Deep Learning-Based Mixture-of-Experts Model for Enhanced Network Traffic Classification

Chandroth Jisi
Dept. of AI Convergence Network
Ajou University
Suwon, South Korea
jisichandroth@ajou.ac.kr

Byeong-hee Roh
Dept. of AI Convergence Network
Ajou University
Suwon, South Korea
bhroh@ajou.ac.kr

Jehad Ali
Dept. of AI Convergence Network
Ajou University
Suwon, South Korea
jehadali@ajou.ac.kr

Maira Khalid
Dept. of AI Convergence Network
Ajou University
Suwon, South Korea
mairakhalid@ajou.ac.kr

Ahmed Raza Mohsin
Dept. of AI Convergence Network
Ajou University
Suwon, South Korea
ahmedraza@ajou.ac.kr

***Abstract*—Traffic classification is crucial for various aspects of network management, including network security, Quality of Service (QoS), and resource allocation. While deep learning (DL) models have demonstrated effectiveness in traffic classification, each type of DL model captures different feature sets, potentially missing essential information. For example, a dense network might excel at identifying general patterns, while a CNN focuses on spatial features, and an LSTM captures temporal dependencies. However, relying on a single DL model may result in incomplete feature extraction. To address this limitation and enhance classification performance, we propose an ensemble model that combines three DL architectures: a dense network, a CNN, and an LSTM. This ensemble method leverages the strengths of each model, enabling comprehensive feature extraction that incorporates spatial, temporal, and generalized patterns. By integrating these diverse feature sets, our ensemble model improves traffic classification accuracy and enhances overall system performance.**

***Index Terms*—Traffic classification, Deep learning, Mixture of Expert, Ensemble learning.**

I. Introduction

Traffic classification is a crucial aspect of modern network management, facilitating enhanced network security, Quality of Service (QoS) optimization, resource allocation, and anomaly detection. By categorizing network traffic, classification methods enable the prioritization of critical data flows, such as latency-sensitive applications like video conferencing and VoIP, while identifying potential security threats and anomalies [1]. In high-demand environments, such as cloud and edge computing, traffic classification dynamically allocates resources based on flow requirements, improving performance and reliability [2]. Additionally, in software-defined networks (SDN), traffic classification predicts optimal routes to maintain QoS across diverse flows [3], [4]. Despite its importance, challenges remain, particularly with the increasing prevalence of encrypted traffic protocols, which limit traditional inspection-based methods, and the need for scalable real-time classification to address evolving traffic patterns and applications [2], [5].

Deep learning (DL)-based traffic classification methods have become popular for their ability to extract complex patterns from data with high accuracy [6]. For instance, convolutional neural networks (CNNs) are effective at capturing spatial features, while recurrent neural networks (RNNs) excel at analyzing sequential patterns [7]. However, single DL models often fail to capture the full range of featuresspatial, temporal, and generalizednecessary for comprehensive analysis. Ensemble learning addresses this limitation by combining multiple models to leverage their individual strengths [8]. The Mixture-of-Experts (MoE) approach extends this concept by dynamically weighting the contributions of specialized models, or "experts," through a gating network. This selective emphasis on relevant experts enhances accuracy and adaptability, making MoE particularly effective for the diverse and complex requirements of modern network traffic classification [9].

To address the limitations of feature extraction when using a single deep learning method, we employ an ensemble approach. In this paper, we propose a MoE model that combines three well-known deep learning architectures: dense networks, CNNs, and long short-term memory networks (LSTMs). The main contributions of this paper are as follows:

- We propose a Mixture-of-Experts (MoE) model that combines dense networks, CNNs, and LSTMs to harness their unique feature extraction capabilities. This approach enables the model to capture a comprehensive range of feature types, leveraging the strengths of each architecture to extract spatial, temporal, and generalized features from the data.
- We introduce Full Input with Specialized Models strategy, where each expert processes the complete input data through its unique architecture, enabling it to interpret the information in distinct ways. This approach allows each

