ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/amaresan-venkatesan/291

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science

https://urfpublishers.com/journal/artificial-intelligence

Vol: 1 & Iss: 2

Research Article

Error Rates, Timestamp, and Flow Duration Analysis-Based Attack Pattern Identification Framework Using EAE-DBSCAN

Amaresan Venkatesan*

Citation: Venkatesan A. Error Rates, Timestamp, and Flow Duration Analysis-Based Attack Pattern Identification Framework Using EAE-DBSCAN. *J Artif Intell Mach Learn & Data Sci 2023*, 1(2), 1273-1280. DOI: doi.org/10.51219/JAIMLD/amaresan-venkatesan/291

Received: 02 April, 2023; **Accepted:** 18 April, 2023; **Published:** 20 April, 2023

***Corresponding author:** Amaresan Venkatesan, USA, E-mail: v.amaresan@gmail.com

ABSTRACT

For ensuring Network Security (NS) against malicious activities, Dedicated Link Aggregation (DLA) in Computer Network Traffic (CNT) optimizes data transmission with increased bandwidth and reliability. Nevertheless, the traditional works failed to identify the Attack Patterns (APs) centred on timestamps, Error Rates (ERs), and flow duration, thereby resulting in inefficiencies in threat detection in NS. Thus, this paper proposes Ensemble Adaptive Entropy Density-Based Spatial Clustering of Applications with Noise (EAE-DBSCAN) and MeDecay Heuristic-based Radial Basis Function Networks (MDH-RBFN) techniques to identify patterns and classify the normal and malicious traffic, respectively. Primarily, the data is pre-processed, followed by DLA utilizing EAE-DBSCAN and feature extraction. After that, by using EAE-DBSCAN, the patterns are identified from the extracted features for enhanced network performance. Subsequently, utilizing MDH-RBFN, the data is categorized as normal and malicious traffic with a Mean Absolute Error (MAE) of 0.0025. Here, the malicious traffic is blocked, whereas the non-attacked data is encrypted. Thereafter, the traffic level is predicted for non-attacked traffic data as low, medium, high, very high, and extreme. At last, the required loads are balanced for storing data in the cloud.

Keyword: Reed-Solomon Quantum turbo Codes Cryptography (RSQ2C), Generalized Bell-IIFuzzy Inference System (GBIIFIS), Weighted Round Robin with Overflow Handling (WR2QH), Computer Network Traffic (CNT), Time Series Analysis, Pattern Identification (PI), and Dedicated Link Aggregation (DLA).

1. Introduction

To increase bandwidth, enhance reliability, and improve load balancing across high-demand environments, multiple link connections are integrated into a single logical connection by DLA in CNT (Abbasi, et al. 2021) (Choi, et al. 2021). Traffic in networks may arise owing to the multiple link connections (Weerakody, et al., 2021). In this, normal traffic refers to legitimate data (Li, et al. 2021), whereas malicious traffic comprises unauthorized or harmful data packets (Lindemann et al., 2021). Therefore, the malicious traffic must be blocked for secured data transmission.

The malicious traffic data are detected and blocked by the prevailing Machine Learning and Deep Learning methods, namely Decision Trees (Liu, et al. 2021) and Long-Short Term Memory (LSTM) (Drewek-Ossowicka, et al., 2021). But, the scalability and real-time responsiveness (Ensafi, et al. 2022, Barrera-Animas et al., 2022) are impacted by challenges like high computational demands and data requirements (Ruiz, et al. 2021). In addition, the conventional works failed to identify the APs centered on timestamps, ERs, and flow duration, thus leading to inefficiencies in threat detection. Thus, in the proposed work, EAE-DBSCAN and MDH-RBFN techniques are leveraged to efficiently identify and block malicious traffic.

1.1 Problem Statement

The limitations in traditional works are explained as follows,

» None of the prevailing works identified the APs centred on timestamps, ERs, and flow duration, thereby causing inefficiencies in threat detection.

» The traffic level prediction for non-attacked data was not concentrated on in many works, thus leading to inefficient resource allocation and network congestion.

» Non-aggregation of dedicated links in (Fotiadou, et al. 2021) resulted in suboptimal network performance, bandwidth inefficiencies, and limited scalability.

» The non-attacked data in (Balamurugan, et al. 2022) were stored in the cloud in unencrypted form, which led to potential security breaches.

» Anomaly and Normal traffic were identified utilizing unprocessed data in (Shen et al., 2021), which caused misclassified results.

» The proposed work's objectives are detailed below,

» The proposed work identified the traffic APs centered on timestamps, ERs, and flow duration utilizing EAE-DBSCAN.

» The traffic level prediction is performed using the GBΠFIS technique.

» DLA is carried out using the EAE-DBSCAN algorithm.

» Data encryption is done using RSQ2C to improve NS.

» Data preprocessing is done for enhancing the classification process.

The paper is structured as: The related works are discussed in Section 2, the proposed methodology is described in Section 3, the results and discussion are presented in Section 4, and lastly, the proposed work is concluded in Section 5 with future development.

