[1]Arshiya S. Ansari

[1] Hanadi Saad Al Harbi

# Cyber Threat Prediction Model for Transferring Data in Wireless Edge Computing Platforms Using Deep Learning

Journal of Electrical Systems

***Abstract:*** - Wireless edge computing or WEC which is becoming more popular to transmit computation-intensive projects while improving end users' quality of service. Edge devices may access cloud services and functionality via WEC. However, owing to the current spike in attack activity, the expansion of the Internet of Things (IoT) aims poses serious cyber security issues. Hardware and deep learning solutions are being developed to detect cyber-attacks, data dumping and traffic situations in edge networks. Deep models can outperform their shallow counterparts in terms of learning due to the massive amounts of data generated by IoT devices the application of DL models finds assistance in several fields, with the WEC serving as the approach's clear benefactor for traffic forecasting and attack detection. To address these issues, this study offers a unique DL-based traffic forecast approach that combines a cyber-attack prediction strategy with an information transmission mechanism. The suggested approach consists of three basic procedures, intrusion detection, data unloading and traffic forecasting. In this research, we concentrate on intrusion detection. Initially, we preprocessed the data using the Standard Scalar technique. Principal Component Analysis (PCA) is utilized in feature extraction to minimize dimensionality while preserving important information. Less relevant characteristics are progressively removed for feature selection using recursive feature elimination (RFE). Finally, we proposed a novel Rectified Hyperbolic Long Short-Term Layered ConvoNeuronet (RH-LSTM-CNN) approach for intrusion detection. The suggested strategy outperforms the current approaches with accuracy (98.33%), recall (98%), precision (98%), and f1 score (98%).

***Keywords:*** Cyber security deep learning, Neural Network, Computing, Intrusion Detection, Rectified Hyperbolic Long Short-Term Layered ConvoNeuronet (RH-LSTM-CNN), Attack and Detection.

## 1 Introduction

Enhancing the performance of Wireless Edge Computing (WEC) platforms technologies that integrate computing capabilities at the system's edge to provide computational capacity near the data source requires the efficient and secure transmission of data [Zhou, Jadoon and Shuja (2021)]. In the ever-changing realm of wireless edge computing, when gadgets produce immense quantities of data instantaneously, the capacity to transmit this data becomes utmost importance. The growth of Internet of Things (IoT) gadgets, along with the widespread use of cell phone technology has accelerated the demand for decentralized computing solutions like the WEC platform [Zhou, Liu Q and Zeng (2020)]. These platforms facilitate real-time processing, minimizing reliance on centralized servers in the cloud and enabling quick decision-making at the connection's edge, but the success of these platforms relies on the effective transfer of information among gadgets and the computing architecture [Kumar Dhabliya, Agarwal et al. (2022)]. To harness the capabilities of WEC, it is crucial to address the issues that arise as data quantities increase. A significant challenge in transferring information in WEC is the wide variety of network-connected devices. These devices differ in their physical capabilities but in their communication methods and data formats [Zhang, Xia, Liu et al. (2021)]. To achieve a harmonious integration of diverse gadgets, it is necessary to develop new methods that can effectively connect these devices and facilitate an effortless and standardized movement of information. Standardization processes, such as "Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP)," have emerged as crucial enablers, providing efficient and light-weight communication techniques tailored for resource-constrained devices found in edge environments [Awotunde, Chakraborty, and Adeniyi (2021)]. Furthermore, the protection of data during transmission is a vital issue that should not be underestimated. The inherent wireless characteristics of edge computing expose vulnerabilities, making data subject to detection and illegal access [Balasubramaniam, Vijesh Joe, Sivakumar et al. (2023)].

Data integrity and privacy must be protected throughout transmission via encrypted techniques, robust authentication procedures and reliable communication protocols. Developments in these fields are essential to bolstering data transfer procedures and ensuring the security and unalter ability of sensitive data. The design of WEC platforms is crucial in determining the dynamics of information transport. Fog computing, a derivative of computing at the edge, includes intermediate points termed fog nodes that enhance the efficiency of data processing and transmission. These nodes serve as intermediaries among devices at edge and cloud infrastructure, facilitating efficient data transport and minimizing latency [Li, Zhao, Tao et al. (2022)]. Using this hierarchical approach to computation results in a more orderly and efficient process, which in turn improves the WEC's overall efficacy by optimizing data flow. A surge in the usage of edge caching and Content Delivery Networks (CDNs)

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

*Corresponding Author: Arshiya S. Ansari. Email: ar.ansari@mu.edu.sa

to improve data transmission efficiency has coincided with an increase in the demand for WEC systems [Esmaeili , Goki, Masjidi et al. (2022)]. These solutions reduce the amount of time that recurring data transmissions from distant servers are required by pre-caching frequently utilized data close to the edge devices. Edge caching optimizes system efficiency by reducing delay and usage of bandwidth, ensuring that important data is quickly accessible for immediate processing [Hatamian, Tavakoli, Moradkhani. (2021)]. The constrained bandwidth in wireless networks may lead to reduced data transfer speeds, impeding the smooth transmission of substantial amounts of information between edge devices and centralized computer resources. Moreover, being susceptible to interference can lead to the loss of packets and delays, which might impact the dependability of data transmissions. In addition, there may be difficulties regarding privacy and security when confidential information is transmitted over wireless connections [Li, Zhu, Chu et al. (2020)]. This requires the use of strong authentication and encryption methods to reduce possible risks. The objective of this research is to improve the security and resilience of WEC environments by implementing a cyber-threat prediction model, specifically the RH-LSTM-CNN to analyze patterns and anomalies in data transfers.

### *1.1 Contribution of this study*

- This study introduces a novel DL-based traffic forecasting method for WEC that combines information transfer with cyber-attack prediction. The objective is to improve cybersecurity in WEC through traffic forecasting, data unloading, and intrusion detection.
- The study uses advanced data preprocessing to increase the accuracy of intrusion detection. It preprocesses data using the Standard Scalar technique, extracts features using PCA, and applies RFE for feature selection to reduce dimensionality while maintaining important information by gradually eliminating less significant features.
- The study proposes novel neural network architecture called Rectified Hyperbolic Long Short-Term Layered Convo Neuronet (RH-LSTM-CNN) for intrusion detection. This innovative architecture outperforms existing approaches, achieving high levels of accuracy, recall, f1 score, and precision in detecting cyber-attacks in edge networks.