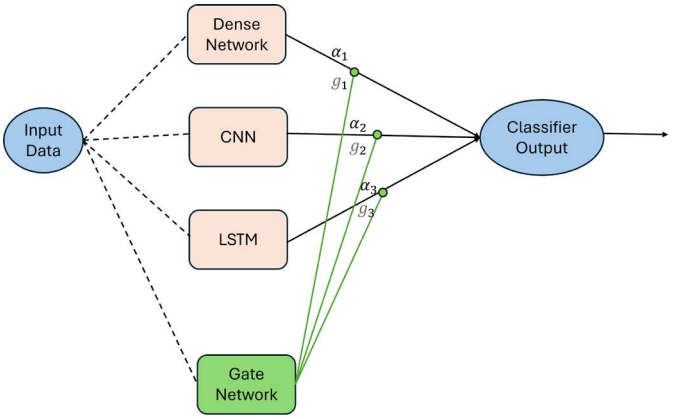

Fig. 1. Proposed MoE architecture

model to learn diverse patterns from the data, leveraging the strengths of its specific architecture to capture a wide range of feature types.

- We present a cooperative prediction approach, which combines outputs from all experts. The gating network orchestrates this process, balancing the contributions from each expert to optimize the final prediction.

II. Proposed Method

We propose a novel Mixture-of-Experts (MoE) model for network traffic classification that combines multiple deep learning architectures to capture diverse feature types. Traditional deep learning models often excel at specific tasks, such as convolutional neural networks (CNNs) identifying spatial features or long short-term memory networks (LSTMs) capturing temporal patterns, but struggle to generalize across complex data. Dense networks, meanwhile, offer generalized representations across a broad feature space. The proposed MoE model integrates Dense Network, CNN, and LSTM architectures into a cooperative ensemble, leveraging their specialized strengths to enhance feature extraction and improve classification performance.

The Mixture of Experts (MoE) architecture shown in Figure 1 is composed of two main components: expert models, and a gating network. The input data is fed into the system, allowing each expert model to interpret the same data independently. The expert models include a Dense Network, a CNN, and a LSTM network, each specialized in extracting different feature types. The gating network then assigns dynamic weights to each experts output based on the inputs characteristics, allowing the model to adaptively emphasize certain experts over others.

The core of our model lies in its Full Input with Specialized Models strategy, where each expert model comprising a Dense Network CNN, and LSTM processes the entire input independently, extracting unique patterns based on its architecture. Let the input data be represented as $X \in R^{n \times m}$, where n is the number of samples and m is the number of features per sample. Each input sample is processed independently by all expert models:

The Dense Network captures generalized patterns across the input data, with layers structured to condense high-level features through a series of fully connected layers. The Dense Network output be denoted as

$$\alpha_1 = f_{\text{Dense}}(X),$$

$$f_{\text{Dense}}(\mathbf{X}) = \text{Softmax}(\mathbf{W}_D \cdot \phi_D(\mathbf{X}) + \mathbf{b}_D), \quad (1)$$

where

$$\phi_D(\mathbf{X}) = \text{ReLU}(\mathbf{W}_{D2} \cdot \text{ReLU}(\mathbf{W}_{D1} \cdot \mathbf{X} + \mathbf{b}_{D1}) + \mathbf{b}_{D2}), \quad (2)$$

and $\mathbf{W}_D, \mathbf{b}_D, \mathbf{W}_{D1}, \mathbf{W}_{D2}, \mathbf{b}_{D1}, \mathbf{b}_{D2}$ are the weights and biases of the Dense Network.

The *CNN* extracts spatial features through a series of convolutional layers that emphasize local, spatial relationships within the data. This is particularly useful for traffic patterns where spatial structure or patterns in sequences are significant. The CNN output is represented as

$$\alpha_2 = f_{\text{CNN}}(X),$$

where the CNN layers focus on identifying spatial dependencies by convolving across the input sequences.

$$f_{\text{CNN}}(\mathbf{X}) = \text{Softmax}(\mathbf{W}_C \cdot \phi_C(\mathbf{X}) + \mathbf{b}_C), \quad (3)$$

where

$$\phi_C(\mathbf{X}) = \text{Flatten}(\text{MaxPooling}(\text{ReLU}(\text{Conv1D}(\mathbf{X})))), \quad (4)$$

and $\mathbf{W}_C, \mathbf{b}_C$ are the weights and biases, while $\text{Conv1D}, \text{MaxPooling}, \text{Flatten}$ represent convolutional layers, pooling layers, and flattening operations, respectively.