2. Literature Survey

(Fotiadou, et al. 2021) presented a DL-based approach for threat detection and control of traffic flow in NS. In this, based on patterns that were automatically learned through LSTM, the traffic flows were controlled. Hence, the malicious traffic data was effectively identified by the framework. Yet, misclassifications were caused by the unprocessed data usage, thereby hindering the effectiveness of the DL-based threat detection method.

(Balamurugan et al., 2022) accomplished an Enhanced Deep Reinforcement Learning (EDRL) algorithm to enhance Network Traffic (NT) analysis and prediction. In this framework, the EDRL technique was used to analyze and predict different types of NT, comprising unencrypted and encrypted data traffic. However, during NT analysis, high computational complexities in this framework caused increased latency.

(Shen, et al. 2021) introduced a Decentralized Applications (DApp) fingerprinting approach utilizing Graph Neural Networks (GNNs) to efficiently identify users' visits to specific DApps by analyzing encrypted NT. To preserve multiple-dimensional features in bidirectional client-server interactions, a Traffic Interaction Graph was used as an information-rich representation. However, due to the slow learning process, the framework had issues with adaptability to traffic changes and time efficiency.

(Khan, et al. 2022) established a Bayesian model that automatically analyzed the abnormal traffic flow patterns. In this work, Distributed Denial of Service attacks and Flash Crowds in Wireless Sensor Networks were also distinguished. In addition, high traffic caused by malicious attacks and legitimate spikes in user activity was differentiated in this model. This method failed to encrypt the non-attacked data despite efficient categorization, thus causing security breaches. Therefore, the entire work performance was hindered.

(Dong, 2021) developed a Cost Sensitive Support Vector Machine (CMSVM) to accurately identify application types in internet traffic using network flow level characteristics. Moreover, in this work, the dynamic assignment of weights enhanced the classification performance. Nevertheless, as CMSVM failed to handle highly imbalanced datasets, it attained an increased error rate. Hence, it decreased the accuracy of predicting the traffic flow levels.

3. Proposed Methodology for Pattern Identification and Traffic Level Prediction in Dedicated Link Aggregation

In Figure 1, the structural diagram of the proposed EAE-DBSCAN and MDH-RBFN techniques is shown.

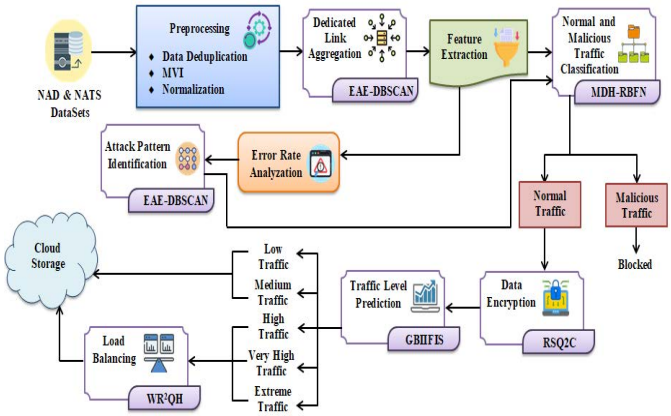


Figure 1: Structural Diagram of the Proposed Work.

3.1. Data Collection

The proposed work begins by collecting the data from two datasets, such as Network Anomaly Detection (NAD) and Network Analytics Time Series (NATS). It is shown as,

$$C^{data} = C^{1}, C^{2}, \ldots\ldots, C^{f} \quad (1)$$

Where, the total number of collected data $\left(C^{data}\right)$is signified as (f).

3.2. Preprocessing

After that, by using Data Deduplication (DD), Missing Value Imputation (MVI), and Normalization techniques, the collected data $\left(C^{data}\right)$ is preprocessed as shown below,

The duplicate copies are removed from $\left(C^{data}\right)$ using deduplicated data size $\left(C^{dedup}\right)$. Hence, the reduced data $\left(R^{data}\right)$ is articulated as,

$$R^{data} = \frac{\left(C^{data} - C^{dedup}\right)}{C^{data}} \quad (2)$$

Further, MVI fills in the missing data for completeness as exhibited below,

$$M^{\forall} \leftarrow \varphi\left(R^{data}\right) \quad (3)$$

Here, the process to fill in the missing values is specified as (φ), and the data after the imputation of missing values is notated as $(M^{\dot{\imath}})$.

Afterward, based on minimum and maximum values, the data is normalized (N^{υ}), which is shown as,

$$N^{\upsilon} = \frac{M^{\dot{\imath}} - M^{\dot{\imath}}_{\min}}{M^{\dot{\imath}}_{\max} - M^{\dot{\imath}}_{\min}} \quad (4)$$

Here, the minimal and maximal values of $(M^{\dot{\imath}})$ are represented as $(M^{\dot{\imath}}_{\min}, M^{\dot{\imath}}_{\max})$, and (D^{P}) is the preprocessed data.

