Leveraging Large Language Models for VNF Resource Forecasting

Jing Su, Suku Nair
AT&T Center for Virtualization
Southern Methodist University
Dallas, USA
{suj, nair}@smu.edu

Leo Popokh
Hewlett Packard Enterprise
Dallas, USA
leonid.i.popokh@hpe.com

2024 IEEE 10th International Conference on Network Softwarization (NetSoft) | 979-8-3503-6958-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/NetSoft60951.2024.10588943

***Abstract*—The evolution of the Network Function Virtualization (NFV) paradigm has revolutionized the way network services are deployed, managed, and scaled. Within this transformative landscape, Virtual Network Function (VNF) resource prediction emerges as a cornerstone for optimizing network resource allocation and ensuring service reliability and efficiency. Traditional resource forecasting methods often struggle to adapt to the dynamic and non-linear nature of changes in resource consumption patterns in modern telecommunication networks. We address this challenge by leveraging the inherent pattern recognition and next-token prediction capabilities of Large Language Model (LLM) without requiring any domain-specific fine-tuning. Our study utilizes Llama2 as the foundation model to evaluate the performance against widely used probability-based models on a public VNF dataset that encompasses real-world resource consumption data of various VNFs for comparative analysis. Our findings suggest that LLM offers a highly effective alternative for VNF resource forecasting, demonstrating significant potential in enhancing network resource management.**

***Index Terms*—NFV, VNF, Resource Prediction, Large Language Model, Generative AI**

I. Introduction

In the rapidly evolving landscape of network technology, Virtual Network Functions (VNFs) have become a cornerstone in the architecture of modern telecommunication systems. The concept of VNF stems from the broader framework of Network Functions Virtualization (NFV), which aims to decouple network functions from proprietary hardware appliances, allowing them to be hosted on standard server hardware as virtual machines (VMs) or containers. This paradigm shift not only enhances flexibility and scalability but also introduces complexities in resource management and allocation. The quality of network services is directly influenced by the ability to allocate resources effectively. VNF resource forecasting plays a critical role in maintaining high service quality and reliability. By anticipating resource demands, it is possible to proactively scale resources up or down, thereby avoiding service degradation or interruptions. This not only ensures a consistent user experience but also enhances the overall reliability of the network services.

VNF resource forecasting involves predicting the resource usage of VNF instances to facilitate decision-making for automatic adaptation of physical resources. This can trigger actions such as horizontal scaling, vertical scaling, or migration requests by NFV management and orchestration (NFV-MANO). Accurate VNF resource usage prediction is an essential first step in a VNF Resource Allocation (VNF-RA) pipeline [1]. This challenge is further compounded by the integration of Software-Defined Networking (SDN), which introduces additional layers of abstraction and control over network resources. Accurate prediction of VNF resource requirements is crucial for optimizing the utilization of underlying physical resources. By forecasting the resource needs of VNFs, network operators can minimize underutilization and overprovisioning, leading to significant cost savings and enhanced operational efficiency. This predictive approach enables a more responsive and cost-effective allocation of computing, storage, and networking resources, which is essential in a highly dynamic NFV environment.

Traditional time series forecasting models such as Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ETS) have been widely used for resource forecasting in the past [2]–[4]. ARIMA models aim to describe the autocorrelations present within datasets, while ETS models rely on delineating both the trend and seasonality in the data. However, with the increasing complexity and dynamic nature of network traffic, these traditional methods often struggle with complex patterns seen in VNF resource usage, especially when there are multiple overlapping cycles or trends. Therefore, many research teams have been exploring Deep Learning techniques like Long Short Term Memory (LSTM) [1], [5]–[7] and Deep Reinforcement Learning (DRL) [8] to improve upon these traditional mechanisms.

Recent advancements in Large Language Models (LLMs) have opened new avenues for addressing these challenges. The core mechanism powering these models is next-token prediction, which enables them to anticipate the most likely subsequent item in a sequence of data. Though primarily developed for linguistic purposes, this ability has broader applications in various data-driven forecasting tasks. By analyzing historical data and identifying patterns, these models can make informed predictions about future resource requirements.

