LPMLP-Based Framework for IPsec VPN Cloud Gateway with Advanced Network Monitoring and Issue Resolution

Vinay Gugueoth, guguv@amazon.com

***Abstract*—Nowadays, VPN technology is widely used in cloud and hybrid network communication that makes use of algorithms and tunneling to meet different security requirements. However, existing cloud VPN gateways often lack advanced monitoring capabilities and struggle to identify and resolve network connectivity and performance issues. Hence, LPMLP adapted Secure cloud VPN Gateway with Network Monitoring and Issue Resolution is proposed. Here, the VPN raw log data is taken as input and pre-processed. Then, Apache Spark is used to handle the big unstructured data. The patterns are extracted from the structured data using the KMP technique. From the patterns, correlation using PCC is computed and events are scored based on the correlation value. Next, RADFCM clustering is used to group the correlated anomalies. Based on the clustered result, the SPBERT summarizer is used to summarize the issues in the network. In the meantime, time series features are extracted from the structured data, and optimal features are selected via the RLCEPO technique. On the other side, the metrics data are captured and they undergo preprocessing followed by PCC-based correlation analysis. Finally, the summarized issues, selected features, along with correlated anomaly metrics are given to the LPMLP classifier to perform the recommendation. The experimental evaluation proved the efficiency of the proposed framework.**

***Index Terms*—Virtual Private Network (VPN), Knutt Morris Pratt (KMP), Rectangular Area Division Fuzzy C-Means Clustering (RAD-FCM), Scaled Polynomial Bidirectional Encoder Representations from Transformers (SPBERT), Refraction Learning Continuous Emperor Penguin Optimizer (RLCEPO), Linearly Paralyzed Multi-Layer Perceptron (LPMLP).**

I. Introduction

The significant increment in the number of Internet users caused the rapid evolution of networking systems, e.g., cloud computing, hybrid connectivity between on-premises data center and cloud networks, and network virtualization [1]. Ensuring secure connectivity and maintaining privacy are problems for both users and service providers in networking systems. However, VPNs are one of the most efficient ways to ensure the security and attain anonymity in the cloud networking as well as the Internet [2]. In the untrusted network of the Internet, avoiding sniffing attacks and preserving data integrity are the major goals of VPNs [3], [4]. Security experts use VPNs for sharing Intranet services on a public network with authentication and authorization [5]. Internet Protocol Security (IPsec) is one of the oldest but still most widespread VPN protocols [6]. Recently, VPNs are becoming increasingly complex to support the heterogeneous needs of a variety of Internet applications, which led to an exponential growth in data traffic over heterogeneous networks [7]. Thus, automation is necessary to manage these complex heterogeneous networks. Any automated network management system that aims to enhance resource consumption planning, network performance, quality of service, and cyber security typically starts with network traffic classification [8]. Network classification and analysis techniques are classified into two main groups, active methods, and passive methods. In order to get information on the state of the network, active methods entail creating and injecting probe traffic into the network. Passive techniques, such as fault tolerance and troubleshooting can be used to carefully monitor traffic, especially in post-event scenarios [9]. Then, payload-based traffic classification and port-based traffic classification were introduced. However, when working with encrypted payloads, they were incredibly prone to errors [10]. To overcome the limitation of port-based and payload-based approaches, classical machine learning (ML) techniques have been given considerable attention [11]. The most important component of ML algorithms is the learning ability incurred. This ability is gained through experience and a refinement process [12]. In some works, ML algorithms are employed by determining the pattern in the network data captured. However, pattern identification of video streaming, file transfer, email, and browsing is a challenge in such ML-based classification [13]. Hybrid learning with a focus on Deep Learning (DL) has recently attracted many researchers in network traffic classifications. The accuracy of these hybrid models plays a significant role in traffic forecasting and classification [14]. However, these models suffer from information loss, over-fitting, and increased model complexity. Moreover, these models could not concentrate on network-related issues resolving [15] such as connectivity, performance and anomaly detection. Hence, an LPMLP-based recommendation model for a secure IPsec VPN cloud Gateway is proposed in this paper.

A. Problem Statement

Existing VPN-based works had the following drawbacks:

- Existing VPN cloud gateways struggled to proactively find and resolve network-related issues. This leads to potential security vulnerabilities, performance bottlenecks, and operational inefficiencies in cloud networking.
- Existing VPN network traffic monitoring systems failed to perform well for a large volume of traffic and the

extraction of patterns about network behavior was not considered.
- Metric patterns and network issues are the important parameters that affect the performance of the IPsec VPN cloud gateways, which are not considered in any of the existing works.
- Differentiating the IPsec VPN cloud gateway network anomaly features is a challenging task because of the encrypted packet structure.

To overcome these challenges, this paper proposed an intelligent network monitoring and issue-resolving model based on LPMLP. The main contributions of the framework are,

- To identify and resolve network-related issues, the LPMLP model is proposed.
- To handle big data, Apache Spark is used and the patterns of the network were extracted from the data.
- To enhance the recommendation process, issue summarization using SPBERT is done.
- To differentiate anomaly data, Anomaly clustering is performed based on correlated event scoring.

The organization of this paper is structured as follows: Section II analyses the various prior works related to the proposed method. The proposed methodology is discussed in section III. Section IV analyses the performance of the proposed methodologies. Finally, section V ends the paper with a conclusion.