3.3. Dedicated Link Aggregation

Subsequently, the dedicated links are aggregated from (D^{P}) by identifying the dense regions in traffic data using the DBSCAN technique. Nevertheless, due to fixed parameters, DBSCAN struggles with varying density clusters. Hence, Ensemble Adaptive Entropy (EAE) technique, which dynamically adjusts MinPts and Epsilon based on local density estimates and quantifies cluster uncertainty, is used. The algorithmic steps are explained below,

Primarily, the (l) numbers of (D^{P}) are signified as,

$$D^{P} = D^{1}, D^{2}, \ldots\ldots D^{l} \ \ where \ (P = 1 \ to \ \ l) \quad (5)$$

Further, the core points required for grouping are centered on MinPts and Epsilon. Here, the EAE technique with min-max adaptive parameter (ε) is used for Epsilon determination, which is shown as,

$$\Phi = \frac{1}{D^{P}} \left\{ \varepsilon_{\min} + \left(1 - A^{mea} \times \log_{2}\left(D^{P}\right)\right) \right\} \times \left(\varepsilon_{\max} - \varepsilon_{\min}\right) \quad (6)$$

Where, (Φ) and (A^{mea}) signify the optimal epsilon value and the local density-based adaptive measure for (D^{P}), respectively.

After that, the core points $(C^{1}_{Pts}, C^{2}_{Pts})$ are calculated based on MinPts (ϑ) using minimum and maximum of (Φ), which are equated as,

$$C^{1}_{Pts} = \Phi_{\max} \times \vartheta \quad (7)$$

$$C^{2}_{Pts} = \Phi_{\min} \times \vartheta \quad (8)$$

Subsequently, the noise points (that do not come under the boundary of core points) are removed as,

$$P^{noise} \rightarrow \left(< \vartheta\right) \ \forall \left(\Phi_{\max}, \Phi_{\min}\right) \quad (9)$$

Here, the noise points to be ignored for aggregation are notated as (P^{noise}). Thus, $(L^{\mathbb{g}})$ specifies the optimally aggregated dedicated links.

3.4. Feature Extraction and Error Rate Analyzation

Thereafter, the features like flow_duration, protocol_type, service_flag, dst_host_count, src_bytes, dst_bytes, num_failed_logins, srv_count, dst_host_count, dst_host_srv_count, class, Timestamp, Outbound_Utilzation, and more are extracted from $(L^{\mathbb{g}})$ and are denoted as (E^{fea}).

Further, by using error count, ERs are analyzed to enhance the overall traffic security and performance. Thus, the analyzed ERs (E^{rate}) are represented as,

$$E^{rate} = \frac{S^{\mathbb{E}} + R^{\mathbb{E}}}{T^{\mathbb{E}}} \times 100 \quad (10)$$

Where, the synchronization and rejection error count based on NTs is signified as $(S^{\mathbb{E}}, R^{\mathbb{E}})$, and the total connections (error count) are denoted as $(T^{\mathbb{E}})$.

3.5. Pattern Identification

Based on timestamp, analyzed ERs, and flow duration, the APs are identified after ER analysis to avoid inefficiencies in threat detection using EAE-DBSCAN, which is explained in (section 3.3). Thus, $(I^{Pattern})$ represents the identified APs.

Then, the malicious and normal traffic data are categorized as explained in the below sections.

3.6. Normal and Malicious Traffic Classification

Further, to categorize the malicious and normal traffic data, the identified APs $(I^{Pattern})$ and extracted features (E^{fea}) are inputted to Radial Basis Function Networks (RBFN). Nevertheless, underfitting or oversensitivity issues could be caused by the poor selection of width parameters in RBFNs, thereby degrading the model's accuracy. Hence, the MeDecay Heuristic (MDH) technique, which adaptively sets width parameters for improved stability and generalization, is introduced. Figure 2 exhibits the MDH-RBFN classifier.

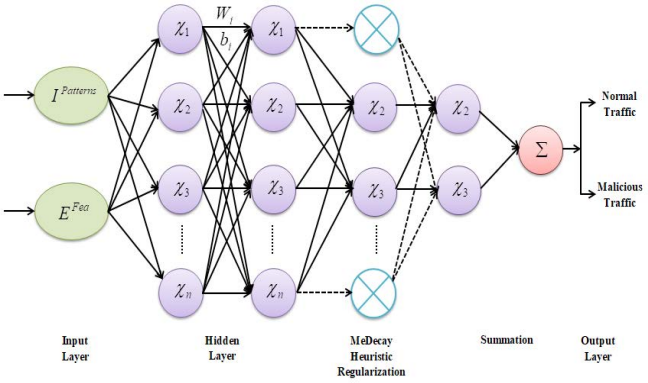


Figure 2: MDH-RBFN classifier

Input State Calculation

Primarily, the inputs $(I^{Pattern})$ and (E^{fea}) are combinedly signified as (λ^{g}). Thus, the (x) numbers of (λ^{g}) are articulated as,

$$\lambda^{g} = \lambda^{1}, \lambda^{2}, \ldots\ldots \lambda^{x} \ \ where \ (g = 1 \ to \ \ x) \quad (11)$$

In addition, the center point is initialized based on the input data (λ^{g}), which is shown as,

$$r^{\mathbb{f}} \leftarrow \lambda^{g} \quad (12)$$

Where, the center point based on the Radial Basis Function (RBF) is represented as $(r^{\mathbb{f}})$.