Lastly, the LSTM captures temporal dependencies in the data, identifying sequential patterns that may indicate particular trends over time. This temporal representation is denoted as

$$\alpha_3 = f_{\text{LSTM}}(X).$$

$$f_{\text{LSTM}}(\mathbf{X}) = \text{Softmax}(\mathbf{W}_L \cdot \phi_L(\mathbf{X}) + \mathbf{b}_L), \quad (5)$$

where

$$\phi_L(\mathbf{X}) = \text{ReLU}(\text{LSTM}(\mathbf{X})), \quad (6)$$

and $\mathbf{W}_L, \mathbf{b}_L$ are the weights and biases, while LSTM represents the hidden states computed by the LSTM layer.

Thus, each model (Dense, CNN, and LSTM) contributes a distinct perspective, extracting general, spatial, and temporal features, respectively.

To combine the outputs of these specialized experts, a gating network dynamically assigns weights to each experts output, producing weight values g_1, g_2, and g_3 that are based on the relevance of each expert to the input data. Mathematically, the gating network is formulated as:

$$[g_1, g_2, g_3] = \text{softmax}(W_g X + b_g),$$

where W_g and b_g are the weights and bias parameters of the gating networks final dense layer, and the softmax function

ensures that $g_1 + g_2 + g_3 = 1$. This gating mechanism determines the contribution of each expert, making the model adaptive to varying input patterns.

The final prediction y_{pred} is computed as a weighted sum of the expert outputs, represented by:

$$y_{\text{pred}} = g_1 \cdot \alpha_1 + g_2 \cdot \alpha_2 + g_3 \cdot \alpha_3.$$

This cooperative prediction, controlled by the gating network, ensures that the model dynamically emphasizes the most relevant features from each expert, achieving a balanced and adaptable approach to classification across diverse data patterns.

The proposed model utilizes the categorical cross-entropy loss ($\mathcal{L}$) function to calculate the classification error, ensuring effective optimization during the training process.

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log(\hat{y}_{ij}), \tag{7}$$

where y_{ij} is the true label for class j and $\hat{y}_{ij}$ is the predicted probability for class j.

To prevent overfitting, a regularization term $\mathcal{R}$ is added to the loss function:

$$\mathcal{R} = \lambda \sum_{w \in \mathbf{W}} \|w\|^2, \tag{8}$$

where λ is the regularization strength, and $\mathbf{W}$ represents all the trainable weights in the model.

The total loss $\mathcal{L}_{\text{total}}$ for the MoE model is given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \mathcal{R}. \tag{9}$$

III. Performance analysis

A. Dataset and Experimental Setup

For this study, the ISCX VPN-nonVPN dataset was utilized to evaluate the performance of the proposed MoE model. This benchmark dataset provides labeled traffic flows categorized into VPN and non-VPN classes, encompassing diverse application types such as browsing, streaming, and file transfer, with detailed packet-level features like length and inter-arrival time. All experiments were conducted on a 12th Gen Intel(R) Core(TM) i5-12600K PC with a 3.69 GHz CPU and 48 GB of RAM. Data preprocessing was carried out using Python libraries `Pandas` and `NumPy`, while the MoE model was implemented with `TensorFlow`/`Keras` for model building and `scikit-learn` for evaluation.

Table I provides a comprehensive description of the proposed MoE model architecture. It highlights the layers, parameters, activation functions, and output shapes for each component, including the Dense Network, CNN, LSTM, and Gating Network.

B. Results

Figure 2 shows the comparative accuracy of individual deep learning models, Dense Network, CNN, and LSTM, alongside our proposed Mixture-of-Experts (MoE) model. The results indicate that the MoE model outperforms each individual model, achieving an accuracy of approximately 99.3%. This superior performance is due to the ensemble approach of the MoE model, which combines the unique strengths of each expert model. By integrating these models, the MoE model captures a broader range of feature types such as general, spatial, and temporalthat a single model alone might miss. Additionally, the MoE models cooperative prediction strategy, driven by a gating network, adaptively assigns weights to each expert, emphasizing the most relevant features for each input. This dynamic weighting enables the MoE model to achieve higher classification accuracy overall.

