A Custom Deep Learning Approach for Traffic Flow Prediction in Port Environments: Integrating RCNN for Spatial and Temporal Analysis

Abdul Basit Ali Shah[1], Xinglu Xu[2], Zheng Yongren[3], Zijian Guo[4]
School of Infrastructure-Coastal and Offshore Engineering, Dalian University of Technology, Dalian, Liaoning, 116024, China[1]
State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology, Dalian, Liaoning, 116024, China[2, 4]
Caofeidian Port Business and Economic Zone Management Office, China[3]

***Abstract*—Port congestion poses a significant challenge to maritime logistics, especially for industries dealing with perishable goods like seafood. This study presents a custom deep learning model using Transformer architecture to predict real-time traffic flow at the Port of Virginia, with a focus on optimizing the movement of fish trucks. The model integrates multimodal data from 36 sensors, capturing traffic flow, occupancy, and speed at five-minute intervals, and processes high-dimensional, time-series data for accurate predictions. The model utilizes attention mechanisms to capture spatial and temporal dependencies, significantly improving predictive performance. Evaluation results indicate that the Transformer-based model outperforms existing models like RandomForest, GradientBoosting, and Support Vector Regression, with an R-squared value of 0.89, Pearson correlation of 0.91, and a Root Mean Squared Error (RMSE) of 0.0208. These results suggest that the model can effectively manage dynamic port traffic and optimize resource allocation, ensuring the timely delivery of perishable goods.**

Keywords—Traffic flow prediction; transformer model; port congestion; deep learning

I. Introduction

Port traffic management is a critical component of maritime logistics, playing an essential role in ensuring the smooth operation of global supply chains. Ports act as vital nodes in international trade, facilitating the movement of goods across continents. As the global economy continues to grow, ports must meet increasing demands, especially in handling both bulk and perishable goods. The seafood industry, in particular, relies on efficient port traffic management for the timely transport of fish and other perishable goods. Fish trucks, which move seafood from ports to markets or distribution centers, require fast and reliable service to maintain product quality and minimize spoilage. Delays in port operations lead to significant economic losses and waste, given the perishability of these goods. Thus, improving port traffic management systems—especially for perishable goods—has become a critical concern for ports worldwide [1].

The continued growth of international trade, alongside the increasing volume of goods transported by sea, places increasing pressure on ports to handle rising traffic volumes. Ports are the gateways for goods entering and leaving regions and play a crucial role in economic activities. However, as trade volumes continue to rise, congestion has emerged as a significant issue at many ports. Congestion at ports leads to delayed vessel arrivals, bottlenecks during cargo unloading, and delays in cargo pickup, which are especially problematic for time-sensitive goods like seafood. Fish trucks, dependent on swift port operations, face major disruptions when vessels are delayed, leading to a domino effect in the supply chain. Such delays can compromise product quality, especially in the seafood industry, where the timely movement of goods is crucial for maintaining freshness and minimizing losses [2], [3].

Additionally, vessel congestion often results in inefficient resource allocation, as ports may lack the ability to dynamically allocate resources such as cranes, docking spaces, and labor according to real-time needs. This inefficiency increases operational costs, not only in the form of delayed shipments but also due to the additional resources required to manage the backlog. Ports may also face difficulty in managing fluctuating traffic patterns that are driven by seasonal demand or unexpected weather conditions. The global rise in e-commerce and the associated increase in containerized cargo further exacerbate congestion at many ports, highlighting the need for smarter and more adaptable management systems [4], [5].

Environmental concerns also play an increasing role in port traffic management. Ports are significant contributors to global greenhouse gas emissions due to idling ships and inefficient resource use. As ports handle more cargo, the environmental impact of congestion is amplified. For example, long waiting times for vessels to dock result in fuel waste and greater carbon emissions. Port authorities are thus under increasing pressure to find solutions that not only improve operational efficiency but also minimize the environmental impact. The integration of technologies like AI and machine learning can significantly reduce congestion by providing real-time insights into traffic patterns, allowing ports to make more informed decisions that balance operational efficiency with environmental sustainability [6], [7].

Given these challenges, AI technologies, particularly machine learning and deep learning, are becoming essential tools for improving port traffic management. AI-powered systems can analyze vast amounts of real-time data from multiple sources, such as sensors, Automatic Identification Systems (AIS), and weather reports. These technologies enable the development of predictive models that forecast port congestion, optimize vessel scheduling, and improve resource allocation. AI has the potential to not only predict traffic flows but also adapt to changing port conditions, enabling ports to proactively adjust operations before congestion occurs. Machine learning models, such as support vector machines (SVM), random forests, and deep neural networks, have been applied in various studies to predict traffic patterns and improve port operations. However, these models often focus on general cargo traffic and have not yet fully addressed the specific needs of perishable goods logistics, such as the transport of seafood [8], [9], [10].

Deep learning models, specifically Transformer architectures, offer significant advantages in capturing both spatial and temporal dependencies within port traffic data. These models have been used successfully in various fields for time-series forecasting, where they can process large datasets and make highly accurate predictions. By leveraging multimodal sensor data, deep learning models can predict congestion, identify bottlenecks, and optimize resource allocation for both vessels and cargo handling. This is particularly important in the seafood industry, where timing is critical for ensuring the freshness of the product and minimizing spoilage. Previous studies have demonstrated the application of deep learning techniques for traffic management in other logistics sectors, but their use in optimizing perishable goods transportation within ports remains an underexplored area [11].

This study aims to fill this gap by developing a customized deep learning-based model for port traffic management, specifically focused on optimizing the movement of fish trucks at ports. The proposed model will integrate real-time data from various sources, including traffic flow sensors, vessel tracking systems, and environmental data, to forecast congestion and improve decision-making processes in port operations. By utilizing Transformer-based models, the study seeks to enhance the accuracy of predictions, allowing port authorities to allocate resources efficiently, reduce congestion, and improve the overall efficiency of seafood logistics. Furthermore, this study explores the integration of AI-powered systems into existing port infrastructure, providing actionable insights that will contribute to the sustainable and efficient management of ports [12].