The remainder of this study is divided into the following sections: Part 2, discusses the related works and the approaches are discussed in part 3. The experiment's findings are presented in part 4, and lastly, the study is concluded in part 5.

## 2 Related works

The study [Diro and Chilamkurti (2018)] presented a deep learning (DL)-based distributed attack detection method for the IoT. The technology showed effectiveness in identifying various cyberthreats on IoT devices, offering an effective method to improve cybersecurity in IoT settings. The research [Gopalakrishnan, Ruby, Al-Turjman et al. (2020)] investigated a DL-based approach that includes cyberattack detection for data offloading in mobile edge computing. They offered a novel strategy, demonstrating that the DL improved productivity and security in mobile edge computing systems. [Gumaei, Al-Rakhami, Hassan et al. (2021)] presented a system that integrated a blockchain with a “deep recurrent neural network (DRNN)” and edge-based computing for 5G-capable drone recognition and airplane mode recognition. The experimental findings demonstrated that the performance of the DRNN model outperformed then-current leading research on identifying drone and flight modes. [Alkhudaydi, Krichen, Alghamdi. (2023)] presented a DL method for anticipating IoT cybersecurity attacks. They utilized innovative DL techniques to show the predictive skills could enhance IoT security, with the focused-on attack prevention and prediction.

Security architecture for “Multi-Access Edge Computing (MEC)” infrastructure based on “Software-Defined Networking and Network Function Virtualization (SDNFV)” was proposed by the author [Krishnan, Duttagupta, Achuthan. (2019)]. They focused on threat monitoring and offered an approach for using SDNFV to improve MEC security. The study [Lv, Chen, Lou et al. (2021)] investigated the application of ML to intelligent edge computing in smart cities. They provided insights into the future of smart city technology and focused on including ML algorithms to increase intelligence and efficiency.

[Al-Taleb and Saqib (2022)] proposed a hybrid ML model as a novel method for intelligent cyber threat detection in smart cities. They aimed to improve urban cybersecurity measures by utilizing novel techniques for threat identification and mitigation. The research [Grover, Alladi, Chamola et al. (2021)] investigated the integration of edge computing and DL to create a secure multitier network for the “Internet of Vehicles (IoV).” They offered a significant contribution to the sector by concentrating on improving IoV security using innovative technologies. A DL-inspired “Intrusion Detection System (IDS)” designed for “Edge Computing Environments (ECE)” was presented in the paper [Aldaej, Ahanger, Ullah. (2023)]. The method described in sensors demonstrates enhanced security protocols and adaptability in edge computing environments by utilizing innovative approaches.The study [Zhang, Lu, Li. (2021)] investigated methods for utilizing the relationship between edge-based technologies and the technologies that provide secrecy to safeguard private company data. The study presented a conceptual framework and technical structure that focused on the utilization of secure data technology. Finally, the detection of anomalous data was fundamentally precise. The article [Alotaibi, Ahmed, Kamel. (2023)] recommended the implementation of automated processes and a self-customized "Intelligent Automation Detection Model (IADM)"

in an IoT based smart city's distributed computing infrastructure, to recognize and halt any detrimental linked system activity and reaction. Based on the findings, the RF, K-NN, and AB models demonstrated superior accuracy compared with other algorithms.

[Assiri and Ragab. (2023)] offered the best DL method for identifying cyberattacks in an IoT setting with blockchain technology. They examined practical ways to improve cyber security in IoT ecosystems focused-on DL and blockchain integration for reliable threat detection. They provided insightful information about the blockchain, IoT and cybersecurity were evolved. [Dutta, Choraśm, Pawlicki et al. (2020)] presented a combined method that utilized DL models, including the "Deep Neural Network (DNN)" and LSTM, along with an advanced classifier, such as "logistic regression", practicing the principle of stacking simplification. The suggested methodology employed two stages to improve the capabilities of detecting anomalies in networks. [Samy, Yu , Zhang. (2020)] presented a robust, distributed system that used DL to identify cybersecurity risks related to the IoT. The system, which used threat sensors on fog nodes, demonstrated high detection rates as well as efficacy in response time and precision, recognizing a variety of cyber-attacks in several categories. The article [Arif, Ajesh, Shamsudheen et al. (2022)] developed a robust and power-saving computational offloading strategy employing LSTM. The task forecasting method was used to offload computing on portable devices and the edge cloud schedule scheme helped to optimize the edge computing offload model by moving tasks. Ensuring that every component allocated to a particular task preserved private during the whole process improved the security of LSTM.

## 3 Methodology

This section presents the intrusion detection dataset, preprocessing, feature extraction, and feature selection. We also discuss RH-LSTM-CNN for cyber threat prediction. Figure 1 shows the flowchart for methodology.

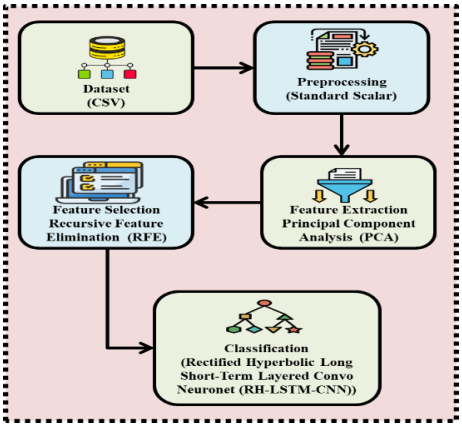


**Figure 1:** Flowchart for methodology

### *3.1 Dataset*

We gathered the intrusion detection dataset in "Comma-Separated Values (CSV)" format from Kaggle (intrusion detection (kaggle.com)).It is a popular file format for tabular data storage, every line of the file represents a row, and values are separated by commas or other delimiters inside each row. CSV-formatted intrusion detection datasets refer to data that is arranged tabularly, with each row of the table detailing a particular event or observation about network activity and possible security concerns. The table's columns stand for various characteristics or qualities connected to each occurrence. Figure 2 shows the sample dataset.