Hidden State Calculation

Here, using Euclidean Distance (ED), the distance is calculated between (λ^g) and $(r^{\hbar})$, and the calculated distance (d^E) is equated as,

$$d^E = \sqrt{\sum\left(r^{\hbar} - \lambda^g\right)^2} \quad (13)$$

After that, the hidden state (χ) is computed for each $(r^{\hbar})$ as,

$$\chi = \exp\left(-\frac{\left(d^E\right)^2}{2(\wp)^2}\right) \quad (14)$$

Here, the exponential function is signified as $(\exp)$, and the width parameter is notated as $(\wp)$. Here, $(\wp)$ is adaptively set utilizing the MDH technique as,

$$\wp = \xi\left(r^{\hbar} - \lambda^g\right) + \frac{\hbar}{2} \times \varpi\left\|r^{\hbar} - \lambda^g\right\|^2 \quad (15)$$

Where, the loss function is represented as (ξ), the regularization parameter is specified as $(\hbar)$, and the median function is denoted as (ϖ).

Output State Calculation

The output (ψ) is further computed utilizing (χ) and weights (w) assigned for each (λ^g) as shown below,

$$\psi = \sum w \times \chi\left(d^E\right) \quad (16)$$

$$\psi \rightarrow M^{traffic}, N^{traffic} \quad (17)$$

Here, the malicious and normal traffic data in the cloud environment are notated as $\left(M^{traffic}, N^{traffic}\right)$.

Pseudo code of MDH-RBFN

Input: Combined input, (λ^g)

Output: Malicious and Normal traffic, $\left(M^{traffic}, N^{traffic}\right)$

Begin

Initialize iterations, $\left(\tau, \tau^{\max}\right)$

While $\left(\tau < \tau^{\max}\right)$

Initialize $\left(\lambda^g\right), \left(r^{\hbar}\right)$

Calculate ED, $d^E = \sqrt{\sum\left(r^{\hbar} - \lambda^g\right)^2}$

Evaluate width parameter,

$$\wp = \xi\left(r^{\hbar} - \lambda^g\right) + \frac{\hbar}{2} \times \varpi\left\|r^{\hbar} - \lambda^g\right\|^2$$

Compute, $\psi = \sum w \times \chi\left(d^E\right)$

End while

Return $\rightarrow M^{traffic}, N^{traffic}$

End

Here, to prevent security breaches, $\left(M^{traffic}\right)$ are blocked; for enhanced security, $\left(N^{traffic}\right)$ are encrypted.

3.7. Data Encryption

$\left(N^{traffic}\right)$ are encrypted after categorization by using Quantum Cryptography (QC), which provides unconditional security based on the laws of quantum mechanics. However, the random qubit sequence in QC decelerates key generation, particularly over longer distances. Thus, Reed-Solomon Turbo Codes (RSTC), which correct single-qubit errors and enhance key generation, are utilized.

Initially, for faster key generation, a quantum key $\left(q^{key}\right)$ is generated using RSTC. Here, to create a unique key, $\left(q^{key}\right)$ analyzes the public and private keys $\left(p^{key}, r^{key}\right)$ and is shown as,

$$q^{key} = \frac{p^{key}\left(2^{r^{key}} - 1\right)}{\left(p^{key} + r^{key}\right) - p^{key}} \quad (18)$$

$$r^{key} = \left(p^{key} \times \Im\right) \quad (19)$$

Where, the coefficient of $\left(r^{key}\right)$ is specified as $(\Im)$. Then, $\left(N^{traffic}\right)$ are encrypted $\left(y^{encrypt}\right)$ as exhibited below,

$$y^{encrypt} = N^{traffic} \times \alpha(\Gamma) \quad (20)$$

Here, the polarization factor that converts all $\left(N^{traffic}\right)$into photons (Γ) for respective encryption is specified as (α).

Finally, for secure communications, eavesdropping $\left(D^{eaves}\right)$ is checked as equated below,

$$D^{eaves} = \begin{cases} \mathrm{E} & when \quad \left[N^{traffic}\left(y^{encrypt}\right) = 1\right] \\ \mathrm{E}^* & when \quad \left[N^{traffic}\left(y^{encrypt}\right) = 0\right] \end{cases} \quad (21)$$

Here, (E) and $\left(\mathrm{E}^*\right)$represent the absence of $\left(D^{eaves}\right)$ when the value is 1 and the presence of $\left(D^{eaves}\right)$ when the value is 0, respectively.