In the context of VNF resource forecasting, LLMs can be utilized to analyze and predict network resource requirements. By appropriately tokenizing the VNF resource consumption data, these models can leverage their next-token prediction

979-8-3503-6958-8/24/$31.00 ©2024 IEEE

capabilities to forecast future resource needs more accurately and efficiently than traditional methods. This approach not only harnesses the advanced predictive capabilities of LLMs but also aligns with the dynamic and automated nature of NFV and SDN environments. Our study leverages pre-trained foundation models (PFMs) for VNF resource forecasting. These models have been trained on vast datasets, encompassing a wide range of knowledge domains, which enables them to generate predictions with a high degree of accuracy and relevance. The next-token prediction ability of LLMs is particularly beneficial for forecasting tasks, as it allows the models to extrapolate future states from sequential data. Llama2 is an open-source LLM that includes model weights for pre-trained and fine-tuned language models, ranging from 7 billion to 70 billion parameters. It is one of the state-of-the-art LLM and outperforms other open-source language models on many external benchmarks, including reasoning, coding, proficiency, and knowledge tests [9].

This paper aims to explore the application of pre-trained LLMs for effective VNF resource forecasting. We delve into how the generative abilities of these models, rooted in next-token prediction, can be adapted to the unique challenges of forecasting in NFV and SDN contexts. We explore the challenges and opportunities presented by this innovative approach, aiming to demonstrate its efficacy in enhancing the operational efficiency of telecommunication networks. Consequently, we aim to provide a novel perspective on optimizing network function virtualization through the lens of advanced generative AI methodologies.

II. Methodology

A. Problem Analysis and Modeling

In an NFV environment, $\{x_t\}_{t\in\mathbb{Z}}$ with $x_t \in \mathbb{R}^n$ denotes the historical resource consumption data known up to the current time t. The VNF resource forecasting task is finding a prediction function g to predict a given length of future resource consumption $\{\hat{x}_t\}$ begins at $t+1$ as $\hat{x}_{t+\Delta t} = g(x_t, \Delta t)$. This work uses the LLM next-token prediction ability as the function g for resource consumption forecasting.

Fig. 1 depicts an example of integrating LLM for future VNF resource forecasting in the NFV environment. The NFV-MANO can thereby make informative allocation decisions to ensure efficient resource utilization. In this workflow, the serializer will preprocess the collected VNF resource historical consumption data $\{x_t\}$ for further processing. The objective of preprocessing is to ensure a uniform distribution and consistent format of the data. It will also serialize the signedness, radix, and separation mark of the incoming data. The data are then passed through an encoder, which processes the input in preparation for the transformer block. The encoder will create a string representation of the input. After that, the input string will be tokenized and broken down into tokens that the model can understand. The tokens in LLM are often numerical representations of words or characters. After tokenization, the tokens will be embedded into vectors of continuous values. These vectors are designed to capture more information about the tokens, such as their meanings, semantic relationships, and their context within the text. Each token is initially represented by an integer, as determined by the tokenization process. For each token, its integer identity is used to look up its vector in the embedding matrix. This vector is a dense representation with real numbers, capturing the semantic properties of the token.

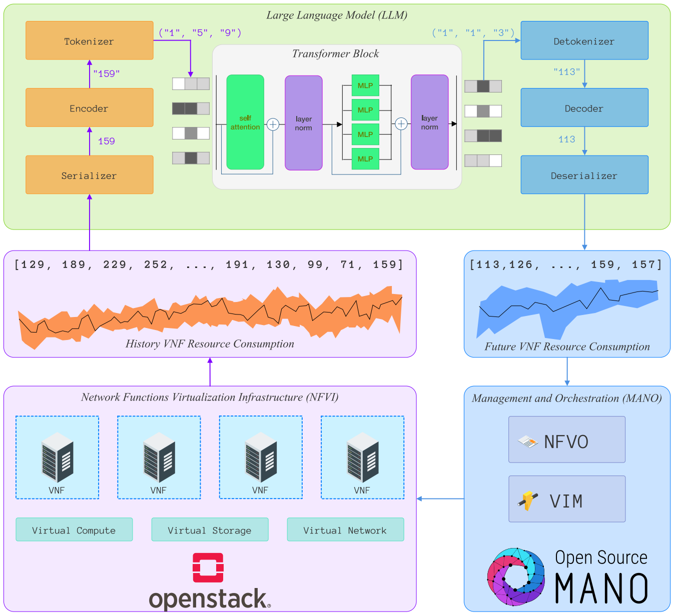

Fig. 1. A high-level overview of integrating LLM for future VNF resource forecasting for efficient services orchestration.

