International Journal of Computing and Digital Systems
2025, VOL. 17, NO.1, 1-19

http://dx.doi.org/10.12785/ijcds/1571039804

Dynamic Fast Convergence Improvement using Predictive Network Analysis

Mohammed Hussein Ali[1]

[1]Engineering Department, Collage of Engineering, Aliraqia University, Iraq

Received 16 June 2024, Revised 30 October 2024, Accepted 28 November 2024

Abstract: In today's digital landscape, network infrastructure is crucial for the daily operations of every firm. Service outages may occur owing to disruptions stemming from changes in network topology. This article presents research that addresses these concerns by optimizing network convergence in the Spanning Tree Protocol (STP) with predictive analytics and automated adjustments. Abstract: This research presents a framework that integrates machine learning methods (ARIMA) with link prediction and graph embedding to reduce convergence time and enhance network stability. Real-time network monitoring, coupled with predictive analytics, allows dynamic modifications, significantly minimizing downtime. The authors claim that their methodology yields a 70% accuracy in forecasting short convergence times and 80% accuracy in predicting conjugation terminality's. It accurately predicts substantial outages in 85% of all dimensions and optimizes resource management with moderate precision for low usage (75%), and high utilization (70%). The findings indicate that the improved STP may diminish downtime, augment resilience, and boost resource utilization uplink efficiency, making this technique a viable option for near real-time network management.
Keywords: Predictive Network Analysis , STP, dynamic environments, vital real-world issue

1. Introduction

A. Overview

In today's linked digital landscape, the efficacy of network infrastructure is essential for corporate activities. Disruptions caused by alterations in network topology, such as connection failures or reconfigurations, may lead to significant service outages, resulting in substantial financial losses, decreased productivity, and dissatisfied customers. At the conclusion of all, it suffices to mention that in this age, keeping your network up and running is not a choice but a necessity for enterprises. Many STP implementations may take too long to converge when the network topology changes, producing performance concerns while also endangering access for key services needing an always-on connection.

B. Problem Statement

Very crucial in operations for networks, we require short convergence time since the uptime of a service relies on it. The slightest network failure may lead to considerable losses, such as financial losses, reputational damage, and diminished productivity/community. In this paper, we present a novel approach to the problem: incorporating predictive network analytics into Layer 2, specifically STP. Our upgraded STP solution automatically adjusts the convergence parameters on-the-fly by using actual and historical data, delivering shorter recovery times, reducing downtimes while dealing with topological changes to boost robustness. Rather than responding like typical STP solutions, which modify depending on changing network circumstances, we employ predictive analytics to proactively increase performance. Table 4 illustrates that our new combination of machine learning approaches, i.e., ARIMA, link prediction, and graph embedding, is competent to offer predictions for network behavior in real time as well as convergence times, which make the network more flexible and efficient. This assures quicker convergence, higher stability, and more efficient resource usage compared to conventional STP techniques.

C. Objectives

In recent work, there has been a lot of investigation into ways to increase network performance. As an example, "Comprehensive analyses of image forgery detection methods from traditional to deep learning approaches" traces the progression of strategies in digital spaces, and similarly, with "A GAN-Based Model for Deep fake Detection in Social Media," it considers entirely new machine investigations that advance across dynamic contexts. Research that has helped to enhance the efficiency of network systems in different domains [37, 38], such as "Energy-Efficient and Congestion-Thermal Aware Routing Protocol for WBAN"

E-mail address: Mohammed.h.ali@aliraqia.edu.iq

(Chen et al. We add to the literature of network convergence, building upon these findings by incorporating a prediction model for real time alterations on networks.

D. Outline of Paper

The rest of this paper is arranged as follows: Section 2 discusses the history and related studies, concentrating on classical STP and newer efforts in network convergence. This is followed by the Proposed Solution section, in which we outline our new approach and its key pieces, ending with a methodology segment that details how both were implemented along with their assessment. The remainder of the study provides experiments for validation in Section III, and a discussion with regard to benefits limits implementation concerns. defies scalability implications is addressed with future work mentioned.

2. Literature Review

A. Traditional STP Implementations and Limitations

Traditional implementations of the Spanning Tree Protocol (STP) have long been essential in network infrastructure [1]. It provides a technique to prevent loops and guarantee network stability [2]. notwithstanding their broad use. These standard procedures are not without their limits[3]. mainly highlighted by difficulties such as delayed convergence [4].

STP operates by detecting and terminating duplicate routing paths inside a network to eliminate loops, thus preventing data packets from keep circulating in a network[5]. traffic jam or failure on the network. Although it has been successful in theory, this method of convergent path-selection is rather slow. specifically, in bigger or more complicated networks [6]. In that case, during topology changes such as link failures or network reconfigurations, most STP implementations might need a long period to converge [7]. this brings about a short disconnection of network with compromised performance[8].