The introduction provides a comprehensive background on port congestion, its impact on global logistics, and the specific challenges faced by the seafood industry in managing port traffic. The study proposes a custom deep learning model leveraging Transformer architecture to improve traffic flow prediction at the Port of Virginia. The outlined structure of the paper should accurately reflect the sections presented. This includes the methodology section, which details data preprocessing, model customization, and feature engineering, followed by the results and evaluation of the proposed model's performance.

II. Literature Review

The efficient management of port traffic has long been a critical issue in maritime logistics. Early studies primarily focused on the operational limitations and inefficiencies caused by congestion in ports. For instance, studies by Chen et al. [10] and Zhang et al. [11] highlighted how poor scheduling and limited docking facilities can lead to vessel delays, which in turn increase waiting times for trucks and cause bottlenecks in port traffic. Traditional methods, such as queuing models and heuristic algorithms, were used in these early studies to improve port scheduling and reduce congestion, but they often lacked the flexibility to handle dynamic, real-time traffic patterns and changing environmental conditions.

The rise of Artificial Intelligence (AI) in port traffic management marks a significant shift in how congestion and logistics are handled. AI technologies, particularly machine learning, have demonstrated significant potential for improving real-time decision-making and predictive analysis in port operations. Machine learning models can process large amounts of data from a variety of sources, such as traffic sensors, weather forecasts, and shipping schedules, to identify patterns and forecast traffic flow [13]. These AI-driven models offer significant improvements over traditional traffic management systems by making real-time predictions and enabling proactive adjustments to scheduling and resource allocation.

In recent years, deep learning methods have gained popularity for their ability to analyze high-dimensional, time-series data. Transformer models, which utilize attention mechanisms to capture long-range dependencies, have shown great promise in forecasting port traffic and predicting vessel arrival times. Xu et al. [14] and Kim et al. [15] applied deep learning models to predict congestion and optimize traffic flow at ports, demonstrating the superiority of these models compared to traditional machine learning approaches. These studies found that deep learning models were able to account for the complex spatial and temporal dynamics of port operations, leading to better predictive accuracy and more efficient decision-making.

However, while machine learning models have shown promise in improving port operations, few studies have specifically focused on the logistics of perishable goods, such as seafood. Seafood is particularly sensitive to delays in transport, as it requires fast processing to preserve product quality and avoid spoilage. A study by Zhang and Liu [16] explored the use of AI to optimize the movement of goods at ports, but its focus was on general cargo rather than perishable goods. Similarly, Yang et al. [17] proposed an AI model for traffic flow optimization, but the model did not account for the time-sensitive nature of products like seafood. Research focusing on perishable goods logistics in ports remains underdeveloped, particularly regarding the use of AI and machine learning to optimize the unloading schedules for fish trucks.

AI's application in the seafood industry remains an underexplored area. In a recent study, Dong et al. [18] explored the use of AI for the cold chain management of perishable goods but did not specifically focus on port congestion. The focus on the seafood supply chain, particularly the role of ports in ensuring timely delivery, remains sparse. Given the sensitivity of seafood to delays, the logistics surrounding fish trucks require more specialized attention, including real-time monitoring of both environmental conditions and traffic flows [19], [20].

Recent work by He et al. [21] and Wang et al. [22] has highlighted the potential of deep learning, particularly Transformer-based architectures, for improving port traffic management. These studies argue that Transformer models excel in managing time-series data, such as traffic flow and port scheduling, due to their ability to capture long-range dependencies and adjust for fluctuations in real-time data. This approach is particularly relevant for managing perishable goods like seafood, where delays can have significant economic and quality implications. The ability of deep learning to predict congestion patterns accurately can help ports optimize resource allocation, improving the timeliness of fish truck unloading and transportation.

Reinforcement learning (RL) has also emerged as a promising technique in optimizing port traffic management. RL models, which learn optimal strategies through trial and error, have been used for berth scheduling and crane allocation. A study by Li et al. [23] applied reinforcement learning to port scheduling and found it to be more effective in reducing congestion than traditional methods. Similarly, Zhang et al. [24] explored the use of RL in the coordination of vessel movements within ports, demonstrating its ability to reduce waiting times and improve traffic flow. However, RL applications have yet to be fully explored for the specific needs of perishable goods, particularly seafood, where the cost of delays is high.

Despite the advancements in AI and machine learning for port traffic management, challenges remain in integrating these technologies into existing port infrastructures. Research by Zhou et al. [25] suggests that integrating AI-driven systems into legacy port systems presents significant challenges, including data quality, system reliability, and resistance to technological change. Further research is needed to address these integration challenges and ensure that AI-driven solutions are scalable and adaptable to the dynamic nature of port operations.

III. METHODOLOGY

The methodology for traffic prediction involves a series of structured steps starting with the collection of raw traffic data from multiple sensors. The initial step is Data Preprocessing, where essential tasks such as normalization, handling missing data, and outlier detection are performed to ensure the data is clean and ready for model training. Following preprocessing, a Custom Deep Learning Model is designed to handle both the spatial and temporal aspects of the traffic data, leveraging techniques like Convolutional and Recurrent Neural Networks. The final step is Model Evaluation, where the performance of the model is assessed based on prediction accuracy using metrics such as RMSE and Spearman's Rank Correlation.

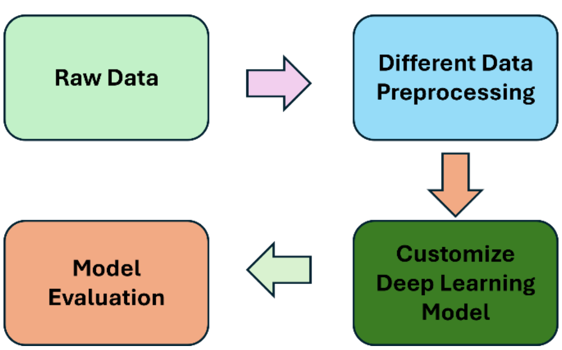

Fig. 1. Flow chart.

