STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **14**, September 2025, pp 0–15.
Published online in International Academic Press (www.IAPress.org)

Hybrid Deep Learning Technique for Cybersecurity Detection and Classification

Akhila Reddy Yadulla1, Bhargavi Konda, Mounica Yenugula, Vinay Kumar Kasula*, Chaitanya Tumma

Department of Information Technology, University of the Cumberlands, KY, USA

Abstract Nowadays, cyber threats (CT) evolve rapidly, and this necessitates developing strong and intelligent prediction models that are effective for the detection and classification of cyber security (CS). Hence, a new Elman Crayfish network (ECFN) is proposed to predict and classify CT. In this study, a Kaggle CS threat dataset is trained with Python to develop a more effective classification model. The dataset undergoes a data refinement stage, where noisy data is preprocessed to improve precision. In order to effectively choose the features, a Crayfish Optimization Algorithm is applied in a spatiotemporal feature analysis to select the relevant attributes that contribute to classification. The ECFN utilizes these chosen features to predict CT more effectively. Finally, the detected attacks are classified, and the performance is measured to obtain high accuracy and reliability in detecting CT. The developed method improves CS protection by optimizing the selection process and improving the accuracy of classification. The model's performance is evaluated with metrics like F score, accuracy, recall, precision, and error rate, and the comparison of the results with existing approaches proves its efficiency.

Keywords Cyber threat, cyber security, Crayfish optimization, Elman neural network

DOI: 10.19139/soic-2310-5070-2491

1. Introduction

Cyber security (CS) is a crucial area in the modern digital age, which works towards protecting systems, networks, and information against CA [1]. Organizations, businesses, governments, and countries are all at great risk of CA, which is becoming more common, severe, complex, and diversified [2]. As technology continues to rely more on machines, people and institutions are exposed to various types of security risks, such as data breaches, identity theft, malware, and ransomware [3]. A prediction platform that can forecast risky behavior and attacks before they happen is now imperative. Prevention of damage is hard with existing methods of attack detection, which inform security administrators when an attack occurs [4]. CS employs various techniques, including encryption, firewalls, intrusion detection systems, and security policy, for protection from risks and security of confidential information [5]. As threats to cyberspace evolve, attackers exploit vulnerabilities in software, networks, and human actions in order to access, manipulate, or delete information [6]. CA refers to malicious activities aiming to gain unauthorized access, inflict damage, or interrupt computer systems and infrastructures. There are multiple types of threats: Phishing, denial-of-service (DoS), malware, and advanced persistent threats [7]. Intruders might utilize highly sophisticated methods like social engineering and zero-day exploits to infiltrate systems [8]. With an increase in the frequency of CA, it becomes mandatory to implement intelligent and automated defensive strategies that will effectively detect and respond to the threat [9].

*Correspondence to: Vinay Kumar Kasula (Email:vinaykasula.phd@ieee.org). Department of Information Technology, University of the Cumberlands, KY, USA.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Machine Learning (ML) and Deep Learning (DL) have proved to be powerful tools in CS to identify and neutralize cyber threats [10]. ML involves learning to detect abnormalities and patterns in huge data to provide applications with the ability to identify potential threats through automation [11]. ML-based threat detection models offer the capability of detecting anomalies and classification distinctions. DL is a sub-discipline of ML that employs neural networks (NN) with multiple layers for processing intricate patterns in data [12]. DL models are particularly beneficial in handling large-scale data sets of CA discovering complex threats [13]. The application of ML and DL in CS first enhances threat detection with accuracy by analyzing vast amounts of data, reducing errors. It learns and updates on a continuous basis from evolving threats, increasing the resilience against new forms of attacks [14]. Besides its advantages in identifying unknown or emerging threats by analyzing behavioral anomalies, it also has disadvantages [15]. One of the most significant disadvantages is the high computational cost and resource usage of training deep learning models.

Additionally, the ML models are controllable by malicious actors using adversarial attacks to modify training data [16]. It strongly depends on training data quality [17]. If the training data is inferior or biased, the models will yield inaccurate or unjust results. Also, it cannot fully replace human intelligence and requires security to confirm and interpret alarms [18]. However, CS is imperative in protecting against evolving cyber threats. CA keeps improving in complexity, and therefore, organizations need to implement advanced security measures [19]. While Artificial intelligence (AI) provides great promise for enhancing threat detection and response, these offer much value but also present challenges that have to be implemented judiciously and monitored [20]. There are several bio inspired optimization methods, such as Lion optimization (LO), particle swarm optimization (PSO), grey wolf Optimization (GWO), genetic algorithm (GA), Whale Optimization (WO), and so on that exist for all mathematical applications. Even though, the specific reason for selecting the crayfish optimization is due to the unique fitness function, which is competitive for the best shelter location. In this present research work, this finest behaviour is utilized to find the threat features in the trained data. The competitive best shelter location finding helps to find the threat more accurately than other bio-inspired models. As cyber threats keep evolving, the interfacing of AI-driven solutions is required to create robust and dynamic security frameworks. The key contribution of the work is explained.

- Initially, the cyber-threat dataset was collected from Kaggle and trained in Python.
- Hence, a novel ECFN has been developed as a predictive and classification system.
- Consequently, the noisy elements in the data are filtered, and the Crayfish optimization selects the essential attributes.
- Subsequently, as per the selected attributes, the Crayfish fitness function predicts the CT.
- Finally, the attack is classified by the ECFN, and the performance is evaluated.

The second part of this paper includes current relevant work, while the third part explains the system challenge. The fourth portion develops the system challenge, and the fifth section discusses the case study and performance validation. Finally, the sixth part concludes the work.

2. Related Work

Some recent related papers are described;

Albakri et al. [21] have developed an ML-based CT detection (CTD) and classification method by using a hybrid metaheuristic with blockchain (HMB). The technique employs a hybrid enhanced glow-worm swarm system for choosing features, a hunter-prey for optimum parameter selection, a quasi-recurrent model for CAP detection, and an Ethereum Blockchain for assault detection. When tested, the system's performance demonstrated the highest accuracy of 99.74% but faced challenges in complexity.