II. LITERATURE SURVEY

Aswad [16] used the time-related features on the Apache Spark and artificial neural networks to classify the VPN traffic flow. 80% of the dataset was used to train the system, and the other 20% was preserved for testing and validation. There were 50 training iterations and 10 cross-validation iterations. The technique prevented unnecessary processing and flooding in standard VPN traffic. However, the classification accuracy of Non-VPN was low. Almomani [17] introduced a classification model of VPN encrypted traffic with ensemble learning algorithms. Three machine learning (ML) techniques—Random Forest, neural network, and Support Vector Machine (SVM)—were used in the model to classify VPN and Non-VPN traffic. The results of the experiment demonstrated that the model correctly distinguished between VPN and non-VPN traffic. But, it couldn't perform deep packet inspection due to performance and privacy requirements. In [18] focused on enabling the maximum utilization of the reserved bandwidth by protecting the VPN site from flooding attacks. The strategy defended against both insider attacks and outsider attacks on the allotted bandwidth. The insider attacks probabilistic model was simulated, and the real packet loss was examined. The deployment and implementation of the method were simple. But, the attacks in Remote Access VPN were not focused. In [19], two Deep Learning (DL) models were suggested to classify the traffic into VPN and Non-VPN traffic. The models carried out the experiment goals by preprocessing the traffic samples into session pictures using Convolutional Auto-Encoding (CAE) and Convolutional Neural Network (CNN), respectively. The experimental results showed that the models outperformed traditional identifying strategies. But, the pre-processing of data was not very much focused. Parchekani et al. [20] presented an end-to-end traffic categorization technique to distinguish between different traffic classes, including VPN traffic, in the three layers of the Open Systems Interconnection (OSI) paradigm. Using MLP and Recurrent Neural Network (RNN), the cascade neural network concentrated on two metrics: class scores and distance from the centers of the classes. The outcomes proved that the framework was effective. But, the network-related issues were not concentrated. Afuwape et al. [21] suggested various algorithms to classify and detect VPN traffic. The classification process made use of the Bagging Decision Tree and Gradient Boosting algorithms, and the outcomes were encouraging. Additionally, it was discovered that the classifiers performed better when the network traffic flows were created using various values of the time parameters. But, the computation was slow while handling large data.

III. PROPOSED IPSEC VPN FRAMEWORK

In this paper, a secure IPsec VPN cloud gateway model with Intelligent Network Monitoring and Issue Resolution is proposed. The output of correlated analysis, optimized features, and language summarization are given to the proposed LPMLP model to provide the final report. The processes involved in this framework are illustrated in Fig.1. The proposed framework starts by collecting the input VPN raw log data. This log data contains a series of run-time information that occurred in the system. The collected input VPN data L is described as, $L_i = \{l_1, l_2, .., l_q\}$, where, l_q refers to the q^{th} number of log data collected.

A. Pre-processing of Raw Log Data

The collected VPN raw log data L undergoes preprocessing to enhance the quality of the raw data as it may contain inconsistent data values. Preprocessing is performed by two steps, namely null value removal and deduplication.

- **Null value Removal:** Here, the rows and columns of the data L containing null values are removed to increase the performance of the classifier.
- **Deduplication:** Then, in this process, the redundant data present in L are eliminated. Thus, the raw data is preprocessed and is denoted as L_{pre}.

B. Big Data Structuring and Handling

As the data L_{pre} contains a large amount of data, more time will be taken to process each set of data during the classification. So, to structure and handle the big data, Apache Spark is used here. Apache Spark is an in-memory cluster computing technology that reads the data and performs operations in Resilient Distributed Dataset(RDD). The RDD performs two operations, such as transformation and action. The function of Apache spark is described as,

$$A = map(L_{pre}) \tag{1}$$

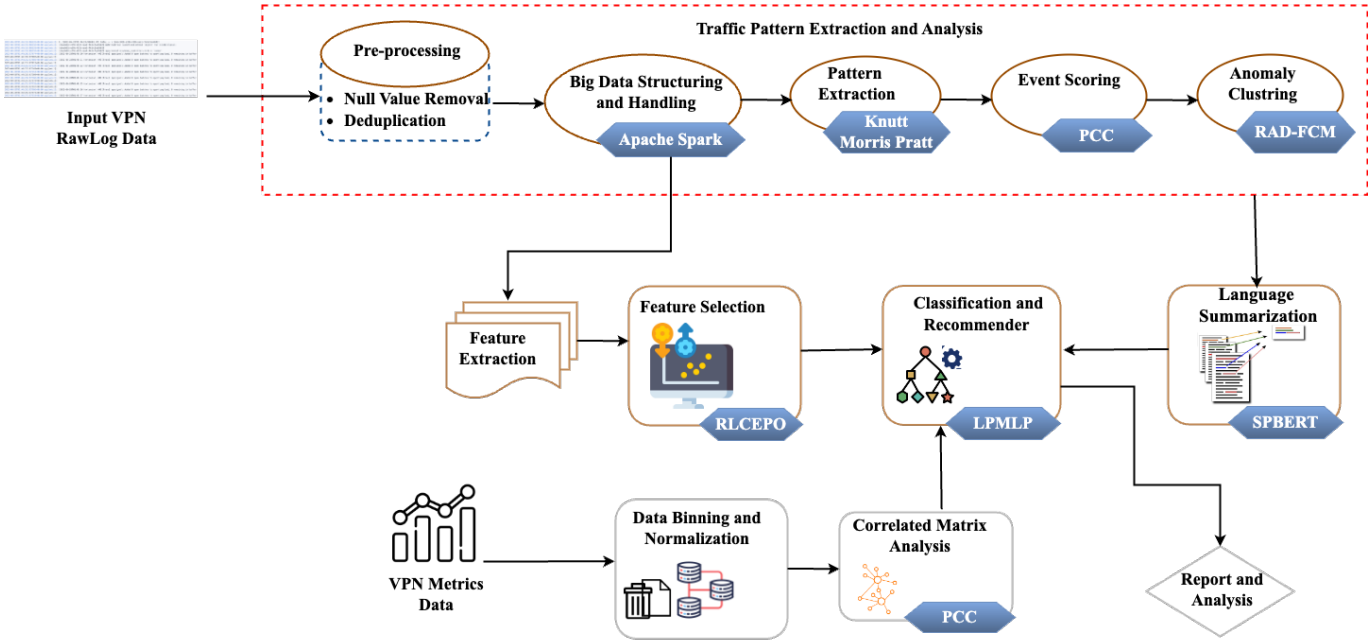

Fig. 1. Structural Design of Proposed Methodology

where, A denotes the mapped output by the map function map. Then, all the mapped data are stored in an RDD, which is more efficient during multiple access queries.

