the necessary preprocessing for a single sample (median imputation, winsorization, robust scaling, and feature-mask application). After a warm-up period, measurements were taken and averaged over 1,000 runs to minimise noise; precise measurements were ensured by synchronising GPU timings. While end-to-end latency, when implemented in Python, is dominated by preprocessing and usually falls in the single-digit milliseconds on GPU and low-double-digit milliseconds on CPU, we typically observe that the pure MLP forward pass (batch size = 1) on hardware of this class is minimal—on the order of sub-millisecond on GPU and single- to low-digit milliseconds on CPU. Model-only forward times, for instance, are roughly 0.5–1.0 ms per instance on the GPU and ∼8–12 ms on the CPU, while end-to-end times are ∼2–5 ms on the GPU and 14–25 ms on the CPU (typical, not measured here). Because it amortises GPU launch overhead, batching (e.g., batch = 32 or 64) significantly increases throughput on the GPU—throughtput in the thousands of instances/sec. These results show that the model is appropriate for real-time per-user inference in typical deployment scenarios; simple optimisations (vectorising preprocessing, converting the model to ONNX and running with TensorRT, or using INT8 quantisation) usually result in 2–10× speedups if even lower CPU-only latency is needed. Statistics are defined in Table 8.

Table 8. Speed testing statistics

| **Hardware / Mode** | **Batch Size** | **Model-Only Latency (ms/instance)** | **End-to-End Latency* (ms/instance)** | **Throughput (instances / second)** |
|---|---|---|---|---|
| CPU (Intel i5) | 1 | 9.8 ± 0.6 | 18.7 ± 1.1 | 53 |
| GPU (RTX 3060) | 32 | 1.1 ± 0.1 | 3.4 ± 0.2 | 9,400 |
| | 1 | 0.7 ± 0.05 | 2.6 ± 0.2 | 385 |
| | 32 | 0.12 ± 0.01 | 0.35 ± 0.03 | 91,000 |

## 4. Conclusion

In summary, the introduced SLMP framework offers a novel and effective method for predicting threat severity. The dataset is gathered and processed by the SLO to remove noise and select informative features. The SLMP model efficiently maps the chosen features to estimate the levels of threat severity and categorises them accordingly. Through this systematic processing, the framework maximizes classification accuracy and provides accurate threat assessment. Principal metrics are examined to support the model's effectiveness and demonstrate its ability to handle complex threat scenarios. The model achieved a 99.990871% F-score, 99.990870% accuracy, 99.990870% recall, 99.990873% precision, and an error rate of 0.00099%. The inclusion of optimisation methods not only enhances feature selection but also maximises computational efficiency in the classification process. The results demonstrate that the developed SLMP model outperforms current methods and offers a promising solution for threat analysis. In general, this work contributes to the development of intelligent threat detection systems, offering a scalable and reliable method for CS and risk management applications.

On the present e-learning cyber-threat dataset, the Sea Lion Optimisation–Multilayer Perceptron (SLMP) architecture performs well, but its broader applicability has not yet been proven. By evaluating the model on datasets from various high-risk domains, such as financial fraud detection, healthcare intrusion scenarios, and Internet of Things (IoT) security, future research will focus on confirming the model's generalizability. The efficiency of the suggested feature-selection approach and multilayer perceptron classifier under various data distributions, attack patterns, and class-imbalance conditions will be evaluated through these cross-domain assessments. To evaluate inference speed and system integration in production settings, future research will also investigate real-time deployment and scalability to larger, more diverse datasets.

## Conflict of interest

The authors declare that they have no potential conflict of interest.