Alzubi et al. [22] A Federated learning-based CTD (FLbCTD) system is developed. Black widow optimization is used to hyper-parameter tune and takes characteristics out of binary input images in the MobileNetv2 model. For malware detection and classification, a group of voting-based classifiers is created, along with long short-term memory (LSTM) and gated recurrent unit (GRU) approaches. It performs well but faces challenges in capturing complex behavior.

Shahin et al. [23] have presented a CTD that uses Gradient Boosting in conjunction with Attention LSTM (GBAL) and a fully connected network to identify irregularities. The model's capacity to recognize various attack types is demonstrated by the effective detection of CS risks in seven distinct devices. It has the potential to improve cyber security risk identification. The limitation is it requires a large number of datasets.

Behiry et al. [24] have developed a hybrid K-means clustering with Singular Value Principal Component (KCSVPC) approach. It employs methods such as enhanced K-means clustering information gain for attribute extraction and Singular Value and Principal Component for feature reduction. Three datasets are used to assess the feed-forward NN technique. It exhibits excellent accuracy and dependability but fails to capture temporal dependencies and introduces noise.

Duraibi and Alashjaee [25] study proposes an Enhanced Mayfly (EMF) utilizing a Hybrid DL technique. Data normalization, EMF-based feature selection, the dipper-throated optimization for optimum hyper parameter selection, and LSTM-based Deep Stacked Sequence-to-Sequence Autoencoder model for Identification are all employed. The analysis and comparisons demonstrate the developed technique's superiority over current approaches. It still faces limitations in information loss, impacting the model's ability to detect sophisticated attacks.

Table 1. summary of the literatures

Author	**Methods**	**Advantages**	**Disadvantages**
Albakri et al. [21]	HMB	The model attained high accuracy rate	Increases the complexity
Alzubi et al. [22]	FLbCTD	It performs well in prediction	Face challenges in capturing complex patterns
Shahin et al. [23]	GBAL	It improves CS risk identification	Requires a large amount of data
Behiry et al. [24]	KCSVPC	It exhibits excellent accuracy and dependability	This method fails to capture temporal dependencies and introduces noise.
Duraibi and Alashjaee [25]	EMF	It enhances the prediction superiority	It leads to information loss

The summary of the discussed literature is exposed in table 1, the common drawbacks noted through the reviewed literature are poor feature extraction and prediction accuracy. Hence, to address these issues, the hybrid model is used based on the Elman and the crayfish optimal model. Here, the incorporation of the crayfish optimal features in the Elman network provides the tuned outcome by extracting the features more accurately and affords the finest prediction scores.

3. Problem Statement

In recent days, CTD has evolved as a major concern for CS. Hence, many DL techniques have been developed to solve the issues. However, it faces challenges in overcoming the problems of the current techniques. Some methods, even with high accuracy, are plagued by excessive computational complexity as a result of the incorporation of multiple optimization and selection routines. Such complexity renders its scalability. Also, some models fail to handle complex behavioural patterns, especially in dynamic attack cases, diminishing their ability to detect adaptive CT. Some feature extraction models introduce noise and lose temporal dependencies, affecting the capacity to analyze attack patterns effectively.

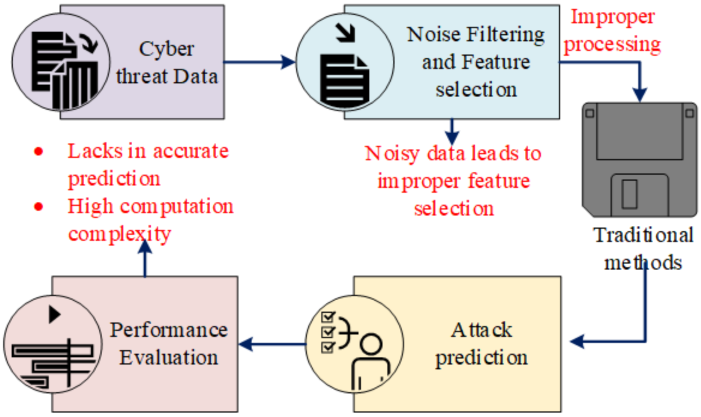

Cyber threat Data
Noise Filtering and Feature selection
Improper processing
Lacks in accurate prediction
High computation complexity
Noisy data leads to improper feature selection
Traditional methods
Performance Evaluation
Attack prediction

Figure 1. System model with problem

Other methods have some major problems associated with information loss during feature selection and reduction steps. Although optimization-based feature selection improves model performance, it removes meaningful information, impacting the Identification of complex and dynamic CT. The Problem statement is displayed in Figure 1.

4. Proposed Methodology

As CT is evolving as a major concern in CS, predicting and classifying it plays a crucial role. Hence, a novel Elman Crayfish network (ECFN) has been developed with prediction and classification features. Initially, the dataset has been collected from the kaggle. Hence, the data has been refined to improve its quality. Here, the noise elements are removed, and this data is used for further processing. Moreover, the spatiotemporal feature analysis is performed to select the necessary attributes using Crayfish optimization. Finally, the CT is predicted and classified. The proposed architecture is displayed in Figure 2. The developed ECFN model is evaluated using some of the criteria, such as F score, accuracy, recall, precision, and error rate, and compared with current approaches.

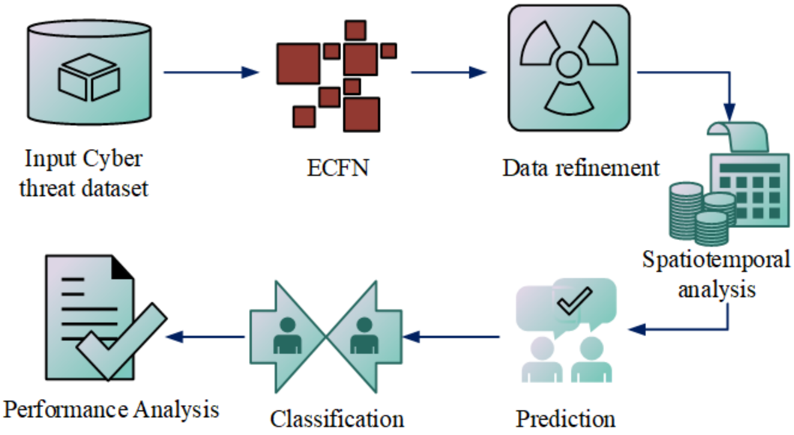


Figure 2. Proposed architecture

4.1. ECFN process

The proposed model is developed by combining Cray fish optimization (CFO) [26] with the Elman neural network. The optimization fine-tunes the network for improving the classification accuracy, layers of the proposed model is exposed in Figure 3 and the hyperparameters variables are defined in table 2.