$$RDD_i = \Omega(A) \tag{2}$$

where, $i = 1, 2, ..., n$ denotes the number of RDDs and Ω is the group by key function. Finally, the reduce action red is performed as,

$$L_r = red\Big(\sum RDD_i\Big) \tag{3}$$

where, L_r denotes the reduced data. Thus, the unstructured big data is transformed into a standard format.

C. Pattern Extraction

In this phase, the patterns of the events that occurred in the time-series data L_r are extracted using the KMP technique. This technique is a string-matching algorithm, which performs data matching from left to right. KMP processes the data L_r and matches the events, such as problems that occurred in the VPN network, their corresponding slots, etc. The steps involved in the KMP algorithm are as follows:

- Initially, a pattern of an event is selected from L_r. Then, a variable to be matched is selected from a string containing an array of letters.
- This pattern is compared with that particular string based on the selected variable.
- If the pattern matches with the string, this pattern is stored using the KMP algorithm and the pattern does not shift and continues the matching process.
- If the pattern does not match, then the pattern will shift one position to the right.
- Compare the pattern with the next string. This is continued until the entire pattern in the strings is matched.

Thus, the time-series data is processed and the various patterns P_i are extracted. The extracted patterns are described as, $P_i = \{p_1, p_2, ...p_r\}$, $i = 1, 2, ..., r$. Where, P_r denotes the r^{th} number of patterns extracted.

D. Event Scoring

After pattern extraction, the patterns P_i are correlated using the Pearson Correlation Coefficient (PCC) technique. Using the resultant correlation values, all the anomaly events are scored. The correlation values δ_i are computed as,

$$\delta_i = \frac{\sum_{i=1}^{r}(p_i - \bar{P})(p_{i+1} - \bar{P})}{\sqrt{\sum_{i=1}^{r}(p_i - \bar{P})^2(p_{i+1} - \bar{P})^2}} \tag{4}$$

where, $\bar{P}$ denotes the mean value of all the patterns. Thus, based on δ_i, the anomaly scoring for each event is performed.

E. Anomaly Clustering

In this phase, the correlated event scores are grouped using the RAD-FCM algorithm based on the anomaly. The conventional FCM provided better results for overlapped data sets. But, the disadvantages of FCM lie in the selection of the initial cluster centroids, which are sensitive to noise and outliers and had a great impact on the clustering results. Therefore, rectangular area division-based computation is developed to initialize the centroids. RAD-FCM algorithm works by assigning membership to each value in δ_i, corresponding to each cluster center. The data points are assigned to the cluster center based on the similarity between them. The centroids ς are obtained with the help of RAD as,

$$\varsigma = \left\{\frac{[\delta_1 + md(\delta_i)]}{2}, \frac{[\delta_2 + md(\delta_i)]}{2}\right\} \tag{5}$$

here, md denotes the midpoint of the rectangular area. After obtaining the centers, the fuzzy membership is computed using the expression,

$$f_{ij} = \frac{1}{\sum(\varepsilon_{ij})^{\frac{2}{\iota-1}}} \tag{6}$$

where, f_{ij} is the membership of the i^{th} event score to j^{th} centroid, ε denotes the Euclidean distance, and ι is the fuzziness index of range $[1, \infty]$. Now, the centroids are updated as,

$$\varsigma_j = \frac{\Sigma(f_{ij})^{\iota} * \delta_i}{\Sigma(f_{ij})^{\iota}} \tag{7}$$

The main aim of RAD-FCM is to minimize the objective function concerning centroids and memberships. The objective function Θ is expressed as,

$$min(\Theta) = \sum (f_{ij})^{\tau} \|\delta_i - \varsigma_j\|^2 \tag{8}$$

All processes are repeated until the minimum Θ value is obtained from all centroids. Finally, the events are grouped. The formed clusters Z are denoted as, $Z = \{Z_{at}, Z_{non}\}$. Where, Z_{at} and Z_{non} denote the attacked data and non-attacked data. Algorithm 1 describes the procedure of the RAD-FCM technique.

Algorithm 1 RAD-FCM

Input: Event scores (δ_i)
Output: Cluster set Z
Begin
Initialize objective function Θ
Compute the cluster centers as, $\varsigma = \left\{\frac{[\delta_1 + md(\delta_i)]}{2}, \frac{[\delta_2 + md(\delta_i)]}{2}\right\}$
for $(i = 1)$ to $\left(\| \varsigma_j^i - \varsigma_j^{i-1} \| < \Theta \right)$ **do**
  Compute membership matrix as $f_{ij} = \frac{1}{\sum(\varepsilon_{ij})^{\frac{2}{\iota-1}}}$
  Update the centroids
  Assign data points to the nearest centroid
end for
if $\delta_i = Z$ **then**
  Terminate the process
else
  Compute new centriod
end if
Return Z
End

F. Language Summarization

Here, based on the anomaly grouping results Z, the issues in the network are summarized using the SP-BERT text summarizer. The BERT model is available and is pre-trained in more languages than other models. But, it is slow for training because it is immense due to the variation of the activation function. Hence, scaled polynomial constant unit activation is proposed here.