Current LLMs widely adopt Generative Pre-trained Transformer (GPT) for next-token prediction and generation. GPT models are built on the Transformer architecture [10], which employs the multi-head attention mechanism. Fig. 2 illustrates the text generation process of LLMs. The workflow starts with the input text, which undergoes tokenization. During this step, each word is converted into a unique integer token. The final output is a vector representing the likelihood of each candidate token being the next word. Finally, the token with the highest probability is typically selected as the next word in the generated text. These steps are recursively called until an end-of-sequence token occurs.

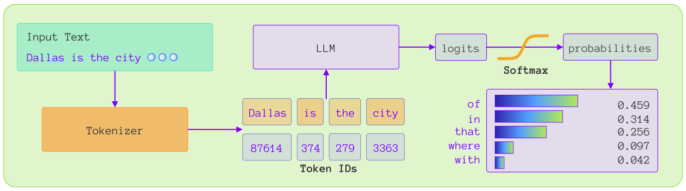

Fig. 2. Example of large language models for text generation.

In the proposed framework, the model initially computes output probabilities for subsequent tokens. Utilizing its trained probability distributions, it selects the token with the highest likelihood as the next output. This token is subsequently reintegrated into the model as input, facilitating the prediction of further tokens. The sequence of predicted tokens undergoes a detokenization and decoding process, ultimately being transformed into a deserialized representation of anticipated VNF resource utilization $\{\hat{x}_t\}$. Fig. 3 presents the flow of an LLM employed for VNF resource consumption forecasting. The process starts with input data consisting of numerical values representing resource consumption metrics. These metrics undergo serialization and are converted to a string format with configured base and precision. The serialized data are then encoded with specified numbers and data entity separators. The data are then tokenized, and each digit is assigned a unique token identity. The LLM ingests these token identities, producing logits representing the raw predictions for the next possible values. The higher the logit, the higher the confidence in the corresponding token being the appropriate next value in the sequence. A softmax layer converts these logits into probabilities, signifying the likelihood of each predicted outcome. This process ultimately outputs a probability distribution for the next predicted resource consumption values, aiding in forecasting future VNF resource requirements. These steps are recurrently invoked until the specified forecast horizon is fulfilled.

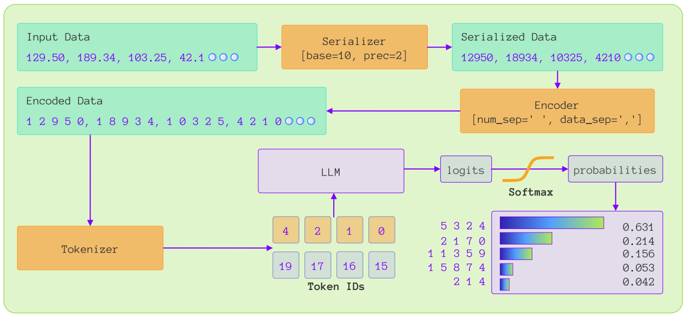

Fig. 3. Workflow of large language models for VNF resource forecasting.

B. Open-source LLMs

In the rapidly evolving field of LLMs, the advent of open-source models like Llama2 marks a pivotal shift. These models are critical in democratizing access to advanced AI technologies. Open-source LLMs offer transparent and collaborative development pathways, enabling a more comprehensive range of researchers and developers to contribute to and scrutinize the technology.

Llama2, developed by Meta AI in 2023 [9], stands out as a noteworthy example. It builds upon the foundation laid by its predecessor Llama1, offering improved performance and scalability. Notably, Llama2 is designed to match or even surpass the proficiency of leading closed-source models in certain domains. This is achieved through a combination of extensive pre-training on diverse and large text corpora and advanced fine-tuning techniques. The model comes in three sizes (7B, 13B, and 70B parameters) catering to different computational needs and applications.

The pre-training process for Llama2 involves self-supervised learning on a vast dataset comprising two trillion tokens. This foundational training equips the model with a broad understanding of language and context. Following this, Llama2 undergoes a fine-tuning process using a blend of supervised learning and Reinforcement Learning from Human Feedback (RLHF). This stage incorporates human-annotated examples and instructional datasets, refining the model's capabilities in dialogue and specific task performances. Incorporating the potential of Llama2 for resource consumption prediction adds another layer of utility to this open-source LLM. While LLMs like Llama2 are primarily designed for understanding and generating human language, their underlying capabilities can be adapted for various specialized tasks, including VNF resource forecasting.

VNF resource forecasting involves analyzing sequential data points, often collected over time, to forecast future values. LLMs can be instrumental in this domain due to their proficiency in pattern recognition and sequence prediction. In this work, we will utilize Llama2 as the foundation model and evaluate all its different size variants.