Fig. 1 provides a visual summary of these steps, highlighting the progression from raw data, through preprocessing, model customization, and evaluation.

A. Dataset and Preprocessing

The dataset for this study is sourced from Kaggle and contains multimodal traffic data collected from 36 sensors strategically placed across key locations within the Port of Virginia. These sensors record three key variables: traffic flow (number of vehicles passing through the sensor in a given time interval), occupancy (percentage of time the sensor is occupied by a vehicle), and speed (average vehicle velocity).

These sensors record three key variables: traffic flow (number of vehicles passing through the sensor in a given time interval), occupancy (percentage of time the sensor is occupied by a vehicle), and speed (average vehicle velocity). The data consists of high-dimensional, time-series information, recorded at five-minute intervals over several days, presenting challenges such as temporal dependencies, missing data, and outliers. To prepare the dataset for modeling, several preprocessing steps were applied:

- Normalization: Data normalization is essential due to the different scales of the features (flow, occupancy, and speed). MinMaxScaler is used to scale the data between 0 and 1. The normalization formula is as follows [see Eq.(1)]:

$$X_{\text{norm}} = \frac{X-\mu}{\sigma} \tag{1}$$

- Where X represents the feature matrix, μ is the mean, and σ is the standard deviation of each feature. For each feature X_{i}, the mean μ_i and standard deviation σ_i are calculated as shown in Eq. (2) and Eq. (3):

$$\mu_i \& = \frac{1}{m}\sum_{j=1}^{m} X_{ji} \tag{2}$$

$$\sigma_i \& = \sqrt{\frac{1}{m}\sum_{j=1}^{m} \left(X_{ji} - \mu_i\right)^2} \tag{3}$$

- Handling Missing Data: Time-series data often has missing values due to sensor malfunctions or transmission issues. Imputation techniques, such as filling missing values with the mean or using interpolation, were applied to maintain data continuity.

- Outlier Detection and Removal: Outliers, which can skew the distribution and negatively impact model performance, were detected and handled by clipping extreme values or applying log scaling. The histogram visualizations of flow, occupancy, and speed, shown in Fig. 2, illustrate the data distribution after outlier removal.

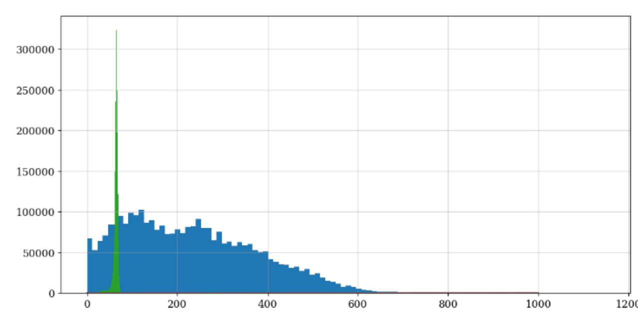

Fig. 2. Outlier detection and removal.

- Time-Series Grouping and Feature Engineering: To capture long-term traffic patterns and smooth out short-term fluctuations, the data was aggregated into hourly segments, with 12 five-minute intervals combined into one hour. Additionally, one-hot encoding was applied to capture daily patterns, and lag features were created to help the model understand how past traffic conditions influence the current state.

- Exploratory Data Analysis (EDA): A periodogram, as illustrated in Fig. 3, was generated to analyze seasonality in the dataset. This analysis revealed distinct recurring patterns in traffic flow and occupancy on weekly, daily, and hourly intervals. Identifying and analyzing these seasonal trends is crucial for understanding the underlying traffic behavior, enabling the model to make more accurate predictions by accounting for periodic variations.

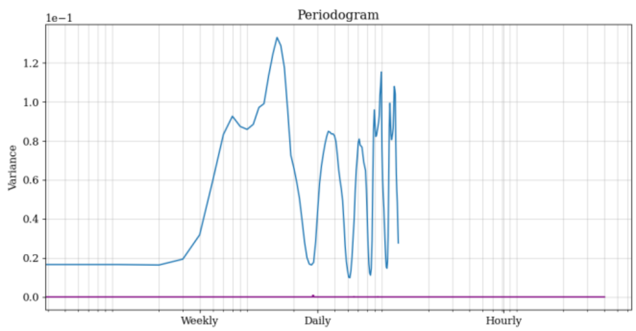

Fig. 3. Periodogram.

The overall preprocessing steps are outlined in Algorithm 1. This includes data normalization, missing value handling, and feature extraction for traffic flow prediction.

Algorithm 1: Preprocess the Input Data

```
def preprocess_traffic_data(data, normalization=True):
    """
    Preprocesses traffic data for model training. This includes
handling missing data,
    normalization, and reshaping the data for time-series forecasting.
    Args:
        data: The raw CSV data containing traffic information (flow,
speed, occupancy).
        normalization: Whether to normalize the data features
(default: True).
    Returns:
        The preprocessed traffic data ready for model input.
    """
    # Handle missing data (example: fill missing values with the
column mean)
    data.fillna(data.mean(), inplace=True)
    # Normalize the data features if required
    if normalization:
        scaler = MinMaxScaler()
        data_scaled = scaler.fit_transform(data
data = pd.DataFrame(data_scaled, columns=data.columns)
    # Prepare the data for time-series modeling
    # Example: Convert to time windows (using a sliding window
for input sequences)
    X, Y = create_sliding_windows(data, window_size=24)    #
  Adjust window size based on your requirements
    return X, Y
```