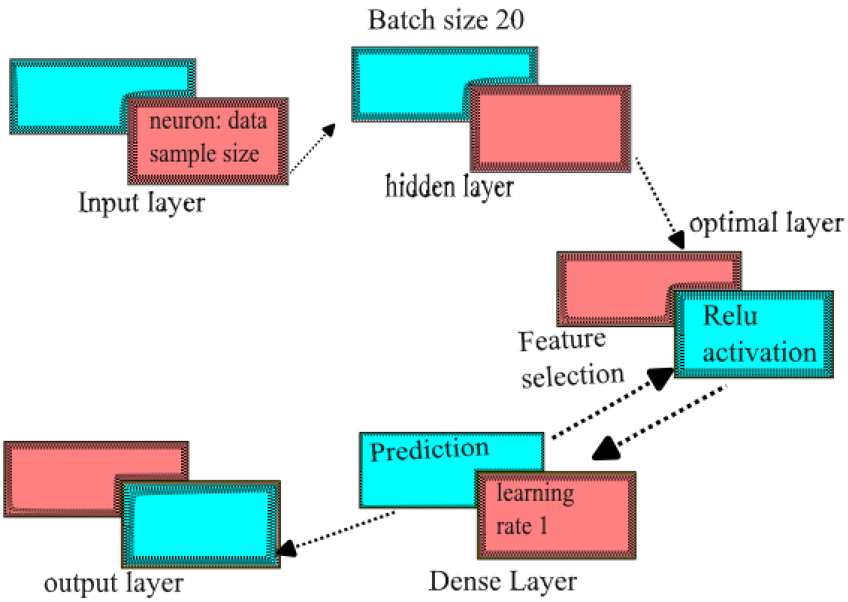


Figure 3. layers of proposed model

Table 2. Hyper-parameter variables

Parameter specification	
Activation function	ReLU
Optimizer	Carry fish
Tuning model	Fine tuning
Learning rate	0.001
Batch size	25
Hidden layers	3
Filters	4

4.1.1. Data refinement Data refinement is an important preprocessing technique within CS to enhance the quality of gathered CT data by removing noise, inconsistency, and unwanted information. Raw data, in most cases, comprises outliers, missing values, and redundant items, which refinement helps remove while keeping important information intact to process it further. At first, data initialization is performed. The data initialization is executed by Eqn. (1).

$$CT_I = CT_1, CT_2, CT_3 ... CT_n \tag{1}$$

The cyber threat dataset is denoted as CT, and the Initialization function is denoted as I. After the initialization process, data refinement is performed. It is done by using the CFO. The data refinement is computed by Eqn. (2)

$$D_R = \frac{\alpha(CT) - \delta * (CT - CT_{noise})}{CT_i} \tag{2}$$

The data refinement variable is denoted as D_R, α denotes the noise reduction function, δ denotes the optimizations step size movement controlling variable, and CT_{noise} denotes the noise. For processing the data refinement in more accurate way, min-max scalar with regularization concept was utilized. It can help in reducing algorithm complexity and the occurrence of overfitting.

4.1.2. Spatiotemporal Analysis Spatiotemporal feature analyses are executed to identify patterns and correlations in cyber data for predicting and classifying. This aids in more effective threat detection and prediction. The CFO improves this process by efficiently discovering the most important spatiotemporal features, enhancing detection accuracy, and reducing computational overhead. CFO behavior best searches for subsets of features by maintaining exploration-exploitation to ensure the model pays attention to important threat indicators. This leads to an adaptive and CT forecast system that is able to respond to changing patterns of attack. Feature analysis is executed in Eqn. (3). Here, the spatiotemporal features were defined by analyzing the traffic patterns. In any smart application, making traffic is the key syndrome for getting affected link, which is termed as harmful. So, it is considered as the critical factor for cyber threat detection. To understand the attack occurrence at the specific time interval, spatiotemporal analysis was performed. In addition, to view the exact attack occurrence time, time window strategy [30] was utilized.

$$STA = \frac{CT(F_i) + \gamma(F_{best} - F_i) + \lambda(F)}{F_i^{t+1}} \tag{3}$$

The spatiotemporal feature analysis is denoted as STA the features denoted as F_i at each iteration i. The exploration function is denoted as γ. The exploitation function is denoted as λ and the best features is defined as F_{best}. Here, the needed features were stored in the carry fish best shelter finding memory, while executing the exploitation function, the needed features were retrieved and extracted by matching the stored features in the carry fish [26]. This best shelter finding operation is given in the Elman network to tune the hyper parameters variables and operations.

4.1.3. Threat Classification Cyber threat prediction and classification are the processes of examining the spatiotemporal features to identify and predict CT. Prediction is the process of determining threats from past threat patterns, whereas classification identifies threats as belonging to specific categories. The threat is predicted by tracing the anomalies by the ECFN. It is computed by Eqn. (4).

$$P = W + b\frac{A_t}{\eta(ST_A)} \tag{4}$$

Here, the prediction variable is denoted as P weights assigned is denoted as W, b denotes the bias term, A_t denotes the anomaly types, and η denotes the tracing variable. The ENN is employed for categorization as it handles temporal dependencies and improves classification accuracy. The classification is executed by Eqn. (5).

$$Qaa22C = \begin{cases} if(P=0)\,Phishing \\ if(P=1)\,DoS \\ if(P=2)\,MitM \\ if(P=3)\,SQLI \\ if(P=4)\,CSS \\ if(P=5)\,Ransomware \\ if(P=6)\,Password \\ if(P=7)\,ZDE \\ if(P=8)\,DDoS \end{cases} \tag{5}$$

The classification variable is denoted as C. The classification is performed for 9 classes ranging from 0 to 8. *DoS* denotes the denial of service, *MitM* denotes the Man in the middle, *SQLI* denotes the Structured Query