C. Zero-shot Forecasting

Zero-shot involves applying the LLM to a forecasting task without any task-specific training. The LLM relies solely on its pre-existing knowledge and understanding gained during its initial training phase.

LLM and GPT have been demonstrated as zero-shot time series forecasters in the work of Gruver *et al.* (2023) [11]. Our approach was inspired by this work, particularly their innovative zero-shot forecasting approach. While adhering to the basic structure proposed by them, our model uniquely applies tokenization techniques to network data patterns., offering new insights into VNF resource forecasting.

Particularly pertinent in the NFV domain, where network conditions and requests are highly variable due to fluctuating user demands, zero-shot forecasting presents a valuable tool for predicting resource requirements in previously unseen scenarios, thereby circumventing limitations inherent in the training phase.

III. Evaluation

A. Datasets

We are using a public VNF dataset from Knowledge-Defined Networking (KDN) [12] with CPU consumption of read-world VNFs when operating under real traffic for the evaluation. The original dataset provided a MATLAB loading program for each category. We are using the same procedure to load the data into Python. This dataset used Open Virtual Switch (OVS) and Snort as network components. OVS is a widespread virtual switch implementation [13], and Snort is effective for network intrusion detection in SDN [14]. This dataset consists of three categories of VNFs as follows:

1) OVS: CPU consumption of an OVS connected to an SDN controller functioned as an SDN-enabled switch. This category consists of 1153 data points after being loaded.

2) Firewall: CPU consumption of an OVS configured with firewall rules operated as an SDN-enabled firewall. This category consists of 560 data points after being loaded.

3) Snort: CPU consumption of a Snort with the initial configuration. This category consists of 604 data points after being loaded.

These VNFs were deployed as VMs with two additional VMs connected via gigabit links for traffic generation and reception. Network traffic was sourced from a campus network serving approximately $30,000$ users, and the data was captured in 20-second intervals.

In our evaluation, we partitioned each dataset into two distinct subsets: a training set and a validation set. This was done in an $80:20$ ratio, adhering to standard machine learning model validation practices. The training set, comprising the first 80% of the data, was utilized for model fitting, allowing them to learn the patterns and seasonalities. The remaining 20%, designated as the validation set, served to evaluate the model's performance on unseen data, ensuring generalizability and robustness.

B. Baselines

In this evaluation, we use ARIMA, ETS, and Theta, three prominent models in the realm of time series forecasting, as baselines to evaluate the performance for comparison. ARIMA, an integration of autoregressive and moving average models, is adept at capturing a wide range of time series data structures, making it the most general class of models for forecasting a time series and comparable with deep neural network approaches [15]. ETS, on the other hand, extends exponential smoothing to capture trends and seasonality more effectively and has been demonstrated to be effective and recommended for cellular traffic prediction [16]. The Theta method has gained attention due to its simplicity and superior forecasting accuracy. It has been confirmed by many empirical studies and forecasting competitions to perform well [17].

For the implementation, we use AutoARIMA, AutoETS, and AutoTheta from the StatsForecast framework [18], which are automated versions of their respective forecasting models. These automated models are designed to streamline the process of model selection and hyperparameter tuning, which are crucial for achieving maximum performance.

C. Evaluation Metrics

In the realm of forecasting, it is imperative to employ robust and reliable metrics to evaluate the accuracy and effectiveness of predictive models. We select Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) as the evaluation metrics, which are widely recognized and utilized for this purpose.

In evaluating these metrics, a lower value indicates better model performance for all four metrics. By examining these metrics collectively, we aim to provide a comprehensive assessment of the forecasting performance, capturing different aspects and impacts of forecasting errors.

D. Effectiveness Analysis

During the evaluation phase, the AutoARIMA, AutoETS, and AutoTheta models maintained their default hyperparameters. Each Llama2 variant was set with a consistent temperature hyperparameter of 1.0. Furthermore, in the prediction process, the number of samples was fixed at 5 for both statistical and LLM models, facilitating the quantification and capture of intrinsic uncertainty in the forecasts.