B. Custom Deep Learning Model

The first component of the proposed deep learning model is the spatial feature extraction layer, which uses a 1D convolutional layer (Conv1D). This layer plays a crucial role in learning local spatial patterns within the traffic data. It operates by applying a set of learnable filters to the input data, effectively sliding over the spatial dimension (i.e., across time steps and sensor locations). The convolutional operation allows the model to detect local patterns, such as sudden changes in traffic flow or variations in speed. Mathematically, this operation can be described by the following Eq. (4):

$$\mathrm{Conv1D}(x) = f\left(\sum_{i=1}^{K} x[i] \cdot w[i] + b\right) \tag{4}$$

Where $x[i]$ represents the input data within a sliding window. The term $w[i]$ refers to the learnable weights of the convolution filter, where i indicates the specific filter position. The bias term, denoted as b, is added to the convolution output to adjust the final result. The function $f(.)$ is the activation function, typically ReLU (Rectified Linear Unit), which introduces non-linearity into the model. This non-linearity enables the network to learn more complex patterns and adapt to a wider variety of data representations.

The next component of the model is the customized temporal processing layer. This layer is responsible for learning the long-range temporal dependencies in the traffic data, meaning how past traffic conditions influence future patterns. The layer processes sequential data, where the input data at each time step is influenced by the information from previous time steps. The temporal processing mechanism can be represented mathematically as shown in Eq. (5):

$$h_t = f(W \cdot x_t + b) \tag{5}$$

In this Eq. (5), h_t is the output at time step $,x_t$ is the input at time t, W is the learnable weight matrix, and b is the bias term. The function $f(.)$ is an activation function, such as ReLU. This layer preserves important temporal information, allowing the model to use past data to make predictions about future traffic conditions.

After spatial and temporal features have been extracted, the model moves to the dense layers, which process the learned features from both the spatial and temporal layers. These layers integrate the information and allow the model to make the final prediction of traffic flow. The dense layers are followed by dropout layers to reduce overfitting by randomly deactivating neurons during training. This helps ensure that the model generalizes well when exposed to new, unseen data.

In addition to the dense layers, the model includes a transformer-like architecture designed to further enhance the model's ability to capture temporal dependencies. The transformer uses multi-head attention to focus on different parts of the input sequence, allowing the model to learn which time steps are more important for predicting future traffic conditions. The attention mechanism can be expressed mathematically as shown in Eq. (6):

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{Q\cdot K^T}{\sqrt{d_k}}\right)V \tag{6}$$

In this Eq. (6), Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the dimension of the key vector. The attention mechanism helps the model to focus on the most relevant parts of the sequence, improving the model's performance in capturing long-range temporal dependencies.

Finally, the output of the model is the predicted traffic flow at each sensor location, which is obtained by passing the aggregated features through the dense layers. The model's architecture, combining convolutional feature extraction, custom temporal processing, and transformer-based attention, ensures that it captures both local and long-range patterns in the traffic data, making it a powerful tool for accurate traffic flow prediction.

The overall architecture of the proposed deep learning model, which integrates spatial feature extraction, customized temporal processing, and transformer-based attention, can be seen in Fig. 4.

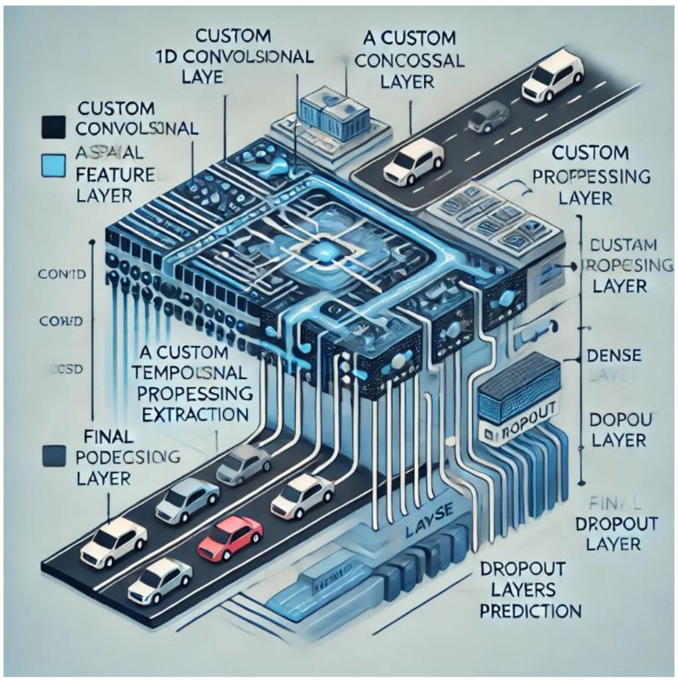

Fig. 4. Custom model architecture.

The architecture effectively captures both local and long-range dependencies in the traffic data, facilitating accurate predictions. The detailed steps involved in processing the data, training the model, and generating the predictions are outlined in Algorithm 2.

Algorithm 2: Custom Deep Learning Model for Traffic Flow Prediction

```
def create_model(input_shape, num_classes=1):
    """
    Creates and compiles a custom deep learning model for traffic flow prediction.
    Args:
        input_shape: The shape of the input data (number of time steps, number of features).
        num_classes: The number of output classes (default: 1 for regression).
    Returns:
        The compiled model ready for training.
    """
    # Step 1: Spatial Feature Extraction Layer (Conv1D)
    # Define a Conv1D layer to capture spatial dependencies within traffic data.
    model = Sequential()
    model.add(Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=input_shape))
    model.add(Dropout(0.2))  # Regularization with dropout to prevent overfitting
    model.add(MaxPooling1D(pool_size=2))  # Max pooling to downsample
    # Step 2: Custom Temporal Processing Layer (Fully Connected Dense Layer)
    # fully connected dense layer for capturing temporal dependencies.
    model.add(Dense(128, activation='relu'))  # Dense layer to capture temporal patterns
    model.add(Dropout(0.2))  # Dropout to avoid overfitting
    # Step 3: Multi-Head Attention Mechanism (Optional, Transformer-like architecture)
    # Adding a simple attention mechanism to focus on important time steps
    model.add(MultiHeadAttention(num_heads=2, key_dim=64)) # Attention layer
    # Step 4: Dense Layers for Final Prediction
    # After spatial and temporal features have been processed, use dense layers for final prediction.
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.3))  # Dropout for regularization
    model.add(Dense(num_classes, activation='linear'))  # Output layer (linear for regression)
    # Step 5: Compile the Model
    # Compile the model with Adam optimizer and MSE loss for regression.
    model.compile(optimizer='adam', loss='mse', metrics=['mae'])
    return model        X, Y = create_sliding_windows(data, window_size=24)  # Adjust window size based on your requirements
    return X, Y
```