In SPBERT, the token of each word with an issue is determined and represented within different classes, and each class is separated by the separation function. The tokens tk are represented as, $tk = \lfloor t_1, t-2, ..., t_q \rfloor$. Where, t_q refers to the q^{th} tokenized word. Now, these tokens are given to the embedding layer, which performs token embedding, segment embedding, and position embedding. Embeddings are the representation of words in vector form.

- **Token embedding:** Here, the words are transformed into a fixed dimensional vector with a class and separation function added to the beginning and end of the sentences.
- **Segment embedding:** The segment embedding for a word is done to classify the different inputs with binary coding.
- **Position embedding:** Position embedding is employed to discern between a word's contextual meaning and the meaning of the sentence according to how the word is positioned.

The token, segment, and position embeddings are summed up and given to the SPBERT transformer encoder layer, which contains twelve transformers with twelve attention mechanisms. The encoder encodes the words and the decoder determines the significant keywords and gives contextual embeddings. The encoded output is given to the output layer, which contains a simple classifier model with a fully connected layer and SP constant unit activation. The loss ℓ in the classifier output is calculated as,

$$\ell = \frac{1}{2}(\Im - \sum I) \tag{9}$$

where, $\Im$ depicts the target score, and I is the output embedding, which is summarized based on the SP activation function Φ as,

$$\Phi = a\wp\Big(\frac{I}{c} + b\Big) - a\wp(b) \tag{10}$$

where, $\wp$ refers to the scaled polynomial, and the constants $b \in (0, 1)$ and $a, c > 0$. Thus, by using SPBERT, the issues I are summarized.

G. Feature Extraction

Meanwhile, the time series features, such as date, source, suricata ID, flow ID, event, input IP address, Destination IP address, protocol, description, etc are extracted from the structured data L_r. The extracted feature set $\Re$ is described as, $\{\Re = \Re_1, \Re_2, ..., \Re_m\}$. Where, $\Re_m$ denotes the m^{th} number of features extracted from the data L_r.

H. Feature Selection

Then, from the feature set $\Re$, the optimal features are selected using the RLCEPO algorithm. Continuous Emperor Penguin Optimizer (CEPO) is one of the highest-performing meta-heuristic algorithms of recent times imposed the gathering behavior of emperor penguins. However, the paramount challenge in CEPO is that it is prone to stagnation in local optima. To mitigate this issue, Refraction Learning (RL) strategy is proposed to enhance the ability to jump out of local optimum. In RLCEPO, the features $\Re$ are considered as the penguin population. The main aim of the algorithm is to find the best mover and update their position. Initially, the penguins create a huddle in a polygon shape. The wind direction plays a vital role in determining the boundaries of the huddle which is expressed as, $\eta = \nabla v$. Where, η denotes

the gradient, ∇ is thenabla operator, and v is the velocity of the wind. Based on the direction of the wind, the potential function ρ is estimated as,

$$\rho = v + \kappa * rd \tag{11}$$

where, κ denotes the constant and rd is a random vector. The position of the penguin is updated based on the position of the penguin having the best fitness. Here, the event scoring values δ_i are considered as the fitness F, which is computed as, $F = max(\delta_i)$.

The purpose of the huddle is to maintain a high temperature in it during winter. The difference in temperature H between in and out of the boundary is formulated as,

$$H = \left(h - \frac{E}{e - E}\right) \tag{12}$$

$$h = \begin{cases} 0, & \gamma > 1 \\ 1, & \gamma < 1 \end{cases} \tag{13}$$

where, h denotes the temperature profile, e refers to the current iteration, E is the maximum number of iteration, and γ denotes the polygon's radius. Then, the distance $dist$ between the members of the huddle is computed based on the refraction learning strategy σ as,

$$dist = Abs\Big(A(\overrightarrow{C})\overrightarrow{\Re_{best}(e)} - \overrightarrow{B}.\overrightarrow{\Re(e)}\Big) * \sigma \tag{14}$$

$$\sigma = \big(\overrightarrow{C} + \overrightarrow{B}\big)/2 + \big(\overrightarrow{C} + \overrightarrow{B}\big)/\big(2Kd\big) - \overrightarrow{\Re_{best}}/Kd \tag{15}$$

where, A denotes the members' force in influencing the best position, Abs means absolute value, $\overrightarrow{\Re_{best}}$ denotes the best penguin, Kd, are random numbers of range [0, 1], and $\overrightarrow{C}$ and $\overrightarrow{B}$ are vectors utilized to avoid collisions, which are computed as,

$$\overrightarrow{C} = \big(\vartheta \times \big(H + \Re_{gr(acc)}\big) \times rand\big) - H \tag{16}$$

$$\overrightarrow{B} = rand() \tag{17}$$

$$\Re_{gr(acc)} = Abs\big(\overrightarrow{\Re} - \overrightarrow{\Re_{best}}\big) \tag{18}$$

where, ϑ denotes the movement parameter, $\Re_{gr}(acc)$ refers to the grid accuracy of the polygon, and $rand()$ defines the random number of range [0, 1]. The social force is $A(\overrightarrow{C})$ formulated as,

$$A(\overrightarrow{C}) = \sqrt{\left(\mu.\chi^{-\frac{e}{w}} - \chi^{-e}\right)^2} \tag{19}$$

where, μ and w are the control parameters that maintain exploitation and exploration effectively, χ refers to the expression function. Then, the positions of the penguins are modified based on the new best-fit penguin's position. This updation is formulated as,

$$\overrightarrow{\Re}(e+1) = \overrightarrow{\Re_{best}(e)} - dist.\overrightarrow{C} \tag{20}$$

The process is continued until the termination criterion is satisfied. Finally, the best solution (i.e., the optimal features $\Re_{best}$) is obtained by the RLCEPO algorithm. Algorithm 2 explains the procedure of the RLCEPO technique.

