AlertFusion-OptiNet: An Advanced SIEM Alert Management System for IoT Environments using CMRO and AlertQ-Net

Abdullah Alenizi[1,*]

[1]Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Al-Majmaah 11952, Saudi Arabia

Email: aalenizi@mu.edu.sa

Abstract

SIEM, which stands for Security Information and Event Management, is a collection of services and solutions that give businesses the capacity to gather, examine, and handle security-related data in real time from all areas of their IT infrastructure. This study presents AlertFusion-OptiNet, a sophisticated SIEM alert management architecture intended for effective alert handling and intrusion detection. The proposed CMRO algorithm (a hybrid of Coot Bird Optimization and Mug Ring Algorithm) is used to select the best features after the system integrates data from multiple sources (raw logs, network traffic, and security alerts), applies preprocessing to eliminate redundancy and inconsistencies, and extracts features using techniques like LDA, GloVe, statistical analysis, and DWT. PCA is then used to reduce dimensionality. The shortcomings of current intrusion detection systems include delayed alert replies, poor feature selection, and ineffective management of heterogeneous datasets. Two-channel CNNs, LSTM, and Bi-RNNs are used in AlertFusion-OptiNet's hybrid detection model to improve accuracy and real-time detection, while AlertQ-Net uses reinforcement learning to handle and monitor alerts continuously. The proposed AlertFusion-OptiNet accomplished 99.43% and outruns SOTA models.

Keyword: Security Information and Event Management; Intrusion Detection; Deep Learning; Reinforcement Learning; Alert Management; Hybrid Optimization

1. Introduction

In a time when information systems are widely used in many different industries, cyber risks to businesses are becoming more and more frequent. Improving current detection methods has become essential due to the ever-changing nature of attacks (Farrel et al, 2024). SIEM systems are essential tools for monitoring, analyzing, and managing security incidents across an organization's network. They collect and aggregate data from various sources like firewalls, Intrusion Detection Systems (IDS), and application logs to provide a centralized view of security activities (Uccello et al, 2024). To enhance intrusion detection capabilities, SIEMs are now integrating Machine Learning (ML) and DL algorithms. These algorithms are highly effective in identifying complex patterns of malicious behavior, automating anomaly detection, and reducing false positives (Thepa et al, 2024). By leveraging historical data and real-time inputs, ML/DL models can detect sophisticated threats that traditional rule-based systems might miss (Muneer et al, 2024). The use of ML and DL in SIEM for intrusion detection offers several significant advantages. One of the most important is improved detection accuracy (Tendikov et al, 2024). Normally, ML models learn from historical attack patterns and apply this knowledge to identify new, evolving threats (Tashfeen, 2024).

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

Automation and real-time analysis are key strengths as DL models can process large volumes of data faster than humans, identifying potential breaches in real-time without human intervention (Singh et al, 2024). Anomaly detection capabilities are also greatly enhanced, as unsupervised learning models can flag unusual behaviors or patterns that fall outside normal operational baselines. Furthermore, ML/DL models have the potential to adapt and evolve, learning from new data to continually improve detection performance (Shaik & Shaik 2024). Despite the benefits, there are several challenges in applying ML and DL to SIEM systems. A major issue is the need for high-quality training data. ML/DL models require vast amounts of labeled data to function effectively, but obtaining such datasets in cybersecurity can be difficult due to privacy concerns and limited access to real-world attack data (Sania et al, 2024). Another challenge is handling false positives, while ML can reduce these over time; initial models still produce many false alarms, leading to alert fatigue for security teams. Model complexity and resource demands are also barriers (Muhammad et al, 2023).

DL models, in particular, require substantial computational resources and expertise to train and maintain, which may not be feasible for all organizations. Several limitations exist in current implementations of ML and DL for intrusion detection in SIEM systems (Moukafih et al, 2020). One limitation is the black-box nature of DL models, where the reasoning behind an alert or detection is not easily interpretable by security analysts (Azmi et al, 2021). This lack of transparency makes it difficult to trust the model's output fully. Additionally, adversarial attacks pose a risk, where attackers intentionally manipulate data to deceive ML models (Pulyala, 2023). Another limitation is the lack of contextual awareness, while ML models detect anomalies; they often struggle to incorporate broader contextual information that distinguishes between false positives and true threats. However, the integration of threat intelligence feeds with ML models improves detection accuracy by providing real-time context (Esseghir et al, 2022). Furthermore, the combination of ML/DL with behavioral analytics and context-aware AI enhances precision in detecting insider threats and Advanced Persistent Threats (APTs), making SIEM systems even more robust and reliable (Kothandaraman et al, 2023). Traditional SIEM systems are ineffective in managing real-time alerts and identifying complicated intrusions due to the growing complexity and number of cyberattacks. These algorithms frequently have trouble processing various, high-dimensional datasets, selecting features, and responding quickly, which results in less-than-ideal threat identification. To overcome these obstacles, AlertFusion-OptiNet combines sophisticated preprocessing, the CMRO method for optimal feature selection, and hybrid deep learning models for precise and effective intrusion detection. AlertQ-Net incorporates reinforcement learning to provide continuous and adaptive alert monitoring and prioritizing. The necessity to provide a scalable, precise, and real-time SIEM solution to meet the needs of contemporary cybersecurity is inspired by this architecture. The AlertFusion-OptiNet model effectively integrates various data sources, detects intrusions in real time, and optimizes alert management using cutting-edge AI approaches to improve security in complex and dynamic real-world network environments. Its flexibility guarantees a quick and precise reaction to threats in a variety of changing network topologies, including those based on the Internet of Things. With these advancements, this paper utilizes an ensemble of DL models for intrusion detection and RL for alert management systems. The key contributions are

- The study suggests AlertFusion-OptiNet, a novel multi-stage framework that combines cutting-edge modules for intrusion detection, dimensionality reduction, feature selection, and data preprocessing to effectively handle massive security data.
- The most pertinent features from multi-source datasets are chosen using a new CMRO technique that combines the Mug Ring technique (MRA) and Coot Bird Optimization (CBO), improving detection accuracy and lowering processing overhead.
- The intrusion detection system greatly increases detection accuracy and robustness by integrating Bi-RNNs for anomaly detection, LSTM for temporal sequence analysis, and Two-Channel CNNs for pattern recognition.
- The study presents AlertQ-Net, a reinforcement learning-based method for continuous monitoring and real-time warning prioritizing that makes it possible to manage security alerts effectively and adaptable.

This article is structured as a recent literature on intrusion detection in Section II. Section III explains the proposed architecture. Experimentation and results are given in Section IV. Section V concludes the research.

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

2. Literature Review

In 2024, Sheeraz et al. addressed an attack detection mechanism using an Optimised Correlator (OC) to enhance the performance of the SIEM system. A revolutionary correlation engine substituted the conventional regex matching sub-module with a novel high-performance multiple regex matching library dubbed Hyperscan for simultaneous log data scanning. In 2020, Moukafih et al. established a majority system based on a reliability method that combined weak learners, such as basic Feedforward Neural Networks (FFNNs) to provide great detection capabilities while requiring little computational power. The outcomes of the experiments demonstrated that the model performs better than intricate, resource-intensive DL models. In 2020, Al-Duwairi et al. suggested an Internet of Things (IoT) botnet Distributed Denial of Service (DDoS) attack detection and mitigation system based on SIEM. By keeping an eye on particular packet types coming from compromised IoT devices, this system was able to identify and stop DDoS attack activity from these devices.

In 2024, Amru et al. explained an ensemble model for smart homes to detect intrusions. Using eXtreme gradient boosting (XGBoosting), the categorization challenge of attacks was regarded as a predictive modeling problem. Using an ensemble technique, models were added one after the other to rectify faults until no further gains in performance were possible. In 2024, Ahmed offered an IDS in Wireless Sensor Networks (WSNs) using a Support Vector Machine (SVM) in conjunction with Stochastic Gradient Descent (SGD). To enhance the performance of recommendation systems, the research also suggested integrating context knowledge, which considered user preferences as well as system attributes or circumstances. In 2023, Ban et al suggested an Instance-Weighted SVM (IWSVM) that makes use of cutting-edge data visualization and ML techniques such as an event-segmenting algorithm and a cost-sensitive learning method for breaking alert fatigue. This was attained by filtering and correlating alerts and by speeding up the triage process. In 2024, Sharma et al provided a DL model such as CNN, and Deep Neural Network (DNN) to classify different types of attacks in the dataset for intrusion detection. To reduce the number of features and select the most crucial elements, 2 distinct DL models were developed. In 2024, Turukmane & Devendiran presented a Mud Ring-assisted multilayer SVM (M-MultiSVM) for automated intrusion detection. Pre-processing of the obtained data was done followed by the Advanced Synthetic Minority Oversampling Technique (ASmoT) to lessen the issue of class imbalance. The M-MultiSVM classified the various attack types and MRA adjusted the hyperparameters.

In 2023, Hromada et al. introduced the basic security and safety concepts along with the necessity of their convergence. For a better understanding of this, the Converged Resilience Assessment (CRA) technique was a mathematical model. Security Information and Event Management (SIEM) and Physical Security Information Management (PSIM) systems are then explained as technological ideas that may be applied to resilience evaluation. In 2024, Ghadermazi et al. suggested a novel system that considers these things and makes use of mathematical optimization techniques and machine learning to dynamically increase throughput throughout work shifts. The framework assigns analysts to alerts with matching attributes, creates clusters of related alerts, and automates the identification and elimination of some benign signals to achieve efficiency.

