Development of an Intelligent Service Delivery System to Increase Efficiency of Software Defined Networks

Serik Joldasbayev[1], Saya Sapakova[2], Almash Zhaksylyk[3],
Bakhytzhan Kulambayev[4], Reanta Armankyzy[5], Aruzhan Bolysbek[6]
International Information Technology University, Almaty, Kazakhstan[1]
Al-Farabi Kazakh National University, Almaty, Kazakhstan[1]
Department of Computer Engineering, International University of Information Technology, Almaty, Kazakhstan[2, 5]
Satbayev University, Almaty, Kazakhstan[3]
Turan University, Almaty, Kazakhstan[4]
Bachelor Student at International Information Technology University, Almaty, Kazakhstan[6]

Abstract*—The burgeoning complexity in network management has garnered considerable attention, specifically focusing on Software-Defined Networking (SDN), a transformative technology that addresses limitations inherent in traditional network infrastructures. Despite its advantages, SDN is often susceptible to bottlenecks and excessive load issues, underscoring the necessity for more robust load balancing solutions. Previous research in this realm has predominantly concentrated on employing static or dynamic methodologies, encapsulating only a handful of parameters for traffic management, thereby limiting their effectiveness. This study introduces an innovative, intelligence-led approach to service delivery systems in SDN, specifically by orchestrating packet forwarding—encompassing both TCP and UDP traffic—through a multi-faceted analysis utilizing twelve distinct parameters elaborated in subsequent sections. This research leverages advanced machine learning algorithms, notably K-Means and DBSCAN clustering, to discern patterns and optimize traffic distribution, ensuring a more nuanced, responsive load balancing mechanism. A salient feature of this methodology involves determining the ideal number of operational clusters to enhance efficiency systematically. The proposed system underwent rigorous testing with an escalating scale of network packets, encompassing counts of 5,000 to an extensive 10,000,000, to validate performance under varying load conditions. Comparative analysis between K-Means and DBSCAN's results reveals critical insights into their operational efficacy, corroborated by juxtaposition with extant scholarly perspectives. This investigation's findings significantly contribute to the discourse on adaptive network solutions, demonstrating that an intelligent, parameter-rich approach can substantively mitigate load-related challenges, thereby revolutionizing service delivery paradigms within Software-Defined Networks.

***Keywords*—Load balancing; machine learning; server; classification; software**

I. Introduction

In the contemporary digital landscape, the explosive growth in data traffic, coupled with the increasing reliance on cloud services and the Internet of Things (IoT), has rendered traditional networking architectures both obsolete and inadequate [1]. These legacy systems, characterized by their rigidity and hardware-dependency, present significant hurdles in catering to the dynamic nature of modern data traffic and the need for real-time decision-making [2]. It is within this challenging context that Software-Defined Networking (SDN) [3] has emerged as a beacon of innovation, enabling unprecedented levels of network management and adaptability by abstracting the control logic from the underlying physical infrastructure.

However, the benefits of SDN, particularly its centralized control and programmable network behavior, also bring forth complex challenges, chief among them being effective load balancing [4]. The conventional load balancing mechanisms, with their static nature, fail to comprehend and adapt to the erratic behavior of contemporary network traffic [5], leading to suboptimal utilization of network resources, potential network bottlenecks, and compromised service quality [6]. Thus, there is an exigent need for a more intelligent, scalable, and adaptive load balancing solution, capable of interpreting complex network environments and autonomously optimizing traffic distribution to enhance overall network performance.

Recognizing these challenges, this research paper explores the integration of machine learning (ML) techniques into the SDN architecture, specifically aimed at revolutionizing the load balancing processes [7]. Machine learning, with its capability to analyze vast datasets [8], identify patterns [9], and make predictive decisions [10], presents a promising solution to the intricate problem of dynamic load balancing. By applying ML algorithms, the research intends to enable SDN controllers to make real-time traffic routing decisions based on data-driven insights, thereby significantly improving resource allocation [11], reducing latency [12], and enhancing the user experience [13].

The paper begins by establishing the foundational concepts critical to this discourse, including an overview of Software-Defined Networking (its architecture, functionalities, and significance) and the inherent challenges of traditional load balancing strategies. This backdrop is essential for understanding the transformation that SDN brings to network management and the subsequent complexities, particularly in maintaining efficient traffic flow and equitable server workloads.

Following this, the discussion shifts focus to the core proposition of this study: the application of machine learning in enhancing SDN load balancing. Herein, the paper will dissect various machine learning models suitable for this application, considering their strengths and potential limitations in real-time decision-making and predictive analysis. The discussion extends to the adaptation of these ML models within the SDN framework, detailing the process from the initial stages of data collection and processing, to the advanced stages of algorithm training, validation, and deployment.

In application, the integration of ML into SDN for load balancing manifests as a dynamic, self-learning system capable of monitoring network conditions, predicting traffic fluctuations, and preemptively redistributing loads across servers to prevent imbalance and congestion [14]. The system's ability to learn continuously from network behavior and traffic patterns marks a significant advancement over traditional methods, essentially transforming reactive responses into proactive strategies.

To substantiate the theoretical discourse, the paper will present empirical evidence derived from simulated SDN environments, demonstrating the practical efficacy and reliability of ML-augmented load balancing [15]. These experiments highlight the performance improvements in terms of reduced network latency, efficient resource utilization, enhanced traffic management, and overall service quality.

Conclusively, this research contributes to the academic and practical realms of network management by advocating for a synergistic approach between machine learning and Software-Defined Networking. The insights drawn from this study underscore the potential of machine learning not just as a tool for load balancing, but as a comprehensive solution for creating intelligent, self-sustaining, and highly efficient network infrastructures, setting a new standard for future network operations and research endeavors.

II. RELATED WORKS

A. *Understanding Software-Defined Networking and the Load Balancing Dilemma in Servers*

The transformation of networking through Software-Defined Networking (SDN) is well-documented in literature, providing a shift from traditional, hardware-centric networks towards a flexible, software-driven approach [16]. The SDN model, advocating for the separation of the control plane from the data plane, introduces programmability into network management, thereby offering centralized control and real-time decision-making [17]. Despite its advanced capabilities, SDN faces inherent challenges, particularly in load balancing, a critical component for maintaining efficient server performance and network stability [18]. Fig. 1 demonstrates traditional IP network load balancing architecture.

In the realm of server environments, especially, load balancing acts as a linchpin, determining the operational robustness and reliability of network services. Traditional load balancing methods, as critiqued in [19], often rely on predetermined, static policies, lacking the flexibility and intelligence to adapt to changeable network traffic, leading to issues like server overload or underutilization. Moreover, the study in [20] highlights the inefficacy of these methods in contemporary data-intensive scenarios, underscoring the need for more sophisticated, context-aware solutions. Fig. 2 demonstrates a software defined network model for load balancing.

B. *Machine Learning – A Paradigm Shift in Network Management*

Exploring beyond conventional methodologies, recent studies have illuminated the role of machine learning (ML) in redefining network management. The comprehensive review in [21] illustrates ML's capabilities in pattern recognition, anomaly detection, and predictive analysis, marking a significant departure from rule-based systems towards adaptive, autonomous operations. Specifically, machine learning algorithms can analyze extensive datasets, learning and evolving through experiences without explicit programming for every contingency [22].

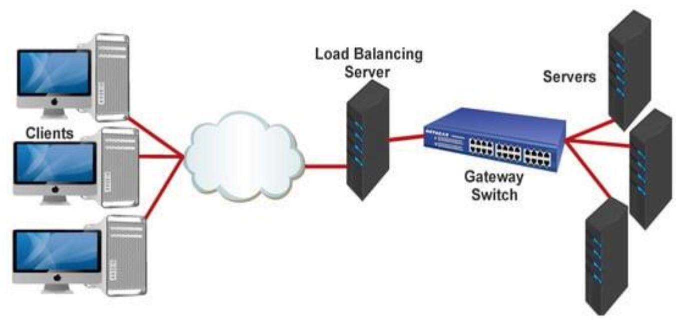

Fig. 1. Topology of self-organizing map.

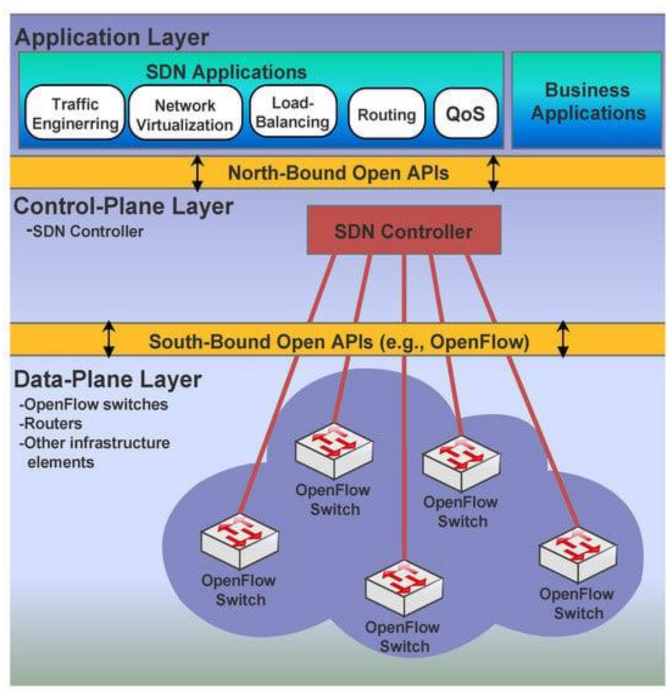

Fig. 2. Software defined network model for load balancing.