Algorithm 2 RLCEPO

Input: Extracted features ($\Re$)
Output: Optimal features ($\Re_{best}$)
Begin
Initialize input data ($\Re$), parameters $\overrightarrow{C}$, $\overrightarrow{B}$ and A, number of iterations E
Compute fitness of esch individual (F)
Select best solution
while $\big(e < E\big)$ **do**
  Update temperature profile
  if $\big(\gamma > 1\big)$ **then**
    Set $h = 1$
  else
    Set $h = 0$
  end if
  Compute temperature difference (H)
  Compute distance between members of the huddle $dist = Abs\Big(A(\overrightarrow{C})\overrightarrow{\Re_{best}(e)} - \overrightarrow{B}.\overrightarrow{\Re(e)}\Big) * \sigma$
  for each individual **do**
    Calculate collision avoidance parameters $\overrightarrow{C}$ and $\overrightarrow{B}$
    Formulate the social force A
    Update position of penguins
  end for
  Obtain fitness of update positions (F)
  if $F(\overrightarrow{\Re}(e+1)) > F(\overrightarrow{\Re_{best}(e)})$ **then**
    Update $\overrightarrow{\Re}(e+1)$ as best-fit position
  else
    Keep previous solution $\overrightarrow{\Re_{best}(e)}$
  end if
  Set $e = e + 1$
end while
Return $\Re_{best}$
End

I. Processing of Metrics Data

On the other end, the metrics data D related to the VPN network are captured. This gathered data undergoes two steps, namely data binning and normalization in order to decrease the complexity of the classifier.

Data Binning: It is a pre-processing technique, which helps to reduce the impact of the smallest observational mistakes. Here, the original data values are split up into discrete intervals, or bins, and are then swapped out for a generic value produced for that bin.

Normalization: Here, the binned data D_{bin} is structured using the normalization process. The data normalization Λ is formulated as,

$$\Lambda = \frac{D_{bin} - \varphi(D_{bin})}{SD} \tag{21}$$

where, φ denotes the mean value and SD is the standard deviation.

Correlated Metrics Analysis After processing the data, the normalized data Λ undergoes correlation analysis using

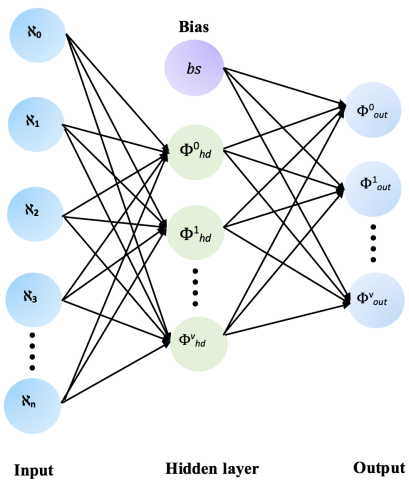

Fig. 2. LPMLP

the PCC technique. The metrics data are correlated as,

$$\lambda = \frac{\sum_{i=1}^{n} \left(\Lambda_i - \bar{\phi}\right)\left(\Lambda_{i+1} - \bar{\phi}\right)}{\sqrt{\sum_{i=1}^{n} \left(\Lambda_i - \bar{\phi}\right)^2 \left(\Lambda_{i+1} - \bar{\phi}\right)^2}} \tag{22}$$

where, λ denotes the correlation values of the anomaly metrics and is the mean value of the normalized data.

J. Classification and Recommender

In this final phase, the summarized issues I, optimal features $\Re_{best}$, and correlated anomaly metrics λ are given to the Linearly Paralyzed Multi-Layer Perceptron (LPMLP) classifier to find the issues in VPN and recommendations. The MLP model is suitable for solving complex non-linear problems with large data handling capability. Also, it makes the prediction quicker once it is trained. The only drawback with MLP is that it requires more hyperparameter tuning. Hence, to overcome this issue, linear saturated activation function is used with paralyzed neuron percentage-based parameter optimization. The combination of such activation and parameter tuning reduces the complexity. The network architecture of the proposed LPMLP is given in Fig.2. The input data $(\aleph \rightarrow \{I, \Re_{best}, \lambda\})$ is given to the input layer. Here, the input layer prepares the input for further processing by the other layers. Then, the prepared data is given to the hidden layers Φ_{hd}, where the hidden neurons perform the following process,

$$\Phi_{hd} = \sum_{i=1}^{v} \aleph_i \omega + bs \tag{23}$$

where, bs depicts the bias value, ω depicts the total number of hidden neurons, and denotes weight value, which is obtained by paralyzed neuron-based optimization as,

$$\omega = argmin\left[\frac{1}{v}\sum \aleph_i\right] \tag{24}$$

The Linear Saturated Activation Function θ of the hidden neuron is computed as,

$$\theta(\Phi_{hd}) = exp(\aleph) + \Xi \tag{25}$$

where, Ξ denotes the activating constant. The processed features from the hidden neurons are summed up and given to the output layer to predict the output Φ_{out} of the corresponding input. The output of the LPMLP is given as,

$$\Phi_{hd} = \theta\Big(\sum_{i=1}^{v} \Phi_{hd}\omega + bs\Big) \tag{26}$$

After the output value is predicted, the loss function $Loss$ is calculated to find the error in the classifier as,

$$Loss = \frac{1}{v}\sum_{i=1}^{v}\Big(O_{out} - \Phi_{out}\Big) \tag{27}$$

here, O_{out} denotes the threshold output. If the estimated error value is less than or equal to the threshold, then the output class will be considered; otherwise, the training continues by changing the weight values. The output class reports the issues within the VPN gateway and recommends solutions for the corresponding issues. For example, the output is given as that the VPN network is going down due to multiple reasons like large packet size, high CPU Utilization, etc., and the solution to overcome this problem. The Efficiency of the proposed recommender is analyzed in the following section.