Pseudo code of RSQ2C

Input: Normal traffic data, $\left(N^{traffic}\right)$

Output: Encrypted data, $\left(y^{encrypt}\right)$

Begin

Initialize iterations, $\left(\tau, \tau^{\max}\right)$

While $\left(\tau < \tau^{\max}\right)$

Initialize $\left(p^{key}, r^{key}\right)$

Generate $\left(q^{key}\right)$,

$$q^{key} = \frac{p^{key}\left(2^{r^{key}} - 1\right)}{\left(p^{key} + r^{key}\right) - p^{key}}$$

Encrypt $\left(N^{traffic}\right)$,

$$y^{encrypt} = N^{traffic} \times \alpha(\Gamma)$$

Check $\left(D^{eaves}\right)$

End while

End

3.8. Traffic Level Prediction

By using the Fuzzy Inference System (FIS), the traffic level is predicted after encrypting the data. But FIS has tuning difficulty of membership function and control rules. Thus, the Generalized Bell-Π (GBΠ) membership function, which balances interpretability and effectiveness in FIS by addressing the tuning challenges, is used. The working steps of GBΠFIS are detailed as follows,

Initially, the rules $(\aleph)$ are set centered on the if-then condition as,

$$\aleph = \begin{cases} if\ O^{ut} = 0-20\%, then\ L^{traffic} \\ if\ O^{ut} = 20\%-40\%, then\ M^{traffic} \\ if\ O^{ut} = 40\%-60\%, then\ H^{traffic} \\ if\ O^{ut} = 60\%-80\%, then\ VH^{traffic} \\ if\ O^{ut} = 80\%-100\%, then\ E^{traffic} \end{cases} \quad (22)$$

Here, the condition states that if outbound utilization $\left(O^{ut}\right)$ is $(0-20\%),(20\%-40\%),(40\%-60\%),(60\%-80\%)\ and\ (80\%-100\%)$, then low, medium, high, very high, and extreme traffic $\left(L^{traffic}, M^{traffic}, H^{traffic}, VH^{traffic}, E^{traffic}\right)$ is predicted.

After that, to overcome the tuning difficulties, the GBΠ membership function $\left(G^{\Pi}\right)$ is assigned for fuzzy and is shown as,

$$G^{\Pi} = \frac{\aleph}{1 + \left(\frac{|\aleph|}{a}\right)^{2b}} \quad (23)$$

Here, the scaling parameters of the GBΠ function are specified as (a, b).

Further, using a fuzzy relationship $\left(R^{Fuz}\right)$, the final decision $\left(F^{D}\right)$ is obtained as shown below,

$$F^{D} = \frac{\sum y^{encrypt} \times R^{Fuz}(\aleph)}{G^{\Pi}} \quad (24)$$

Subsequently, the crisp output $\left(C^{out}\right)$ is obtained from $\left(F^{D}\right)$ and is exhibited as,

$$C^{out} = \frac{\sum F^{D} \times G^{\Pi}}{\sum G^{\Pi}} \quad (25)$$

Therefore, the traffic levels $\left(L^{traffic}, M^{traffic}, H^{traffic}, VH^{traffic}, E^{traffic}\right)$ are categorized based on rules.

3.9. Load Balancing

Subsequent to categorization, $\left(L^{traffic}, M^{traffic}\right)$ data are directly stored in the cloud for future usage, whereas $\left(H^{traffic}, VH^{traffic}, E^{traffic}\right)$ are balanced utilizing the WR²QH technique.

Primarily, the weights are assigned $\left(w^{\alpha}\right)$ for each server/ link/data $\left(l^{server}\right)$.

After that, the total weight is determined for each $\left(l^{server}\right)$ as (W).

Then, the portion of traffic $\left(P^{TRA}\right)$ is calculated as,

$$P^{TRA} = \frac{w^{\alpha}}{W} \times I^{TRA} \quad (26)$$

Here, the total incoming traffic is specified as $\left(I^{TRA}\right)$.

Thereafter, the traffic is handled in $\left(l^{server}\right)$ based on capacity and is shown below,

$$n^{server} = \sum\left[\left(c^{server} + 1\right)\% \times W\right] \quad (27)$$

Where, the current and next server are signified as $\left(c^{server}, n^{server}\right)$, and $\left(B^{data}\right)$ denotes the balanced data, which are then stored in the cloud for future usage. In further sections, the performance assessment of the proposed work is described.

4. Results and Discussion

In this section, the performance assessment of the proposed and traditional techniques is compared. In addition, the entire work is implemented in the PYTHON platform.

4.1. Dataset description

NAD and NATS datasets, which are gathered from publicly available sources, are used in the proposed work. Here, the NAD and NATS datasets together contain 73367 data with 42 features and 2 classes, such as normal and anomaly traffic. From the datasets, the proposed work used 45697 data (80%) and 27670 data (20%) for training and testing, respectively.

4.2. Performance analysis of the proposed work

Here, the proposed EAE-DBSCAN, RSQ2C, GBΠFIS, and MDH-RBFN are analogized with existing techniques and related works. The proposed MDH-RBFN's performance assessment is shown below,

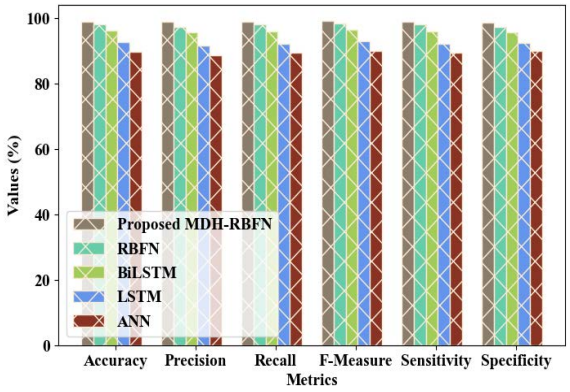


Figure 3: Performance Assessment of the Proposed MDH-RBFN.