In study [23], the authors evaluate various machine learning models, highlighting their suitability for different network scenarios based on accuracy, computational requirements, and ease of implementation. These models, ranging from supervised learning algorithms like decision trees and neural networks to unsupervised techniques like clustering, have found applications across network security, traffic classification, and resource management [24]. Particularly, the predictive and adaptive nature of ML models positions them as ideal candidates for managing unpredictable, high-volume network traffic in real-time [25].

C. Synergizing Machine Learning with SDN for Enhanced Load Balancing

The intersection of machine learning with SDN, especially in the context of load balancing, is a relatively nascent yet rapidly evolving field of research. Several pioneering studies have demonstrated the feasibility and benefits of this integration. For instance, the work in [26] introduces an ML-based framework that empowers the SDN controller with data-driven intelligence to dynamically manage network loads, optimizing both the distribution and routing of traffic. By monitoring network conditions and making informed decisions, this approach mitigates common issues such as bottlenecks and uneven server workloads.

A similar study in [27] exploits the predictive capabilities of ML to forecast traffic patterns and potential hotspots in the network, enabling proactive load balancing measures before servers are critically impacted. Here, machine learning models are trained using historical network data, achieving the foresight necessary for anticipating and preparing for future traffic conditions. Another notable research [28] adopts reinforcement learning, an approach where algorithms learn optimal strategies through trial and error, fostering a self-adjusting, resilient load balancing mechanism. Fig. 3 demonstrates an intelligent software defined network architecture with machine learning techniques.

D. Potential Challenges and Ethical Implications

However, the implementation of ML-based solutions in SDN load balancing is not without its challenges. References [29] and [30] discuss technical hurdles, including the need for large training datasets, algorithmic complexity, and the intensive computational power required for real-time analysis and decision-making. Furthermore, concerns regarding data privacy and security emerge, considering the sensitive nature of network traffic data, necessitating robust encryption and privacy preservation techniques [31].

The literature also addresses the broader ethical implications of integrating ML into network systems. The autonomous decision-making aspect, while contributing to efficiency, raises questions about accountability and transparency in machine decisions, especially in scenarios leading to service denial or prioritization [32]. Studies like [33] argue for the establishment of ethical guidelines and regulatory frameworks to ensure that ML-driven networking adheres to principles of fairness, privacy, and non-discrimination.

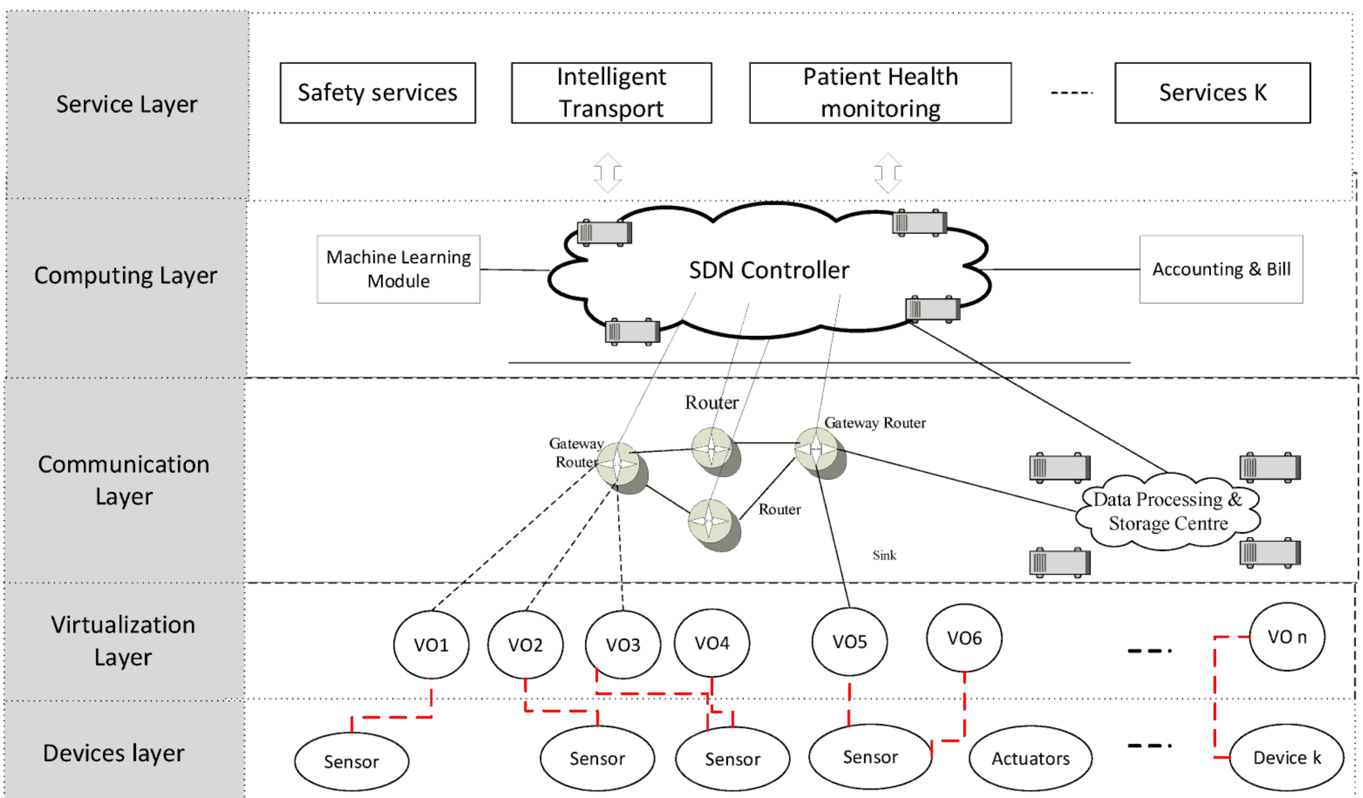

Fig. 3. Intelligent software defined network architecture with machine learning.

In conclusion, the synergy between machine learning and Software-Defined Networking opens a new frontier in network management, particularly in addressing the perennial issue of load balancing. While current research, as explored, has laid a substantial groundwork in this field, indicating notable improvements in network performance and resource optimization, it is evident that further explorations and solutions are necessary [34-36]. Future studies will need to delve deeper into overcoming the practical and ethical challenges present, pushing the boundaries to realize the full potential of ML-integrated SDN systems. This endeavor not only involves the technical refinement of algorithms and systems but also the establishment of standards and protocols that align with ethical norms and societal values.

III. Materials and Methods

In this study, we employed Jupyter Notebook as the primary platform for executing our research methodology, aiming to innovate within the realm of Software-Defined Networking (SDN) by integrating a multifaceted parameters approach. The data pivotal to our research was sourced from Kaggle.com, originating from Universidad Del Cauca Popayan, Colombia, which provided a rich foundation for our empirical analysis. A visual representation of the methodology adopted is detailed in Fig. 4, elucidating the sequential steps involved in the proposed technique.

The essence of our proposed methodology initiates when a client propels a request targeted at a specific service from the server. This request is strategically directed first to the Software Load Balancer, where our uniquely designed algorithm is embedded. The initial phase of the algorithmic process involves the extraction of flow statistics, encompassing elements such as IP addresses, port data, inter-arrival times, and more, a task accomplished utilizing the CIC Flowmeter [30].

Subsequent to the accumulation of these integral features, the algorithm proceeds to compute the cluster value associated with the incoming request. This calculated value plays a crucial role in the ensuing step, where the request is systematically channeled to the most fitting server, optimizing efficiency and resource allocation [31]. This mechanism ensures not only equitable server load distribution but also a refined, responsive interaction between the client and server systems. Fig. 5 illustrates of topology of self-organizing map.

The forthcoming sections of this paper promise a comprehensive dissection of each component within our methodology. By unfolding the intricate layers of our approach, we intend to provide clear insights into the functionality of the proposed system, emphasizing its potential to revolutionize load balancing in Software-Defined Networks through intelligent, parameter-sensitive mechanisms [32]. This exploration will underline the practical implications of our research, contributing significantly to the existing body of knowledge in this dynamic field of study.

A. Proposed Method

Within the framework of our proposed methodology, a distinct approach is employed concerning the handling of client-initiated requests for specific services directed at the server. Initially, these requests are routed towards the Software Load Balancer, which incorporates a specially designed algorithm for this precise function. The primary action of this algorithm involves the acquisition of detailed flow statistics through the application of the CIC Flowmeter. This step is critical as it gathers comprehensive data, including various nuanced network traffic features essential for the subsequent stages. Fig. 6 demonstrates an algorithmic structure of the proposed method.