| | duration | protocol_type | service | flag | src_bytes | dst bytes | land | wrong_fragment | urgent | hot | ... | dst_host_srv_count | dst_host_same_srv_rate | dst_host_diff_srv_rate | dst_host_same_src_port_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | udp | 12 | 2 | 36 | 0 | 0 | 0 | 0 | 0 | - | 171 | 1.00 | 0.00 | |
| **1** | 0 | icmp | 25 | 4 | 0 | 0 | 0 | 0 | 0 | | - | 44 | 0.17 | 0.05 | 0 |
| **2** | 0 | icmp | 48 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | - | 20 | 0.08 | 0.06 | 0 |
| **3** | 0 | icmp | 50 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | - | 27 | 0.11 | 0.07 | 0 |
| **4** | 0 | icmp | 50 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | - | 1 | 0.01 | 0.64 | 0 |
| | - | ... | ... | | - | | | - | | | | - | - | - | |
| **5995** | 0 | icmp | 20 | 2 | 6932 | 0 | 0 | 0 | 0 | 0 | | 46 | 0.15 | 0.06 | 0 |
| **5996** | 0 | icmp | 7 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | 18 | 0.07 | 0.06 | 0 |
| **5997** | 0 | icmp | 55 | 2 | 980 | 326 | 0 | 0 | 0 | 0 | | 153 | 0.53 | 0.10 | 0 |
| **5998** | 0 | icmp | 25 | 2 | 222 | 397 | 0 | 0 | 0 | 0 | - | 110 | 1.00 | 0.00 | 0 |
| **5999** | 0 | icmp | 25 | 2 | 330 | 6299 | 0 | 0 | 0 | 0 | | 255 | 1.00 | 0.00 | 0 |

6000 rows x 42 Columns

**Figure 2:** Sample dataset

### *3.2 Data preprocessing using standard scalar approach*

Standard Scalar preprocessing scales and normalizes IDS data to improve the accuracy of cyber threat prediction. A preprocessing method called Standard Scalar is used in machine learning to standardize a dataset's characteristics. Rescaling the features to have the characteristics of a standard normal distribution with a mean ($\boldsymbol{\mu}$) of **0** and a standard deviation ($\boldsymbol{\sigma}$) of **1** is the process of standardization. For each feature, this is accomplished by removing the mean and dividing the result by the standard deviation. Eq. (1) can be used to standardize its scaling as given a feature $\boldsymbol{Y}$

$$z = \frac{Y-\mu}{\sigma} \quad (1)$$

Where $z$ is the standardized value of $Y$, $\mu$ is the mean of the feature's values, $Y$is the original value of the feature and $\sigma$ is the deviation of the feature's values.

This procedure is automated for every feature in a dataset using the standard scalar included in scikit-learn or related tools. Each feature in the training set has its mean and standard deviation determined, these numbers are used to transform the training and test sets. For each feature in the modified data, the mean will be equal to zero and the standard deviation will be one. Figure 3 depicts the result of data preprocessing

| duration | protocol_type | service | flag | src_bytes | dst_bytes | land | wrong_fragment | urgent | hot | ... | dst_host_srv_count | dst_host_same_srv_rate | dst_host_diff_srv_rate 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 0.121569 | 2 | - 1.218905 | - 0.516243 | - 0.039395 | - 0.140927 | - 0.022366 | - 0.091615 | 0.0 | - 0.091195 | *** | 0.519570 | 1.079356 | - 0.447489 |
| 1 - 0.121569 | 0 | - 0.432085 | 1.231757 | - 0.039517 | - 0.140927 | - 0.022366 | - 0.091615 | 0.0 | - 0.091195 | | - 0.632389 | - 0.772249 | - 0.188345 |
| 2 - 0.121569 | 0 | 0.959981 | 1.231757 | - 0.039517 | - 0.140927 | - 0.022366 | - 0.091615 | 0.0 | - 0.091195 | | - 0.850082 | - 0.973026 | - 0.136517 |
| 3 - 0.121569 | 0 | 1.081030 | - 1.390243 | - 0.039517 | - 0.140927 | - 0.022366 | - 0.091615 | 0.0 | - 0.091195 | | - 0.786588 | - 0.906100 | - 0.084688 |
| 4 - 0.121569 | 0 | 1.081030 | 2.105757 | - 0.039517 | - 0.140927 | - 0.022366 | - 0.091615 | 0.0 | - 0.091195 | ... | - 1.022422 | - 1.129185 | 2.869546 |

5 rows X 42 columns

**Figure 3:** The result of data preprocessing

### *3.3 Extracting feature using principal component analysis (PCA)*

PCA enhances cyber threat prediction by reducing data dimensionality, identifying important features and raising anomaly detection accuracy. The primary goal of PCA was to create a transformed linear sequence for real data with related features. As a result, a collection of innovative information with few characteristics was added to the "Principal Component Load (PCL)" matrix to reflect the real data. It is appropriate for the process of dimensionality reduction for multidimensional data.