The forecasting results compared to the validation dataset are shown in Fig. 4, and Table I details the benchmarking results across all models. The dataset length reveals that the OVS dataset is approximately double the size of both the Firewall and Snort datasets, consequently leading to a prediction length that is also twice as long for OVS. We made three key observations: 1) LLM-based models demonstrate outstanding forecasting accuracy compared to statistical models. LLMs benefit from their ability to leverage large-scale data during training, enabling them to capture complex patterns and dependencies often missed by statistical approaches. 2) Smaller LLM models exhibit reduced capability in long-term forecasting compared to larger models. This performance degradation can be attributed primarily to the reduced parameter counts, which constrain their ability to capture and model the extensive and intricate patterns necessary for accurate long-term predictions. 3) Although larger models are typically expected to learn and represent complex data patterns better, the 70B model exhibits inferior performance compared to the 7B and 13B models, specifically in the Firewall and Snort datasets. This unexpected behavior of the 70B model could be attributed to alignment interventions such as RLHF during pre-training. While aimed at improving model safety and alignment with human values, these interventions may inadvertently prioritize certain types of data handling or response patterns that do not align well with the general type of forecasting tasks.

In summary, our analysis indicates that LLM models, particularly the moderate size (i.e., 13B), offer balanced forecasting accuracy, computational efficiency, and adaptability.

IV. Conclusions and Future Work

This paper has demonstrated the viability of using a pre-trained LLM, specifically Llama2, for zero-shot VNF resource forecasting within NFV and SDN environments. Our evaluation indicates that Llama2, despite not being fine-tuned, can effectively predict resource requirements due to its substantial next-token prediction ability, potentially surpassing traditional forecasting methods in accuracy and efficiency. This approach offers a promising new direction for network resource management, leveraging the advanced capabilities of LLMs to handle the complexities of modern network traffic.

For future work, we aim to explore the integration of LLM with real-time network management systems for dynamic resource allocation. Another area of interest is refining data preprocessing and encoding methods to enhance prediction accuracy further. Besides, extending this approach to other

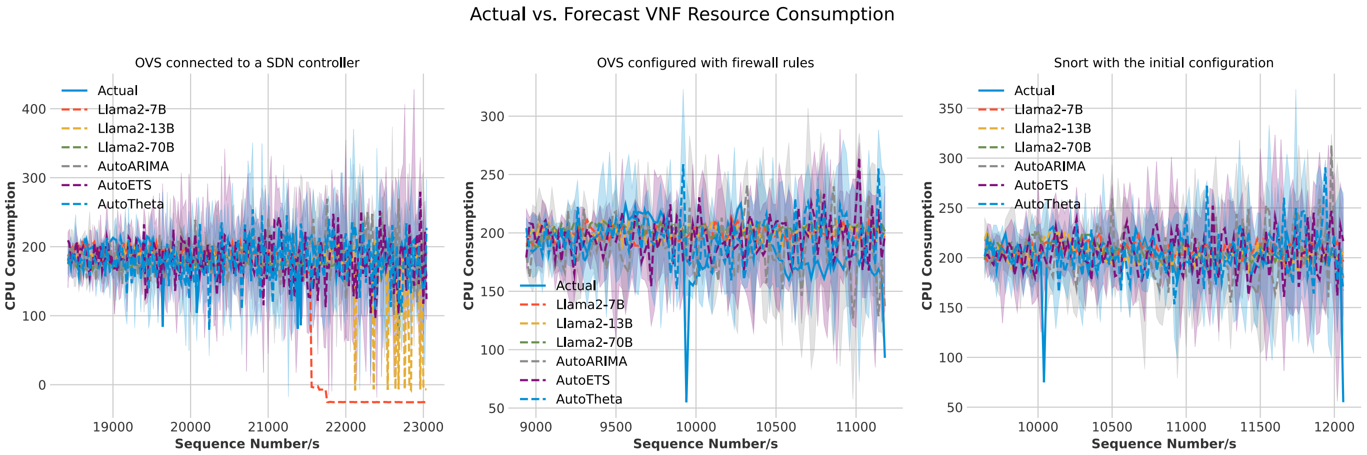

Fig. 4. Comparison among actual validation dataset and forecasting results.

TABLE I
VNF Resource Forecasting Task on KDN dataset.