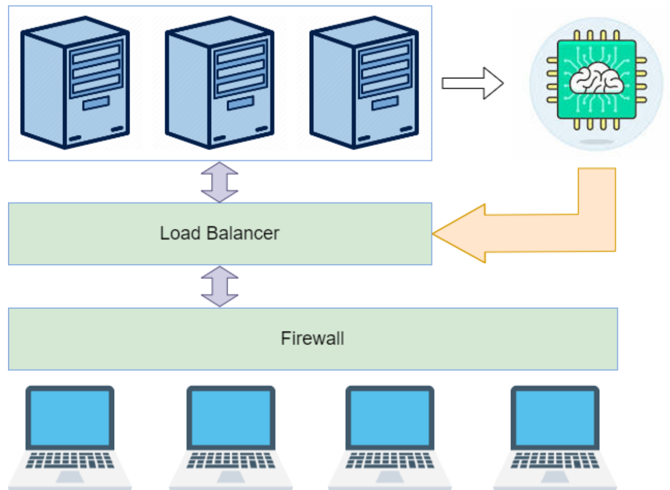

Fig. 4. Sequential steps of a load balancer.

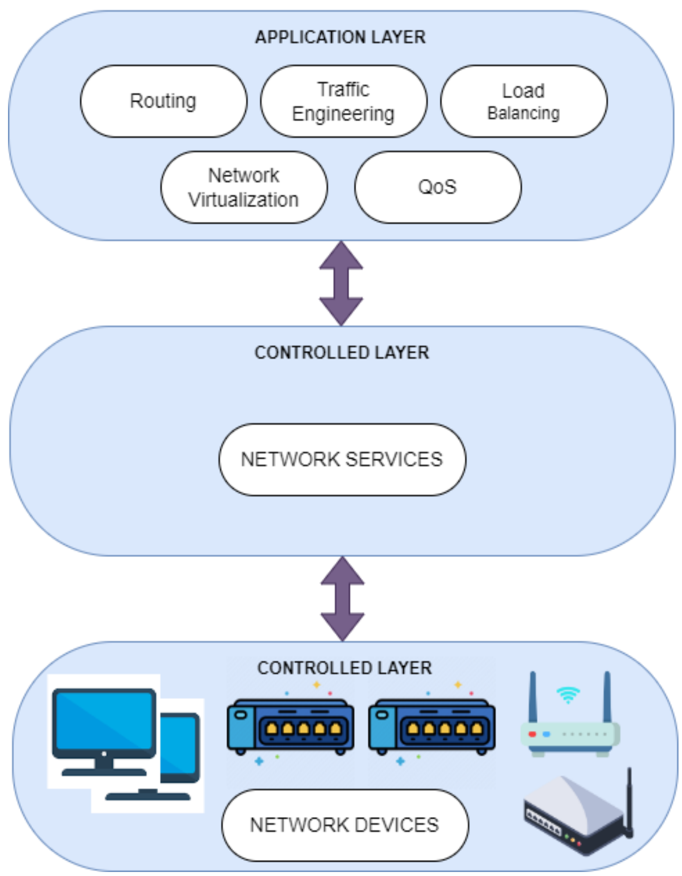

Fig. 5. Topology of self-organizing map.

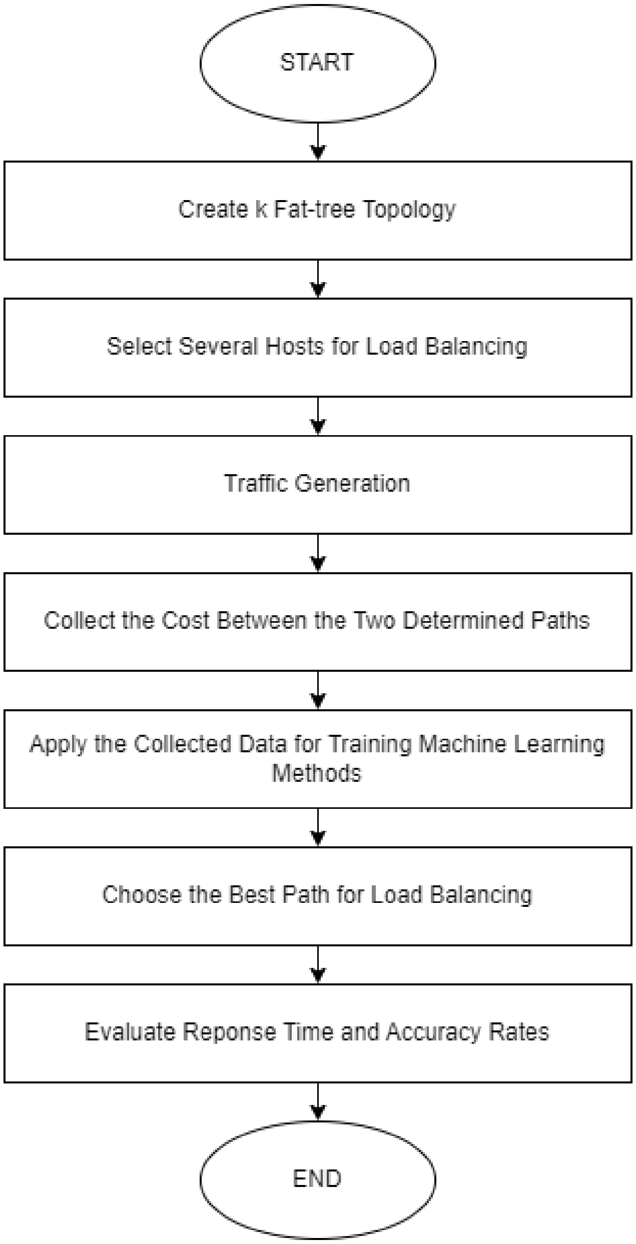

Fig. 6. Algorithmic structure of the proposed method.

Following the data collection, a sophisticated process is initiated where the algorithm, utilizing techniques from the KMeans clustering method, diligently computes the cluster value corresponding to each request [33]. This phase is integral for the intelligent distribution of network requests. In conjunction, the system conscientiously assesses the current load of the targeted server, specifically gauging the number of requests it is currently processing against its known capacity thresholds.

If the server's load is ascertained to be below the predefined acceptable threshold, the request is then responsibly forwarded [34]. However, if the contrary is identified — indicating a potential overload scenario — the system activates a contingency mechanism. Instead of overburdening the server, the request is strategically redirected to intermediate nodes within the network infrastructure [35]. This judicious decision ensures the prevention of any single point's overload, thereby maintaining a harmonious, efficient network flow and service delivery.

This intricate process exemplifies the core operational strategy of our load balancing methodology, tailored for Software-Defined Networks (SDN) and enhanced via machine learning techniques. The subsequent section of this study will delve deeper into the specifics of the dataset employed, illuminating how this foundational element plays a pivotal role in informing and guiding the algorithm's decision-making processes. Through this, we aim to underscore the feasibility and effectiveness of integrating advanced machine learning techniques into traditional load balancing practices, setting a new benchmark for efficiency and intelligent network management.

B. Dataset

In the course of our research, we sourced our data from an extensive dataset available on the Kaggle platform, originating from network traffic captured at Universidad Del Cauca in Popayan, Colombia, specifically during the morning and evening hours of the year 2017 [36-37]. This comprehensive dataset encompasses a total of 3,577,296 captured packets. For the purposes of our study, a subset consisting of 100,000 packets was extracted to serve as the training material for our machine learning model.

The original dataset is characterized by a complexity of 87 distinct features, each providing insights into various aspects of the network packets. However, for the scope of our work focused on load balancing, we distilled our feature set, selecting 12 critical features instrumental in our analysis and subsequent processing [38]. These features, detailed henceforth, provide the granular data necessary for the intricate understanding and effective management of network load dynamics:

Source.IP: Represents the IPV4 address originating from the client.

Destination.IP: Specifies the IPV4 address intended for receipt by the destination.

Source.Port: Denotes the unique port number used by the source.

Destination.Port: Identifies the respective port number at the destination.

Flow.Duration: Captures the cumulative duration of the flow in milliseconds (ms).

Flow.IAT.Std: Measures the standard deviation of packet inter-arrival times within the flow, reflecting the variability in packet transmission intervals.

Fwd.IAT.Std: Indicates the standard deviation of inter-arrival times of packets forwarded from source to destination, offering insights into the consistency of sent packets.

Bwd.IAT.Std: Contrasts the above by presenting the standard deviation of inter-arrival times for packets received from the destination.

Packet.Length.Std: Accounts for the standard deviation of packet lengths, acknowledging the variability in the size of packets traversing the network.

Down.Up.Ratio: Analyzes the ratio of download to upload traffic, a critical factor in understanding network load.