IV. RESULTS AND DISCUSSION

To evaluate the efficacy of the proposed system, the performance analysis, as well as comparative analysis, is carried out in this section. All the experiments were conducted on the working platform of PYTHON.

A. Database Description

The proposed methodology utilized the ISCXVPN-2016 [22] network traffic dataset for performance validation. The VPN-non-VPN dataset contains the traffic content of web-browsing, Email, chat, streaming, file transfer, Voice over Internet Protocol (VoIP), and Point to point (P2P) related to corresponding applications. The dataset contains 14 traffic categories with descriptions of different types of traffic generated.

B. Performance Analysis of Recommender

Here, the performance of the proposed LPMLP classifier is compared with the existing MLP, Radial Basis Function Networks (RBFN), Restricted Boltzmann Machines (RBM), and Deep Belief Networks (DBN) based on quality metrics. Fig.3 illustrates the detection rate and the recommendation efficiency of the proposed and existing classifiers. The proposed LPMLP attains better performance for both metrics, where the enhancement is achieved for 2.14% and 2.16% than the existing MLP. Hence, the analysis concludes that the introduction of the Linear Saturated Activation Function significantly improved the quality of the proposed recommendation model.

In Fig.4, the results of the proposed LPMLP and existing networks based on Mean Absolute Percentage Error (MAPE), and Mean Square Error (MSE) are analyzed. In comparison, the proposed model attains a minimum error rate of 6.6589% (MSE) and 5.2978% (MAPE). But, the other existing models obtained higher error values. Hence, the inclusion of paralyzed

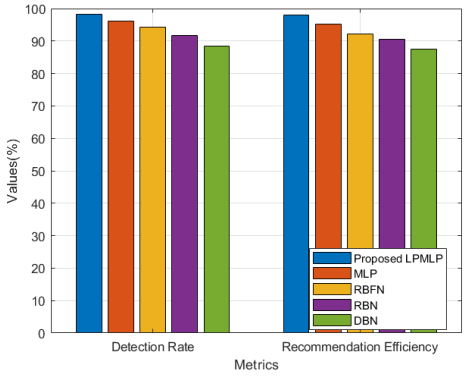

Fig. 3. Performance Measure of Proposed LPMLP

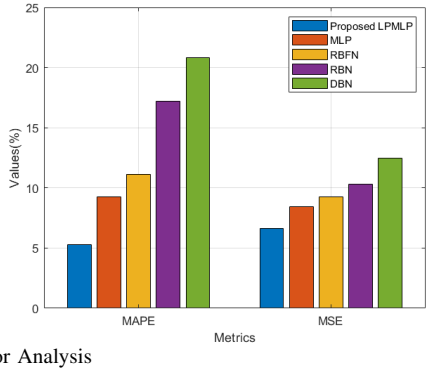

Fig. 4. Error Analysis

neuron percentage has a strong potential in improving the performance of the proposed model by providing less variability in classification results.

TABLE I
PERFORMANCE VALIDATION BASED ON ACCURACY

Techniques	Accuracy (%)
Proposed LPMLP	98.6748
MLP	96.1845
RBFN	93.2659
RBM	90.4878
DBN	88.6545

Table I compares the accuracy values obtained by the proposed and existing models. The accuracy of the proposed LPMLP model is 98.6748%. But, the accuracies of the existing MLP, RBFN, RBM, and DBN are decreased by 2.49%, 5.40%, 8.19%, and 10.02%, respectively. This shows the superiority of the proposed system. The modification of activation and parameter optimization has improved the overall performance of the recommendation model.

In above Fig. 5, the outcomes of the proposed and the existing methods are analyzed. The conventional MLP, RBFN, RBM, and DBN models offer lower precision, recall, and F-measure values. But, the proposed method achieves higher performance values of 97.6589% (precision), 98.4578% (recall), and 97.2659% (F- measure). Overall, the performance results reveal that the Linear Saturated Activation Function in conventional MLP has improved the recommendation performance. Fig.6 shows the performance of proposed and existing classifiers using the performance measures, such as False Positive Rate (FPR), False Negative Rate (FNR), and

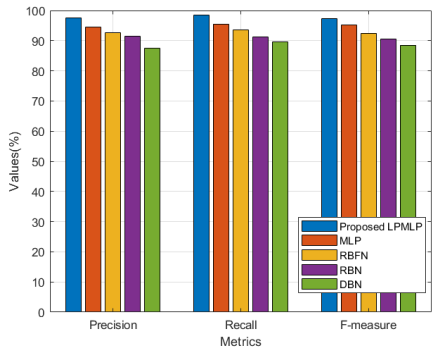

Fig. 5. Performance analysis of the proposed model

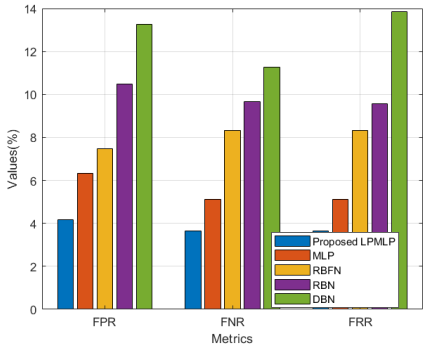

Fig. 6. Performance Measure based onFPR, FNR, and FRR

False Reject Rate (FRR). While analyzing the Fig.6, the FPR (4.1578%), FNR (3.6589%), and FRR (3.6548%)attained by the proposed model is very much lower than the existing methods. Minimum values of these three metrics showed the efficiency of the proposed model in classification and recommendation.