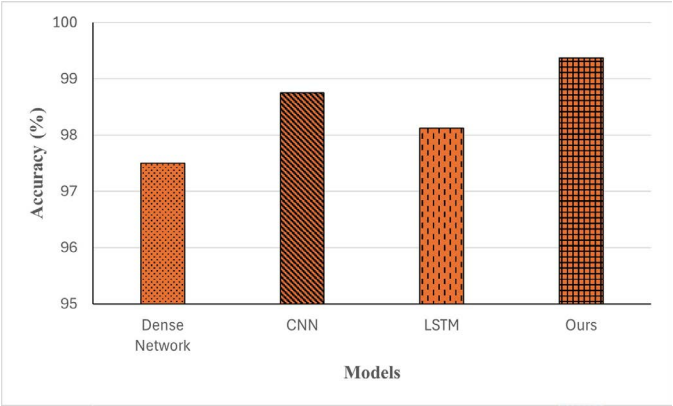

Fig. 2. Overall accuracy of the models

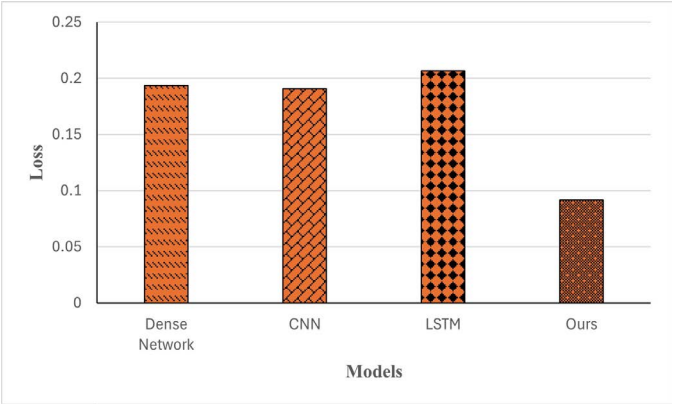

Fig. 3. Loss of the models

Figure 3 illustrates a comparison of the loss values for individual deep learning modelsDense Network, CNN, and LSTM, along with our proposed Mixture-of-Experts (MoE) model. The MoE model achieves noticeably lower loss compared to each individual model, highlighting its improved alignment with the data and overall performance.

Figure 4 displays the accuracy trends of individual models and our proposed model over a series of training epochs. It is evident that while each models accuracy improves with additional epochs, the MoE model consistently achieves the highest accuracy at each stage of training. By the end of training, the MoE model surpasses the individual models, stabilizing at a near-perfect accuracy level. This superior performance of

TABLE I
DETAILED ARCHITECTURE OF THE PROPOSED MOE MODEL WITH m=40, NUMBER OF CLASSES =20

Component	Layer Type	Parameters	Activation Function	Output Shape
Dense Network	Dense Layer 1	128 units, kernel size: 40	ReLU	(n, 128)
	Dropout Layer	Dropout rate: 0.3	-	(n, 128)
	Dense Layer 2	64 units	ReLU	(n, 64)
	Dropout Layer	Dropout rate: 0.3	-	(n, 64)
	Dense Layer 3	32 units	ReLU	(n, 32)
	Output Layer	1 ×20	Softmax	(n, 20)
CNN	Conv1D Layer	64 filters, kernel size: 3, stride: 1	ReLU	(n, 38, 64)
	MaxPooling1D Layer	Pool size: 2	-	(n, 19, 64)
	Dropout Layer	Dropout rate: 0.3	-	(n, 19, 64)
	Flatten Layer	-	-	(n, 1216)
	Dense Layer	64 units	ReLU	(n, 64)
	Output Layer	1 ×20	Softmax	(n, 20)
LSTM	LSTM Layer	64 units	-	(n, 64)
	Dropout Layer	Dropout rate: 0.3	-	(n, 64)
	Dense Layer	32 units	ReLU	(n, 32)
	Output Layer	1 ×20	Softmax	(n, 20)
Gating Network	Dense Layer 1	64 units, kernel size: 40	ReLU	(n, 64)
	Output Layer	3 ×20	Softmax	(n, 3)