Active.Std: Observes the standard deviation of the time periods wherein the packets are actively engaged in transmission before transitioning to inactivity.

Idle.Std: Mirrors the active standard deviation by monitoring the idle periods preceding packet transmission activity.

This illustration is imperative, granting stakeholders an overview of the data's distribution and tendencies, thereby underlining the empirical foundation upon which subsequent load balancing strategies are developed and validated. This rigorous approach ensures that the machine learning model is not only grounded in reality but also attuned to the nuances of network traffic behavior, essential for the optimization of load balancing in Software-Defined Networks [39].

In this study, an intricate understanding of the dataset's characteristics is pivotal, ensuring a robust foundation for subsequent analyses. This is achieved through a detailed statistical breakdown of the dataset, focusing on several key metrics that offer insights into each data column's properties [40]. These specific parameters are integral to comprehending the underlying data structure and are crucial for the preparatory phases of data handling, particularly data preprocessing [41-45]. The parameters include:

Count: This represents the total number of entries or elements present in a specific column. It is fundamental in identifying the data volume and assessing if any missing values are in the dataset that may skew analysis or require imputation.

Mean: This metric provides the average value of the data in a particular column, offering a central value around which the data points tend to cluster. This is crucial for understanding the typical or 'normal' value, helping identify outliers or trends that deviate from the expected pattern.

Standard Deviation (std): The standard deviation indicates the dispersion or variability of the data points in a column around the mean. A higher standard deviation signifies greater variability, and conversely, a lower standard deviation suggests that the values cluster closely around the mean. Understanding this aspect is vital for predicting the consistency of data and managing expectations for anomaly detection.

Minimum (min): This value highlights the smallest number recorded in a data column. Identifying minimum values is critical, especially in scenarios where certain thresholds or limits must not be breached for operational or security reasons.

Percentiles (25th, 50th, and 75th): These values delineate the distribution of data in quartiles, indicating where a particular data point stands relative to the rest of the dataset. The 25th percentile represents the lower quartile, the 50th percentile (or median) denotes the middle value, and the 75th percentile signifies the upper quartile. Analyzing these metrics assists in understanding the data's spread, skewness, and the presence of potential outliers.

Maximum (max): Conversely to the minimum, this value represents the highest value in a column. Recognizing the maximum values is essential, particularly in contexts where operations or functionalities are sensitive to value surges, requiring optimizations or safeguarding measures.

These comprehensive statistics serve as the bedrock for data preprocessing, as they collectively offer a multi-dimensional view of the data's behavior. By understanding these fundamental aspects, researchers and analysts can make informed decisions in the subsequent stages of the study. The ensuing section delves deeper into 'Data Preprocessing,' where these statistical insights are employed to refine the dataset, ensuring it is primed for the ensuing stages of machine learning model development and deployment. This process is critical, as the quality of data preprocessing significantly influences the accuracy and reliability of outcomes in data-driven initiatives.

C. Data Preparation

Data preprocessing stands as a cornerstone in the realm of data analytics and machine learning, pivotal in refining raw data into a more digestible and analyzable format, enhancing the efficacy of subsequent operations [46]. This transformative process encompasses several stages, each critical to enhancing the data's quality and, consequently, the outcomes derived from it [47-49]. The stages integral to data preprocessing include:

Data Cleaning: This initial stage addresses the dataset's hygiene, identifying and rectifying missing, incomplete, or irrelevant records and anomalies. It ensures coherence and consistency in the dataset, preparing it for more accurate analysis. Methods like imputation, noise filtering, or outright removal of corrupted data fall under this category.

Data Transformation: Post-cleaning, data transformation or scaling procedures are employed, often to streamline the data attributes across a common scale or format. This harmonization is crucial for algorithms to interpret the data accurately, ensuring that variances in measurements, magnitudes, or formats don't lead to biases or misinterpretations. Techniques such as normalization, aggregation, or encoding categorical variables are typical examples of data transformation.

Data Reduction: Given the extensive volumes of data in contemporary use, data reduction techniques are essential to distill the information and retain only the most relevant attributes for analysis. This step enhances computational efficiency and focuses the analysis on the data aspects most critical to the objectives. Methods like dimensionality reduction, feature selection, and binning are commonly utilized for this purpose.

In the context of our research methodology, these preprocessing steps are meticulously applied. Initially, an examination for missing or noisy data is conducted, alongside an assessment of each parameter's data type. This evaluation is facilitated by modules from Python's extensive libraries, particularly pandas for data manipulation and Seaborn for statistical visualization.

Upon identifying irregularities, we implement hashing on the datasets using the 'apply map()' method inherent in pandas, converting data into a fixed-size value or index. This process streamlines the data structure, enhancing its manageability and security.

Subsequently, to ensure uniformity in data scales, especially when dealing with parameters with varying measurement units or ranges, normalization is carried out. Here, we employ the min-max scaling technique, a standard form of feature scaling that restructures the data values so they reside within a specified range, typically between 0 and 1. This scaling is vital, making sure no particular feature dominates others due to its scale, and it works by re-scaling the range of features while maintaining the same relative distribution and relation between values.

The min-max scaler operates on a straightforward principle, adjusting values along the dataset's minimum and maximum values according to the formula [50]:

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

where, Xnorm is the normalized value, X represents the original value, and Xmin and Xmax denote the minimum and maximum values in the original data feature, respectively.

This formula's application rescales the data, ensuring a balanced representation across features, enhancing the subsequent machine learning models' performance by reducing the potential bias that disproportionate data might introduce. The ensuing sections will delve into how these preprocessed data contribute to building a robust, intelligent system.

D. Machine Learning Model Training

In the realm of machine learning, particularly in clustering methodologies, understanding the concept of distance is crucial as it is fundamentally linked to how algorithms, such as KMeans and DBSCAN, group data points into distinct clusters based on inherent similarities or differences [51]. Both these algorithms employ the concept of Euclidean distance, a measure indicative of the 'straight line' distance between two points in a multi-dimensional space, often visualized through geometric planes in basic examples, but in practice, applicable to spaces with any number of dimensions.

Euclidean distance is pivotal because it quantifies the disparity between two entities, which is foundational for clustering. Essentially, a smaller Euclidean distance between two points signifies greater similarity, indicating that they belong to the same cluster or group. Conversely, a larger Euclidean distance is indicative of considerable dissimilarity, often signifying that the points pertain to different clusters [52].

In an n-dimensional space, which is common in machine learning applications due to the multiple features or variables considered, the Euclidean distance between two points—say, Point A and Point B—is calculated using the following mathematical formula [53]:

$$d = \sqrt{\left(A_1 - B_1\right)^2 + \left(A_2 - B_2\right)^2 + \ldots + \left(A_n - B_n\right)^2} \tag{2}$$

Or more succinctly,

$$d = \sqrt{\sum_{i=1}^{n} \left(A_i - B_i\right)^2} \tag{3}$$

Here, d represents the Euclidean distance, while Ai and Bi correspond to the coordinates of Points A and B, respectively, in the n-dimensional space. Each term within the summation accounts for a single dimension's contribution to the total distance, and the aggregation of these individual disparities across all dimensions provides a comprehensive measure of the overall distance between the two points.

This calculation becomes particularly significant in our context, where KMeans and DBSCAN are utilized for clustering based on flow statistics represented through 12 parameters. By leveraging Euclidean distance, these algorithms can effectively assess and quantify the similarity or dissimilarity between various data points (or flows), subsequently enabling the informed and logical grouping of these points into cohesive clusters.

This clustering, based on quantifiable measures, ensures a methodical and data-driven approach to categorization, essential for nuanced tasks like network traffic management, anomaly detection, or optimization in a Software-Defined Network (SDN) setting. It forms the basis for further analyses and decision-making processes that hinge on the understanding of data relationships and classifications in the multidimensional space that the dataset occupies.

IV. Experimental Results

In the investigative analysis encompassing the study, distinct scenarios were methodically examined to elucidate the performance metrics of various advanced computational algorithms, specifically focusing on Logistic Regression (LR), Support Vector Machine (SVM) with hyperparameters C valued at 1 and 100 respectively, Artificial Neural Networks (ANN), and Deep Learning (DL) techniques [54-56]. These algorithms were subjected to rigorous evaluation to discern their efficacy and responsiveness within the constructed scenarios.

The first scenario's results are meticulously documented, with the response times explicitly recorded in milliseconds. These outcomes are visually represented in Fig. 7. This format provides a clear, comparative analysis of the algorithms' performance under the conditions stipulated in the first scenario.

Conversely, the second scenario, crafted to perhaps challenge the algorithms under a different set of conditions or parameters, similarly culminates in a set of data denoting the response times, also captured in milliseconds. The results from this separate analytical run are graphically mapped out in Fig. 8. This dual-format display of results ensures a thorough interpretation of the data, aiding in the comparative scrutiny essential for holistic understanding.