Language injection, *CSS* denotes the Cross-site scripting, *ZDE* denotes the Zero-day Exploits, and *DDoS* denotes the Distributed DoS.

```
Start
Step 1: Dataset initialization()
    int $CT_I$;
    // Initializing the dataset by crayfish population initialization function in Eqn. (1)
Step 2: Data refinement()
    int $D_R, \alpha, \delta^*, CT_{noise}$;
    // Initializing the data refining variables
    $DR \rightarrow$ |controlling$(CT)$ - noisy elements|
    // Noisy elements are removed and data is refined
Step 3: Spatiotemporal Analysis ()
    int $ST_A, F_i, \gamma, \lambda, F_{\text{best}}$
    // $F_{\text{best}}$ is the best search location of Caryfish in Eqn. (3) [28]
    $ST_A|D_R(\text{essentialattributes}) \rightarrow$ dense layer
    // needed attributes are selected
Step 4: Classification()
    int $P, W, b, A_t, \eta, C$;
    // Initializing the cyber threat prediction functions
    $P \rightarrow$ tracing$(ST_A) \times$ weights $+$ bias
    // Cyber threat is predicted
    if ($P = 0$)
        Phishing
    if ($P = 1$)
        DoS
    if ($P = 2$)
        MitM
    if ($P = 3$)
        SQLI
    if ($P = 4$)
        CSS
    if ($P = 5$)
        Ransom ware
    if ($P = 6$)
        Password
    if ($P = 7$)
        ZDE
    if ($P = 8$)
        DDoS
    // Cyber threat is classified
End
```

The algorithm for the work is provided in a pseudo-code format, and the entire workflow is displayed sequentially in Figure 4.

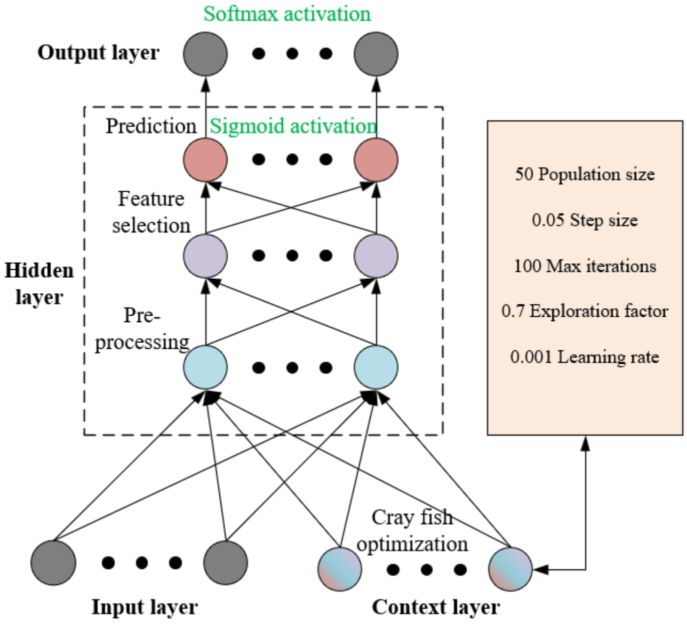


Figure 4. ECFN Flowchart

5. Results and Discussion

The Python environment on Windows 10 is used to verify the ECFN. First, the CS Threat datasets are collected from Kaggle. The proposed framework includes data collection and removing noisy elements, identifying informative attributes, and detecting and classifying threats. The parameter specification used for implementing the proposed framework is described in Table 3.

Table 3. Operation specification

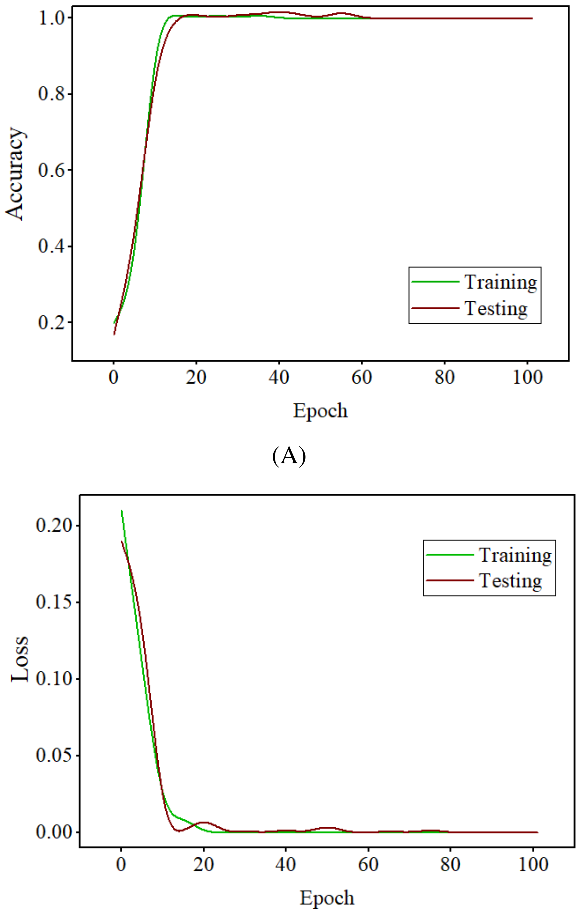
Metrics	Parameters
Program	Python
Version	3.7.14
Operating System	Windows 10
Network	Elman neural network
Optimization	Crayfish
Dataset	Cyber security threat

5.1. Case study

The dataset named as cyber threat for new malware data dataset has been collected from the Kaggle website (https://www.kaggle.com/datasets/zunxhisamniea/cyber-threat-data-for-new-malware-attacks). Then, the proposed ECFN is developed. The collected dataset has been divided into 70% for training and 30% for testing. The data divided for testing is used to analyze the performance of the model in Identification. The description of the dataset is given in Table 4. In addition, while processing the class imbalance data, the stratified sampling [29] method was utilized for making different subclasses.