TABLE II
COMPARISON OF TRAINING TIME

Techniques	Training Time (ms)
Proposed LPMLP	58647
MLP	63984
RBFN	67481
RBM	72348
DBN	78653

The performance analysis of the proposed LPMLP and the existing methods with respect to the training time of the networks is represented in table II. The time consumed for training should be low to prove the efficiency of the neural networks. Accordingly, the proposed LPMLP method consumed a training time of 58647ms, whereas the existing methods showed an increased time difference than the proposed network. From the analysis, it is clear that the proposed model has efficiently reduced the training time to a greater extent and yields better performance than the existing methods.

C. Performance Analysis of Clustering

The performance of the proposed RAD-FCM technique is assessed with the prevailing methods, such as FCM, K-Means, Partition Around Medoid (PAM), and Clustering Large Applications (CLARA) based on the clustering time and

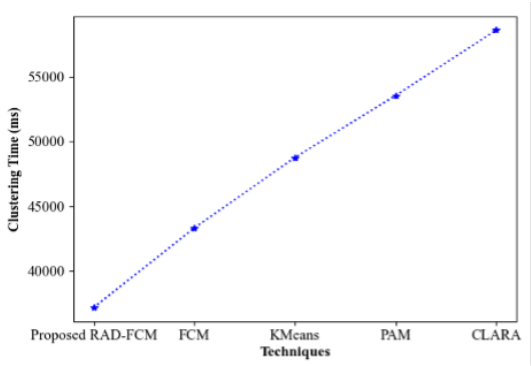

Fig. 7. Clustering Time Analysis

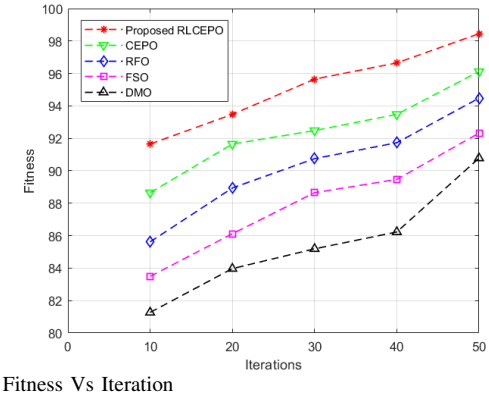

Fig. 8. Fitness Vs Iteration

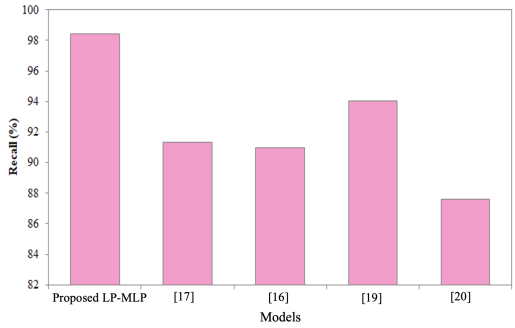

Fig. 9. Recall Analysis with Literature Papers

accuracy. The time consumed for clustering by the proposed model and the existing models is shown in Fig.7. The time taken for clustering must be low to prove the efficiency of the clustering technique. The proposed RAD-FCM method consumed a clustering time of 37152ms, which is lower compared to the existing models having clustering times of 43298ms (FCM), 48756ms (K-Means), 53548ms (PAM), and 58611ms (CLARA). So, from the results, it is concluded that the RAD computation in the proposed model provided efficient results in anomaly clustering. The outcomes of the proposed and prevailing techniques based on clustering accuracy are shown in table III. The analysis clearly shows that the clustering accuracy of the proposed model is 1.76% greater than the existing FCM. Similarly, the proposed RAD-FCM outperformed the other models too. Thus, the centroid computation using the rectangular area division process in RAD-FCM has improved the accuracy of the cluster formation.

TABLE III
CLUSTERING ACCURACY ANALYSIS

Techniques	Clustering Accuracy(%)
Proposed RAD-FCM	97.88
FCM	96.12
K-Means	94.48
PAM	90.65
CLARA	87.37

D. Performance Analysis of Feature Selection

This section compares the performance of the proposed RLCEPO algorithm with the prevailing CEPO, Red Fox Optimization (RFO), Fish Swarm Optimization (FSO), and Dwarf Mongoose Optimization (DMO) Algorithms. Fig.8 depicts the fitness values attained at different iterations for feature selection. Here, the fitness of the proposed model corresponds to the number of 50 iterations is 98.45, which is higher than the existing methods. Similarly, the proposed model obtained higher fitness than the existing ones for all number of iterations. Thus, the analysis concludes that the Refraction Learning strategy in the proposed optimization had helped in jumping out of the local optimum, resulting in efficient feature optimization.

E. Comparative Assessment Based on Literature Papers

In this phase, the performance of the proposed model is validated with the existing works suggested by [16], [17], [19], [20] based on recall values obtained by using the ISCXVPN2016 dataset. Fig.9 depicts the recall values obtained by the proposed and existing models. The proposed model has considered the time series features, their issues summarization, and metrics data to identify and resolve issues in VPN networks. The novel steps included in the proposed work have improved the recall value to 98.45%. The existing works performed network traffic classification but failed to identify the issues, which in turn impacted their recall values. Thus, the analysis clearly shows the superiority of the proposed work.