In Figure 3, the performance assessment of the proposed MDH-RBFN and the traditional RBFN, Bidirectional Long-Short Term Memory (BiLSTM), LSTM, and Artificial Neural Network (ANN) techniques is shown. Here, the proposed MDH-RBFN attained high Accuracy (99.05%), Precision (98.95%), Recall (99.01%), F-measure (99.17%), Sensitivity (99.01%), and Specificity (98.64%) values than prevailing techniques. This enhanced performance is owing to the utilization of MDH, which adaptively sets width parameters centered on penalizing deviations.

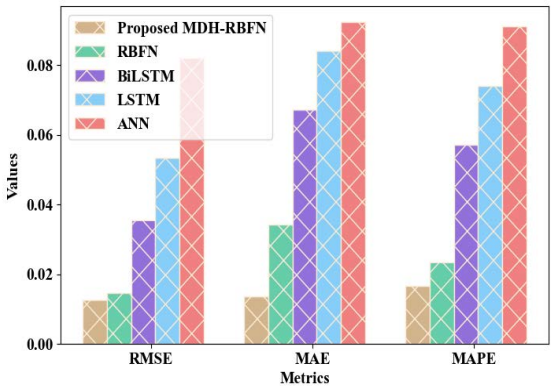


Figure 4: Comparative Analysis based on MAE, MAPE and RMSE.

Table 1: Training Time Analysis.

Techniques	Training Time (ms)
Proposed MDH-RBFN	34252
Existing RBFN	56893
Existing BiLSTM	89436
Existing LSTM	97803
Existing ANN	107845

The MAE, Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Training Time (TT) of the proposed MDH-RBFN and traditional RBFN, BiLSTM, LSTM, and ANN techniques are illustrated in Figure 4 and Table 1. Here, the proposed MDH-RBFN has minimum MAE (0.0135), MAPE (0.0167), RMSE (0.0127), and TT (34252ms) values due to the enhanced performance, whereas the traditional techniques exhibit degraded performance due to the oversensitivity issues.

As shown in Figure 5, the Pattern Identification Time (PIT) and Aggregation Time (AT) of the proposed EAE-DBSCAN are analogized with the traditional DBSCAN, K-Means Clustering (KMC), Fuzzy C Means (FCM), and K-Nearest Neighbor (KNN) techniques. Here, the proposed EAE-DBSCAN attained minimum PI (3578ms) and AT (2389ms). But, the traditional DBSCAN, KMC, FCM, and KNN attained average maximum PIT (8493ms) and AT (7457ms). The proposed technique is enhanced since EAE usage in DBSCAN dynamically adjusts the parameters in varying-density datasets.

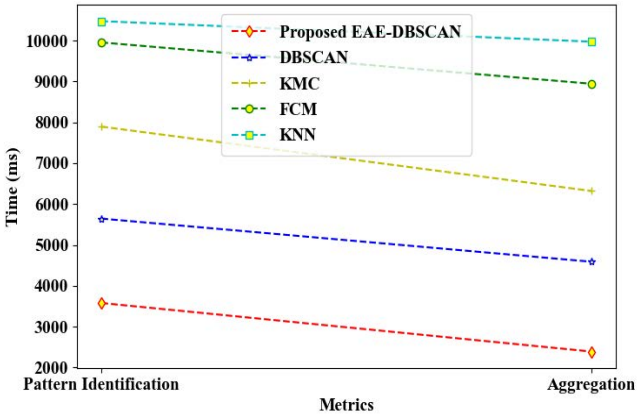
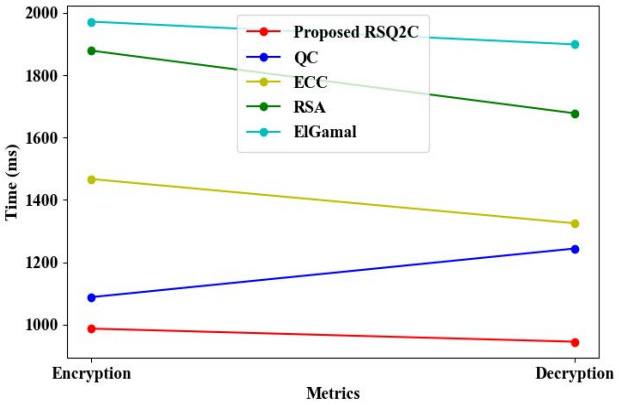
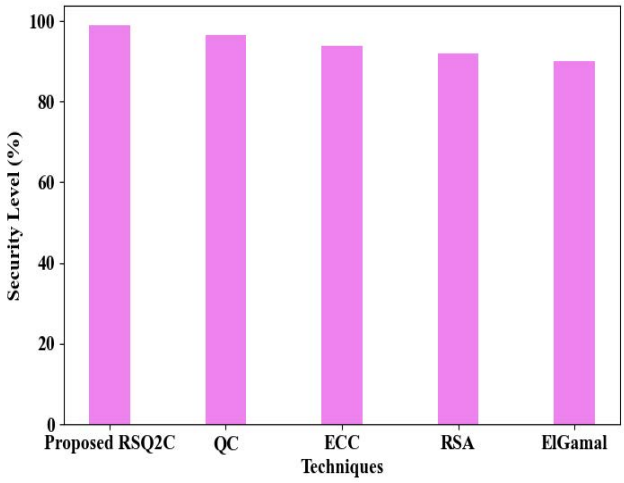