In 2023, Zahid et al. employed a module to intercept network traffic from IoT devices and forward it to a machine-learning model for identification and forecasting. The ML model is sent to a server for decoding via a Wazuh agent after being embedded in JSON log format. The feature set for event monitoring is established after an analysis of industrial protocols. At the Wazuh end, dynamic and custom rules are created to produce warnings for abnormalities found by the protocols or ML model. In 2022, Coppolino et al. suggested SIEM tool specifically improves the technology's state of the art in two areas: integrity and privacy. Two of the most promising technologies for trustworthy computing are used in conjunction to make the breakthroughs, specifically: Reliable Execution Setting Homomorphic Encryption (HE) and TTE. A real-world use case of a smart hospital—one with high IT usage and demanding security requirements was utilized to validate this model.

3. Problem Statement

Normally, intrusion detection in SIEM systems faces several challenges, particularly in handling large volumes of data generated by modern IT infrastructures. On the other hand, DL models, while powerful, are computationally intensive, leading to scalability issues and increased processing times in real-time systems. In addition, the extraction of features increases complexity, making the system more prone to errors if feature selection or preprocessing is not optimally managed. Ensuring that the system adapts to diverse environments and evolving threats without suffering from high false positive or false-negative rates is another critical challenge. Several optimization techniques, though designed to enhance the performance of SIEM systems, often struggle with real-world deployment due to the dynamic

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

nature of cyber threats. They overfit specific threat patterns, reducing generalization to unseen attacks. Additionally, maintaining the balance between detection accuracy and resource efficiency is difficult. Furthermore, the limited availability of labeled attack data for training DL models restricts the effectiveness of the systems, making them less adaptable to novel and advanced threats. For this reason, this paper implements an advanced alert management system using SIEMs with ensembled DL approaches. Table I summarizes the aims and limitations of previous research for intrusion detection using different methods.

Table 1: aims and limitations of previous research for intrusion detection using different methods

Authors/Year	Methods	Aim	Advantages	Limitations
Sheeraz et al in 2024	OC	To enhance SIEM system performance	21 and 2.5 times faster and more efficient than a Simple Event Correlator (SEC)	Need to include more attacks on varied datasets
Moukafih et al in 2020	FFNNs	To offer accuracy with computational efficacy	89% accuracy	Further improvement was needed concerning the accuracy
Al-Duwairi et al in 2020	SIEM-based IoT Botnets	To efficiently identify DDoS attacks	Achieved considerable accuracy	Need to focus on varied data features
Ahmed in 2024	SVM-SGD	To enhance IDS in WSNs	96% accuracy	Only a few features were considered for experimentation
Amru et al in 2024	XGBoosting	To identify intrusions into smart homes	94% precision	Exposed high computational complexity
Ban et al in 2023	IWSVM	To detect critical security incidents	99% accuracy	Need to reduce misclassifications
Sharma et al in 2024	CNN-DNN	To develop an IDS for IoT	Considerable accuracy was achieved	Revealed complex computation
Turukmane & Devendiran in 2024	M-MultiSVM	To present an IDS using ML	99% and 97% accuracy for CSE-CIC-IDS 2018 and UNSW-NB15 datasets	Failed to combine the varied data from different datasets

4. A Novel Intrusion Detection Model in SIEMS

A. Proposed Architecture

Fig. 1 demonstrates the overview of the proposed AlertFusion-OptiNet. This research introduces AlertFusion-OptiNet, a novel framework for efficient alert management and intrusion detection. The framework consists of several key modules, including data gathering and preprocessing, multi-feature extraction and fusion, dimensionality reduction, feature selection, intrusion detection, and alert management. At first, data are collected from multiple sources (i.e., 3 different datasets) including raw logs, network traffic, and security alerts. Cleansing the gathered data eliminates redundancy and missing values via MI, and log normalization ensures uniformity. Further, significant features are

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

extracted using LDA, and GloVe for dataset 1, statistical features for dataset 2, and DWT for dataset 3, and are fused based on weighted importance. Dimensionality reduction is achieved via PCA, while the best features are selected using the proposed CMRO algorithm, which hybridizes CBO and MRA. Intrusions are detected through AlertFusion-OptiNet, a hybrid model combining 2-channel CNNs for pattern recognition, LSTM for temporal sequence analysis, and Bi-RNNs for anomaly detection. Finally, alert management is handled via AlertQ-Net, an RL approach for continuous monitoring.

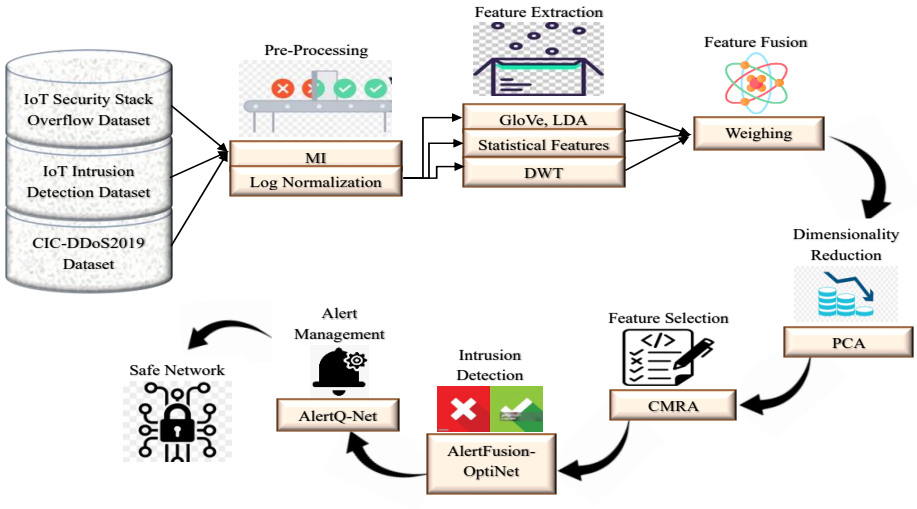


Figure 1. Overview of Proposed AlertFusion-OptiNet Model

B. Data Collection

Data Collection involves gathering raw logs, network traffic, and security alerts from multiple sources, including firewalls, Intrusion Detection/Prevention Systems (IDS/IPS), and endpoint devices. These data sources provide crucial information on network activities, security events, and system performance. This diverse input enables a comprehensive view of potential threats and suspicious behavior across the network, forming the foundation for accurate intrusion detection and alert management. The datasets used are IoT Security Stack Overflow, IoT Intrusion Detection, and CIC-DDoS2019 which is accessible through https://www.kaggle.com/datasets/shibli007/iot-security-stack-overflow-dataset, https://www.kaggle.com/datasets/subhajournal/iotintrusion, https://www.kaggle.com/datasets/aymenabb/ddos-evaluation-dataset-cic-ddos2019, and https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15 respectively.

C. Pre-Processing

The role of preprocessing is to prepare raw data for analysis by cleaning, normalizing, and structuring it, ensuring consistency and accuracy. This step enhances the model's ability to extract relevant features, improving the overall performance of IDS and alert management systems.

1) MI: It is a simple technique used to replace missing values in a dataset with the mean of the available values in the same feature (column) (Joel et al, 2024). This helps to preserve the dataset size while handling missing data as given in Eq. (1), in which x_μ refers to the imputed value, x_i means to existing non-missing values in the feature, and $\aleph$ stands for non-missing values count.

$$x_\mu = \frac{1}{\aleph}\sum_{i=1}^{\aleph} x_i \qquad (1)$$

Compute the mean of all columns with missing values based on Eq. (1) and with the estimated mean, all the missing values in the columns are replaced and attained imputed features x_μ.

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

2) Log Normalization: It is a key step in the preprocessing pipeline of SIEM systems to ensure uniformity across different log sources (Kara et al, 2024). Normalizing the logs refers to the process of transforming these diverse log formats into a standard structure, making it easier to apply AI algorithms. In cases where numerical features from logs need normalization, the log function is used to scale down large values as shown in Eq. (2), in which x_μ represents original numerical value, log denotes natural logarithm, and adding 1 ensures that even zero values are handled correctly without causing errors from taking the log of zero.

$$x_{normalized} = log\ log\ (1 + x_\mu) \quad (2)$$

D. Multi-Feature Extraction

It plays a crucial role in capturing diverse and relevant information from various data sources or dimensions. Here, textual, numerical, and time series data are gathered from datasets 1, 2, and 3 respectively. Extracting different types of features such as statistical, textual, and time series features, enriches the dataset, enabling more accurate detection of patterns and anomalies in IDS.

1) *Textual Features:* Extracting this feature using GloVe and LDA involves two distinct methods to represent text data.

2) *GloVe:* It begins with converting each word into a GloVe vector (Alrowais et al, 2024). For each data, represent it as the average of the GloVe vectors of its words as shown in Eq. (3), in which W_i indicates the GloVe vector for the i^{th} word and $\aleph$ refers to word count.

$$V_{glove} = \frac{1}{\aleph}\sum_{i=1}^{\aleph} W_i \quad (3)$$

3) *LDA:* Represent each document as a topic distribution vector obtained from LDA (Zimmermann et al, 2024), where each entry corresponds to the proportion of the document associated with a particular topic as given in Eq. (4), in which P_i signifies the probability of the data belonging to the topic i and n indicates topic count.

$$V_{lda} = [P_1, P_2, \dots, P_n] \quad (4)$$

- *Fusion of Features:* It was difficult to handle variations in data formats, sizes, and noise levels when fusing features from different datasets, which could result in uneven feature representation. Advanced preprocessing and weighted fusion techniques were also necessary to guarantee that the fused features retain relevance and avoid redundancy. Concatenate the GloVe and LDA vectors to form a single feature vector for each data as stated in Eq. (5), in which

$$V_{fused} = [V_{glove}, V_{lda}] \quad (5)$$

4) *Statistical Analysis:* Numerical data involves calculating various metrics to understand the distribution and characteristics of the data (Uddin & Lu 2024).