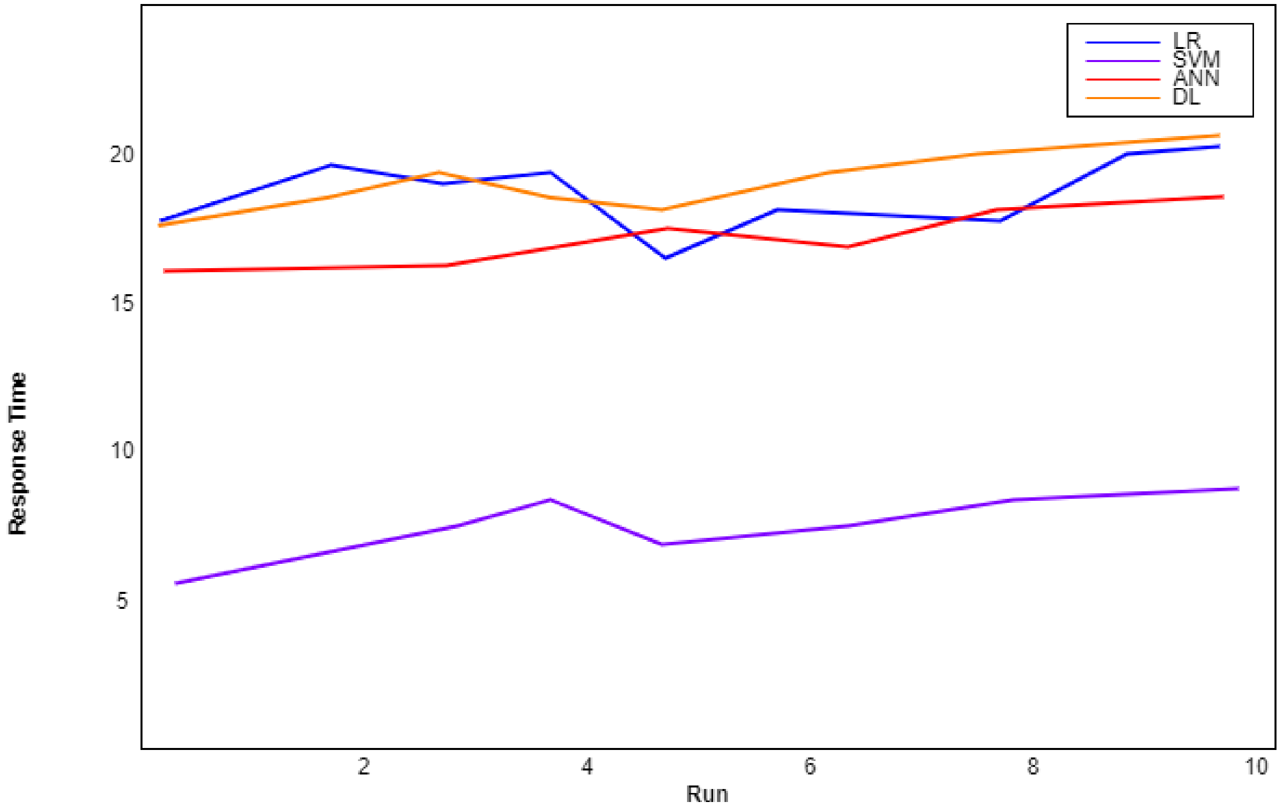

Fig. 7. Obtained results using machine learning methods in first case study.

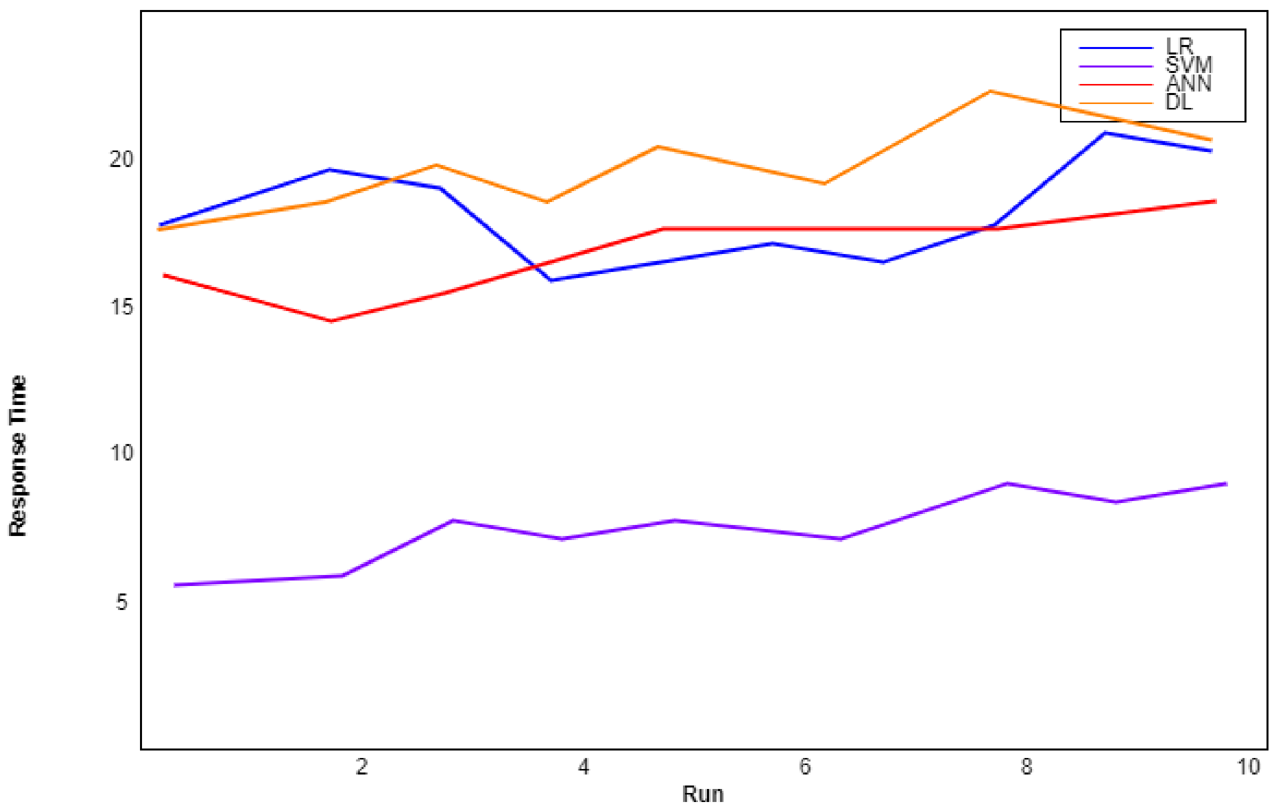

Fig. 8. Obtained results using machine learning methods in second case study.

Furthermore, an average response time was computed for each scenario, providing a consolidated metric that accounts for variance and anomalies by diluting them across multiple runs. This average was derived by executing each algorithm 100 consecutive times and then calculating the mean response time from the total accumulated data. This process, therefore, ensures the reliability and consistency of the results, acknowledging that a single run can be susceptible to anomalous external influences [57].

Such a meticulous approach to data representation not only underscores the rigorousness of the testing environment but also provides readers with a clear, unambiguous representation of each algorithm's performance. This systematic presentation and analysis are crucial for informed decision-making, further research, and practical applications of these algorithms in real-world scenarios. Fig. 9 demonstrates ROC curves of the applied machine learning methods for software defined networks.

Performance assessment stands as a pivotal component in the realm of machine learning (ML), particularly when it involves the critical evaluation of classification problems. One of the paramount tools in this evaluative arsenal is the Receiver Operating Characteristic (ROC) curve, an instrumental diagnostic graph that elucidates the competency of a classification model by displaying the trade-off between sensitivity (true positive rate) and specificity (false positive rate) across various thresholds [58].

The essence of the ROC curve lies in its capacity to illustrate the probability curve for distinct classes, thereby providing insightful commentary on the model's predictive acumen. It offers a graphical representation that is integral in comprehending how adeptly a model discerns between classes, underlining the probabilities that it correctly identifies true positives and true negatives. This finesse in estimation is paramount, as it significantly influences the decision-making processes.

In the context of the research at hand, the ROC curves were employed as an evaluative measure for distinct scenarios, each formulated to test the mettle of specific algorithms: Support Vector Machine (SVM) with regularization parameters C = 1 and C = 100, Artificial Neural Networks (ANN), and Deep Learning (DL). These scenarios, differentiated possibly by their unique datasets, conditions, or objectives, necessitated an assessment methodology that would impartially and accurately reflect the performance of each algorithm.