Create the observational matrix $U$ concerning the actual data. After $N$ observations, the $k$variables $\theta_1, \theta_2, \dots, \theta_k$ determine the matrix $U$. The sample information from the dataset is numerically estimated in every row, and the number $n$ in the column denotes the number of samples that were in Eq. (2).

$$U = \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1h} \\ U_{21} & U_{22} & \cdots & U_{2h} \\ \vdots & \vdots & \ddots & \vdots \\ U_{r1} & U_{r2} & \cdots & U_{rh} \end{bmatrix} \quad (2)$$

For the observation matrix, data processing should be centralized. Also, compute the sample mean (Eq. (3)).

$$\overline{U_x} = \frac{1}{n}\sum_{y=1}^{r} y_{yx} \quad (3)$$

Moreover, Eq. (4) defines the standard deviation $T_x$ for the dataset in the WEC platforms. It measures the dispersion of data points around the mean $\overline{u_x}$ . Here n is the number of data points. $u_{yx}$ Represents individual data points and $u_x$ is their mean

$$T_x = \sqrt{\frac{1}{n}\left(u_{yx} - \overline{u_x}\right)^2} \quad (4)$$

Using Eq. (5),

$$\widetilde{u_{yx}} = \frac{u_{yx}-\overline{u_x}}{T_x}(y = 1,2, \dots, r, x = 1,2, \dots, h) \quad (5)$$

By performing the centralized data processing, the standardized matrix $\overline{u_x}$ is created. Calculate a sample correlation matrix based on Eq. (6).

$$4 \quad K = \frac{1}{n}\widetilde{U}^S\widetilde{U} \quad (6)$$

Eq. (7) is used to calculate each component of $H$.

$$k_{yx} = \frac{\sum_{l=1}^{r}(u_{yx}-\overline{u_x})(u_{yx}-\overline{u_x})}{\sqrt{\sum_{l=1}^{r}(u_{yx}-\overline{u_x})(u_{yx}-\overline{u_x})^2 \sum_{l=1}^{r}(u_{yx}-\overline{u_x})(u_{yx}-\overline{u_x})^2}} \quad (7)$$

The eigenvalue and eigenvector of $H$. Determine the $k$ feature values of $H$, which are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \dots \geq 0$. Consequently, calculating the involvement rate for each key component using Eq. (8).

$$q_y = \frac{\lambda_y}{\lambda_1+\lambda_2 \ldots +\lambda_h}(y = 1,2, \ldots, h) \quad (8)$$

Choose the highest q major component that achieves 90%, yielding the PCA results of $\lambda_{q=1}$. The eigenvectors $f_1, f_2, \ldots, f_o$ are calculated using the values for the descending order features $1,2, \ldots, k$. Choose the PCL's top $q$ features vector (Eq. (9)).

$$W_{h\times o} = (f_1, f_2, \ldots, f_o) \quad (9)$$

Using the PCL matrix $W^S_{h\times o}$ and Eq. (10), generate new primary variables $k_1, k_2, \ldots, k_q$ by changing the real data.

$$\begin{bmatrix} k_1 \\ \vdots \\ k_o \end{bmatrix} = K^S_{h\times o} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_o \end{bmatrix} \quad (10)$$

The matrix's dimensions shifted from $\boldsymbol{ktop}$ following the linear change, considerably identifying the volume of the information received. Figure 4 displays the outcome of extracting feature.

| | PCA1 | PCA2 | PCA3 | PCA4 | PCA5 | Target |
|---|---|---|---|---|---|---|
| 0 | 2.821581 | -0.685037 | -0.325219 | -2.270078 | 0.958324 | 1 |
| 1 | -3.442154 | -1.412826 | -0.017742 | 0.023688 | 0.315944 | 0 |
| 2 | -4.347374 | -1.070552 | 0.024193 | -0.117273 | -0.212242 | 0 |
| 3 | -0.513409 | 5.802153 | -0.007896 | 0.912308 | -0.592352 | 0 |
| 4 | -0.451528 | 5.307523 | 0.060586 | -0.511676 | 1.554621 | 2 |

**Figure 4:** The result of extracting feature

### *3.4 Feature selection using recursive feature elimination (RFE)*

RFE improves cyber threat prediction by optimizing model performance and efficiency through iteratively deleting less relevant information. RFE is a feature selection method that seeks to find the most significant features in a dataset by eliminating the less significant features. When the target number of features is attained, the model is trained on the subset of features and the least significant elements removed. The model provides a rating that is used to evaluate the significance of attributes.

- Select a model: First, select an algorithm for learning that rates the significance of various features. While RFE applies to a wide range of techniques, it is commonly selected for linear models.
- Training Models: Determine each feature's relevance score by training the selected model across the full dataset.
- Ranking the features: Rank the features according to the important scores that are received. Features that have lower significance scores are less significant.
- Eliminate the least significant element: Remove from the dataset the feature that has the lowest relevance score.

Training Models and Eliminate the least significant element steps should be repeated until the required number of features is obtained. Eq. (11) represents the RFE.

$$\mathrm{RFE}(\mathrm{X}, \mathrm{y}, \mathrm{M}, \mathrm{k}) \quad (11)$$

Where **X** is the original feature matrix, **y** is the target variable, **M** is the ranking feature, and **k** is the desired number of features to be selected the result of feature selection is shown in Figure 5.

Selected Features: ['PCA1', 'PCA2', 'PCA5']

| | PCA1 | PCA2 | PCA5 | Target |
|---|---|---|---|---|
| 0 | 2.821581 | -0.685037 | 0.958324 | 1 |
| 1 | -3.442154 | -1.412826 | 0.315944 | 0 |
| 2 | -4.347374 | -1.070552 | -0.212242 | 0 |
| 3 | -0.513409 | 5.802153 | -0.592352 | 0 |
| 4 | -0.451528 | 5.307523 | 1.554621 | 2 |

**Figure 5:** The result of feature selection

### *3.5 Rectified hyperbolic long short-term layered convo neuronet (RH-LSTM-CNN) for classification*

The RH-LSTM-CNN hybrid model leverages local pattern capture and long-term dependency modeling in sequential data to integrate Conv1D and LSTM layers for efficient cyber threat prediction.

*3.5.1 One-dimensional convolutional neural network( CONV1D)*

The Conv1D model improves threat predictions and strengthens cybersecurity defenses by detecting cyber threats utilizing sequential data patterns. In this study to detect network intrusions using a Conv1D with the layers of the Conv1Dare allocated to pooling, dropout, activation functions, and Conv1D. Hyper-parameters including the amount of CNN layers, the quantity of neurons present in each stratum, the filter's size, and configuring the Conv1D involves setting the subsampling factor for each layer.