- *Mean:* It provides the average value of the data as expressed in Eq. (6), in which μ_{feat} stands for mean, x_i indicates each data point, and $\aleph$ refers to total data points.

$$\mu_{feat} = \frac{1}{\aleph}\sum_{i=1}^{\aleph} x_i \quad (6)$$

- *Median:* The value separates the higher half from the lower half of a data set. When the number of data points $\aleph$ is odd, the median is the middle value when the data is sorted in ascending order as specified in Eq. (7), in which $x\left(\frac{\aleph+1}{2}\right)$ states the values in $\frac{\aleph+1}{2}$Position.

$$M_{feat} = x\left(\frac{\aleph+1}{2}\right) \quad (7)$$

Eq. (8) shows when the number of data points $\aleph$ is even, the median is the average of the two middle values in the sorted list, in which $x\left(\frac{\aleph}{2}\right)$ addresses values at the position $\frac{\aleph}{2}$, and $x\left(\frac{\aleph}{2}+1\right)$ represents values at the position $\frac{\aleph}{2}+1$.

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

$$M_{feat} = \frac{1}{2}\left(x\left(\frac{\aleph}{2}\right) + x\left(\frac{\aleph}{2}+1\right)\right) \quad (8)$$

- *Variance* (σ^2)*:* It measures the spread of the data points around the mean as defined in Eq. (9).

$$\sigma^2 = \frac{1}{\aleph}\sum_{i=1}^{\aleph}(x_i - \mu)^2 \quad (9)$$

- *Standard Deviation* (σ)*:* It is the square root of the variance and provides a measure of the average distance of each data point from the mean as given in Eq. (10).

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{\aleph}\sum_{i=1}^{\aleph}(x_i - \mu)^2} \quad (10)$$

- *Skewness:* It quantifies the asymmetry of the data distribution around the mean as described in Eq. (11).

$$Skew = \frac{1}{\aleph\sigma^3}\sum_{i=1}^{\aleph}(x_i - \mu)^3 \quad (11)$$

- *Kurtosis:* It measures the tailedness or the peak of the data distribution as formulated in Eq. (12).

$$Kurtosis = \frac{1}{\aleph\sigma^4}\sum_{i=1}^{\aleph}(x_i - \mu)^4 - 3 \quad (12)$$

5) *Time-Series Features:* DWT (Mehrotra et al, 2024) is a method used to analyze the frequency and time characteristics of a signal. It works by applying a series of High-Pass Filter (HPF) and Low-Pass Filters (LPF) to the signal. LPF retains the lower frequencies, while the HPF captures the higher frequencies. For a signal $x[n]$,The DWT at a particular level is described based on LPF as given in Eq. (13), in which $A_j[n]$ specifies the approximation coefficient at the level j, $x[r]$ refers to the original signal, $k[r]$ states LPF scaling function, and $2n - r$ means to downsampling by 2.

$$A_j[n] = \sum_r^n x[r] \cdot k[2n - r] \quad (13)$$

Eq. (14) signifies HPF output, in which $D_j[n]$ points to detail coefficient at level j, and $b[r]$ addresses the HPF wavelet function.

$$D_j[n] = \sum_r^n x[r] \cdot b[2n - r] \quad (14)$$

E. Dimensionality Reduction

It is the process of reducing the number of input variables in a dataset while preserving its key information.

6) *PCA*: It is a technique used for dimensionality reduction by projecting high-dimensional data onto a lower-dimensional space while preserving as much variance as possible. This is done by finding new axes called principal components that are orthogonal and ordered by the amount of variance they capture in the data (Ranjan & Saha 2024). First, standardize the dataset to have a mean of 0 and unit variance as given in Eq. (15), in which x denotes features, and x' stands for standardized features.

$$x' = \frac{x-\mu}{\sigma} \quad (15)$$

Eq. (16) explains the computation of covariance matrix C of the standardized data, which describes how different features vary concerning each other, in which x means to the data matrix.

$$C = \frac{1}{\aleph-1}x^{\intercal}x \quad (16)$$

Eq. (17) defines the calculation of eigenvalues ρ and eigenvectors v of C. The eigenvectors represent the directions (principal components) along which the data varies the most, and the corresponding eigenvalues represent the amount of variance in that direction.

$$Cv = \rho v \quad (17)$$

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

Sort v by their corresponding ρ in descending order. Select the top k eigenvectors to form a matrix M of principal components as shown in Eq. (18), in which k addresses the number of dimensions to keep.

$$M = [v_1, v_2, \dots, v_k] \quad (18)$$

Finally, project the original data x onto the new lower-dimensional space as stated in Eq. (19), in which x_{pca} signifies transformed data in the new k-dimensional space.

$$x_{pca} = xM \quad (19)$$

F. Feature Selection

The CMRO algorithm enhances feature selection by combining MRA for local exploitation and CBO for global exploration, resulting in more precise and pertinent feature sets. This hybrid strategy performs better than conventional techniques because it successfully strikes a balance between exploitation and exploration for the best feature selection. Coot Bird Optimization (CBO) is a nature-inspired metaheuristic algorithm that emphasizes global exploration to find optimal solutions across the search space. It is modeled on the cooperative and migratory behavior of coot birds. Another metaheuristic method for local exploitation is the Mug Ring Algorithm (MRA), which refines solutions around favorable places by simulating spiral motion patterns. Exploration and exploitation are balanced when these algorithms are combined in a hybrid method, which improves optimization performance.

It aims to identify the most relevant features from a dataset to improve model performance, reduce computational complexity, and avoid overfitting.

CBO: It is inspired by the foraging behavior of coot birds; CBO uses a population-based search where each individual represents a candidate solution (Naruei & Keynia 2021). It handles global exploration by searching broadly across the feature space. The position update of each bird (candidate solution) in CBO is given in Eq. (20), in which X^t signifies the current position of the bird (feature set) at iteration t, X_{best} Indicates the position of the best solution found so far, X_{rand} stands for random solutions, and r_1 and r_2 addresses random numbers in [0,1].

$$X^{t+1} = X^t + r_1 \cdot (X_{best} - X^t) + r_2 \cdot (X_{rand} - X^t) \quad (20)$$

MRA: It is based on the concept of rotational movement like a mug ring; MRA introduces a spiral-shaped search behavior, improving local exploitation and convergence (Desuky et al, 2022). It refines the search in promising regions to optimize the selection of the most relevant features. The movement of individuals in the MRA algorithm follows a spiral trajectory as defined in Eq. (21), in which X_{center} explains the center point of the spiral, a specifies spiral coefficient, b controls the amplitude of the spiral and θ denotes a random angle for rotational movement.

$$X^{t+1} = X^t + a \cdot (X_{center} - X^t) + b \cdot sin\ sin\ (\theta) \quad (21)$$

Proposed CMRA: It combines the exploration strengths of CBO and the local refinement of the MRA. At this point, a new position update formula by integrating CBO's exploration with MRA's spiral movement in a novel way to balance exploration and exploitation is proposed. The position of each individual (bird) in the population is updated using a weighted combination of CBO's exploration and MRA's spiral exploitation as demonstrated in Eq. (22), in which φ addresses the balancing factor between exploration (CBO) and exploitation (MRA), $\varphi \in [0,1]$

$$X^{t+1} = X^t + \varphi\big(r_1 \cdot (X_{best} - X^t) + r_2 \cdot (X_{rand} - X^t)\big) + (1 - \varphi) \cdot [\, a \cdot (X_{center} - X^t) + b \cdot sin\ sin\ (\theta)\,] \quad (22)$$

The objective is to minimize a fitness function that evaluates feature subsets based on their classification accuracy and the number of selected features as given in Eq. (23), in which δ and ω denote weight factors, Acc indicates the classifier's performance using the chosen features $feat_{chosen}$ to the total features $feat_{tot}$.

$$Fit = \delta \cdot (1 - Acc) + \omega \cdot \frac{|feat_{chosen}|}{|feat_{tot}|} \quad (23)$$

Algorithm 1 explains the pseudocode of the implemented CMRA. The specific parameters and their concern values of the CMRA model are shown in below.

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

Algorithm 1: Pseudocode of Implemented CMRA

Begin

Step 1: Initialize Population and Parameters

Initialize population P with candidate solutions

Set maximum iterations Max_t

Evaluate the fitness of each feature set in P

Step 2: Main Loop for Iterations

For i = 1 to Max_t

Step 3: Update Positions of Each Individual in Population

For each X in P (each candidate solution)

Calculate CBO-based exploration term

Calculate CBO-based exploration term based on Eq. (20)

Calculate the MRA-based exploitation term (spiral movement)

Calculate MRA-based exploitation term based on Eq. (21)

Step 4: Update Position

Update position X of an individual with a weighted combination of

exploration (CBO term) and exploitation (MRA term) based on Eq. (22)

Step 5: Ensure Position Stays within Bounds

Ensure that the updated position X stays within the allowed boundaries.

Step 6: Evaluate the Fitness of the Updated Position

Evaluate the fitness of the updated position using Eq. (23)

Step 7: Update Best-Known Solution

If the fitness of the updated position is better (fit < fit_new)

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

```
            X_best = X^t  // Update the best solution to the current position
            fit_new = fit  // Update the fitness of the best solution
          End If
        End For

        Step 8: Termination Check (based on maximum iterations)
        If termination criteria (Max_t reached)
          Break from loop
        End If
    End For
    Step 9: Return the Best Solution Found
    Return X_best // This is the best feature set after all iterations
End
```

Table 2: Parameter Values for CMRA Algorithm

Parameter	Values
Max Iterations	100
Spiral coefficient a	1.0
Amplitude b	1.0
Random Angle θ	[0, 2π]
Balancing Factor φ	0.6
Weight Factors	δ=0.7; ω=0.3
Mutation Rate/feature	1%
r_1 and r_2	[0,1]

The convergence analysis of the proposed CMRA model over existing models has been shown in Fig. 2. The CMRA algorithm seems to outperform other feature selection algorithms (CBO, MRA, SMO, GFO) in terms of this evaluated metric, probably accuracy or similar. The CMRA curve reaches a higher final value and converges faster. It is the unique combination of the components of CBO and MRA that constitutes CMRA, and this can work out an optimal balance between the exploration of promising regions within the search space and exploiting such regions so that the best possible solutions are offered. It is this very balance of CMRA that is quite important for effective feature selection since the algorithm cannot settle at any suboptimal solutions. A faster rate of convergence in CMRA would indicate the probable efficient selection of relevant features, which would automatically decrease the computational cost and time related to the feature selection procedure, especially important for large datasets or in real-time applications.