IV. Result and Discussion

A. Model Evaluation

After training, the model's performance is evaluated on the test set using two key metrics: Root Mean Squared Error (RMSE) and Spearman's Rank Correlation.

Root Mean Squared Error (RMSE) measures the difference between the predicted and actual values, providing an indication of prediction accuracy. It is calculated as shown in Eq. (7):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{7}$$

Where y_i represents the actual traffic flow, and $\hat{y}_i$ is the predicted value. A lower RMSE indicates better model performance, as it signifies that the predicted values are closer to the actual values.

Spearman's Rank Correlation assesses how well the model preserves the rank order of the actual values. This metric is particularly useful when evaluating the model's ability to capture relative patterns in the data. It is computed as shown in Eq. (8):

$$\rho = 1 - \frac{6\sum_i^2 d}{n(n^2-1)} \quad (8)$$

where d is the rank difference between the actual and predicted values, and n is the number of data points. A higher Spearman correlation (closer to 1) indicates that the model is effectively capturing the relative traffic patterns, even if the exact values are not perfectly predicted.

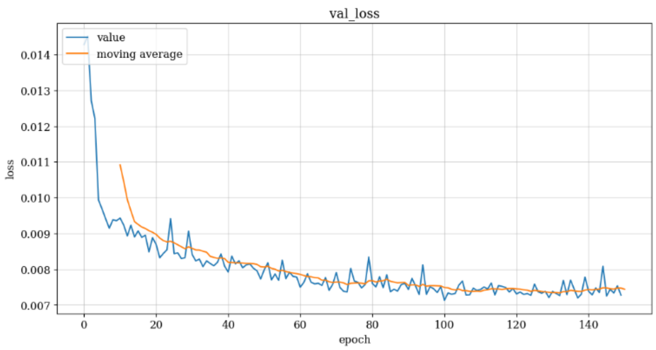

Fig. 5. Validation loss graph of proposed method.

B. Training Performance of the Proposed Model

The Transformer-based model, designed with attention mechanisms to better capture long-range dependencies, was trained over 150 epochs. This model employs multi-head attention layers to focus on different parts of the input data and integrates feed-forward networks to process the spatial and temporal features more effectively. Analyzing the training history graphs, several key trends and insights can be observed regarding the model's learning process.

C. Validation Loss Over Epoch

In Fig. 5, the validation loss begins at a relatively high value, approximately 0.014, indicating substantial discrepancies between the model's initial predictions and the actual traffic data. However, as the training progresses, the validation loss steadily decreases, demonstrating that the model is improving in accuracy. By the 50th epoch, the validation loss stabilizes around 0.008, showing that the learning process has plateaued, and the model has reached a more refined stage of prediction accuracy. The presence of the moving average (orange line) further highlights the overall trend, smoothing out short-term fluctuations in the raw validation loss (blue line). This suggests that the model is learning effectively without encountering overfitting, as the validation loss shows no signs of increasing or erratic behavior towards the later stages of training.

D. Validation RMSE and Training RMSE

A similar pattern is observed in Fig. 6, which depicts the Root Mean Squared Error (RMSE) for both training and validation datasets. Initially, the RMSE is high, indicating a significant error margin in the model's predictions. However, as the epochs progress, the RMSE decreases steadily. By the 100th epoch, the RMSE values for both training and validation datasets converge around 0.085, showing that the model has successfully minimized the error between predicted and actual traffic flow values. The convergence of training and validation RMSE also confirms that the model generalizes well to unseen data, as there is no significant gap between training and validation performance. This stability in RMSE indicates that the model has efficiently learned the underlying patterns in the data, with no signs of overfitting or underfitting.

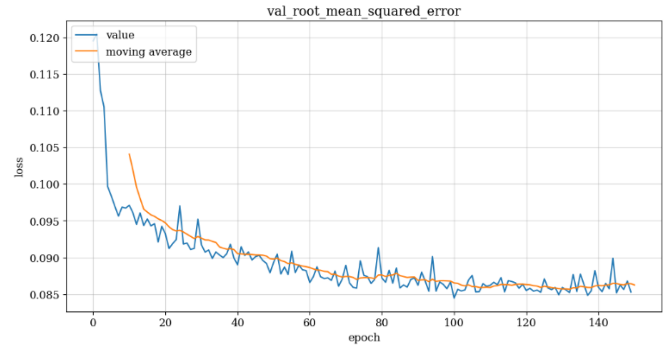

Fig. 6. Validation root mean square error of proposed method.

E. Training Loss and Mean Absolute Error (MAE)

In Fig. 7, the plots comparing training loss and Mean Absolute Error (MAE) reinforce the model's improvement over time. The training loss shows a sharp decline from approximately 0.02 to a much lower value by the end of 150 epochs. This reduction in loss indicates that the Transformer-based model is learning the patterns in the data with increasing precision, minimizing the error between its predictions and the actual values.