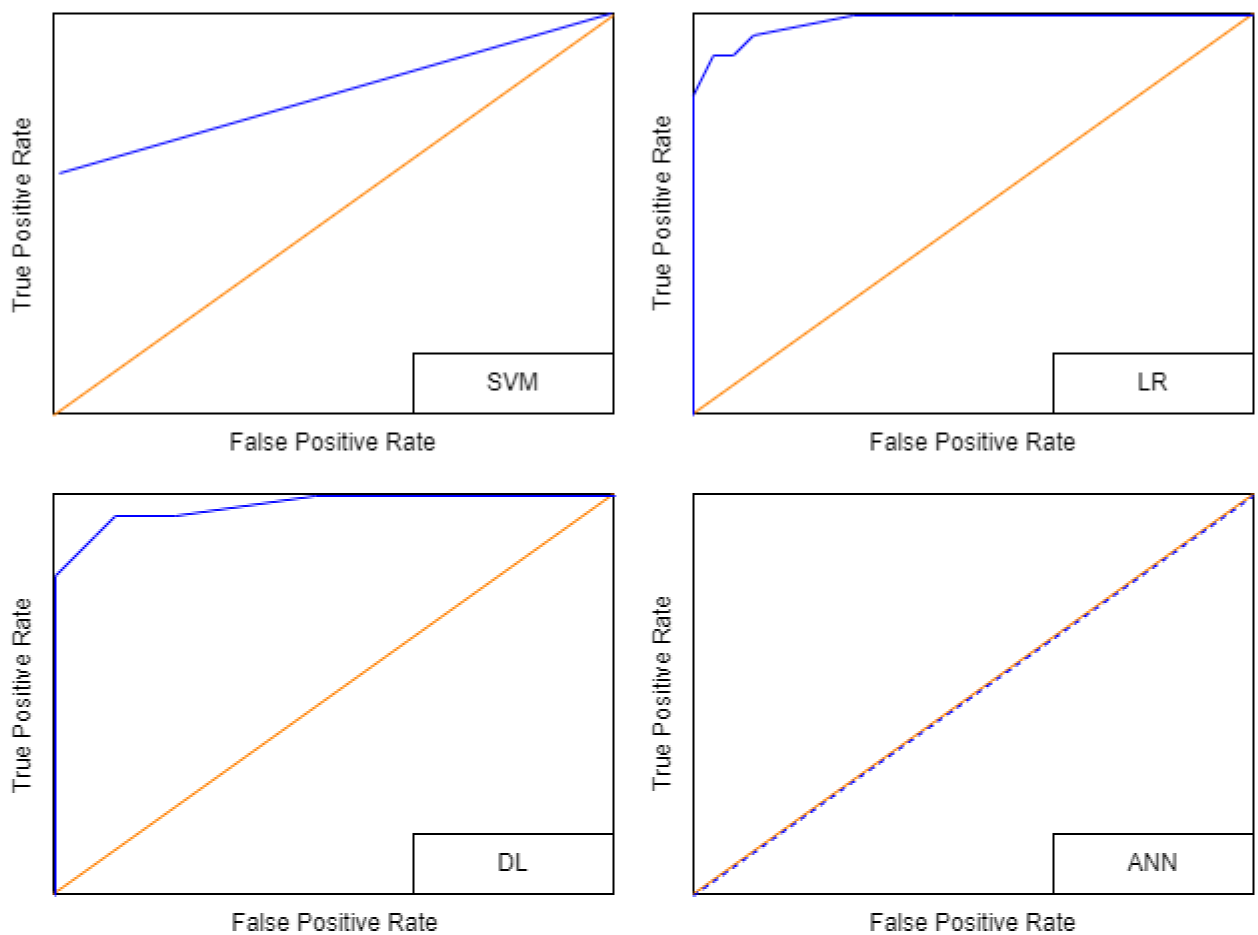

Fig. 9. ROC curves of different methods in applying machine learning for SDN.

The results of this meticulous evaluation are visually encapsulated in Fig. 9. Each figure corresponds to a different scenario and presents the ROC curves for the algorithms, thereby showcasing a comparative analysis of their performance in terms of sensitivity and specificity. This side-by-side portrayal of the ROC curves is instrumental in not only understanding the individual performance nuances of SVM, ANN, and DL within each scenario but also in drawing insightful inferences from their comparative capabilities [59].

Such analytical representations are invaluable, as they transcend mere numerical accuracy and delve into the model's ability to maintain robustness across varying class distributions and thresholds. This depth of analysis is imperative for the practical application of machine learning models, ensuring they are vetted not just on the scaffold of accuracy, but on their holistic performance and reliability in diverse operational environments.

V. Discussion

In the complex and rapidly evolving field of machine learning applied within Software-Defined Networks (SDNs), the current research initiates a nuanced dialogue, offering insights into a pioneering approach that leverages advanced algorithms for enhanced load balancing [60]. This discussion section delves into the multifaceted aspects of the study, critically analyzing the advantages, confronting the challenges, acknowledging limitations, and envisioning future trajectories.

A. Advantages

The integration of machine learning in SDN heralds a transformative phase in network management, primarily by introducing predictive analytics, automation, and adaptability. Firstly, the model's ability to predict traffic patterns and potential security threats stands as a testament to its predictive prowess, which traditional networking models lack [61]. By analyzing and learning from historical network data, the system can anticipate future loads, enabling proactive adjustments.

Moreover, automation in load balancing, facilitated by the employed machine learning techniques, significantly reduces the need for human intervention, thereby minimizing human errors and operational costs. It also frees up valuable human resources for more complex, creative tasks, enhancing productivity [62].

Furthermore, the adaptability inherent in machine learning models ensures that the system evolves with the changing network scenarios and traffic conditions. This dynamism stands in stark contrast to the static nature of traditional network configurations and is integral to managing the unpredictable, ever-changing demands on modern networks.

B. Challenges

However, the path to such integration is strewn with challenges. One of the foremost is the complexity involved in training the models. The chosen machine learning algorithms require substantial computational power and time, especially as the system scales, and the data volume grows.

Data security and privacy are other critical concerns. The necessity to collect, store, and analyze vast amounts of network data opens potential vulnerabilities for confidential information, necessitating robust security protocols that could increase overhead costs [63].

Moreover, the system's efficiency is contingent on the quality of the data fed into it. Inconsistent, incomplete, or biased data can skew the machine learning models' outcomes, leading to inaccurate predictions and suboptimal load balancing.

C. Limitations

Reflecting on the limitations of the current study underscores areas for improvement. The research predominantly focused on theoretical modeling and controlled environments, which may not fully simulate the unpredictability and heterogeneity of real-world network traffic [64].

The study's efficacy in a live environment, subjected to various unforeseen variables and security threats, remains unverified. Thus, the research, though sound in its theoretical foundation, necessitates empirical testing within a practical, operational network setting.

Another notable limitation is the selection of features and parameters for the machine learning models [65]. While the research justifies the chosen variables, there remains a question of whether other overlooked parameters could influence the load balancing process. Also, the algorithms' performance has been evaluated in isolation, without considering the potential synergies or conflicts that could arise from an integrated, multi-algorithm approach.

D. Future Directions

Addressing these limitations and challenges outlines the future trajectory of this research area. One immediate step is the transition from a controlled environment to real-world testing. Implementing the proposed model within operational SDNs will provide invaluable insights into its practical applicability and resilience, especially in the face of security threats and anomalous traffic patterns [66].

Considering the exponential growth in global data traffic, future studies must also explore models that can seamlessly scale, without a corresponding exponential demand on resources [67]. Such research could delve into more resource-efficient machine learning algorithms or hybrid models that combine the strengths of multiple algorithms to achieve more effective load balancing with lesser computational demand.

The aspect of security, given its centrality in network operations, also calls for dedicated exploration. Future research endeavors could focus on developing integrated security protocols within the machine learning models, ensuring that data privacy and network security are intrinsic to the system rather than afterthoughts.

Additionally, given the foundational role of data in machine learning, subsequent studies should investigate comprehensive, unbiased, and representative data collection methods. Enhancing the quality of input data can significantly bolster the accuracy and reliability of the predictions and decisions made by the machine learning models.

On a broader spectrum, there lies immense potential in interdisciplinary research, particularly integrating behavioral sciences with machine learning. Understanding the human aspects of network usage can add another layer of sophistication to predictive models, potentially leading to more intuitive, user-centric network management.

In conclusion, while the current study marks a significant stride towards revolutionizing SDN through machine learning, it also sets the stage for further, more nuanced exploration and innovation. The journey ahead, though demanding, holds the promise of networks that are not just smarter and more efficient but are also more in tune with the human elements they serve. Through collaborative, interdisciplinary, and forward-thinking research, the vision of truly intelligent, secure, and user-focused networks is an achievable horizon.

VI. Conclusion

The journey through this research, from conceptual frameworks to analytical discussions, reflects a profound exploration of integrating machine learning into Software-Defined Networking (SDN) to enhance load balancing. As we draw conclusions, it's imperative to encapsulate the essence of our findings and their implications for future scientific inquiry and practical application in the networking sphere.

This study marked a significant advancement by demonstrating that machine learning algorithms could revolutionize the way network resources are managed, optimizing the distribution of data loads across various pathways. By employing sophisticated algorithms, we unveiled the potential to predict network congestions, dynamically adjust to traffic changes, and improve overall efficiency and user experience. This paradigm shift from traditional methods accentuates a move towards more autonomous, self-sufficient systems capable of sophisticated decision-making processes, essential in the burgeoning era of digital transformation and the Internet of Things (IoT).