Figure 5: Pattern Identification and Aggregation Time Validation.



(a)



(b)

Figure 6: (a) Encryption, Decryption Time, and (b) Security Level validation of the proposed RSQ2C.

As shown in Figures 6 (a) and 6 (b), the proposed RSQ2C is compared with traditional QC, Elliptic Curve Cryptography (ECC), Rivest-Shamir-Adleman (RSA), and ElGamal techniques. Here, RSTC enhances QC by improving error correction and key generation. So, the proposed RSQ2C has a minimum Encryption time of 987ms and a Decryption time of 945ms, with a high Security Level (SL) of 98.85%. However, the traditional approaches had maximum encryption and decryption times with low SL, which degraded the entire work performance.

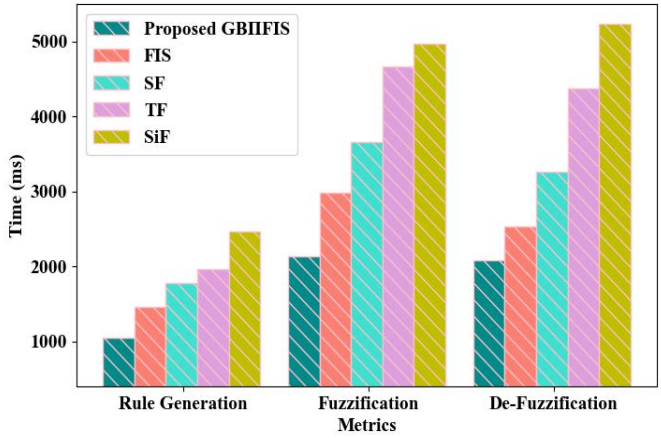


Figure 7: Performance Analysis of the Proposed GBIIFIS.

The performance analysis of the proposed GBIIFIS and the prevailing FIS, Sigmoid Fuzzy (SF), Trapezoidal Fuzzy (TF), and Singleton Fuzzy (SiF) is shown in Figure 7. Here, the proposed work is enhanced in predicting traffic levels by achieving minimum Fuzzification Time (FT= 2132ms), Defuzzification Time (DFT= 2085ms), and Rule Generation Time (RGT= 1045ms). This enhancement is owing to the utilization of GBII in FIS, which renders flexible membership modeling and enhances traffic prediction accuracy. However, the traditional techniques exhibit more FT, DFT, and RGT values than the proposed GBIIFIS, thereby hindering the process of predicting traffic levels.

Table 2: Comparative Analysis with Related Works.

Study	Techniques	RMSE	MAPE
Proposed Work	MDH-RBFN	0.0127	0.0167
(Yang et al., 2021)	ARIMA-BPNN	0.076	0.099
(Bi et al., 2022)	ST-LSTM	0.036	-
(Xu et al., 2021)	AE	0.129	-
(Wan et al., 2022)	LSTM	0.112	-
(Pan et al., 2022)	FPKNet	0.509	0.369

In Table 2, the proposed work is related to prevailing works. Here, for enhanced performance, the proposed work used the MDH technique, thus attaining minimum RMSE and MAPE values. Nevertheless, the conventional (Yang et al., 2021) and (Pan et al., 2022) utilized AutoRegressive Integrated Moving Average model-based Back Propagation Neural Network (ARIMA-BPNN) and Fusion Prior Knowledge Network (FPKNet) with maximum MAPE values. Additionally, (Bi et al., 2022), (Xu et al., 2021), and (Wan et al., 2022) used Savitzky–Temporal-based Long-Short Term Memory (ST-LSTM), AutoEncoder (AE), and LSTM techniques with average maximum RMSE value (0.1724ms) owing to overfitting and slow learning effects. Hence, the proposed work outperformed in analyzing the malicious traffic networks than the traditional works.

5. Conclusion

In this research, the APs based on timestamp, flow duration, and ERs are effectively identified for more enhanced NS. Initially, the data gathered from the datasets were preprocessed. After that, the dedicated links were aggregated within 2389ms. In addition, using EAE-DBSCAN, the APs were identified with a minimum PIT (3578ms). Subsequently, MDH-RBFN categorizes the normal and malicious traffic with 99.05% accuracy and an encryption time of 987ms. At last, the traffic level was predicted with a minimum RGT (1045ms). Hence, the proposed work performed better in predicting malicious traffic for enhanced NS.