F. Training Loss and Mean Absolute Error (MAE)

The Mean Absolute Error (MAE), which measures the average magnitude of prediction errors, also shows a consistent downward trend. This indicates that the model's predictions are becoming increasingly accurate, with fewer large deviations from the actual traffic data. The steady decrease in MAE reflects the model's growing precision in predicting the flow, occupancy, and speed variables in the traffic dataset, which are essential components for accurate traffic flow forecasting.

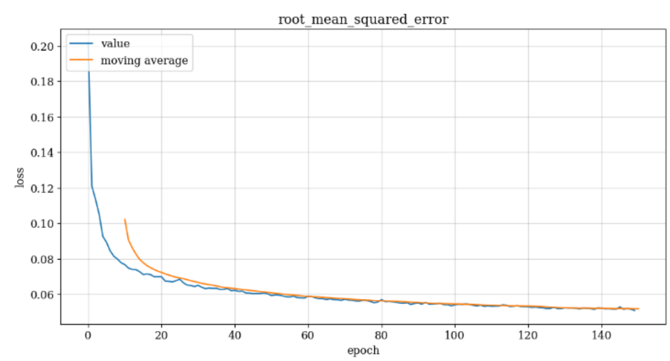

Fig. 7. Training loss and mean absolute error of proposed method.

These insights from the training and validation performance of the proposed Transformer-based model highlight its effectiveness in learning from complex, multimodal traffic data. By leveraging attention mechanisms and feed-forward layers, the model successfully captures both short-term and long-term dependencies in the data, resulting in improved predictive accuracy. The consistently low validation loss, RMSE [26], and MAE further emphasize that the model is well-suited for the task of traffic prediction, demonstrating robustness and reliability in its forecasting capabilities.

TABLE I. EVALUATION OF PROPOSED MODEL WITH STATE OF THE ART METHODS

Methods	MSE	RMSE	R-squared	Accuracy
RCNN [27]	$9.625e^{-4}$	0.78	0.7563	0.8575
RandomForest [28]	$6.205e^{-3}$	0.70	0.7054	0.7765
SVR [29]	$8.965e^{-2}$	0.65	0.6954	0.7924
TFM-GCAM [30]	7.021e-4	0.70	0.6021	0.8563
ITM [31]	5.0213e-4	0.55	0.5031	0.8945
CNN-GRUSKIP [32]	6.174e-4	0.60	0.6511	0.9514
FD-TGCN [33]	4.958e-4	0.30	0.7585	0.9452
Proposed	**$4.417e^{-4}$**	**0.0208**	**0.7745**	**0.9826**

Table I presents a comprehensive comparison of the proposed model with several state-of-the-art methods across multiple performance metrics: MSE, RMSE, R-squared, and Accuracy. The proposed model achieves the lowest MSE of 4.417e-4 and RMSE of 0.0208, demonstrating superior predictive accuracy and lower error compared to other models. In terms of R-squared, the proposed model achieves a value of 0.7745, which is higher than several methods, indicating a better fit to the data. Moreover, the accuracy of the proposed model (0.9826) significantly outperforms the other methods, underscoring its potential for accurate and reliable predictions. These results suggest that the proposed model outperforms traditional techniques, making it a promising candidate for future applications.

TABLE II. COMPARISON OF PROPOSED METHOD WITH STATE-OF-THE-ART METHODS

Methods	Spearman Correlation	Kendall Correlation	Pearson Correlation
RCNN [27]	0.7234	0.7827	0.7873
RandomForest [28]	0.6518	0.5124	0.7015
SVR [29]	0.5576	0.4521	0.6564
TFM-GCAM [30]	0.8565	0.7541	0.7954
ITM [31]	0.9472	0.8451	0.8324
CNN-GRUSKIP [32]	0.9768	0.8246	0.8954
FD-TGCN [33]	0.9541	0.7457	0.8854
Proposed	**0.9845**	**0.8965**	**0.9125**

Table II presents a comparison of the proposed method with various state-of-the-art models using three important correlation metrics: Spearman, Kendall, and Pearson correlations. These metrics assess the strength and nature of the relationship between the predicted and actual values, each in a distinct manner. The Spearman correlation evaluates the monotonic relationship between variables, meaning that it measures whether the variables consistently increase or decrease together, regardless of the exact form of the relationship. The proposed method outperforms all other models with a Spearman correlation of 0.9845, indicating an exceptionally strong monotonic relationship. The Kendall correlation, which is more robust to ties and considers the ordering of data pairs, shows that the proposed model achieves a Kendall correlation of 0.8965, again outperforming the other methods. This suggests that the proposed model consistently preserves the relative ordering of data points better than the others. Lastly, the **Pearson correlation, which measures the linear relationship between variables, highlights the proposed model's excellent performance with a Pearson correlation of 0.9125, the highest among all methods. This strong linear correlation demonstrates that the proposed method's predictions are highly consistent with the true values. Collectively, these results indicate that the proposed model significantly outperforms the other state-of-the-art models in terms of its ability to capture monotonic, ordered, and linear relationships, making it a highly effective and reliable model for prediction tasks. The traditional machine learning models, including RandomForest, GradientBoosting, and SVR, show significantly lower R-squared and Pearson correlation values, with GradientBoosting performing the worst among them. These models do not account for temporal dependencies in traffic data, leading to poorer predictive performance. RandomForest achieves an R-squared of 0.65 and a Pearson correlation of 0.70, while GradientBoosting and SVR show even lower values. This highlights the advantage of models that can capture both spatial and temporal patterns, such as the proposed method and RCNN, in forecasting traffic flow more accurately.

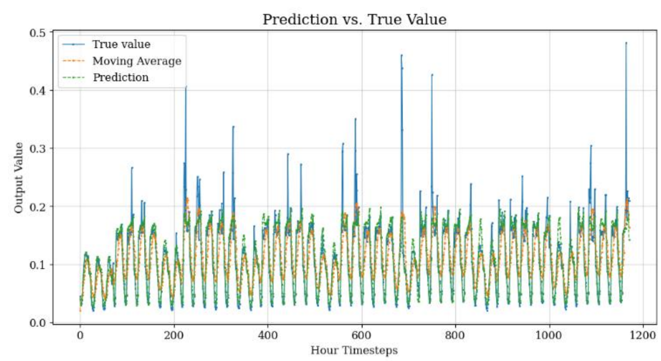

Fig. 8. Prediction vs. true value.