V. CONCLUSION

In this paper, a secure RAD-FCM and LPMLP-adapted IPsec VPN cloud Gateway with intelligent Network Monitoring and Issue Resolution are proposed. The proposed framework undergoes various phases like pre-processing, big data handling, pattern extraction, event scoring, clustering, issue summarization, feature selection, metric data processing, and finally, a recommendation based on the prior processes. Then, the performance of the proposed system is explored, where the proposed RAD-FCM, LPMLP, and RLCEPO approaches were compared with the existing methods to show the efficiency of the proposed model. From the analysis, the recommendation efficiency and issue detection rate of the proposed LPMLP is 97.98% and 98.26%. Further, the proposed model obtained lower MAPE and MSE values. Thus, from the overall evaluation, it is clear that the proposed system has outperformed the comparable systems in terms of all metrics. However, the work did not prioritize the security of data transmission in VPN cloud gateways. In the future, we can propose advanced DL models to ensure secure data transmission in VPNs.

REFERENCES

[1] A. Azab, M. Khasawneh, S. Alrabaee, K.-K. R. Choo, and M. Sarsour, "Network traffic classification: Techniques, datasets, and challenges," *Digital Communications and Networks*, 2022.

[2] S. P. Praveen, T. B. M. Krishna, S. K. Chawla, and C. Anuradha, "Virtual private network flow detection in wireless sensor networks using machine learning techniques," *International Journal of Sensors Wireless Communications and Control*, vol. 11, no. 7, pp. 716–724, 2021.

[3] K. M. A. Kamal and S. Almuhammadi, "Vulnerability of virtual private networks to web fingerprinting attack," in *Advances in Security, Networks, and Internet of Things: Proceedings from SAM'20, ICWN'20, ICOMP'20, and ESCS'20*. Springer, 2021, pp. 147–165.

[4] M. Pudelko, P. Emmerich, S. Gallenmüller, and G. Carle, "Performance analysis of vpn gateways," in *2020 IFIP Networking Conference (Networking)*. IEEE, 2020, pp. 325–333.

[5] R. Bansode and A. Girdhar, "Common vulnerabilities exposed in vpn–a survey," in *Journal of Physics: Conference Series*, vol. 1714, no. 1. IOP Publishing, 2021, p. 012045.

[6] F. Hauser, M. Häberle, M. Schmidt, and M. Menth, "P4-ipsec: Site-to-site and host-to-site vpn with ipsec in p4-based sdn," *IEEE Access*, vol. 8, pp. 139 567–139 586, 2020.

[7] A. Telikani, A. H. Gandomi, K.-K. R. Choo, and J. Shen, "A cost-sensitive deep learning-based approach for network traffic classification," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 661–670, 2021.

[8] M. Elnawawy, A. Sagahyroon, and T. Shanableh, "Fpga-based network traffic classification using machine learning," *IEEE Access*, vol. 8, pp. 175 637–175 650, 2020.

[9] D. Spiekermann and J. Keller, "Unsupervised packet-based anomaly detection in virtual networks," *Computer Networks*, vol. 192, p. 108017, 2021.

[10] A. Shahraki, M. Abbasi, A. Taherkordi, and A. D. Jurcut, "Active learning for network traffic classification: a technical study," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 422–439, 2021.

[11] A. S. Iliyasu and H. Deng, "Semi-supervised encrypted traffic classification with deep convolutional generative adversarial networks," *IEEE Access*, vol. 8, pp. 118–126, 2019.

[12] M. M. Raikar, S. Meena, M. M. Mulla, N. S. Shetti, and M. Karanandi, "Data traffic classification in software defined networks (sdn) using supervised-learning," *Procedia Computer Science*, vol. 171, pp. 2750–2759, 2020.

[13] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (ntma): A survey," *Computer Communications*, vol. 170, pp. 19–41, 2021.

[14] F. Sarhangian, R. Kashef, and M. Jaseemuddin, "Efficient traffic classification using hybrid deep learning," in *2021 IEEE International Systems Conference (SysCon)*. IEEE, 2021, pp. 1–8.

[15] Z. Xu and J. Ni, "Research on network security of vpn technology," in *2020 International Conference on Information Science and Education (ICISE-IE)*. IEEE, 2020, pp. 539–542.

[16] S. A. Aswad and E. Sonuç, "Classification of vpn network traffic flow using time related features on apache spark," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE, 2020, pp. 1–8.

[17] A. Almomani, "Classification of virtual private networks encrypted traffic using ensemble learning algorithms," *Egyptian Informatics Journal*, vol. 23, no. 4, pp. 57–68, 2022.

[18] S. Shunmuganathan, R. D. Saravanan, and Y. Palanichamy, "Securing vpn from insider and outsider bandwidth flooding attack," *Microprocessors and Microsystems*, vol. 79, p. 103279, 2020.

[19] L. Guo, Q. Wu, S. Liu, M. Duan, H. Li, and J. Sun, "Deep learning-based real-time vpn encrypted traffic identification methods," *Journal of Real-Time Image Processing*, vol. 17, pp. 103–114, 2020.

[20] A. Parchekani, S. Nouri, V. Shah-Mansouri, and S. P. Shariatpanahi, "Classification of traffic using neural networks by rejecting: a novel approach in classifying vpn traffic," *arXiv preprint arXiv:2001.03665*, 2020.

[21] A. A. Afuwape, Y. Xu, J. H. Anajemba, and G. Srivastava, "Performance evaluation of secured network traffic classification using a machine learning approach," *Computer Standards & Interfaces*, vol. 78, p. 103545, 2021.

[22] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related," in *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, 2016, pp. 407–414.