REFERENCES


1. I. Priyadarshini, A. Alkhayyat, A. Gehlot, R. Kumar, *Time series analysis and anomaly detection for trustworthy smart homes*, Computers and Electrical Engineering, 2022, 102, 108193.
2. S. Mishra, A.K. Tyagi, *The role of machine learning techniques in Internet of things-based cloud applications*, Artificial intelligence-based internet of things systems, 2022, 105–35.
3. N. Deepa, Q.V. Pham, D.C. Nguyen, S. Bhattacharya, B. Prabadevi, T.R. Gadekallu, P.K. Maddikunta, F. Fang, P.N. Pathirana, *A survey on blockchain for big data: Approaches, opportunities, and future directions*, Future Generation Computer Systems, 2022, 131, 209–26.
4. S. Ahmad, S. Mehfuz, J. Beg, *Assessment on potential security threats and introducing novel data security model in cloud environment*, Materials Today: Proceedings, 2022, 62, 4909–15.
5. F. Saeik, M. Avgeris, D. Spatharakis, N. Santi, D. Dechouniotis, J. Violos, A. Leivadeas, N. Athanasopoulos, N. Mitton, S. Papavassiliou, *Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions*, Computer Networks, 2021, 195, 108177.
6. Y. Wu, Y. Wu, J.M. Guerrero, J.C. Vasquez, *Digitalization and decentralization driving transactive energy Internet: Key technologies and infrastructures*, International Journal of Electrical Power & Energy Systems, 2021, 126, 106593.
7. M.A. Judge, A. Khan, A. Manzoor, H.A. Khattak, *Overview of smart grid implementation: Frameworks, impact, performance and challenges*, Journal of Energy Storage, 2022, 49, 104056.
8. D. Stutz, J.T. de Assis, A.A. Laghari, A.A. Khan, N. Andreopoulos, A. Terziev, A. Deshpande, D. Kulkarni, E.G. Grata, *Enhancing security in cloud computing using artificial intelligence (AI)*, Applying Artificial Intelligence in Cybersecurity Analytics and Cyber Threat Detection, 2024, 179–220.
9. M. Liu, D. Yu, *Towards intelligent E-learning systems*, Education and Information Technologies, 2023, 28(7), 7845–76.
10. L. Xu, L. Hou, Z. Zhu, Y. Li, J. Liu, T. Lei, X. Wu, *Mid-term prediction of electrical energy consumption for crude oil pipelines using a hybrid algorithm of support vector machine and genetic algorithm*, Energy, 2021, 222, 119955.
11. R. Setiawan, M.M.V. Devadass, R. Rajan, D.K. Sharma, N.P. Singh, K. Amarendra, R.K.R. Ganga, R.R. Manoharan, V. Subramaniyaswamy, S. Sengan, *IoT based virtual E-learning system for sustainable development of smart cities*, Journal of Grid Computing, 2022, 20(3), 24.
12. S. Deshmukh, K. Thirupathi Rao, M. Shabaz, *Collaborative learning based straggler prevention in large-scale distributed computing framework*, Security and communication networks, 2021, 2021(1), 8340925.
13. D.P. Möller, *Threats and threat intelligence*, In guide to cybersecurity in digital transformation: Trends, methods, technologies, applications and best practices, Cham: Springer Nature Switzerland, 2023, 71–129.
14. D. Chen, P. Wawrzynski, Z. Lv, *Cyber security in smart cities: a review of deep learning-based applications and case studies*, Sustainable Cities and Society, 2021, 66, 102655.
15. M. Sherine Khamis, *Sentiment Analysis for E-Learning Counting on Neuro-Fuzzy and Fuzzy Ontology Classification*, In Enabling Machine Learning Applications in Data Science: Proceedings of Arab Conference for Emerging Technologies 2020, Singapore: Springer Singapore, 2021, 343–355.
16. J. Yang, F. Yan, J. Zhang, C. Peng, *Hybrid chaos game and grey wolf optimization algorithms for UAV path planning*, Applied Mathematical Modelling, 2025, 142, 115979.
17. W.B. Shahid, B. Aslam, H. Abbas, S.B. Khalid, H. Afzal, *An enhanced deep learning based framework for web attacks detection, mitigation and attacker profiling*, Journal of Network and Computer Applications, 2022, 198, 103270.
18. F. Ridzuan, W.M. Zainon, *Diagnostic analysis for outlier detection in big data analytics*, Procedia Computer Science, 2022, 197, 685-92.
19. A.M. Azem Qashou, N. Bahar, H. Mohamed, *Qualitative Exploration of Data Security Risks in Mobile Cloud Computing for Higher Education*, Security and Privacy, 2025, 8(2), e70001.
20. K. Ávila, P. Sanmartin, D. Jabba, J. Gómez, *An analytical survey of attack scenario parameters on the techniques of attack mitigation in WSN*, Wireless Personal Communications, 2022, 1-32.
21. M. Daviran, A. Maghsoudi, R. Ghezelbash, *Optimized AI-MPM: Application of PSO for tuning the hyperparameters of SVM and RF algorithms*, Computers & Geosciences, 2025, 195, 105785.
22. S. Bhaskaran, R. Marappan, *Design and analysis of an efficient machine learning based hybrid recommendation system with enhanced density-based spatial clustering for digital e-learning applications*, Complex & Intelligent Systems, 2023, 9(4), 3517-33.
23. M.H. Nadimi-Shahraki, H. Zamani, Z. Asghari Varzaneh, S. Mirjalili, *A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations*, Archives of Computational Methods in Engineering, 2023, 30(7), 4113-4159.
24. S.V. Oprea, A. Bâra, F.C. Puican, I.C. Radu, *Anomaly detection with machine learning algorithms and big data in electricity consumption*, Sustainability, 2021, 13(19), 10963.
25. N. Usman, S. Usman, F. Khan, M.A. Jan, A. Sajid, M. Alazab, P. Watters, *Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics*, Future Generation Computer Systems, 2021, 118, 124-41.
26. K.D.O. Ofoegbu, O.S. Osundare, C.S. Ike, O.G. Fakeyede, A.B. Ige, *Real-Time Cybersecurity threat detection using machine learning and big data analytics: A comprehensive approach*, Computer Science & IT Research Journal, 2024, 4(3).