Table 4. Dataset description

Attack types	**Total samples** **100%=750**	**Training** **70%=525**	**Testing** **30%=225**
Phishing	96	67	29
DoS	96	67	29
Man-in-the-Middle	90	63	27
SQL Injection	90	63	27
Cross-Site Scripting	90	63	27
Ransomware	90	63	27
Password Attacks	90	63	27
Zero-Day Exploits	90	63	27
DDoS	18	13	5

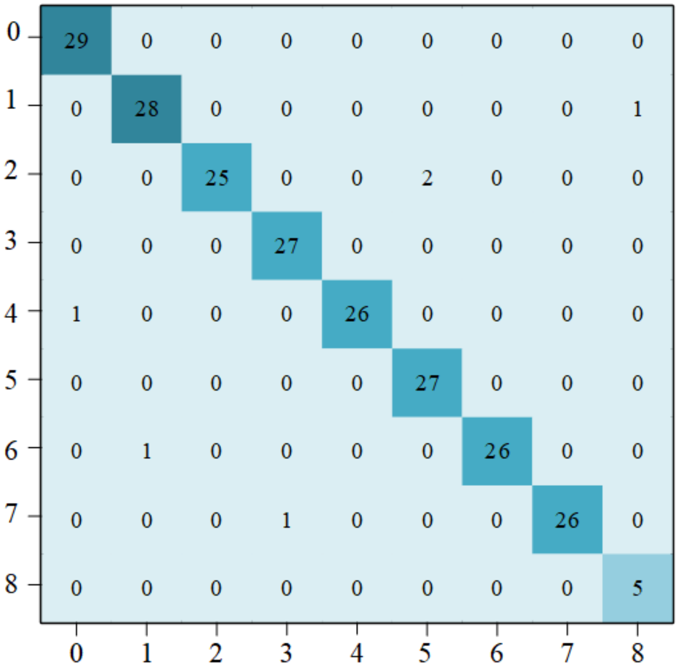


(A)

(B)

Figure 5. (A) Accuracy graph and (B) Loss graph

The accuracy graph and loss graph for the developed model are displayed in Figure 5 (A) and (B), respectively. In the prediction of CT, the accuracy and loss curves are key measures of model performance. The accuracy curve is a measure of how accurately the model identifies threats correctly over training iterations, generally trending upward as the model learns patterns out of the dataset. The loss graph measures the difference between expected and actual results. A smooth drop in the loss graph indicates effective learning. For CT forecasting, a balance between accuracy and loss is important for guaranteeing trustable detection, which is extremely critical in CS.



	0	1	2	3	4	5	6	7	8
0	29	0	0	0	0	0	0	0	0
1	0	28	0	0	0	0	0	0	1
2	0	0	25	0	0	2	0	0	0
3	0	0	0	27	0	0	0	0	0
4	1	0	0	0	26	0	0	0	0
5	0	0	0	0	0	27	0	0	0
6	0	1	0	0	0	0	26	0	0
7	0	0	0	1	0	0	0	26	0
8	0	0	0	0	0	0	0	0	5

Figure 6. Confusion matrix

The confusion matrix for the developed framework is displayed in Figure 6. In this confusion matrix, the miss classification value is 1, thus it can have scored the finest prediction outcome. Here, the model makes 9 classifications ranging from 0 to 8. 0, which denotes phishing. 1 denotes DoS, 2 denotes MitM, 3 denotes SQLI, 4 denotes CSS, 5 denotes ransomware, 6 denotes Password, 7 denotes ZDE, and 8 denotes DDoS. The ECFN framework makes accurate predictions with a few misclassifications.

5.2. *Performance Analysis*

To evaluate the performance of the developed framework, the results are computed with metrics such as F score, Accuracy, Recall, Precision, and error rate, and compared with a few current DL approaches such as Recurrent NN (RNN), Artificial NN (ANN), GRU, and LSTM [27]

5.2.1. F Score and Accuracy The f score measures classification accuracy as it measures recall and precision. It balances both metrics. However, accuracy is a measure of the model's predictive effectiveness. Eqn computes the F score and accuracy. (6) and (7), respectively.

$$\text{F score} = 2 \times \frac{X \times Y}{X + Y} \tag{6}$$

$$\text{Accuracy} = \frac{\mathrm{T_C P + NT_C P}}{\mathrm{T_C P + NT_{CP} + T_{ICP} + NT_{ICP}}} \tag{7}$$

Here, X denotes the precision and Y denotes the recall. T_{CP} denotes the threat correctly predicted, NT_{CP} denotes Non-threat CP, T_{ICP} denotes the threat incorrectly predicted, and NT_{ICP} denotes Non-threat ICP.

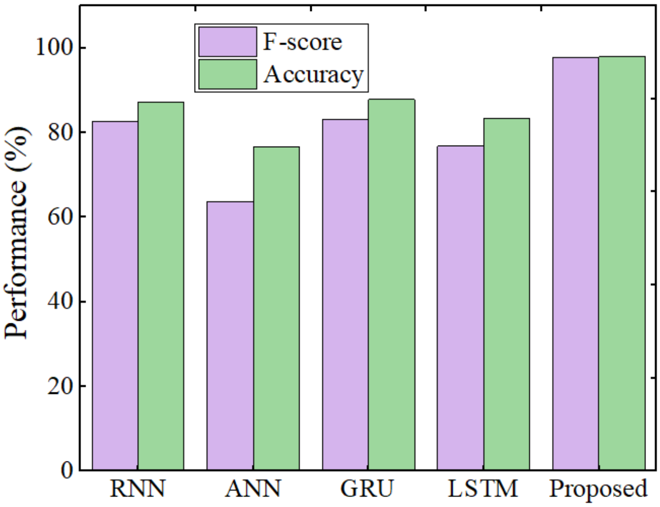


Figure 7. F score and Accuracy Comparison

The comparison results are shown in Figure 7. The existing RNN attained F score of 82.57% and an accuracy of 87.147%, ANN attained F score of 63.602% and an accuracy of 76.594%, the GRU attained F score of 83.175% and an accuracy of 87.806%, and LSTM attained F score of 76.773% and accuracy of 83.418%. The developed model overcomes this by achieving an F score of 97.67% and an accuracy of 97.89%.