However, the research also highlighted critical challenges and limitations, from the complexities of algorithm training and data security concerns to the practical applicability of the proposed model outside simulated environments. These challenges are not terminuses but instead signposts indicating areas requiring further exploration, refinement, and innovation.

Looking forward, the implications of this research are both broad and profound. They suggest an imminent need for robust, real-world testing and the potential for interdisciplinary approaches that could further enrich these technological advances. The prospects of enhanced security measures, scalability considerations, and user-centric adaptations also present exciting, necessary trajectories.

In conclusion, this study does not represent an end but a beginning. It serves as a catalyst for continued exploration and dialogue in the realms of machine learning, networking, and beyond. The confluence of these fields holds significant promise for creating more resilient, efficient, and intelligent networks, poised to support the ever-evolving demands of future digital landscapes.

References

[1] Hamdan, M., Hassan, E., Abdelaziz, A., Elhigazi, A., Mohammed, B., Khan, S., ... & Marsono, M. N. (2021). A comprehensive survey of load balancing techniques in software-defined network. Journal of Network and Computer Applications, 174, 102856.

[2] Omarov, B., & Altayeva, A. (2018, January). Towards intelligent IoT smart city platform based on OneM2M guideline: smart grid case study. In 2018 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 701-704). IEEE.

[3] Muhammad, T. (2022). A Comprehensive Study on Software-Defined Load Balancers: Architectural Flexibility & Application Service Delivery in On-Premises Ecosystems. INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY, 6(1), 1-24.

[4] Rahman, A., Islam, J., Kundu, D., Karim, R., Rahman, Z., Band, S. S., ... & Kumar, N. (2023). Impacts of blockchain in software-defined Internet of Things ecosystem with Network Function Virtualization for smart applications: Present perspectives and future directions. International Journal of Communication Systems, e5429.

[5] Jurado-Lasso, F. F., Marchegiani, L., Jurado, J. F., Abu-Mahfouz, A. M., & Fafoutis, X. (2022). A survey on machine learning software-defined wireless sensor networks (ml-SDWSNS): Current status and major challenges. IEEE Access, 10, 23560-23592.

[6] Wu, J., Dong, M., Ota, K., Li, J., & Yang, W. (2020). Application-aware consensus management for software-defined intelligent blockchain in IoT. IEEE Network, 34(1), 69-75.

[7] Yazdinejad, A., Parizi, R. M., Dehghantanha, A., & Choo, K. K. R. (2020). P4-to-blockchain: A secure blockchain-enabled packet parser for software defined networking. Computers & Security, 88, 101629.

[8] A. Altayeva, B. Omarov, H.C. Jeong and Y.I. Cho, "Multi-step face recognition for improving face detection and recognition rate", Far East Journal of Electronics and Communications, vol. 16, no. 3, pp. 471-491, 2016

[9] Omarov, B., Omarov, B., Shekerbekova, S., Gusmanova, F., Oshanova, N., Sarbasova, A., ... & Sultan, D. (2019). Applying face recognition in video surveillance security systems. In Software Technology: Methods and Tools: 51st International Conference, TOOLS 2019, Innopolis, Russia, October 15–17, 2019, Proceedings 51 (pp. 271-280). Springer International Publishing.

[10] Rawal, B. S., Manogaran, G., Singh, R., Poongodi, M., & Hamdi, M. (2021, June). Network augmentation by dynamically splitting the switching function in SDN. In 2021 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.

[11] Latif, S. A., Wen, F. B. X., Iwendi, C., Li-Li, F. W., Mohsin, S. M., Han, Z., & Band, S. S. (2022). AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems. Computer Communications, 181, 274-283.

[12] Wang, Y., Shang, F., Lei, J., Zhu, X., Qin, H., & Wen, J. (2023). Dual-attention assisted deep reinforcement learning algorithm for energy-efficient resource allocation in Industrial Internet of Things. Future Generation Computer Systems, 142, 150-164.

[13] Cao, B., Sun, Z., Zhang, J., & Gu, Y. (2021). Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3832-3840.

[14] Keshari, S. K., Kansal, V., Kumar, S., & Bansal, P. (2023). An intelligent energy efficient optimized approach to control the traffic flow in Software-Defined IoT networks. Sustainable Energy Technologies and Assessments, 55, 102952.

[15] Poornima, E., Muthu, B., Agrawal, R., Kumar, S. P., Dhingra, M., Asaad, R. R., & Jumani, A. K. (2023). Fog robotics-based intelligence transportation system using line-of-sight intelligent transportation. Multimedia Tools and Applications, 1-29.

[16] Razdan, S., & Sharma, S. (2022). Internet of medical things (IoMT): Overview, emerging technologies, and case studies. IETE technical review, 39(4), 775-788.

[17] Kazmi, S. H. A., Qamar, F., Hassan, R., Nisar, K., & Chowdhry, B. S. (2023). Survey on joint paradigm of 5G and SDN emerging mobile technologies: Architecture, security, challenges and research directions. Wireless Personal Communications, 1-48.

[18] Amiri, Z., Heidari, A., Navimipour, N. J., & Unal, M. (2023). Resilient and dependability management in distributed environments: A systematic and comprehensive literature review. Cluster Computing, 26(2), 1565-1600.

[19] Banafaa, M., Shayea, I., Din, J., Azmi, M. H., Alashbi, A., Daradkeh, Y. I., & Alhammadi, A. (2023). 6G mobile communication technology: Requirements, targets, applications, challenges, advantages, and opportunities. Alexandria Engineering Journal, 64, 245-274.

[20] Ray, P. P., & Kumar, N. (2021). SDN/NFV architectures for edge-cloud oriented IoT: A systematic review. Computer Communications, 169, 129-153.

[21] Sultan, D., Omarov, B., Kozhamkulova, Z., Kazbekova, G., Alimzhanova, L., Dautbayeva, A., ... & Abdrakhmanov, R. (2023). A Review of Machine Learning Techniques in Cyberbullying Detection. Computers, Materials & Continua, 74(3).

[22] Mughaid, A., AlZu'bi, S., Alnajjar, A., AbuElsoud, E., Salhi, S. E., Igried, B., & Abualigah, L. (2023). Improved dropping attacks detecting system in 5g networks using machine learning and deep learning approaches. Multimedia Tools and Applications, 82(9), 13973-13995.

[23] Rahman, A., Islam, M. J., Montieri, A., Nasir, M. K., Reza, M. M., Band, S. S., ... & Mosavi, A. (2021). Smartblock-sdn: An optimized blockchain-sdn framework for resource management in iot. IEEE Access, 9, 28361-28376.

[24] Ribeiro, D. A., Melgarejo, D. C., Saadi, M., Rosa, R. L., & Rodríguez, D. Z. (2023). A novel deep deterministic policy gradient model applied to intelligent transportation system security problems in 5G and 6G network scenarios. Physical Communication, 56, 101938.

[25] Javanmardi, S., Shojafar, M., Mohammadi, R., Persico, V., & Pescapè, A. (2023). S-FoS: A secure workflow scheduling approach for performance optimization in SDN-based IoT-Fog networks. Journal of Information Security and Applications, 72, 103404.

[26] Kashef, M., Visvizi, A., & Troisi, O. (2021). Smart city as a smart service system: Human-computer interaction and smart city surveillance systems. Computers in Human Behavior, 124, 106923.

[27] Qu, Y., Wang, Y., Ming, X., & Chu, X. (2023). Multi-stakeholder's sustainable requirement analysis for smart manufacturing systems based on the stakeholder value network approach. Computers & Industrial Engineering, 177, 109043.

[28] Bourechak, A., Zedadra, O., Kouahla, M. N., Guerrieri, A., Seridi, H., & Fortino, G. (2023). At the Confluence of Artificial Intelligence and Edge Computing in IoT-Based Applications: A Review and New Perspectives. Sensors, 23(3), 1639.

[29] Imam-Fulani, Y. O., Faruk, N., Sowande, O. A., Abdulkarim, A., Alozie, E., Usman, A. D., ... & Taura, L. S. (2023). 5G Frequency Standardization, Technologies, Channel Models, and Network Deployment: Advances, Challenges, and Future Directions. Sustainability, 15(6), 5173.

[30] Abou El Houda, Z., Hafid, A. S., & Khoukhi, L. (2023). Mitfed: A privacy preserving collaborative network attack mitigation framework based on federated learning using sdn and blockchain. IEEE Transactions on Network Science and Engineering.

[31] Sheng, M., Zhou, D., Bai, W., Liu, J., Li, H., Shi, Y., & Li, J. (2023). Coverage enhancement for 6G satellite-terrestrial integrated networks: performance metrics, constellation configuration and resource allocation. Science China Information Sciences, 66(3), 130303.

[32] Sutradhar, S., Karforma, S., Bose, R., & Roy, S. (2023). A Dynamic Step-wise Tiny Encryption Algorithm with Fruit Fly Optimization for Quality of Service improvement in healthcare. Healthcare Analytics, 3, 100177.