	OVS				**Firewall**				**Snort**				*Avg.*			
	MAE	MSE	RMSE	MAPE	MAE	MSE	RMSE	MAPE	MAE	MSE	RMSE	MAPE	MAE	MSE	RMSE	MAPE
Llama2-7B	79.87	15378.76	124.01	42.70	**17.83**	**658.38**	**25.66**	**11.98**	**13.18**	**503.41**	**22.44**	**9.09**	36.96	5513.52	57.37	21.25
Llama2-13B	26.53	2384.19	48.83	15.02	18.26	682.35	26.12	12.23	13.25	540.10	23.24	9.18	19.35	1202.21	32.73	12.14
Llama2-70B	**22.68**	**752.15**	**27.43**	**12.79**	19.65	750.58	27.40	13.03	13.55	542.02	23.28	9.32	**18.63**	**681.59**	**26.03**	**11.71**
AutoARIMA	24.34	1138.38	33.74	14.50	21.75	826.50	28.75	13.32	21.36	910.83	30.18	12.83	22.49	958.57	30.89	13.55
AutoETS	28.61	1350.08	36.74	15.89	21.88	918.47	30.31	13.78	19.26	771.32	27.77	12.09	23.25	1013.29	31.61	13.92
AutoTheta	27.37	1286.07	35.86	15.31	20.77	852.25	29.19	13.22	19.61	790.79	28.12	11.81	22.58	976.37	31.06	13.45

aspects of network management, such as resource consumption anomaly detection, could yield significant benefits.

References

[1] C. St-Onge, N. Kara, and C. Edstrom, “Multivariate outlier filtering for A-NFVLearn: An advanced deep VNF resource usage forecasting technique,” *The Journal of Supercomputing*, vol. 79, no. 14, pp. 16 206–16 232, Sep. 2023.

[2] Y. Xie, M. Jin *et al.*, “Real-Time Prediction of Docker Container Resource Load Based on a Hybrid Model of ARIMA and Triple Exponential Smoothing,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 1386–1401, Apr. 2022.

[3] V. B. Edwin Joseph, M. H. Parvathi *et al.*, “Enable optimal resource utilization for VNF through intelligent cloud platform,” in *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, 2020, pp. 1–6.

[4] D. Prangchumpol, P. Sophatsathit *et al.*, “Resource allocation with exponential model prediction for server virtualization,” *Journal of Digital Information Management*, vol. 13, pp. 385–398, Oct. 2015.

[5] H.-G. Kim, D.-Y. Lee *et al.*, “Machine learning-based method for prediction of virtual network function resource demands,” in *2019 IEEE Conference on Network Softwarization (NetSoft)*, 2019, pp. 405–413.

[6] H.-G. Kim, S.-Y. Jeong *et al.*, “A Deep Learning Approach to VNF Resource Prediction using Correlation between VNFs,” in *2019 IEEE Conference on Network Softwarization (NetSoft)*, Jun. 2019, pp. 444–449.

[7] Z. Zaman, S. Rahman, and M. Naznin, “Novel Approaches for VNF Requirement Prediction Using DNN and LSTM,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019, pp. 1–6.

[8] N. Jalodia, S. Henna, and A. Davy, “Deep Reinforcement Learning for Topology-Aware VNF Resource Prediction in NFV Environments,” in *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov. 2019, pp. 1–5.

[9] H. Touvron, L. Martin *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” 2023.

[10] A. Vaswani, N. Shazeer *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg *et al.*, Eds., vol. 30. Curran Associates, Inc., 2017.

[11] N. Gruver, M. A. Finzi *et al.*, “Large language models are zero-shot time series forecasters,” in *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

[12] A. Mestres, A. Rodriguez-Natal *et al.*, “Knowledge-Defined Networking,” *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, pp. 2–10, Sep. 2017.

[13] R. Yang, X. Chang *et al.*, “Performance modeling of linux network system with open vswitch,” *Peer-to-Peer Networking and Applications*, vol. 13, pp. 151–162, 2020.

[14] S. Badotra and S. N. Panda, “SNORT based early DDoS detection system using Opendaylight and open networking operating system in software defined networking,” *Cluster Computing*, vol. 24, pp. 501–513, 2021.

[15] K. Zhou, W. Y. Wang *et al.*, “Comparison of time series forecasting based on statistical ARIMA model and LSTM with attention mechanism,” in *Journal of Physics: Conference Series*, vol. 1631. IOP Publishing, 2020, p. 012141.

[16] Q. T. Tran, L. Hao, and Q. K. Trinh, “A comprehensive research on exponential smoothing methods in modeling and forecasting cellular traffic,” *Concurrency and Computation: Practice and Experience*, vol. 32, no. 23, p. e5602, 2020.

[17] G. Dudek, “Short-term load forecasting using Theta method,” *E3S Web of Conferences*, 2019.

[18] F. Garza, M. M. Canseco *et al.*, “StatsForecast: Lightning fast forecasting with statistical and econometric models,” *PyCon: Salt Lake City, UT, USA*, 2022.