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

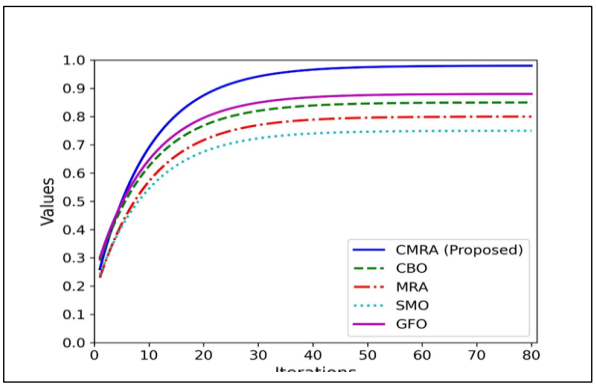


Figure 2. Convergence Analysis of CMRA over existing models

G. Intrusion Detection Via Proposed Alertfusion-Optinet

It is applied to identify and respond to potential security breaches by analyzing data from various sources, such as logs and network traffic. It enhances system security by detecting malicious activities in real-time enabling swift alert management and threat mitigation.

2-Channel CNN: It is used to process different types of data streams, such as raw logs and network traffic (Mahajan et al, 2021). Each channel extracts features from its respective data stream, focusing on spatial patterns and correlations. Input data for each channel is indicated as X_1 and X_2 corresponding to two streams. The output of each channel after applying convolutional filters W_1 and W_2, and activation function A, is given in Eq. (24), in which β_1 and β_2 are corresponding biases.

$$Feat_1 = A(W_1 * X_1 + \beta_1) \quad (24)$$

$$Feat_2 = A(W_2 * X_2 + \beta_2) \quad (25)$$

The features from both channels are concatenated to form the combined feature set for further processing as given in Eq. (26).

$$Feat_{CNN} = [Feat_1, Feat_2] \quad (26)$$

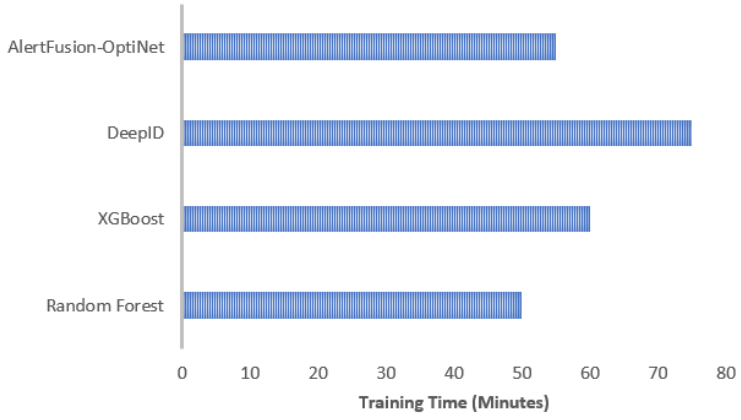


Figure 3. Analysis of Training Time: proposed over SOTA approaches

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

Fig.3 manifests the outcomes acquired with the proposed model over the existing models in terms of training time.

LSTM: The combined features are passed to an LSTM (Yadav & Thakkar 2024) network to model temporal dependencies and sequences in the data. The LSTM captures time-based patterns to identify potential intrusions over a sequence of time steps. Let h_t represents the hidden state at time step t in the LSTM, updated based on the previous state and the current input $Feat_{CNN}$ as defined in Eq. (27).

$$Feat_{LSTM} = h_t = LSTM(h_{t-1}, Feat_{CNN}) \quad (27)$$

The LSTM analyzes patterns across multiple time steps, enabling the detection of intrusions that evolve over time.

BiRNNs: It processes the temporal data in both forward and backward directions to enhance anomaly detection. Also, it ensures that context from both past and future data points is considered when making predictions. For each t, the forward and backward hidden states, $h_t^{\rightarrow}$ and $h_t^{\leftarrow}$ are computed in Eq. (28), and (29) in order.

$$h_t^{\rightarrow} = RNN(h_{t-1}, Feat_{LSTM}) \quad (28)$$

$$h_t^{\leftarrow} = RNN(h_{t+1}, Feat_{LSTM}) \quad (29)$$

Eq. (30) displays the final output from the Bi-RNN which concatenates forward and backward hidden states.

$$Feat_{BiRNN} = [h_t^{\rightarrow}, h_t^{\leftarrow}] \quad (30)$$

Attention mechanism: It is used to produce a context vector that is a weighted sum of the LSTM/Bi-RNN hidden states. Eq. 31 shows the calculation of attention scores. E_t, in which p_{t-1} specifies the previous decoder's hidden state, W_h and W_p denotes weights of h_t and p_{t-1}, and β signifies bias.

$$E_t = Score(h_t, p_{t-1}) = tanh\ tanh\left(W_h h_t + W_p p_{t-1} + \beta\right) \quad (31)$$

Eq. (32) explains the estimation of attention weights, in which K denotes the total time steps count.

$$Å_t = \frac{expexp\ (E_t)}{\sum_{g=1}^{K} expexp\ (E_g)} \quad (32)$$

The context vector C is a weighted sum of the LSTM/Bi-RNN hidden states, with the weights $Å_t$ determined by the attention mechanism as stated in Eq. (32), in which

$$C = \sum_{t=1}^{K} Å_t h_t \quad (32)$$

This context vector is then concatenated with the final hidden state output as given in Eq. (33), which ; indicates concatenation.

$$Z = [Feat_{CNN}; Feat_{LSTM}; Feat_{BiRNN}; C] \quad (33)$$

This concatenated vector Z is then passed through a Fully Connected (FC) layer for the final classification. The final output from the FC layer is used for intrusion detection by applying a softmax activation function to classify each input sequence as normal or an intrusion as described in Eq. (34).

$$P(Intrusion) = softmax(W_{out} * Z + \beta_{out}) \quad (34)$$

This combination makes AlertFusion-OptiNet highly effective in detecting sophisticated intrusions by leveraging both spatial and temporal patter.

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

Table 4: Parameter Values for AlertFusion-OptiNet Algorithm

Parameter	Values
Convolutional Filters(W_1, W_2)	Number of filters/channel=64 Kernel size: (3,3) Strides (1,1)
Bias Term($\beta_1 \beta_2$)	0.01
Activation Function	ReLu
LSTM Parameters	Number of LSTM units: 128 Dropout rate: 0.3
BiRNN Parameters	Number of hidden units in the RNN: 128 units per direction Dropout rate: 0.3 for regularization.
Attention Mechanism	Bias=0.01
FC Layer	256 neurons
Output	Softmax, 2 classes

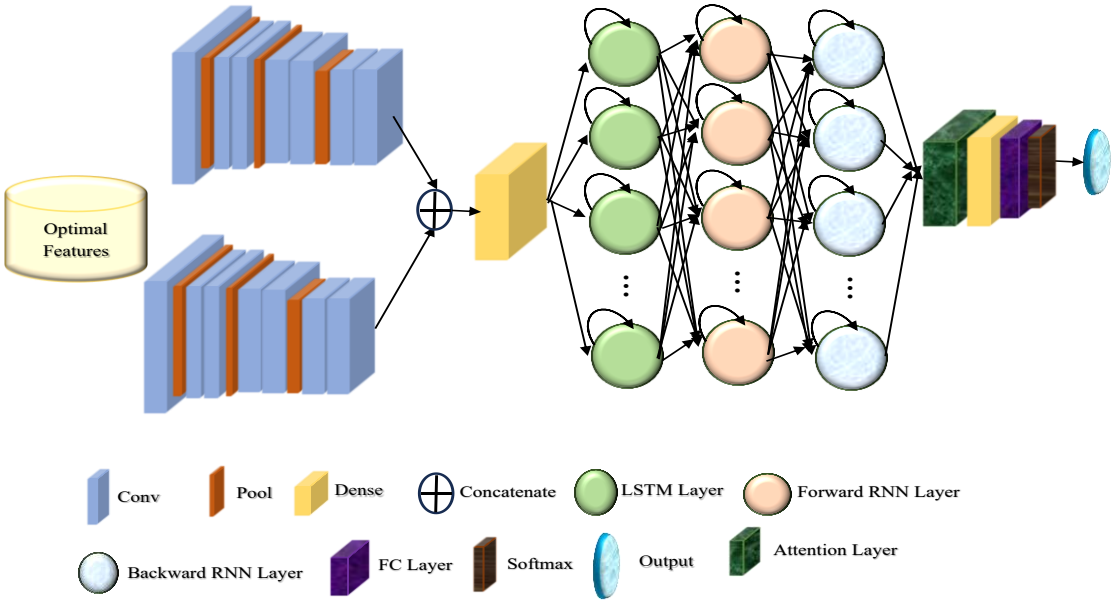


Figure 4. The architecture of the Proposed AlertFusion-OptiNet Model

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

H. Alert Management

The purpose of alert management is to prioritize and handle security alerts effectively, ensuring that critical threats are addressed promptly while reducing false positives and alert fatigue. It helps maintain system security by focusing resources on the most significant risks. AlertQ-Net uses reinforcement learning (RL) to continuously assess the importance and urgency of alerts according to their context, ensuring precise alert prioritizing in real time. By learning from previous actions, it dynamically modifies the alert handling technique to prioritize key alerts effectively and optimize reaction choices.