[33] Al-Turjman, F., Zahmatkesh, H., & Shahroze, R. (2022). An overview of security and privacy in smart cities' IoT communications. Transactions on Emerging Telecommunications Technologies, 33(3), e3677.

[34] Mahi, M. J. N., Chaki, S., Ahmed, S., Biswas, M., Kaiser, M. S., Islam, M. S., ... & Whaiduzzaman, M. (2022). A review on VANET research: Perspective of recent emerging technologies. IEEE Access, 10, 65760-65783.

[35] Omarov, B., Altayeva, A., & Cho, Y. I. (2017). Smart building climate control considering indoor and outdoor parameters. In Computer Information Systems and Industrial Management: 16th IFIP TC8 International Conference, CISIM 2017, Bialystok, Poland, June 16-18, 2017, Proceedings 16 (pp. 412-422). Springer International Publishing.

[36] Zhou, H., Zheng, Y., Jia, X., & Shu, J. (2023). Collaborative prediction and detection of DDoS attacks in edge computing: A deep learning-based approach with distributed SDN. Computer Networks, 225, 109642.

[37] Zhang, J., Liu, Y., Li, Z., & Lu, Y. (2023). Forecast-assisted service function chain dynamic deployment for SDN/NFV-enabled cloud management systems. IEEE Systems Journal.

[38] Priyadarshini, R., & Barik, R. K. (2022). A deep learning based intelligent framework to mitigate DDoS attack in fog environment. Journal of King Saud University-Computer and Information Sciences, 34(3), 825-831.

[39] Das, S. K., Benkhelifa, F., Sun, Y., Abumarshoud, H., Abbasi, Q. H., Imran, M. A., & Mohjazi, L. (2023). Comprehensive review on ML-based RIS-enhanced IoT systems: basics, research progress and future challenges. Computer Networks, 224, 109581.

[40] Mubarakali, A., Durai, A. D., Alshehri, M., AlFarraj, O., Ramakrishnan, J., & Mavaluru, D. (2023). Fog-based delay-sensitive data transmission algorithm for data forwarding and storage in cloud environment for multimedia applications. Big Data, 11(2), 128-136.

[41] Liu, D., Li, Z., & Jia, D. (2023). Secure distributed data integrity auditing with high efficiency in 5G-enabled software-defined edge computing. Cyber Security and Applications, 1, 100004.

[42] Omarov, B., Suliman, A., & Tsoy, A. (2016). Parallel backpropagation neural network training for face recognition. Far East Journal of Electronics and Communications, 16(4), 801-808.

[43] Gong, J., & Rezaeipanah, A. (2023). A fuzzy delay-bandwidth guaranteed routing algorithm for video conferencing services over SDN networks. Multimedia Tools and Applications, 1-30.

[44] Altayeva, A. B., Omarov, B. S., Aitmagambetov, A. Z., Kendzhaeva, B. B., & Burkitbayeva, M. A. (2014). Modeling and exploring base station characteristics of LTE mobile networks. Life Science Journal, 11(6), 227-233.

[45] Shahraki, A., Abbasi, M., Piran, M. J., & Taherkordi, A. (2021). A comprehensive survey on 6G networks: Applications, core services, enabling technologies, and future challenges. arXiv preprint arXiv:2101.12475.

[46] Ahmed, A. A., Malebary, S. J., Ali, W., & Barukab, O. M. (2023). Smart traffic shaping based on distributed reinforcement learning for multimedia streaming over 5G-VANET communication technology. Mathematics, 11(3), 700.

[47] Sahoo, S. K., Mudligiriyappa, N., Algethami, A. A., Manoharan, P., Hamdi, M., & Raahemifar, K. (2022). Intelligent trust-based utility and reusability model: Enhanced security using unmanned aerial vehicles on sensor nodes. Applied Sciences, 12(3), 1317.

[48] Mahmoodi Khaniabadi, S., Javadpour, A., Gheisari, M., Zhang, W., Liu, Y., & Sangaiah, A. K. (2023). An intelligent sustainable efficient transmission internet protocol to switch between User Datagram Protocol and Transmission Control Protocol in IoT computing. Expert Systems, 40(5), e13129.

[49] Fatemidokht, H., Rafsanjani, M. K., Gupta, B. B., & Hsu, C. H. (2021). Efficient and secure routing protocol based on artificial intelligence algorithms with UAV-assisted for vehicular ad hoc networks in intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4757-4769.

[50] Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. Computers and Electrical Engineering, 99, 107810.

[51] Himeur, Y., Elnour, M., Fadli, F., Meskin, N., Petri, I., Rezgui, Y., ... & Amira, A. (2023). AI-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives. Artificial Intelligence Review, 56(6), 4929-5021.

[52] Sangaiah, A. K., Javadpour, A., Ja'fari, F., Pinto, P., Zhang, W., & Balasubramanian, S. (2023). A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things. Cluster Computing, 26(1), 599-612.

[53] Javed, A. R., Shahzad, F., ur Rehman, S., Zikria, Y. B., Razzak, I., Jalil, Z., & Xu, G. (2022). Future smart cities: Requirements, emerging technologies, applications, challenges, and future aspects. Cities, 129, 103794.

[54] Alhayani, B., Kwekha-Rashid, A. S., Mahajan, H. B., Ilhan, H., Uke, N., Alkhayyat, A., & Mohammed, H. J. (2023). 5G standards for the Industry 4.0 enabled communication systems using artificial intelligence: perspective of smart healthcare system. Applied nanoscience, 13(3), 1807-1817.

[55] Tursynova, A., & Omarov, B. (2021, November). 3D U-Net for brain stroke lesion segmentation on ISLES 2018 dataset. In 2021 16th International Conference on Electronics Computer and Computation (ICECCO) (pp. 1-4). IEEE.

[56] Zhang, Q., Li, C., Huang, Y., & Luo, Y. (2023). Effective multi-controller management and adaptive service deployment strategy in multi-access edge computing environment. Ad Hoc Networks, 138, 103020.

[57] bin Salleh, R., Koubaa, A., Khan, Z., Khan, M. K., & Ali, I. (2023). Data plane failure and its recovery techniques in SDN: A systematic literature review. Journal of King Saud University-Computer and Information Sciences.

[58] Xu, X., Li, H., Xu, W., Liu, Z., Yao, L., & Dai, F. (2021). Artificial intelligence for edge service optimization in internet of vehicles: A survey. Tsinghua Science and Technology, 27(2), 270-287.

[59] Khan, A. A., Laghari, A. A., Rashid, M., Li, H., Javed, A. R., & Gadekallu, T. R. (2023). Artificial intelligence and blockchain technology for secure smart grid and power distribution Automation: A State-of-the-Art Review. Sustainable Energy Technologies and Assessments, 57, 103282.

[60] Strinati, E. C., Alexandropoulos, G. C., Sciancalepore, V., Di Renzo, M., Wymeersch, H., Phan-Huy, D. T., ... & Denis, B. (2021, June). Wireless environment as a service enabled by reconfigurable intelligent surfaces: The RISE-6G perspective. In 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit) (pp. 562-567). IEEE.

[61] Haque, A. B., Bhushan, B., & Dhiman, G. (2022). Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends. Expert Systems, 39(5), e12753.

[62] Ghiasi, M., Niknam, T., Wang, Z., Mehrandezh, M., Dehghani, M., & Ghadimi, N. (2023). A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future. Electric Power Systems Research, 215, 108975.

[63] Ali, O., Abdelbaki, W., Shrestha, A., Elbasi, E., Alryalat, M. A. A., & Dwivedi, Y. K. (2023). A systematic literature review of artificial intelligence in the healthcare sector: Benefits, challenges, methodologies, and functionalities. Journal of Innovation & Knowledge, 8(1), 100333.

[64] Mori, S., Mizutani, K., & Harada, H. (2023). Software-Defined Radio-Based 5G Physical Layer Experimental Platform for Highly Mobile Environments. IEEE Open Journal of Vehicular Technology, 4, 230-240.

[65] Neubauer, M., Reiff, C., Walker, M., Oechsle, S., Lechler, A., & Verl, A. (2023). Cloud-based evaluation platform for software-defined manufacturing: Cloud-basierte Evaluierungsplattform für Software-defined Manufacturing. at-Automatisierungstechnik, 71(5), 351-363.

[66] Kube, A. R., Das, S., & Fowler, P. J. (2023). Fair and efficient allocation of scarce resources based on predicted outcomes: implications for homeless service delivery. Journal of Artificial Intelligence Research, 76, 1219-1245.

[67] Hashem, I. A. T., Usmani, R. S. A., Almutairi, M. S., Ibrahim, A. O., Zakari, A., Alotaibi, F., ... & Chiroma, H. (2023). Urban Computing for Sustainable Smart Cities: Recent Advances, Taxonomy, and Open Research Challenges. Sustainability, 15(5), 3916.