The basic operation of a convolution layer is the application of a filter to an input. A feature map that reveals qualities connected to the process of filtering is repeated to obtain the data points. Convolution is a linear procedure that involves multiplying inputs by a certain set of weights. In this case, the kernel multiplies the inputs, which is a 1D array of weights. This process yields an ideal value after every execution and the total result is a set of values together to constitute a feature map.

After the feature map calculation, each value is subjected to the "Rectified Linear Unit (ReLU)" activation procedure. ReLU is a linear activation function that maintains the original input, if the input is negative, it sets to zero. The model's performance is enhanced, learning from training data is accelerated and the vanishing gradient problem is successfully resolved by utilizing the ReLU activation function. Eq. (12) shows the procedure.

$$R(Z) = \max(0, Z) \tag{12}$$

Here the input given to the activation function is indicated by $Z$ and its positive output is denoted by $R(Z)$.

In CNN, pooling layers are frequently used after convolution layers. To reduce the risk of overfitting, sub-sampling is employed to decrease reliance on exact feature map placement in the model. Dependent on complexity and parameter count, the architecture is computed. Pooling filters are selected based on their size, stride and type (e.g. max pooling or average pooling). This results in mapped pooled features. While average pooling determines the average value, max pooling allocates the maximum feature value to each patch.

Dropout layers are introduced as a regularization approach in deep-learning neural networks to mitigate the danger of overfitting. This technique sets inputs to 0 at each step with a predetermined frequency rate during training, hence arbitrarily ignoring certain neurons during processing. Altering the active inputs reduces the possibility of overfitting and Eq. (13) ensures that the increased active inputs have a constant sum.

$$Z = \frac{1}{1-rate} \tag{13}$$

As such, the addition of noise and the burdening of specific nodes cause disruptions to the training process. Specifically, dropout is employed in training and contributes to the network's increased weight, necessitating scaling to the desired dropout rate. The findings are then sent to the output by applying thick layers that fully link to the layer that came before them and incorporate an activation function.

*3.5.2 Long short-term memory network (LSTM)*

Apply LSTM to predict cyber threats and take proactive security steps by utilizing data's sequential patterns. Long-term dependency in RNNs was a problem that utilized to solve. Due to the typical RNN's Backpropagation through Time (BPTT) training method, which results in the vanishing/exploding gradient issue, learning from long sequences can be challenging. Like an LSTM cell, a gated cell was used instead of the RNN cell to be fixed. Figure 6 displays the LSTM cell state.

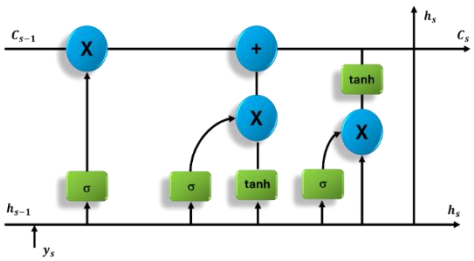


**Figure 6.** LSTM cell state

These gates control information exist, and data is not required to be kept in memory. The LSTM's cell has memory, enabling to recall prior actions. The cell state was the crucial component of LSTMs. Three gates can be used by LSTM to either add or delete information from the cell state. A sigmoid layer judges' information from the cell state to discard at the starting gate, which is a forgotten gate (Eq. (14)).

$$f_s = \sigma\left(W_f \cdot [h_{s-1}, y_s] + a_f\right) \tag{14}$$

The second gate was an input gate that described in Eq.s (15and 16), combines a "tanh" layer to create a vector of newly updated values and a sigmoid layer to determine values that will be updated.

$$i_s = \sigma\left(W_f \cdot [h_{s-1}, y_s] + a_i\right) \tag{15}$$

$$\widetilde{C_s} = \tanh\left(W_f \cdot [h_{s-1}, y_s] + a_c\right) \tag{16}$$

Eq.s (14-16) were used to update the cell state (Eq. (17)) after that.

$$C_s = f_s * C_{s-1} + i_s * \widetilde{C_s} \tag{17}$$

The output of the current state will be calculated using Eq.s (18 and 19), along with the modified cell state and the sigmoid layer that determines the portions of the cell state which can be the final output.

$$o_s = \sigma(W_f \cdot [h_{s-1}, y_s] + a_o) \quad (18)$$

$$h_s = o_s * \tanh(C_s) \quad (19)$$

Data was compressed into the range $(0,1)$ using the $\sigma$ activation function into the range of $(-1,1)$ using the hyperbolic tangent function $(tanh)$. The input vector was $y_s$, the previous hidden state $h_{s-1}$, and the mass matrices $W_f$, $W_i$, $W_c$ and $W_o$were used. Additionally, the procedure makes use of the bias vectors$a_f$, $a_i$, $a_c$, and $a_o$.

The novel approach combines the layers of Conv1D and LSTM, combining the previous layer's capacity to recognize local patterns with the latter's skill in understanding long-term dependencies in sequential data. By incorporating the "$tanh$" and "$ReLU$" activation functions, the model becomes nonlinear and may identify complex connections in the data. As a regularization strategy, a dropout layer is included to avoid overfitting. By combining the best aspects of both convolutional and recurrent architectures, this fusion produces a hybrid model that makes sequence modeling and feature learning. Table 1 displays the overview of the suggested model.

**Table 1:** Overview of the suggested model

| Layer | Output Shape | No. of Parameters |
|---|---|---|
| Conv1D | $X\_train.shape\ [1] - 2,256$ | 1024 |
| LSTM | 256 | 196,608 |
| Dense Layer (with 512 units) | 512 | 131,584 |
| Dropout | 512 | 0 |
| Dense Layer (with 1 unit) | 1 | 513 |

## 4. Experimental results

### *4.1 Experimental setup*

The experimental setup utilized a system that had 8 GB of RAM, running on 64-bit version of Windows 10. "Jupiter and Anaconda" served as the programming software, and the system had 100 GB of available storage on the C drive.

### *4.2 Evaluation criteria*

Figure 7 shows the confusion matrix for the proposed method. Among the total of25 negative incidents, 25 were correctly predicted, while one positive instance was missed. Comparably, 34 of the positive class instances were accurately predicted and none of the negative ones were missed.