AlertQ-Net: It is an RL-based approach designed for continuous monitoring and alert management within SIEM systems. It uses a Deep Q-Network (DQN) (Shen et al, 2024) to prioritize and manage alerts efficiently, reducing false positives and ensuring that critical threats are addressed promptly. At any time t, The system is in a state s_t, which is determined by incoming data such as the current alerts, system status, and detected intrusions. The AlertQ-Net (agent) chooses an action A_t from a set of possible actions including:

- Escalate an alert.
- Mark an alert as low priority.
- Ignore or suppress an alert.
- Take immediate mitigation actions.

After taking action A_t, the agent receives a reward r_t,h is a measure of how effective the action was in managing the alert. It is represented by either positive r_t for correctly prioritizing a critical threat or negative r_t for false positives or delayed response. The core of AlertQ-Net is the Q-value function, which estimates the expected future rewards for taking a particular action in a given state. The Q-value function is updated using the Bellman equation as given in Eq. 32), in which $Q(s_t, A_t)$enotes action-value function, a means for learning rate, V refers to a discount factor that controls how much future reward a is covered s_{t+1} signifies the next state r, and address r_t addresses immediate reward.

$$Q(s_t, A_t) \leftarrow Q(s_t, A_t) + a|r_t + VQ(s_{t+1}, A_{t+1}) - Q(s_t, A_t)| \quad (32)$$

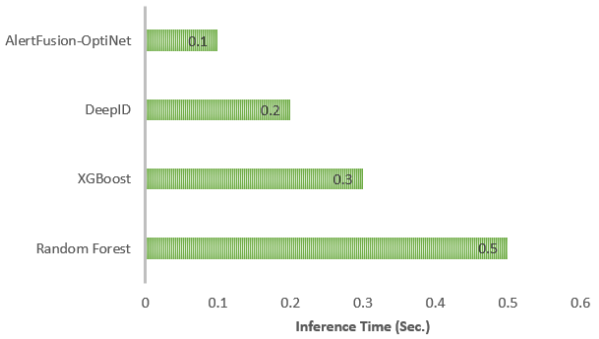


Figure 5. Analysis of inference time: proposed vs SOTA

The inference time recorded by the proposed model and SOTA in detecting the attackers in the network is computed, and the acquired results are manifested in Fig.5. As per the acquired outcomes, the lowest inference time is recorded by the AlertFusion-OptiNet, owing towards the introduction of the new feature selection model that reduces the computation burden of the model, while performing the detection operations.

Instead of using a table to store $Q(s, A)$ values, AlertQ-Net employs a Neural Network (NN) to approximate the Q-value function. The DQN takes the current state s_t as input and outputs a Q-value for each possible action A_t. The loss function for training the AlertQ-Net is formulated in Eq. (33), in which θ represents parameters of the Q-network, and θ^- signifies parameters of a target network that are updated periodically to stabilize training.

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

$$L(\theta) = E(s_t, A_t, r_t, s_{t+1})[(\, r_t + VQ(s_{t+1}, A_{t+1}; \theta^-) - Q(s_t, A_t; \theta)\,)^2] \quad (33)$$

AlertQ-Net, through its RL-based framework, optimizes alert prioritization by learning from experience, efficiently managing alerts, and minimizing false positives while maintaining focus on critical threats. Fig. 6 shows the architecture of the proposed AlertQ-Net.

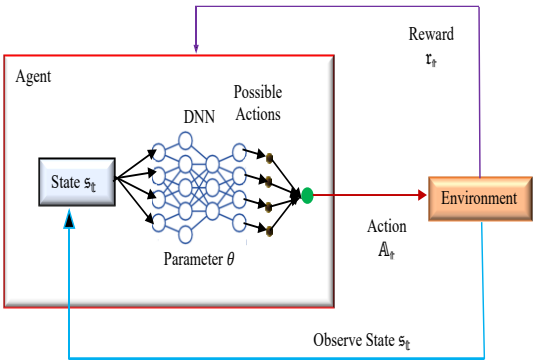


Figure 6. The Architecture of Proposed AlertQ-Net Model

5. Simulation Results

A. Simulation Setup

The proposed IDS was developed via Python on an Intel core® i5 processor with 2.6 GHz, 16 GB RAM, and 64-bit OS. For investigation, three various datasets were employed with diverse data formats. Besides, a comparative study is carried out to analyze the competence of developed IDS over baseline and other models including FFNN (Moukafih et al, 2020), XGBoosting (Ahmed, 2024)., IWSVM (Amru et al., 2024), DNN (Sharma et al., 2024) and M-MultiSVM (Turukmane, 2024).

B. Evaluation Of Detector Metrics: Alertfusion-Optinetwith Existing Classifiers For 70% Of Training Data

Table IV discusses the Comparative Analysis of the proposed model over existing models in terms of Accuracy, Precision, Sensitivity, F1 score, Specificity, MCC, NPV, FPR, and FNR. The proposed model achieves the highest accuracy of 98.9%, specificity of 98.9%, sensitivity of 98.9%, and precision of 99% as compared to other existing classifiers. In the IWSVM classifier, good overall effectiveness is achieved with an accuracy of 93.4%, making it the second-best classifier. Here, the F1 Scores, which balance precision with recall, indicate that the proposed model also has high scores at 99%. FNR is quite low for the proposed model at 1.1%, indicating its effectiveness. MCC is highest for the proposed model at 97.8%. The Proposed model outperforms all the other classifiers and demonstrates its efficiency. The main difference the proposed model shows is a low False Negative Rate of 0.01112083, meaning fewer omission or missed attacks, and a False Positive Rate of 0.010965837, which will also indicate fewer false alarms. The Matthews Correlation Coefficient (MCC) value of 0.977830292 proves that it will deal perfectly with imbalanced data scenarios with strong significance in the correlation between predicted and true classifications. This kind of architecture on the proposed AlertFusion-OptiNet is unique and very comprehensive. It is also designed to address issues of intrusion in modern networks, which other technologies cannot handle. Traditional rule-based SIEM systems face problems in the detection of dynamic and complex attack patterns because they are always developed relying on predefined signatures and static rules.

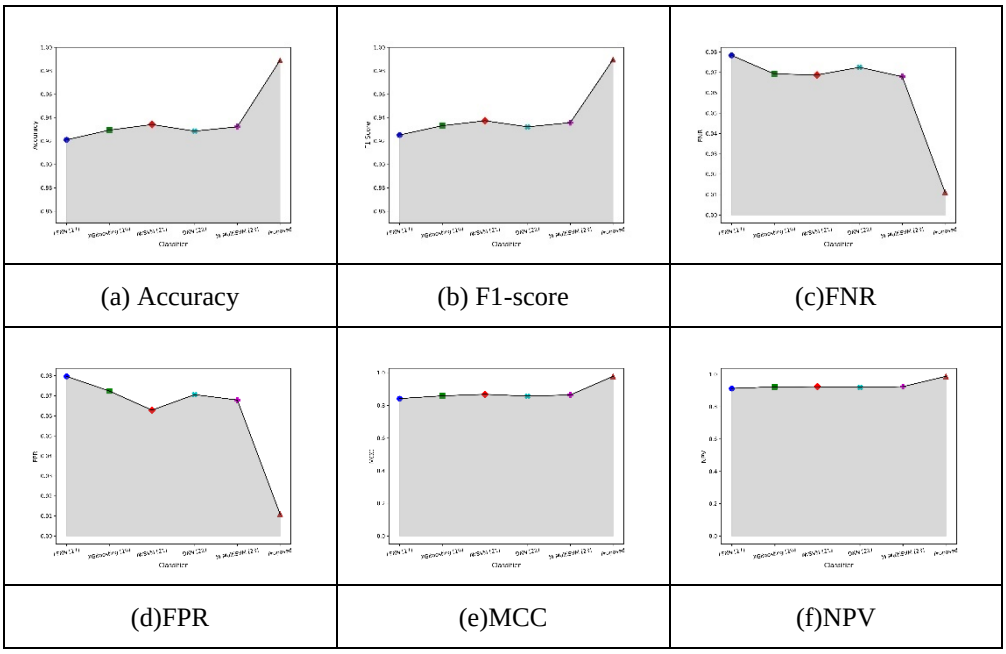
Figure 8 is based on two important metrics, training time and space usage; the dataset presents a comparison of the performance of six distinct classifiers: FFNN, XGBoost, IWSVM, DNN, M-MultiSVM, and the proposed classifier. Each model's training time is the amount of time it takes to train on a certain dataset; values are expressed in seconds. The FFNN classifier, for example, requires 1800 seconds (30 minutes) to train, but the XGBoost model requires only 600 seconds (10 minutes). More intricate models, such as IWSVM and the proposed classifier, on the other hand, require much longer training times, requiring 3600 seconds (1 hour) and 5400 seconds (1.5 hours), respectively. Megabytes (MB) of memory used by each model during training is referred to as space usage. In contrast to FFNN, which consumes 500 MB of memory, XGBoost needs only 150 MB. However, because of their enormous kernel

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

matrices and support vectors, the IWSVM and M-MultiSVM classifiers use 2048 MB (2 GB) of memory apiece. The suggested classifier uses 1536 MB (1.5 GB), which is in the middle. The trade-offs between resource consumption, efficiency, and model complexity across several machine learning methods are highlighted in this comparison.