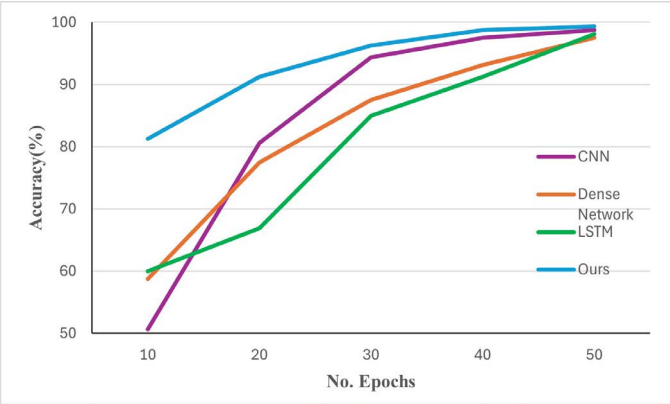

Fig. 4. Accuracy of models vs epochs

the MoE model highlights its ability to leverage the strengths of each expert model through adaptive weighting, capturing a more diverse set of features from the data. Consequently, the MoE model not only converges faster but also achieves greater accuracy than any single model, demonstrating its effectiveness in complex classification tasks.

IV. CONCLUSION

We propose a Mixture-of-Experts (MoE) model for network traffic classification, combining Dense Network, CNN, and LSTM architectures. Each serves as an expert, with a gating network dynamically assigning weights based on input characteristics. This setup leverages the unique feature extraction strengths of each model to enhance classification performance.

To maximize the model's effectiveness, a *full-input-to-expert* strategy was employed, which allowed each expert to process the entire input data independently, providing a comprehensive initialization and ensuring that each expert had access to all relevant features. Furthermore, a *cooperative prediction* approach was implemented, enabling the MoE model to balance the outputs from each expert based on the gating networks weight assignments, thereby optimizing classification accuracy. The experimental results demonstrate that the proposed MoE method effectively classifies multiple traffic classes, achieving an accuracy of 99.3%. This accuracy rate is notably higher than that of individual stand-alone models, confirming the effectiveness of the MoE approach.

ACKNOWLEDGMENT

This work was supported partially by the BK21 FOUR program of the National Research Foundation of Korea funded by the Ministry of Education (NRF5199991514504).

REFERENCES

[1] Chandroth Jisi, Byeong-hee Roh, Jehad Ali, "An effective scheme for classifying imbalanced traffic in SD-IoT, leveraging XGBoost and active learning," in Computer Networks, 2024, 110939, doi.org/10.1016/j.comnet.2024.110939.

[2] P. Wang, X. Chen, F. Ye and Z. Sun, "A Survey of Techniques for Mobile Service Encrypted Traffic Classification Using Deep Learning," in IEEE Access, vol. 7, pp. 54024-54033, 2019, doi: 10.1109/ACCESS.2019.2912896.

[3] J. Ali, H. H. Song and B. -h. Roh, "An SDN-Based Framework for E2E QoS Guarantee in Internet-of-Things Devices," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2024.3465609.

[4] Jisi, C., Roh, B. H., Ali, J. (2024). Reliable paths prediction with intelligent data plane monitoring enabled reinforcement learning in SD-IoT. Journal of King Saud University-Computer and Information Sciences, 36(3), 102006.

[5] W. Lin and Y. Chen, "Robust Network Traffic Classification Based on Information Bottleneck Neural Network," in IEEE Access, vol. 12, pp. 150169-150179, 2024, doi: 10.1109/ACCESS.2024.3477466.

[6] Saadat Izadi, Mahmood Ahmadi, Rojia Nikbazm,"Network traffic classification using convolutional neural network and ant-lion optimization",in Computers and Electrical Engineering, Vol. 101, 2022, doi.org/10.1016/j.compeleceng.2022.108024.

[7] Xinming Ren, Huaxi Gu, Wenting Wei, "Tree-RNN: Tree structural recurrent neural network for network traffic classification," in Expert Systems with Applications, Vol. 167, 2021, doi.org/10.1016/j.eswa.2020.114363.

[8] Ola Salman, Imad H. Elhajj, Ali Chehab, Ayman Kayssi, "Towards efficient real-time traffic classifier: A confidence measure with ensemble Deep Learning," Computer Networks, Volume 204, 2022, doi.org/10.1016/j.comnet.2021.108684.

[9] Jacobs, Robert A., et al. "Adaptive mixtures of local experts." Neural computation 3.1 (1991): 79-87.