5.2.2. Recall and Precision Recall quantifies the percentage of CT that the model correctly labels. It aims to minimize false negatives. Precision is the proportion of true positives (correctly predicted cyber threats) to all instances labeled as threats. Reducing false positives establishes the model's CTD accuracy. Eqn. computes the Recall and Precision in (8) and (9), respectively.

$$\text{Recall} = \frac{\mathrm{T_{CP}}}{\mathrm{T_{CP}} + \mathrm{T_{ICP}}} \tag{8}$$

$$\text{Precision} = \frac{\mathrm{T_{CP}}}{\mathrm{T_{CP}} + \mathrm{NT_{ICP}}} \tag{9}$$

The recall and precision results are evaluated, and a comparison is provided in Figure 8. The current RNN attained a recall of 94.066% and precision of 77.572%, the ANN achieved a recall of 89.682% and precision of 61.794%, the GRU attained a recall of 93.476% and a precision of 78.463%, and the LSTM attained a recall of 92.699% and precision of 75.511%. Moreover, the developed model achieved 97.81% recall and 98.01% precision.

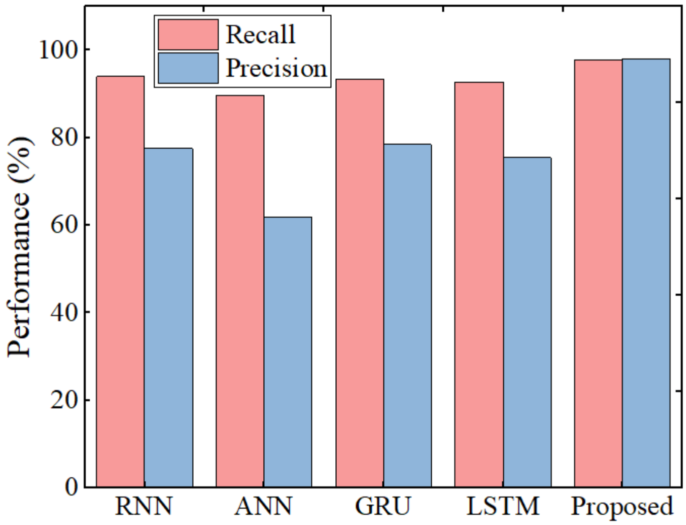


Figure 8. Recall and Precision Comparison

5.2.3. Error rate Error rate refers to the rate of incorrect predictions made by a developed model in the detection of CT. It measures how frequently the model gets an instance wrongly classified. The error rate is computed by Eqn. (10).

$$\text{Error rate} = \frac{\mathrm{T}_{\mathrm{ICP}} + \mathrm{NT}_{\mathrm{ICP}}}{\mathrm{T}_{\mathrm{CP}} + \mathrm{NT}_{\mathrm{CP}} + \mathrm{T}_{\mathrm{ICP}} + \mathrm{NT}_{\mathrm{ICP}}} \tag{10}$$

The Error-rate results are evaluated, and a comparison is provided in Figure 9.

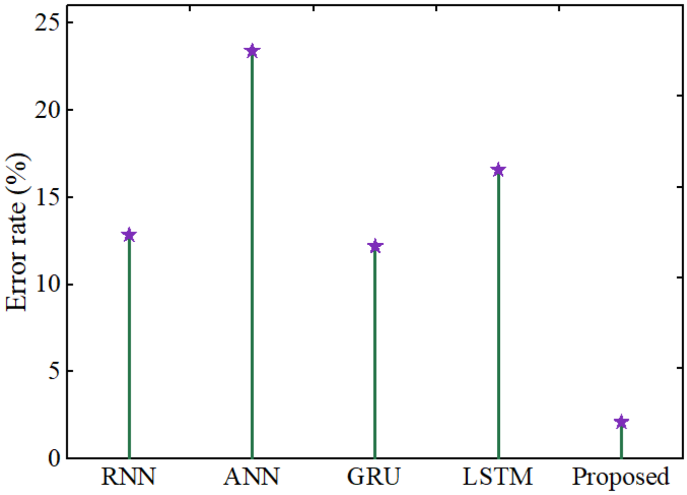


Figure 9. Error Rate Comparison

The error rate for the current RNN was 12.853%, the ANN obtained 23.406%, the GRU obtained 12.194%, the LSTM obtained 16.582%, and the developed ECFN obtained 2.11%, which is comparatively very less. The comparison of the proposed framework with the current approaches is depicted in Table 5.

Table 5. Entire comparison

	F score	Accuracy	Recall	Precision	Error rate	Computational time (ms)	Resource/ memory usage (%)	Training time (ms)	Scalability (%)
RNN	82.57	87.147	94.066	77.572	12.853	89	68	234	62
ANN	63.602	76.594	89.682	61.794	23.406	42	45	183	73
GRU	83.175	87.806	93.476	78.463	12.194	73	76	165	70
LSTM	76.773	83.418	92.699	75.511	16.582	56	39	134	84
Proposed	97.67	97.89	97.81	98.01	2.11	26	23	60	97

5.3. Ethical considerations and limitations

The proposed ECFN model demonstrates a better performance. The hybrid of Crayfish optimization with the Elman neural network tunes the networks and improves the prediction accuracy by removing noise and selecting the necessary attributes. In addition, by changing and setting the required features in the caryfish algorithm memory, it is suitable for real-time large-scale applications like Kaspersky cloud edge data to check and justify the working performance of the proposed algorithm the real-time large network data is adopted from DATA SOURCES — Kaspersky Cyberthreat live map and testing was made. It contains a data poisoning attack and an adversarial attack. The outcome of those processes is mentioned in Table 6. Here, a few different hybridizations were considered, and the comparison was made in both the simulation and real-time environments. Here, the overfitting issues are not raised due to the proper implementation of regularization in the preprocessing layer. All the compared models were executed in the same proposed platform, and the comparison was performed with each other.