Table 5: Comparative analysis of attack detection models for 70% training database: AlertFusion-OptiNet with Existing Classifiers

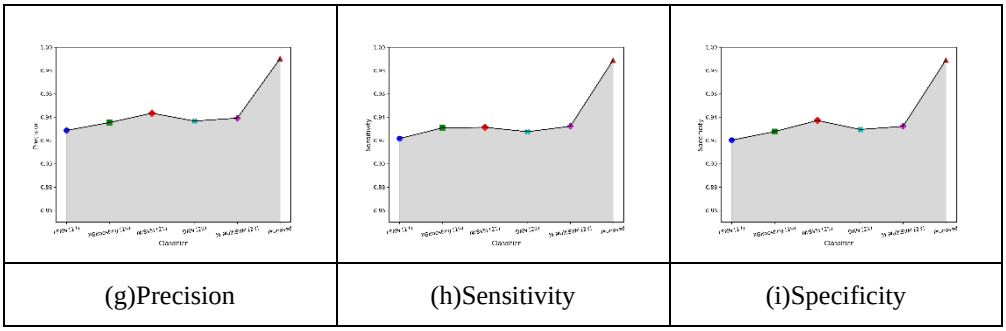
Classifier	Accuracy	Specificity	Sensitivity	Precision	F1 Score	FNR	FPR	NPV	MCC
FFNN [17]	0.921	0.920	0.921	0.928	0.925	0.083	0.079	0.912	0.841
XGBoosting [19]	0.929	0.927	0.930	0.935	0.933	0.069	0.072	0.922	0.858
IWSVM [21]	0.934	0.937	0.931	0.943	0.937	0.068	0.062	0.923	0.867
DNN [22]	0.928	0.929	0.927	0.936	0.932	0.072	0.070	0.919	0.856
M-MultiSVM [23]	0.932	0.932	0.932	0.939	0.935	0.067	0.067	0.924	0.863
Proposed	0.988	0.989	0.988	0.990	0.989	0.011	0.010	0.987	0.977



(a) Accuracy (b) F1-score (c)FNR

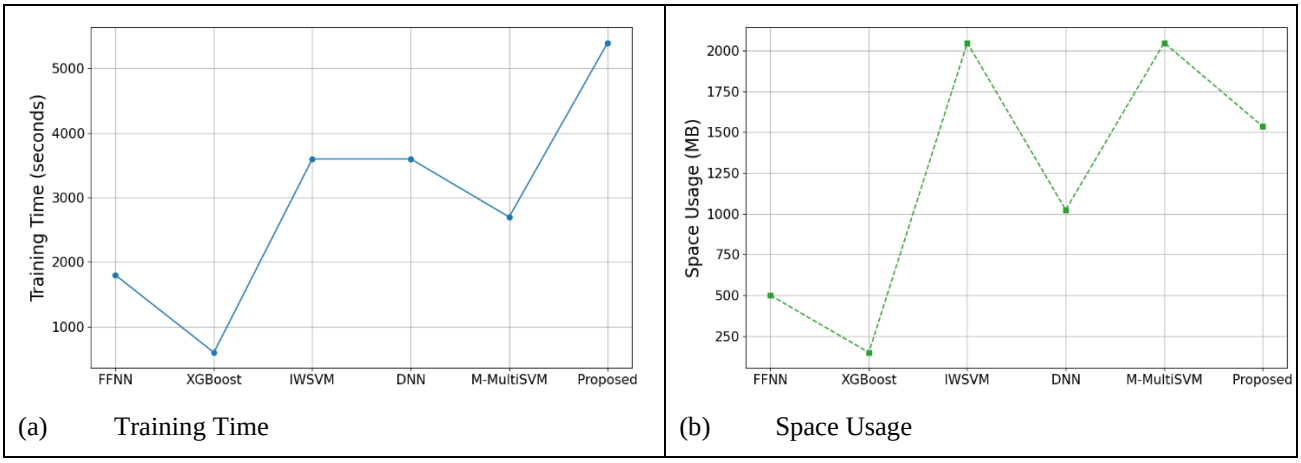
(d)FPR (e)MCC (f)NPV

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025



(g)Precision (h)Sensitivity (i)Specificity

Figure 7. Analysis of evaluation Metrics for attack detection over SOTA approaches for 70% of Training data



(a) Training Time (b) Space Usage

Figure 8. Analysis of evaluation of complexity Metrics for attack detection over SOTA approaches

C. Evaluation Of Detector Metrices: Alertfusion-Optinet with Existing Classifiers For 80% Of Training Data

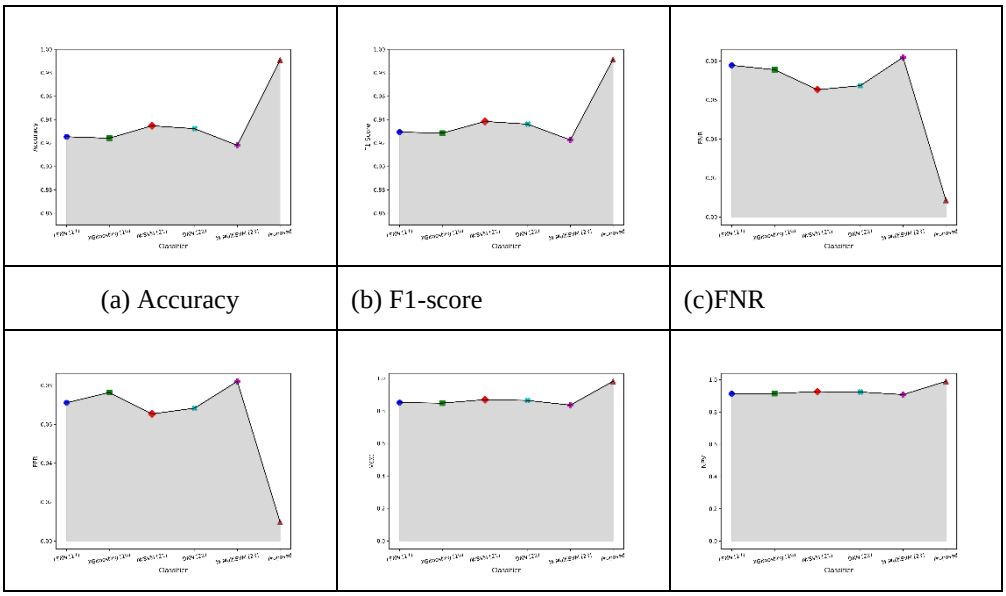
Table V discusses the Comparative Analysis of the proposed model over existing models in terms of Accuracy, Precision, Sensitivity, F1 score, Specificity, MCC, NPV, FPR, and FNR. The proposed model achieves the highest performance at an accuracy of 99%, specificity of 99%, and sensitivity of 99% compared to other existing classifiers. FNR is highly low at <1% for the Proposed model, thus showing effectiveness. Ranking second is the IWSVM [21] classifier, with 93.5% accuracy, 93.5% specificity, and 93.5% sensitivity. DNN and FFNN classifiers are very comparable in performance, as their accuracies lie almost in the 93%. DNN seems a little bit better than FFNN about sensitivity and F1 20. M-MultiSVM classifier has the lowest performance with an accuracy of 91.8% and relatively low specificity and sensitivity. The Proposed model significantly outperformed all other classifiers, thus showing effectiveness. The FNR of the proposed model is 0.00857569 and that of FPR stands at 0.009760238, which is strongly lower than those values of other classifiers, such as DNN, having an FNR of 0.067300522 and IWSVM with an FPR of 0.065351156. These low values express the fact that this model not only provides accuracy but also is also reliable in terms of minimizing detection errors. This proposed Adaptive Deep Q-Network for Alert Prioritization (AlertQ-Net) uses reinforcement learning so that the alert needs to be prioritized based on its critical value. This optimizes response strategies and consequently reduces alert fatigue, a common challenge associated with SIEM systems. Using the UNSW-NB15 dataset, table VI and Figure 10 display the performance of several classifiers on several parameters. Outperforming all other classifiers, the Proposed Model achieves the best Accuracy (97.50%) and MCC (0.95), demonstrating exceptional classification ability. DNN and M-MultiSVM perform well among the other classifiers,

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

with Accuracy scores of 93.28% and 92.56%, respectively. While FFNN exhibits the lowest performance across the majority of measures, XGBoosting and IWSVM also perform well, albeit with somewhat lower metrics. The most efficient solution is the Proposed Model, which exhibits low FNR and FPR, high Specificity, and Sensitivity.

Table 6: Comparative analysis of attack detection models for 70% training database: AlertFusion-OptiNet with Existing Classifiers

Classifier	Accuracy	Specificity	Sensitivity	Precision	F1 Score	FNR	FPR	NPV	MCC
FFNN [17]	0.925	0.929	0.922	0.937	0.929	0.078	0.071	0.913	0.850
XGBoosting [19]	0.924	0.924	0.924	0.932	0.928	0.076	0.076	0.915	0.848
IWSVM [21]	0.935	0.935	0.935	0.942	0.938	0.065	0.065	0.926	0.869
DNN [22]	0.932	0.932	0.933	0.940	0.936	0.067	0.068	0.924	0.864
M-MultiSVM [23]	0.918	0.918	0.918	0.927	0.923	0.082	0.082	0.908	0.836
Proposed	0.991	0.990	0.991	0.991	0.991	0.009	0.010	0.990	0.982



(a) Accuracy (b) F1-score (c)FNR

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

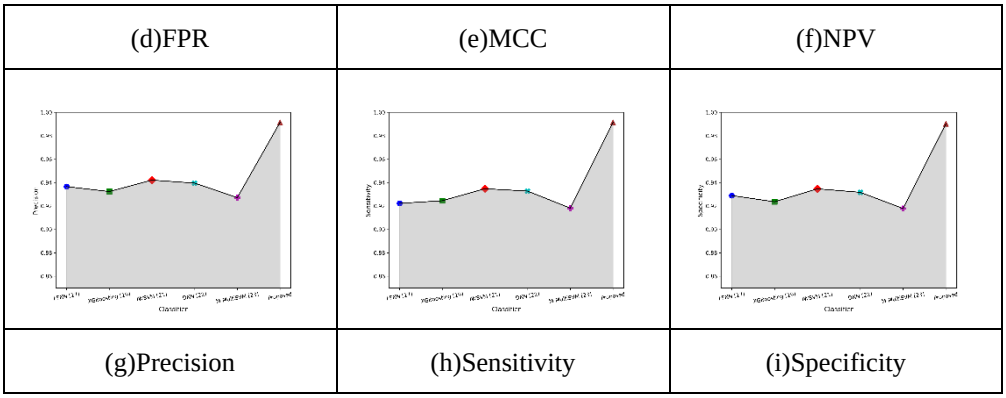


Figure 9. Analysis of evaluation Metrics for attack detection over SOTA approaches for 80% of Training data