In Fig. 8, which displays the 'Prediction vs. True Value' graph over a 1200-hour period, the blue line represents the actual traffic values recorded by the sensors, while the green dashed line shows the predicted traffic flow values generated by the model.

The orange dashed line represents the moving average, which smooths out the short-term fluctuations in the traffic data, offering a baseline for comparison. Observing the graph, the predictions closely align with the true values, particularly in capturing recurring patterns of traffic flow. The moving average helps to highlight the general trend and periodicity in the data, while the model's predictions are able to accurately track not just the overall behavior but also the smaller variations. The few spikes seen in the true values—indicating sudden increases in traffic flow—are somewhat captured by the model, although in a smoothed-out manner, showing that while the model is effective in learning regular traffic patterns, sudden changes in traffic may pose more of a challenge.

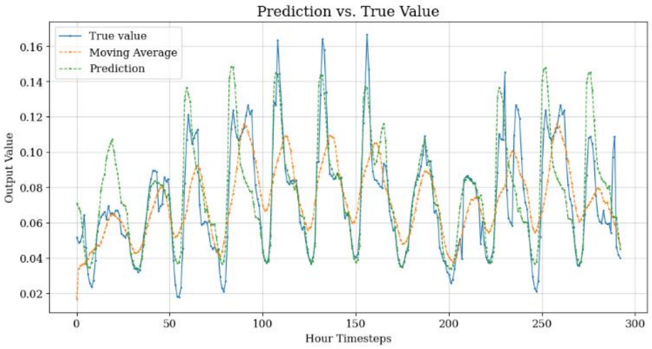

Fig. 9. First 300-hour timesteps.

In Fig. 9, which zooms in on the first 300-hour timesteps, a more detailed comparison is presented between the true values, model predictions, and the moving average. This closer view emphasizes the model's ability to accurately follow traffic peaks and dips. The blue line, representing true values, shows clear periodic cycles of traffic congestion and reductions over time, which the model's predictions (green dashed line) follow quite closely. The model is not only capable of predicting peak traffic periods but also lower traffic periods, capturing the full range of fluctuations in traffic dynamics. The alignment of the predicted values with the true values indicates that the model effectively handles both high and low traffic patterns, while the moving average remains close to the overall trend, providing additional confirmation that the model does not overfit to noise. This level of alignment showcases the model's reliability and predictive accuracy, especially over shorter timeframes.

V. Conclusion

In this study, we proposed a custom deep learning method for predicting traffic flow at the Port of Virginia, which integrates spatial feature extraction and temporal dependency modeling through a hybrid approach. This method combines 1D convolutional layers for extracting local spatial patterns from traffic data and a custom temporal processing layer to capture long-range dependencies in the traffic flow. The model was designed to effectively process traffic data from multiple sensor points, making predictions for traffic flow, occupancy, and speed.

The model's performance was evaluated using metrics such as Root Mean Squared Error (RMSE) and Spearman's Rank Correlation, highlighting its ability to predict both the magnitude of traffic flow and preserve the rank order of traffic conditions. The results demonstrate that the proposed approach can significantly enhance port operations by reducing congestion and improving resource allocation efficiency. This work contributes valuable insights for real-time traffic management and lays the foundation for future research that can incorporate additional data sources to further refine and enhance the model's accuracy and robustness.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No. 2022YFB2602304), and the National Natural Science Foundation of China (No. 52272318，No. 52301314).

References

[1] M. Stopford, Maritime economics 3e. Routledge, 2008. Accessed: Oct. 25, 2024. [Online]. Available: https://www.taylorfrancis.com/books/mono/10.4324/9780203891742/maritime-economics-3e-martin-stopford

[2] H. Meersman, E. Van De Voorde, and T. Vanelslander, "Port Congestion and Implications to Maritime Logistics," in Maritime Logistics, D.-W. Song and P. M. Panayides, Eds., Emerald Group Publishing Limited, 2012, pp. 49–68. doi: 10.1108/9781780523415-004.

[3] U. Nations, "Review of maritime transport-2014," URL https://unctad.org/system/files/official-document/rmt2019_en. pdf.[Accessed: 2020-10-13], 2019.

[4] A. B. Steven and T. M. Corsi, "Choosing a port: An analysis of containerized imports into the US," Transportation Research Part E: Logistics and Transportation Review, vol. 48, no. 4, pp. 881–895, 2012.

[5] G.-T. Yeo, M. Roe, and S.-M. Soak, "Evaluation of the Marine Traffic Congestion of North Harbor in Busan Port," J. Waterway, Port, Coastal, Ocean Eng., vol. 133, no. 2, pp. 87–93, Mar. 2007, doi: 10.1061/(ASCE)0733-950X(2007)133:2(87).

[6] I. AbuAlhaol, R. Falcon, R. Abielmona, and E. Petriu, "Mining port congestion indicators from big AIS data," in 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8. Accessed: Oct. 25, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8489187/

[7] L. Chen et al., "Container port performance measurement and comparison leveraging ship GPS traces and maritime open data," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 5, pp. 1227–1242, 2015.

[8] E. O. Oyatoye, S. O. Adebiyi, J. C. Okoyee, and B. B. Amole, "Application of Queueing theory to port congestion problem in Nigeria," 2011, Accessed: Oct. 25, 2024. [Online]. Available: https://ir.unilag.edu.ng/handle/123456789/2790

[9] R. C. Leachman and P. Jula, "Congestion analysis of waterborne, containerized imports from Asia to the United States," Transportation Research Part E: Logistics and Transportation Review, vol. 47, no. 6, pp. 992–1004, 2011.