Table 6. ECFN Performance in Kaggle data and Real-time data

Methods	CICIDS					Real-time data					
	F score	Accuracy	Recall	Precision	Error rate	F score	Accuracy	Recall	Precision	Error rate	P-value
GWO-LSTM	83	83.1	83	83	16.9	63	63.1	63	63	36.9	0.008
PSO-ANN	78.2	76	77	79.4	24	68.2	68	67	68.4	24	0.03
GA-Elman	92.3	92.3	92.3	92.3	7.7	82.3	80	84.6	—	20	0.01
WO-Elman	92.8	93	91	92.4	7	78.9	78	78.9	—	21.1	0.07
CNN-LSTM	82.3	86	82	82.6	14	65.7	65	64	66.9	35	0.08
LO-Deep belief	77.2	77.2	77	77.4	12.8	57	57	57	—	43	0.005
Transformer model	92	90	91	93	10	84	84	84	—	16	0.005
Capsule Networks	90.5	90.5	90.5	90.5	9.5	81.3	81.3	81.3	—	18.7	0.009
Elman neural network	79.5	79.5	79.5	79.5	10.5	60.1	60	60.2	—	40	0.05
Caryfish optimization	90.5	83.4	90	91	16.6	70.5	73.4	70	71	26.6	0.09
Proposed	97.67	97.89	97.81	98.01	2.11	95	95	95	—	—	0.003

Hence, the proposed system is suitable for real-time application by fixing the desired features in the crayfish optimization. Hence, the efficiency of the validation algorithm is presented in Table 7. Here, the proposed model has shown a few variations between the software analysis and real-time data that maximize the scalability score and justify the applicability of the proposed model in the real-time domain.

Table 7. Validation algorithm efficiency

Methods	Computational time (ms)	Resource/ memory usage (%)	Training time (ms)	Scalability (%)
GWO-LSTM	49	45	101	67
PSO-ANN	33	67	82	73
GA-Elman	68	54	184	81
WO-Elman	43	89	193	83
Caryfish-LSTM	48	63	209	77
LO-Deep belief	34	39	254	90
Transformer model	51	29	261	56
Federated learning	36	31	99	82
Elman neural network	59	28	133	87
Caryfish optimization	83	43	167	64
Proposed	23	20	60	97

To justify the performance of the carry fish optimization, different bio-inspired models like Lion Optimization (LO), Ant Lion Optimization (ALO), Fruit Fly Optimization (FFO), Owl Optimization (OO), GA, PSO, African Buffalo Optimization (ABO), and Chimp optimization Algorithm (COA) were considered and implement in the same platform and the outcomes were compared with carry fish optimization and proposed model in table 8. Here, the optimization performance was measured by performing the zero-day attack detection from the kaggle zero day attack detection data (Zero-Day Attack Detection in Logistics Networks).

Table 8. Performance of optimization algorithms

Methods	Optimization Performance Assessment			
	Computational time (ms)	Resource/ memory usage (%)	Training time (ms)	Scalability (%)
LO	184	75	254	34
ALO	193	97	261	53
FFO	209	64	276	60
OO	254	51	193	57
GA	261	53	209	60
PSO	101	49	254	59
ABO	174	63	217	62
COA	290	64	222	58
Caryfish optimization	83	42	167	64
Proposed	23	20	60	97

The software and hardware requirements for implementing this proposed model is Python, version 3.10, windows 10 operating system, intel core i5 processor CPU, large database, algorithm for attack detection and security model training. Moreover, the hardware requirements are GPU, cloud-edge device, data transmission service, monitoring IoT gadget for attack detection. The code sources with preprocessing details are available in the following link. GitHub - stephenkung/elman_network: elman network by tensorflow, a undergraduate final project.

Limitation: The limitation that was noted while processing the proposed model in the real-time framework is the variation of attack features (zero-day attacks) and increasing the computational complexity. In real-time, capturing the live attacks features takes more time because of this internal processing model. The reason for preserving more computation time is due to the continuous resource usage scenarios because there is no option for fixing the optimal resources in the validation of real-time live attacks. In the future, implementing the future prediction intelligent concept along with the hybrid deep learning and defensive distillation will be the optimal solution.

6. Conclusions

In this research, an ECFN was suggested for the prediction and classification of CT. With the CS threat dataset from Kaggle, the model training was done using Python to improve detection efficiency. A data refining step was employed to eliminate noisy elements and provide high-quality input for the classification process. The Crayfish Optimization Algorithm was also used for spatiotemporal attribute analysis to determine the most crucial attributes for predicting CT. The Crayfish Fitness Function also demonstrated the ability to predict future CT, resulting in an efficient and accurate detection process. Hence, the ECFN model classifies threats efficiently, and its performance was examined. The model attained an F score of 97.67%, an accuracy of 97.89%, a recall of 97.81%, a precision of 98.01%, and an error rate of 2.11%, which demonstrates its efficiency. Therefore, securing the data is essential for security applications. However, the resource usage is not estimated for the real-time live cyber threat data, due to the dynamic variation of threat features and actions. In the future, implementing the future prediction bio-inspired model along with a hybrid deep network will provide the best and optimized resource prediction outcome.