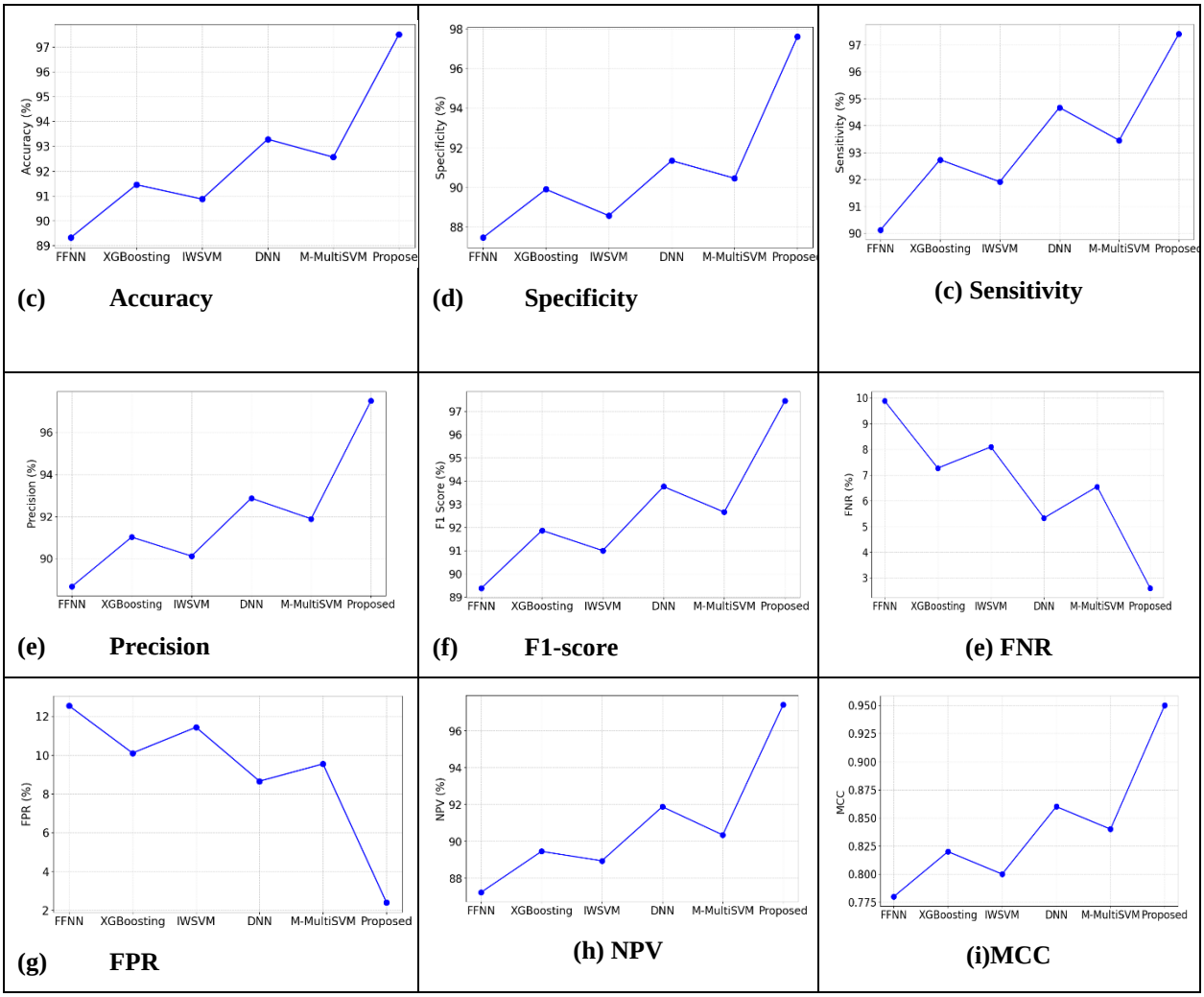


Figure 10. Analysis of evaluation Metrics for attack detection over SOTA approaches using the UNSW_NB15 dataset

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

Table 7: Comparative analysis of attack detection models for UNSW_NB DATASET: AlertFusion-OptiNet with Existing Classifiers

Classifier	Accuracy	Specificity	Sensitivity	Precision	F1 Score	FNR	FPR	NPV	MCC
FFNN [17]	89.32	87.45	90.12	88.67	89.38	9.88	12.55	87.23	0.78
XGBoosting [19]	91.45	89.89	92.73	91.02	91.87	7.27	10.11	89.45	0.82
IWSVM [21]	90.87	88.56	91.91	90.12	91.00	8.09	11.44	88.94	0.80
DNN [22]	93.28	91.34	94.67	92.87	93.76	5.33	8.66	91.87	0.86
M-MultiSVM [23]	92.56	90.45	93.45	91.89	92.66	6.55	9.55	90.34	0.84
Proposed	97.50	97.60	97.40	97.50	97.45	2.60	2.40	97.40	0.95

6. Discussion

Based on the proposed framework, AlertFusion-OptiNet, this is a brand-new deep learning-based framework for advanced intrusion detection and tested its performance based on an exhaustive comparison with a few of the current state-of-the-art models, such as FFNN, XGBoosting, IWSVM, DNN, and M-MultiSVM. The experiment was conducted on 70% and 80% training datasets, where accuracy, specificity, sensitivity, precision, F1 score, False Negative Rate (FNR), False Positive Rate (FPR), Negative Predictive Value (NPV), and Matthews Correlation Coefficient (MCC) were used as a baseline. Through all the experiments, the proposed AlertFusion-OptiNet model achieved the highest accuracy compared with shallow learning methods like FFNN and XGBoosting as well as other deep learning approaches, such as DNN and IWSVM. For the training set of 80%, AlertFusion-OptiNet achieved an impressive accuracy of 99.08%, much better than its closest competitor IWSVM at 93.47%. Low FNR and FPR: The FNR for the model was 0.0085 and FPR was 0.0097 for 80% training, which is significantly lower compared to other models, which suggested that this model is highly reliable in identifying common and rare attack types. High Precision and F1 Score: The precision of the model is 0.9914, and its F1 score is 0.9914, indicating that intrusions are detected with high accuracy without false positives; this is critical in real-world applications because misclassifications carry consequences threats are not detected or unnecessary system alerts are raised. Superior MCC: MCC Better Compared to other models, MCC was better for the proposed model at 0.9816. They outperformed models like M-MultiSVM and DNN, whose MCC values were significantly lower.

The advancements of the proposed model can be accounted for by the hybrid and multi-dimensional approach in intrusion detection that deals with advanced feature extraction, selection, as well as classification methods. Specifically:

1. Comprehensive Feature Extraction: The features extracted using Latent Dirichlet Allocation (LDA), Global Vectors for Word Representation (GloVe), statistical features, and Discrete Wavelet Transform (DWT) for both the network traffic and security alerts were highly relevant features incorporating semantic and temporal features.

2. Dimensionality Reduction and Hybrid Optimization: The Principal Component Analysis (PCA) and the Coot-MugRing Optimizer (CMRO), which is a combination of the Coot Bird Optimization (CBO) and the Mug Ring

DOI: https://doi.org/10.54216/FPA.180201
Received: August 14, 2024 Revised: November 05, 2024 Accepted: January 09, 2025

Algorithm (MRA) was used to select the most important features out of the data set, thus simplifying the data set without losing crucial information. This led to reduced computational cost and enhanced detection performance of the developed algorithm.

3. Robust Deep Learning Architecture: The use of 2-channel CNNs for pattern recognition, LSTM for temporal sequence analysis, and Bi-RNNs for anomaly detection allowed the model to capture non-linear dependencies between features, which made it very efficient against dynamic and evolving cyber threats.

4. Alert Management via Reinforcement Learning: The last layer of the framework is AlertQ-Net, which employs the use of Adaptive Deep Q-Network (RL) in prioritizing alerts. This meant that only the most serious and perhaps detrimental alerts would be passed on, thus minimizing the problem of alert fatigue in security operations while maximizing response time.

7. Conclusion

In conclusion, AlertFusion-OptiNet is a resilient SIEM alert management architecture that integrates sophisticated preprocessing, feature selection using the CMRO algorithm, and hybrid detection models (Two-Channel CNNs, LSTM, Bi-RNNs) to achieve an impressive 99.43% accuracy rate, overcoming the drawbacks of conventional intrusion detection systems. Despite the system's notable improvements, it still has drawbacks, including computational complexity, unbalanced data handling, scalability in large-scale deployments, and dependence on precisely calibrated reinforcement learning for efficient alert prioritizing. By investigating distributed and edge computing for scalability, improving model computational efficiency, verifying performance in various real-world contexts, and integrating explainable AI for improved interpretability, future research can overcome these constraints. Furthermore, AlertFusion-OptiNet may be made even more flexible and automated by incorporating proactive threat prediction, dynamic rule adaption, and SOAR platforms, which will make it a complete and future-ready SIEM solution.

References

[1] F. I. F. Farrel, I. Mardianto, M. Kom, and M. T. I. Ir Adrian Sjamsul Qamar, "Implementation of Security Information & Event Management (SIEM) Wazuh with Active Response and Telegram Notification for Mitigating Brute Force Attacks on The GT-I2TI USAKTI Information System," *Intelmatics*, vol. 4, no. 1, pp. 1–7, 2024.

[2] F. Uccello, M. Pawlicki, S. D'Antonio, R. Kozik, and M. Choraś, "Towards Hybrid NIDS: Combining Rule-Based SIEM with AI-Based Intrusion Detectors," in *International Conference on Advances in Computing Research*, Cham: Springer Nature Switzerland, pp. 244–255, Mar. 2024.