6. Future Recommendation

However, numerous approaches will be implemented in the future to predict the traffic severity for more secure data transmission.

7. References

1. Abbasi, M., Shahraki, A., & Taherkordi, A. (2021). Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey. *Computer Communications*, *170*, 19–41. https://doi.org/10.1016/j.comcom.2021.01.021
2. Balamurugan, N. M., Adimoolam, M., Alsharif, M. H., & Uthansakul, P. (2022). A Novel Method for Improved Network Traffic Prediction Using Enhanced Deep Reinforcement Learning Algorithm. *Sensors*, *22*(13), 1–17. https://doi.org/10.3390/s22135006
3. Barrera-Animas, A. Y., Oyedele, L. O., Bilal, M., Akinosho, T. D., Delgado, J. M. D., & Akanbi, L. A. (2022). Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Machine Learning with Applications*, *7*, 1–20. https://doi.org/10.1016/j.mlwa.2021.100204
4. Bi, J., Zhang, X., Yuan, H., Zhang, J., & Zhou, M. C. (2022). A Hybrid Prediction Method for Realistic Network Traffic With Temporal Convolutional Network and LSTM. *IEEE Transactions on Automation Science and Engineering*, *19*(3), 1–11. https://doi.org/10.1109/TASE.2021.3077537
5. Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines. *IEEE Access*, *9*, 120043–120065. https://doi.org/10.1109/ACCESS.2021.3107975
6. Dong, S. (2021). Multi class SVM algorithm with active learning for network traffic classification. *Expert Systems with Applications*, *176*, 1–11. https://doi.org/10.1016/j.eswa.2021.114885
7. Drewek-Ossowicka, A., Pietrołaj, M., & Ruminski, J. (2021). A survey of neural networks usage for intrusion detection systems. *Journal of Ambient Intelligence and Humanized Computing*, *12*(1), 497–514. https://doi.org/10.1007/s12652-020-02014-x
8. Ensafi, Y., Amin, S. H., Zhang, G., & Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning – A comparative analysis. *International Journal of Information Management Data Insights*, *2*(1), 1–16. https://doi.org/10.1016/j.jjimei.2022.100058
9. Fotiadou, K., Velivassaki, T. H., Voulkidis, A., Skias, D., Tsekeridou, S., & Zahariadis, T. (2021). Network traffic anomaly detection via deep learning. *Information (Switzerland)*, *12*(5), 1–17. https://doi.org/10.3390/info12050215
10. Khan, M. A., Nasralla, M. M., Umar, M. M., Ghani-Ur-rehman, Khan, S., & Choudhury, N. (2022). An Efficient Multilevel Probabilistic Model for Abnormal Traffic Detection in Wireless Sensor Networks. *Sensors*, *22*(2), 1–22. https://doi.org/10.3390/s22020410
11. Li, X., Law, R., Xie, G., & Wang, S. (2021). Review of tourism forecasting research with internet data. *Tourism Management*, *83*, 1–11. https://doi.org/10.1016/j.tourman.2020.104245
12. Lindemann, B., Müller, T., Vietz, H., Jazdi, N., & Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, *99*, 650–655. https://doi.org/10.1016/j.procir.2021.03.088
13. Liu, Z., Zhu, Z., Gao, J., & Xu, C. (2021). Forecast Methods for Time Series Data: A Survey. *IEEE Access*, *9*, 91896–91912. https://doi.org/10.1109/ACCESS.2021.3091162

14. Pan, C., Wang, Y., Shi, H., Shi, J., & Cai, R. (2022). Network Traffic Prediction Incorporating Prior Knowledge for an Intelligent Network. *Sensors*, *22*(7), 1–16. https://doi.org/10.3390/s22072674

15. Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M., & Bagnall, A. (2021). The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. In *Data Mining and Knowledge Discovery* (Vol. 35, Issue 2). Springer US. https://doi.org/10.1007/s10618-020-00727-3

16. Shen, M., Zhang, J., Zhu, L., Xu, K., & Du, X. (2021). Accurate Decentralized Application Identification via Encrypted Traffic Analysis Using Graph Neural Networks. *IEEE Transactions on Information Forensics and Security*, *16*, 2367–2380. https://doi.org/10.1109/TIFS.2021.3050608

17. Wan, X., Liu, H., Xu, H., & Zhang, X. (2022). Network Traffic Prediction Based on LSTM and Transfer Learning. *IEEE Access*, *10*, 86181–86190. https://doi.org/10.1109/ACCESS.2022.3199372

18. Weerakody, P. B., Wong, K. W., Wang, G., & Ela, W. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, *441*, 161–178. https://doi.org/10.1016/j.neucom.2021.02.046

19. Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y., & Sabrina, F. (2021). Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset. *IEEE Access*, *9*, 140136–140146. https://doi.org/10.1109/ACCESS.2021.3116612

20. Yang, H., Li, X., Qiang, W., Zhao, Y., Zhang, W., & Tang, C. (2021). A network traffic forecasting method based on SA optimized ARIMA–BP neural network. *Computer Networks*, *193*, 1–12. https://doi.org/10.1016/j.comnet.2021.108102