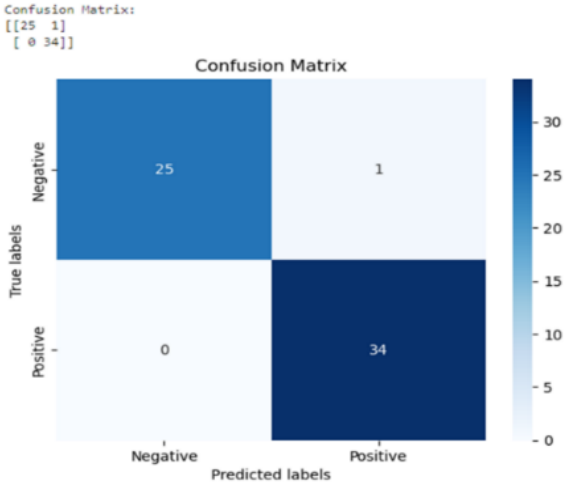


**Figure 7: Confusion matrix for the recommended approach**

In this section, we compared the performance of the suggested method with the existing methods such as "Random Forest (RF), Decision Tree (DT)" [Mighan and Kahani. (2021)], and "Histogram based Gradient Boosting (HBGB)" [Seth, Chahal, Singh et al. (2021)]. The assessment involved measuring various metrics, including accuracy, precision, recall, and f1-score.

#### *4.2.1 Accuracy*

Accuracy in cyber threat prediction refers to a system that predicts a threat about its real occurrence. It calculates the proportion of all forecasts that identify threats. Increased accuracy decreases false negative and positives, which enhances the system's capability to recognize and react to potential threats. For cybersecurity to be effective, reliability is necessary. Accuracy can be defined using the following Eq. (20).

$$Accuracy = \frac{TP+TN}{Totalsamples} \quad (20)$$

- True Positives, or TPs are the cases that were accurately identified as threats.
- True Negatives, or TNs are instances that were accurately identified as non-threats.
- False Positives, or FPs, are instances that are incorrectly identified as threats.

- False Negatives, or FNs, are instances that were incorrectly as non-threats.

Figure 8 and Table 2 illustrate the values of accuracy. Compared to existing techniques RF – 75.2%, HBGB – 97.5%, DT – 75.3%, our proposed method was superior RH-LSTM-CNN -98.33%. In comparison to existing methods, the suggested method RH-LSTM-CNN showed significant improvements in the prediction of cyber threats.

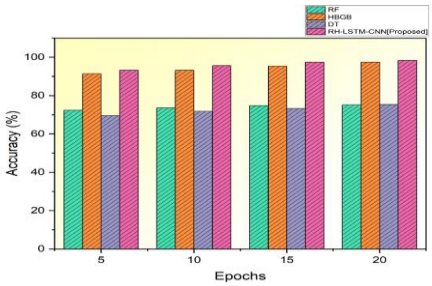


**Figure 8.** Outcome of accuracy

**Table 2:** Result of Accuracy

| Epochs | Accuracy (%) | | | |
|---|---|---|---|---|
| | RF | HBGB | DT | RH-LSTM-CNN[Proposed] |
| 5 | 72.4 | 91.4 | 69.5 | 93.3 |
| 10 | 73.6 | 93.2 | 71.8 | 95.6 |
| 15 | 74.8 | 95.4 | 73.2 | 97.5 |
| 20 | 75.2 | 97.5 | 75.3 | 98.33 |

*4.2.2 Precision*

The precision of cyber threat prediction refers to the percentage of detected threats among all forecasted threats, measured by the accuracy of positive forecasts. Reduced false positives due to increased precision emphasize the system's dependability in detecting real cyber threats, which is essential for efficient cybersecurity actions. To determine the precision with this Eq. (21).

$$Precision = \frac{TP}{TP+FP} \tag{21}$$

Figure 9 and Table 3 demonstrate precision value. Compared to existing methods RF – 70.7%, DT – 73.02%, our proposed method was superior RH-LSTM-CNN -98%. In comparison to existing methods, the suggested method RH-LSTM-CNN showed significant improvements in the prediction of cyber threats.

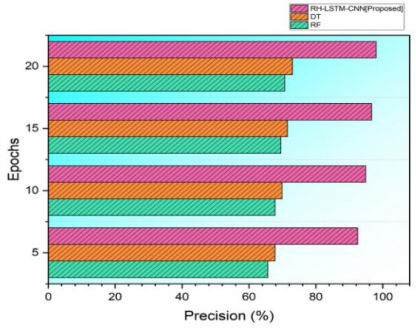


**Figure 9**: Outcome of precision

**Table 3:** Result of precision

| Epochs | Precision (%) | | |
|---|---|---|---|
| | RF | DT | RH-LSTM-CNN[Proposed] |
| 5 | 65.6 | 67.8 | 92.5 |
| 10 | 67.8 | 69.9 | 94.9 |
| 15 | 69.5 | 71.5 | 96.7 |
| 20 | 70.7 | 73.02 | 98 |

*4.2.3 Recall*

Recall, the ability to recognize every significant threat among the true hazards which is measured in the prediction of cyber threats. It assesses the system that detects suspected malicious activity and ensures an approach to detection. When there are fewer unidentified dangers, a strong recall reduces the possibility of detecting potential security breaches. Calculate the recall using this Eq. (22).

$$Recall = \frac{TP}{TP+FN} \tag{22}$$

Figure 10 and Table4 display the values of recall. In comparison to the existing techniques RF – 75.32%, HBGB– 96.7%, DT – 75.26%, our proposed method was superior RH-LSTM-CNN -98%. In comparison to existing methods, the suggested method RH-LSTM-CNN showed significant improvements in the prediction of cyber threats.