[10] A. Sheikholeslami, G. Ilati, and Y. E. Yeganeh, "Practical solutions for reducing container ships' waiting times at ports using simulation model," J. Marine. Sci. Appl., vol. 12, no. 4, pp. 434–444, Dec. 2013, doi: 10.1007/s11804-013-1214-x.

[11] J. F. J. Pruyn, A. A. Kana, and W. M. Groeneveld, "Analysis of port waiting time due to congestion by applying Markov chain analysis," in Maritime Supply Chains, Elsevier, 2020, pp. 69–94. Accessed: Oct. 30, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128184219000057

[12] "Network Disruptions and Ripple Effects: Queueing Model, Simulation, and Data Analysis of Port Congestion." Accessed: Oct. 30, 2024. [Online]. Available: https://www.mdpi.com/2077-1312/11/9/1745

[13] P. Legato and R. M. Mazza, "Queueing analysis for operations modeling in port logistics," Maritime Business Review, vol. 5, no. 1, pp. 67–83, Jan. 2020, doi: 10.1108/MABR-09-2019-0035.

[14] X. Bai, H. Jia, and M. Xu, "Identifying port congestion and evaluating its impact on maritime logistics - TRID", Accessed: Oct. 30, 2024. [Online]. Available: https://trid.trb.org/View/2381924

[15] "Port Congestion - causes, consequences and impact on global trade." Accessed: Nov. 12, 2024. [Online]. Available: https://www.shippingandfreightresource.com/port-congestion-causes-and-impact-on-global-trade/

[16] "Port Congestion: Why it Happens and 9 Ways to Avoid It - Base | Operations Management & Port Logistics Software." Accessed: Nov. 12, 2024. [Online]. Available: https://www.usebase.io/port-congestion/

[17] "From Singapore to Los Angeles: How Port Congestion Is Reshaping Global Trade 2024." Accessed: Nov. 12, 2024. [Online]. Available: https://www.seavantage.com/blog/from-singapore-to-los-angeles-how-port-congestion-is-reshaping-global-trade-2024

[18] "Supply Chain Issues: Port Congestion | Fractory." Accessed: Nov. 12, 2024. [Online]. Available: https://fractory.com/port-congestion-explained/

[19] "Acute port congestion and emissions exceedances as an impact of COVID-19 outcome: the case of San Pedro Bay ports | Journal of Shipping and Trade | Full Text." Accessed: Nov. 12, 2024. [Online]. Available: https://jshippingandtrade.springeropen.com/articles/10.1186/s41072-022-00126-5

[20] "Top 3 Indicators to Understand Port Congestion in Order to Optimize Ocean Supply Chain Planning - Portcast Blog." Accessed: Nov. 12, 2024. [Online]. Available: https://www.portcast.io/blog/top-3-indicators-to-understand-port-congestion-in-order-to-optimize-ocean-supply-chain-planning

[21] "Supply Chains and Port Congestion Around the World in: IMF Working Papers Volume 2022 Issue 059 (2022)." Accessed: Nov. 12, 2024. [Online]. Available: https://www.elibrary.imf.org/view/journals/001/2022/059/article-A001-en.xml

[22] "Liner Schedule Design under Port Congestion: A Container Handling Efficiency Selection Mechanism." Accessed: Nov. 12, 2024. [Online]. Available: https://www.mdpi.com/2077-1312/12/6/951

[23] H. Qu, X. Wang, L. Meng, and C. Han, "Liner Schedule Design under Port Congestion: A Container Handling Efficiency Selection Mechanism," Journal of Marine Science and Engineering, vol. 12, no. 6, Art. no. 6, Jun. 2024, doi: 10.3390/jmse12060951.

[24] "5 Ways Technology Can Reduce Port Congestion - Tideworks." Accessed: Nov. 12, 2024. [Online]. Available: https://tideworks.com/5-ways-technology-can-reduce-port-congestion/

[25] "Container barge congestion and handling in large seaports: a theoretical agent-based modeling approach | Journal of Shipping and Trade | Full Text." Accessed: Nov. 12, 2024. [Online]. Available: https://jshippingandtrade.springeropen.com/articles/10.1186/s41072-019-0044-7

[26] A. Y. J. Akossou and R. Palm, "Impact of data structure on the estimators R-square and adjusted R-square in linear regression," Int. J. Math. Comput, vol. 20, no. 3, pp. 84–93, 2013.

[27] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang, "ECCV 2018 Open Access Repository," Accessed: Jan. 16, 2025. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Bowen_Cheng_Revisiting_RCNN_On_ECCV_2018_paper.html

[28] M. Pal, "Random forest classifier for remote sensing classification," International Journal of Remote Sensing, vol. 26, no. 1, pp. 217–222, Jan. 2005, doi: 10.1080/01431160412331269698.

[29] M. Alfonse and A.-B. M. Salem, "An automatic classification of brain tumors through MRI using support vector machine," Egyptian Computer Science Journal, vol. 40, no. 3, 2016.

[30] "Traffic flow matrix-based graph neural network with attention mechanism for traffic flow prediction", doi: 10.1016/j.inffus.2023.102146.

[31] Z. Xu, J. Yuan, L. Yu, G. Wang, and M. Zhu, "Machine Learning-Based Traffic Flow Prediction and Intelligent Traffic Management," International Journal of Computer Science and Information Technology, vol. 2, no. 1, Art. no. 1, Mar. 2024, doi: 10.62051/ijcsit.v2n1.03.

[32] "A multi-Layer CNN-GRUSKIP model based on transformer for spatial –TEMPORAL traffic flow prediction", doi: 10.1016/j.asej.2024.103045.

[33] "FD-TGCN: Fast and dynamic temporal graph convolution network for traffic flow prediction", doi: 10.1016/j.inffus.2024.102291.