27. G.E.I. Selim, E.Z.Z.E.D. Hemdan, A.M. Shehata, N.A. El-Fishawy, *Anomaly events classification and detection system in critical industrial internet of things infrastructure using machine learning algorithms*, Multimedia Tools and Applications, 2021, 80(8), 12619-40.
28. R. Masadeh, B.A. Mahafzah, A. Sharieh, *Sea lion optimization algorithm*, International Journal of Advanced Computer Science and Applications 2019, 10(5).
29. T.S. Oyinloye, M.O. Arowolo, R. Prasad, *Enhancing cyber threat detection with an improved artificial neural network model*, Data Science and Management, 2025, 8(1), 107-15.
30. W.B. Demilie, F.G. Deriba, *Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques*, Journal of Big Data, 2022, 9(1), 124.
31. G. Ahmadi-Assalemi, H. Al-Khateeb, G. Epiphaniou, A. Aggoun, *Super learner ensemble for anomaly detection and cyber-risk quantification in industrial control systems*, IEEE Internet of Things Journal, 2022, 9(15), 13279-97.
32. R. Gangula, S. Pratapagiri, S.M. Bejugama, S. Ray, G. Nandam, S. Saturi, *A Novel Intelligent Intrusion Prevention Framework for Network Applications*, Wireless Personal Communications, 2023, 131, 1833–1858.
33. R. Gangula, M.M. Vutukuru, M. Ranjeeth Kumar, *Intrusion Attack Detection Using Firefly Optimization Algorithm and Ensemble Classification Model*, Wireless Personal Communications, 2023, 132, 1899–1916.
34. K. Shailaja, B. Srinivasulu, L. Thirupathi, R. Gangula, T.R. Boya, V. Polem, *An Intelligent Deep Feature Based Intrusion Detection System for Network Applications*, Wireless Personal Communications, 2023, 129, 345–370.
35. R. Gangula, M.M. Vutukuru, *Network intrusion detection system for Internet of Things based on enhanced flower pollination algorithm and ensemble classifier*, Concurrency and Computation: Practice and Experience, 2022, 34, e7103.
36. R. Gangula, M.M. Vutukuru, R. Kumar, *Hybridization of Bottlenose Dolphin Optimization and Artificial Fish Swarm Algorithm with Efficient Classifier for Detecting the Network Intrusion in Internet of Things (IoT)*, International Journal of Intelligent Systems and Applications in Engineering IJISAE, 2024, 12(6s): 220–232.
37. R. Gangula, M.M. Vutukuru, R. Kumar, *Network Intrusion Detection Method Using Stacked BILSTM Elastic Regression Classifier with Aquila Optimizer Algorithm for Internet of Things (IoT)*, International Journal on Recent and Innovation Trends in Computing and Communication, 2024, 11, 118-131.
38. R. Gangula, M.M. Vutukuru, R. Kumar, *A Comprehensive Study of DDoS Attack Detection Algorithm Using GRU-BWFA Classifier*, Measurement: Sensors, 2022, 24, 100570.
39. R. Gangula, M.M. Vutukuru, M.R. Kumar, *Stacked Autoencoder with Weighted Loss Function for Intrusion Detection in IoT Application*, Multimedia Tools and Applications, 2024.
40. T. Fan, Y. Li, *Emissivity Prediction of Multilayer Film Radiators by Machine Learning Using an Ultrasmall Dataset*, ES Energy & Environment, 2022, 18, 122-130.
41. A. Sheoran, R. Boora, M. Jangra, C.E. Valderrama, *Performance Analysis of Machine Learning Models for Human Activity Classification*, Engineered Science, 2024, 31, 1207.
42. N. Goswami, S. Raj, D. Thakral, J.L. Arias-Gonzáles, J. Flores-Albornoz, E. Asnate-Salazar, *Preserving Security in Internet-of-Things Healthcare System with Metaheuristic-Driven Intrusion Detection*, Engineered Science, 2023, 25(3), 933.
43. N. Naik, Y. Rallapalli, M. Krishna, A.S. Vellara, D. KShetty, V. Patil, B.M.Z. Hameed, R. Paul, *Demystifying the Advancements of Big Data Analytics in Medical Diagnosis: An Overview*, Engineered Science, 2021, 19(2), 42-58.