REFERENCES

1. O. Abuelamayem, *A Deep Inverse Weibull Network*, *Statistics, Optimization & Information Computing*, vol. 13, no. 4, pp. 1357–1367, 2024.
2. M. A. Amleh and I. F. Al-Freihat, *Prediction of New Lifetimes of a Step-Stress Test Using Cumulative Exposure Model with Censored Gompertz Data*, *Statistics, Optimization & Information Computing*, vol. 13, no. 4, pp. 1368–1387, 2024.
3. T. Fadziso, U. R. Thaduri, S. Dekkati, V. K. Ballamudi, and H. Desamsetti, *Evolution of the Cyber Security Threat: An Overview of the Scale of Cyber Threat*, *Digitalization & Sustainability Review*, vol. 3, no. 1, pp. 1–2, 2023.
4. M. F. Safitra, M. Lubis, and H. Fakhrurroja, *Counterattacking Cyber Threats: A Framework for the Future of Cybersecurity*, *Sustainability*, vol. 15, no. 18, p. 13369, 2023.
5. D. P. Möller, *Intrusion Detection and Prevention*, in *Guide to Cybersecurity in Digital Transformation: Trends, Methods, Technologies, Applications and Best Practices*, Cham: Springer Nature Switzerland, pp. 131–179, 2023.
6. Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, *A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions*, *Electronics*, vol. 12, no. 6, p. 1333, 2023.
7. S. Kumar, M. Dwivedi, M. Kumar, and S. S. Gill, *A Comprehensive Review of Vulnerabilities and AI-Enabled Defense Against DDoS Attacks for Securing Cloud Services*, *Computer Science Review*, vol. 53, p. 100661, 2024.
8. J. M. Couretas, *Cyber Security and Defense for Analysis and Targeting*, in *An Introduction to Cyber Analysis and Targeting*, Cham: Springer International Publishing, pp. 119–150, 2022.
9. I. H. Sarker, *Machine Learning for Intelligent Data Analysis and Automation in Cybersecurity: Current and Future Prospects*, *Annals of Data Science*, vol. 10, no. 6, pp. 1473–1498, 2023.
10. M. Ozkan-Okay, E. Akin, Ö. Aslan, S. Kosunalp, T. Iliev, I. Stoyanov, and I. Beloev, *A Comprehensive Survey: Evaluating the Efficiency of Artificial Intelligence and Machine Learning Techniques on Cyber Security Solutions*, *IEEE Access*, vol. 12, pp. 12229–12256, 2024.
11. Z. Azam, M. M. Islam, and M. N. Huda, *Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree*, *IEEE Access*, vol. 11, pp. 80348–80391, 2023.
12. S. U. Qureshi, J. He, S. Tunio, N. Zhu, A. Nazir, A. Wajahat, F. Ullah, and A. Wadud, *Systematic Review of Deep Learning Solutions for Malware Detection and Forensic Analysis in IoT*, *Journal of King Saud University – Computer and Information Sciences*, vol. 27, p. 102164, 2024.
13. H. Dong and I. Kotenko, *Cybersecurity in the AI Era: Analyzing the Impact of Machine Learning on Intrusion Detection*, *Knowledge and Information Systems*, vol. 19, pp. 1–52, 2025.
14. T. Mazhar, H. M. Irfan, S. Khan, I. Haq, I. Ullah, M. Iqbal, and H. Hamam, *Analysis of Cyber Security Attacks and Its Solutions for the Smart Grid Using Machine Learning and Blockchain Methods*, *Future Internet*, vol. 15, no. 2, p. 83, 2023.
15. D. Javaheri, S. Gorgin, J. A. Lee, and M. Masdari, *Fuzzy Logic-Based DDoS Attacks and Network Traffic Anomaly Detection Methods: Classification, Overview, and Future Perspectives*, *Information Sciences*, vol. 626, pp. 315–338, 2023.
16. M. I. Malik, A. Ibrahim, P. Hannay, and L. F. Sikos, *Developing Resilient Cyber-Physical Systems: A Review of State-of-the-Art Malware Detection Approaches, Gaps, and Future Directions*, *Computers*, vol. 12, no. 4, p. 79, 2023.
17. N. Tran, H. Chen, and J. Bhuyan, *Data Creation and Quality Evaluation for Machine Learning-Based Cyber Intrusion Detection*, *IEEE Access*, vol. 10, pp. 121900–121923, 2022.
18. I. H. Sarker, H. Janicke, A. Mohsin, A. Gill, and L. Maglaras, *Explainable AI for Cybersecurity Automation, Intelligence and Trustworthiness in Digital Twin: Methods, Taxonomy, Challenges and Prospects*, *ICT Express*, 2024.
19. A. Rayhan, *Cybersecurity in the Digital Age: Assessing Threats and Strengthening Defenses*, in *Conference: Cybersecurity Awareness*, pp. 1–26, 2024.
20. D. Kavitha and S. Thejas, *AI Enabled Threat Detection: Leveraging Artificial Intelligence for Advanced Security and Cyber Threat Mitigation*, *IEEE Access*, 2024.
21. A. Albakri, B. Alabdullah, and F. Alhayan, *Blockchain-Assisted Machine Learning with Hybrid Metaheuristics Empowered Cyberattack Detection and Classification Model*, *Sustainability*, vol. 15, no. 18, p. 13887, 2023.

22. O. A. Alzubi, I. Qiqieh, and J. A. Alzubi, *Fusion of Deep Learning-Based Cyberattack Detection and Classification Model for Intelligent Systems*, *Cluster Computing*, vol. 26, no. 2, pp. 1363–1374, 2023.
23. M. Shahin, F. F. Chen, A. Hosseinzadeh, H. Bouzary, and R. Rashidifar, *A Deep Hybrid Learning Model for Detection of Cyberattacks in Industrial IoT Devices*, *The International Journal of Advanced Manufacturing Technology*, vol. 123, no. 5, pp. 1973–1983, 2022.
24. M. H. Behiry and M. Aly, *Cyberattacks Detection in Wireless Sensor Networks Using a Hybrid Feature Reduction Technique with AI and Machine Learning Methods*, *Journal of Big Data*, vol. 11, no. 1, p. 16, 2024.
25. S. Duraibi and A. M. Alashjaee, *Enhancing Cyberattacks Detection Using Dimensionality Reduction with Hybrid Deep Learning on Internet of Things Environment*, *IEEE Access*, 2024.
26. Y. Zhang, P. Liu, and Y. Li, *Implementation of an Enhanced Crayfish Optimization Algorithm*, *Biomimetic*, vol. 9, no. 6, p. 341, 2024.
27. O. A. Alkhudaydi, M. Krichen, and A. D. Alghamdi, *A Deep Learning Methodology for Predicting Cybersecurity Attacks on the Internet of Things*, *Information*, vol. 14, no. 10, p. 550, 2023.
28. H. Jia, X. Zhou, J. Zhang, L. Abualigah, A. R. Yildiz, and A. G. Hussien, *Modified Crayfish Optimization Algorithm for Solving Multiple Engineering Application Problems*, *Artificial Intelligence Review*, vol. 57, no. 5, p. 127, 2024.
29. Z. Wu, Z. Wang, J. Chen, H. You, M. Yan, and L. Wang, *Stratified Random Sampling for Neural Network Test Input Selection*, *Information and Software Technology*, vol. 165, p. 107331, 2024.
30. H. Bei, H. Lin, F. Yang, X. Li, R. Murcio, and T. Yang, *Optimization on Multimodal Network Considering Time Window Under Uncertain Demand*, *IEEE Transactions on Intelligent Transportation Systems*, 2025.