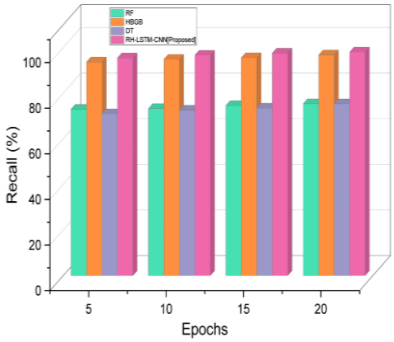


**Figure 10:** Outcome of recall

**Table 4:** Result of Recall

| Epochs | Recall (%) | | | |
|---|---|---|---|---|
| | RF | HBGB | DT | RH-LSTM-CNN[Proposed] |
| 5 | 72.9 | 93.6 | 70.9 | 95.4 |
| 10 | 73.2 | 94.8 | 72.5 | 96.6 |
| 15 | 74.5 | 95.5 | 73.4 | 97.5 |
| 20 | 75.32 | 96.7 | 75.26 | 98 |

*4.2.4 F1 score:*

An F1 score is a metric used to balance recall and precision to predict cyberthreats. Precision assesses the accuracy of positive predictions made by the model. The recall measure assesses the extent to encompass the true positives. The F1 score is an evaluation of a cybersecurity model's effectiveness that is generated by combining false positives and negatives into a single result. Compute the f1 score with this Eq. (23).

$$F1 - score = \frac{2*(precision*recall)}{precision+recall} \quad (23)$$

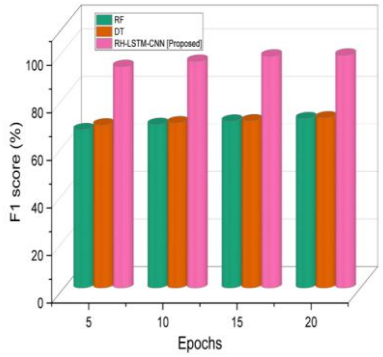


**Figure 11:** Outcome of f1 score

Figure 11 and Table 5illustrate on f1 score value. Compared to existing methods RF – 71.25%, DT– 71.58%, our proposed method was superior RH-LSTM-CNN -98%. In comparison to existing methods, the suggested method RH-LSTM-CNN showed significant improvements in the prediction of cyber threats.

**Table 5:** Result of F1-score

| Epochs | F1 score (%) | | |
|---|---|---|---|
| | RF | DT | RH-LSTM-CNN[Proposed] |
| 5 | 66.6 | 68.6 | 93.2 |
| 10 | 68.8 | 69.5 | 95.4 |
| 15 | 70.2 | 70.3 | 97.6 |
| 20 | 71.25 | 71.58 | 98 |

## 5 Discussion

The specific qualities that support cyber threat projections can be difficult to understand due to interpretability concerns with RF [Mighan and Kahani. (2021)], that could limit effective analysis and decision-making in security-related scenarios. Although HBGB [Seth, Chahal, Singh et al. (2021)] models can be computationally

demanding, which presents difficulties in scenarios involving the real-time prediction of cyber threats. Their resource requirements could restrict their use in specific operational scenarios. Overfitting, which collects noise in the data and produces less accurate predictions, is a risk for DT [Mighan and Kahani. (2021)], in cyber threat scenarios. Their inability to generalize to cases could affect their accuracy. The RH-LSTM-CNN offers superior accuracy in cyber threat prediction, facilitating timely threat identification and efficient pattern recognition by combining rectified hyperbolic tangent activation, LSTM and layered CNN architecture.

## 6 Conclusion

The emergence of Wireless Edge Computing (WEC) offered a useful way to transfer computationally intensive tasks while improving the end user's experience. While edge devices could access cloud services through WEC, the increasing risks connected with the Internet of Things (IoT) made strong cybersecurity measures necessary. The study suggested a novel Deep Learning (DL)-based traffic forecasting method that combined information transfer and cyberattack prediction to overcome these issues. The suggested approach consists of three basic procedures, intrusion detection, data unloading, and traffic forecasting. We gathered the intrusion detection dataset from Kaggle. The study presented a novel intrusion detection method called Rectified Hyperbolic Long Short-Term Layered ConvoNeuronet (RH-LSTM-CNN). This strategy demonstrated superior performance with accuracy (98.33%), recall (98%), precision (98%), and F1 score (98%), showcasing its efficacy in enhancing cybersecurity within the context of WEC and IoT. Resolving dynamic and developing cyber-attacks could present difficulties for RH-LSTM-CNN. Enhancing scalability to large-scale cyber settings and adaptability for threats.

**Acknowledgement:** The author would like to thank the Deanship of Scientific Research at Majmaah University for supporting this work.

**Funding Statement:** This study did not receive any funds from anywhere.

**Author Contributions**: All work is done by all authors equally.

**Availability of Data and Materials:** Not applicable. All references are from Google Scholar.

**Ethics Approval: NA**

**Conflicts of interest:** The author declares that they have no conflicts of interest to report regarding the present study.

## REFERENCE

[1] Zhou S.; Jadoon W.; and Shuja J. (2021): Machine learning-based offloading strategy for lightweight user mobile edge computing tasks. Complexity, pp.1-11.

[2] Zhou C.; Liu Q.; and Zeng R. (2020): Novel defense schemes for artificial intelligence deployed in edge computing environment. Wireless Communications and Mobile Computing, pp.1-20.

[3] Kumar A.; Dhabliya D.; Agarwal P.; Aneja N.; Dadheech P. et al. (2022): Cyber-internet security framework to conquer energy-related attacks on the internet of things with machine learning techniques. Computational intelligence and neuroscience.

[4] Zhang R.; Xia H.; Liu C.; Jiang R.B.; Cheng X.G. (2021): Anti-Attack Scheme for Edge Devices Based on Deep Reinforcement Learning. Wireless Communications and Mobile Computing, pp.1-9.

[5] Awotunde J.B.; Chakraborty C.; and Adeniyi A.E. (2021): Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. Wireless communications and mobile computing, pp.1-17, 2021

[6] Balasubramaniam S.; Vijesh Joe C.; Sivakumar T.A.; Prasanth A.; Satheesh Kumar K. et al. (2023): Optimization Enabled Deep Learning-Based DDoS Attack Detection in Cloud Computing. International Journal of Intelligent Systems.