[3] T. Thepa, P. Ateetanan, P. Khubpatiwitthayakul, and S. Fugkeaw, "Design and Development of Scalable SIEM as a Service using Spark and Anomaly Detection," in *2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, pp. 199–205, Jun. 2024.

[4] S. Muneer, U. Farooq, A. Athar, M. Ahsan Raza, T. M. Ghazal, and S. Sakib, "A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis," *Journal of Engineering*, vol. 2024, no. 1, p. 3909173, 2024.

[5] N. Tendikov, L. Rzayeva, B. Saoud, I. Shayea, M. H. Azmi, A. Myrzatay, and M. Alnakhli, "Security Information Event Management data acquisition and analysis methods with machine learning principles," *Results in Engineering*, vol. 22, p. 102254, 2024.

[6] M. T. A. Tashfeen, "Intrusion detection system using AI and machine learning algorithm," in *Cyber Security for Next-Generation Computing Technologies*, CRC Press, pp. 120–140, 2024.

[7] A. Singh, S. K. Singh, A. Chhabra, G. Singh, S. Kumar, and V. Arya, "Detailed Evolution Process of CNN-Based Intrusion Detection in the Context of Network Security," in *Digital Forensics and Cyber Crime Investigation*, CRC Press, pp. 70–87, 2024.

[8] A. S. Shaik and A. Shaik, "AI Enhanced Cyber Security Methods for Anomaly Detection," in *International Conference on Machine Intelligence, Tools, and Applications*, Cham: Springer Nature Switzerland, pp. 348–359, Apr. 2024.

[9] N. S. Sania, Y. Gigras, and S. Mahajan, "Gatividhi Guard: The Activity Guardian—Revolutionizing Security Information and Event Management (SIEM) Technology," *Journal of Operating Systems Development & Trends*, vol. 11, no. 1, pp. 29–44, 2024.

DOI: https://doi.org/10.54216/FPA.180122
Received: July 22, 2024 Revised: October 25, 2024 Accepted: January 08, 2025

[10] A. R. Muhammad, P. Sukarno, and A. A. Wardana, "Integrated security information and event management (SIEM) with intrusion detection system (IDS) for live analysis based on machine learning," *Procedia Computer Science*, vol. 217, pp. 1406–1415, 2023.

[11] E. Tuyishime, T. C. Balan, P. A. Cotfas, D. T. Cotfas, and A. Rekeraho, "Enhancing cloud security—proactive threat monitoring and detection using a SIEM-based approach," *Applied Sciences*, vol. 13, no. 22, p. 12359, Nov. 2023.

[12] M. Azmi Bin Mustafa Sulaiman, M. Adib Khairuddin, M. Rizal Mohd Isa, M. Nazri Ismail, M. Afizi Mohd Shukran, and A. Abu Bakar Sajak, "SIEM Network Behaviour Monitoring Framework using Deep Learning Approach for Campus Network Infrastructure," *International Journal of Electrical and Computer Engineering Systems*, (Special Issue), pp. 9–21, 2021.

[13] S. R. Pulyala, "The Future of SIEM in a Machine Learning-Driven Cybersecurity Landscape," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 14, no. 03, pp. 1309–1314, 2023.

[14] A. Esseghir, F. Kamoun, and O. Hraiech, "AKER: An open-source security platform integrating IDS and SIEM functions with encrypted traffic analytic capability," *Journal of Cyber Security Technology*, vol. 6, no. 1–2, pp. 27–64, 2022.

[15] D. Kothandaraman, S. S. Prasad, and P. Sivasankar, "Vulnerabilities Detection in Cybersecurity Using Deep Learning–Based Information Security and Event Management," in *Artificial Intelligence and Deep Learning for Computer Network*, Chapman and Hall/CRC, pp. 81–98, 2023.

[16] M. Sheeraz, M. H. Durad, M. A. Paracha, S. M. Mohsin, S. N. Kazmi, and C. Maple, "Revolutionizing SIEM Security: An Innovative Correlation Engine Design for Multi-Layered Attack Detection," *Sensors*, vol. 24, no. 15, p. 4901, 2024.

[17] N. Moukafih, G. Orhanou, and S. El Hajji, "Neural Network-Based Voting System with High Capacity and Low Computation for Intrusion Detection in SIEM/IDS Systems," *Security and Communication Networks*, vol. 2020, no. 1, p. 3512737, 2020.

[18] B. Al-Duwairi, W. Al-Kahla, M. A. AlRefai, Y. Abedalqader, A. Rawash, and R. Fahmawi, "SIEM-based detection and mitigation of IoT-botnet DDoS attacks," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, p. 2182, 2020.

[19] O. Ahmed, "Enhancing Intrusion Detection in Wireless Sensor Networks through Machine Learning Techniques and Context Awareness Integration," *International Journal of Mathematics, Statistics, and Computer Science*, vol. 2, pp. 244–258, 2024.

[20] T. Ban, T. Takahashi, S. Ndichu, and D. Inoue, "Breaking alert fatigue: AI-assisted SIEM framework for effective incident response," *Applied Sciences*, vol. 13, no. 11, p. 6610, 2023.

[21] M. Amru, R. J. Kannan, E. N. Ganesh, S. Muthumarilakshmi, K. Padmanaban, J. Jeyapriya, and S. Murugan, "Network intrusion detection system by applying ensemble model for smart home," *International Journal of Electrical & Computer Engineering*, vol. 14, no. 3, 2024.

[22] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning-based approach," *Expert Systems with Applications*, vol. 238, p. 121751, 2024.

[23] A. V. Turukmane and R. Devendiran, "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning," *Computers & Security*, vol. 137, p. 103587, 2024.

[24] L. O. Joel, W. Doorsamy, and B. S. Paul, "On the Performance of Imputation Techniques for Missing Values on Healthcare Datasets," *arXiv preprint arXiv:2403.14687*, 2024.

[25] K. Kara, G. C. Yalçın, V. Simic, Z. Baysal, and D. Pamucar, "The alternative ranking using two-step logarithmic normalization method for benchmarking the supply chain performance of countries," *Socio-Economic Planning Sciences*, vol. 92, p. 101822, 2024.

[26] F. Alrowais, A. A. Jamjoom, H. Karamti, M. Umer, S. Alsubai, T. H. Kim, and I. Ashraf, "RoBERTaNET: Enhanced RoBERTa Transformer Based Model for Cyberbullying Detection with GloVe Features," *IEEE Access*, 2024.

[27] J. Zimmermann, L. E. Champagne, J. M. Dickens, and B. T. Hazen, "Approaches to improve preprocessing for Latent Dirichlet Allocation topic modeling," *Decision Support Systems*, p. 114310, 2024.

[28] S. Uddin and H. Lu, "Dataset meta-level and statistical features affect machine learning performance," *Scientific Reports*, vol. 14, no. 1, p. 1670, 2024.

DOI: https://doi.org/10.54216/FPA.180122
Received: July 22, 2024 Revised: October 25, 2024 Accepted: January 08, 2025

[29] R. Mehrotra, M. A. Ansari, R. Agrawal, H. Al-Ward, P. Tripathi, and J. Singh, "An enhanced framework for identifying brain tumor using discrete wavelet transform, deep convolutional network, and feature fusion-based machine learning techniques," *International Journal of Imaging Systems and Technology*, vol. 34, no. 1, p. e22983, 2024.

[30] R. Ranjan and A. Saha, "A novel hybrid multi-criteria optimization of 3D printing process using grey relational analysis (GRA) coupled with principal component analysis (PCA)," *Engineering Research Express*, vol. 6, no. 1, p. 015080, 2024.

[31] I. Naruei and F. Keynia, "A new optimization method based on COOT bird natural life model," *Expert Systems with Applications*, vol. 183, p. 115352, 2021.

[32] A. S. Desuky, M. A. Cifci, S. Kausar, S. Hussain, and L. M. El Bakrawy, "Mud Ring Algorithm: A new meta-heuristic optimization algorithm for solving mathematical and engineering challenges," *IEEE Access*, vol. 10, pp. 50448–50466, 2022.

[33] A. Mahajan, V. Singh, R. Srivastav, S. Kapoor, and E. Singh, "Classification of emotions using a 2-channel convolution neural network," in *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, 2021, pp. 1–7.

[34] H. Yadav and A. Thakkar, "NOA-LSTM: An efficient LSTM cell architecture for time series forecasting," *Expert Systems with Applications*, vol. 238, p. 122333, 2024.

[35] Y. Shen, C. Shepherd, C. M. Ahmed, S. Yu, and T. Li, "Comparative DQN-improved algorithms for stochastic games-based automated edge intelligence-enabled IoT malware spread-suppression strategies," *IEEE Internet of Things Journal*, 2024.

[36] M. Hromada, D. Rehak, B. Skobiej, and M. Bajer, "Converged security and information management system as a tool for smart city infrastructure resilience assessment," *Smart Cities*, vol. 6, no. 5, pp. 2221–2244, 2023.

[37] J. Ghadermazi, A. Shah, and S. Jajodia, "A Machine Learning and Optimization Framework for Efficient Alert Management in a Cybersecurity Operations Center," *Digital Threats: Research and Practice*, 2024.

[38] H. Zahid, S. Hina, M. F. Hayat, and G. A. Shah, "Agentless approach for security information and event management in industrial IoT," *Electronics*, vol. 12, no. 8, p. 1831, 2023.

[39] L. Coppolino, L. Sgaglione, S. D'Antonio, M. Magliulo, L. Romano, and R. Pacelli, "Risk assessment driven use of advanced SIEM technology for cyber protection of critical e-health processes," *SN Computer Science*, vol. 3, pp. 1–13, 2022.

DOI: https://doi.org/10.54216/FPA.180122
Received: July 22, 2024 Revised: October 25, 2024 Accepted: January 08, 2025