[7] Li T.; Zhao H.; Tao Y.; Huang D.; Yang C.; Xu S. (2022): Power intelligent terminal intrusion detection based on deep learning and cloud computing. Computational Intelligence and Neuroscience.

[8] Esmaeili M.; Goki S.H.; Masjidi B.H.K.; Sameh M.; Gharagozlou, H. et al. (2022): Ml-ddosnet: IoT intrusion detection based on denial-of-service attacks using machine learning methods and nsl-kdd. Wireless Communications and Mobile Computing.

[9] Hatamian A.; Tavakoli M.B.; Moradkhani M. (2021): Improving the security and confidentiality in the internet of medical things based on edge computing using clustering .Computational Intelligence and Neuroscience.

[10] Li X.; Zhu L.; Chu X.; Fu H. (2020): Edge computing-enabled wireless sensor networks for multiple data collection tasks in smart agriculture. Journal of Sensors, pp.1-9. 2020

[11] Diro A.A.; Chilamkurti N. (2018): Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, vol. 82, pp.761-768.

[12] Gopalakrishnan T.; Ruby D.; Al-Turjman F.; Gupta D.; Pustokhina I.V. et al. (2020): Deep learning enabled data offloading with cyber-attack detection model in mobile edge computing systems. IEEE Access, vol.8, pp.185938-185949.

[13] Gumaei A.; Al-Rakhami M.; Hassan M.M.; Pace P.; Alai G. et al. (2021): Deep learning and blockchain with edge computing for 5G-enabled drone identification and flight mode detection. Ieee Network, vol.35, no.1, pp.94-100.

[14] Alkhudaydi O.A.; Krichen M.; Alghamdi A.D. (2023): A Deep Learning Methodology for Predicting Cybersecurity Attacks on the Internet of Things. Information, vol.14, no.10, p.550.

[15] Krishnan P.; Duttagupta S.; Achuthan K. (2019): SDNFV based threat monitoring and security framework for multi-access edge computing infrastructure. Mobile Networks and Applications, vol.24, pp.1896-1923.

[16] Lv Z.; Chen D.; Lou R.; Wang Q. (2021): Intelligent edge computing based on machine learning for smart city. Future Generation Computer Systems, vol.115, pp.90-99.

[17] Al-Taleb N.; Saqib N.A. (2022): Towards a hybrid machine learning model for intelligent cyber threat identification in smart city environments. Applied Sciences, vol.12, no.4, p.1863.
[18] Grover H.; Alladi T.; Chamola V.; Singh D.; Choo K.K.R. et al. (2021): Edge computing and deep learning enabled secure multitier network for internet of vehicles. IEEE Internet of Things Journal, vol.8, no.19, pp.14787-14796.
[19] Aldaej A.; Ahanger T.A.; Ullah I. (2023): Deep Learning-Inspired IoT-IDS Mechanism for Edge Computing Environments. Sensors, vol.23, no.24, p.9869.
[20] Zhang X.; Lu J.; Li D.(2021): Confidential information protection method of commercial information physical system based on edge computing. Neural Computing and Applications, vol.33, pp.897-907.
[21] Alotaibi N.S.; Ahmed H.I.; Kamel S.O.M. (2023): Dynamic Adaptation Attack Detection Model for a Distributed Multi-Access Edge Computing Smart City. Sensors, vol.23, no.16, p.7135.
[22] Assiri F.Y.; Ragab M.(2023): Optimal Deep-Learning-Based Cyberattack Detection in a Blockchain-Assisted IoT Environment. Mathematics, vol.11, no.19, p.4080.
[23] Dutta V.; Choraś M.; Pawlicki M.; Kozik R. (2020): A deep learning ensemble for network anomaly and cyber-attack detection. Sensors, vol.20, no.16, p.4583.
[24] Samy A.; Yu H.; Zhang H. (2020): Fog-based attack detection framework for internet of things using deep learning. IEEE Access, vol.8, pp.74571-74585.
[25] Arif M.; Ajesh F.; Shamsudheen S.; Shahzad M. (2022): Secure and Energy-Efficient Computational Offloading Using LSTM in Mobile Edge Computing. Security and Communication Networks, pp.1-13.
[26] Mighan S.N.; Kahani M. (2021): A novel scalable intrusion detection system based on deep learning. International Journal of Information Security, vol.20, pp.387-403.
[27] Seth S.; Chahal K.K.; Singh G. (2021): A novel ensemble framework for an intelligent intrusion detection system. IEEE Access, vol.9, pp.138451-138467.
[28] Satish Kumar Alaria, Manish Kumar Mukhija, and Pooja Singh "A Security Approach to Manage a Smart City's Image Data on Cloud," AI-Centric Smart City Ecosystems: Technologies, Design and Implementation (1st ed.), PP: 68-82, (2022). CRC Press. https://doi.org/10.1201/9781003252542.
[29] Alaria, S. K. "A.. Raj, V. Sharma, and V. Kumar."Simulation and Analysis of Hand Gesture Recognition for Indian Sign Language Using CNN"." International Journal on Recent and Innovation Trends in Computing and Communication 10, no. 4 (2022): 10-14.
[30] Satish Kumar Alaria, Prakash Dangi and Pratiksha Mishra. Design and Comparison of LEACH and Improved Centralized LEACH in Wireless Sensor Network. IJRITCC 2021, 9, 34-39.
[31] Satish Kumar Alaria and Abha Jadaun. "Design and Performance Assessment of Light Weight Data Security System for Secure Data Transmission in IoT", Journal of Network Security, 2021, Vol-9, Issue-1, PP: 29-41.
[32] Pratiksha Mishra Satish Kumar Alaria. "Design & Performance Assessment of Energy Efficient Routing Protocol Using Improved LEACH", International Journal of Wireless Network Security, 2021, Vol-7, Issue-1, PP: 17-33.
[33] Ashish Raj, Vijay Kumar, S. K. A. V. S. Design Simulation and Assessment of Prediction of Mortality in Intensive Care Unit Using Intelligent Algorithms. MSEA 2022, 71, 355-367.