The implication of the long convergence time is not only that people will be annoyed, but also that serious problem will arise for the network managers and the enterprises[9]. Extended convergence may cause high latency, packet loss, or even service outages[10]. this can be very frustrating for the users and can prevent the business from performing its necessary activities. Consequently, the importance of maintaining uninterrupted connectivity is especially important in the ever-changing, fast-paced digital world where every delay in network services can lead to financial losses and damage to a company's reputation[8].

Thus, though the STP implementation of the classical version was a foundation of the network stability, its inherent constraints, such as the convergence speed, point to the need for new approaches that would be capable of the successful overcoming of the difficulties[11].

B. Related Research

Advanced network convergence methods are being pursued as one of the critical areas of network management, and the research in this field has been conducted extensively[6]. using the techniques mentioned in this industry[12]. This part provides an outline of already researched studies on the strategies used to encourage network convergence[13]. integration of substitutive methods, and optimization algorithms[14].

Whereas the recent studies have investigated the alternative protocols applying the optimization techniques as the way to mitigate the constraints of standard network convergence procedures[15]. For instance. presented labeled RTDP, an approach aiming at increasing the convergence of real-time dynamic programming [16]. It has significance for boosting network convergence in dynamic contexts. Similarly. presented Rezero, a unique technique that provides quick convergence at wide depths [17]. exhibiting possible uses in network optimization. Moreover, research efforts have concentrated on developing unique algorithms with techniques to optimize network convergence processes[18]. examined approaches to enhance the convergence of simulation-based dynamic traffic assignment, which can have consequences for improving network traffic flow while lowering convergence times [19], proposed a robust, rapid convergence zeroing neural network, which has interesting applications in dynamic systems such as network routing with optimization [20].

Additionally. Explored learning in games and its implications for establishing robust [21]. with fast convergence in dynamic systems. showcasing the potential of game-theoretic techniques in network optimization [22]. Developed an improved MPPT approach for PV systems, emphasizing fast convergence speed. having zero oscillation, which may be customized to maximize energy-efficient network operations [23].

Research on fast convergence and cooperative dynamic spectrum access for cognitive radio networks [24]. allowing creative techniques to boost spectrum efficiency. as convergence speed is in dynamic network environments [25].looked into the quick convergence algorithms for dynamic background modeling, which may be useful in video surveillance.with network anomaly detection [26].

There are also recent developments that deal with the convergence speed of the dynamic systems: for sparse recovery [27], faster convergence rates are obtained for primal-dual systems [28]. Through these studies, the existing body of knowledge on optimization approaches that consider both network convergence and resilience is increased [29]. The literature on network convergence enhancement has a large array of strategies that include substitution of the protocols, optimization algorithms and the dynamic systems approaches [30]. They generate these studies which lead to a great deal of understanding and potential solutions to deal with the issues of network convergence in the present network scenarios [31].

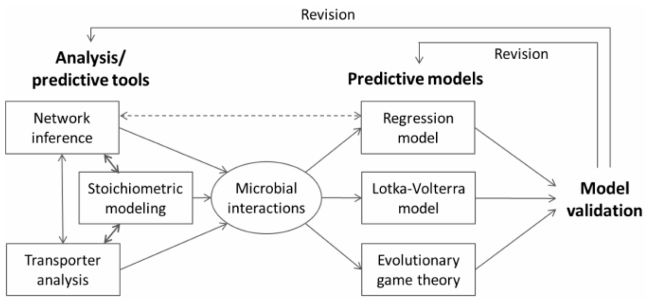

Figure 1. The development of predictive models based on network analysis methodology. which would lead to modeling revisions on a feedback basis utilizing model verification technique

C. Predictive Network Analysis

Predictive network analysis is a vital step in improving network management. the reactive approach to improving the network performance that will be achieved using complex algorithms. Employing predictive modeling approaches. Here goes a discussion of predictive network analysis as a concept. considers its potential opportunities in optimizing network performance and reliability.

Predictive network analysis involves the application of complex algorithms to anticipate network events and behaviors, using performance indicators based on previous data. includes real-time network telemetry. By integrating machine learning, statistical modeling, and data mining approaches, predictive models may foresee possible network difficulties. detect performance constraints. and optimize network setups in advance.

The applications of predictive network analysis are various. with numerous domains. including:

- Fault Prediction with Prevention: demonstrated the application of network analysis on dependency networks to anticipate software flaws. enable proactive efforts to prevent system faults with downtime [32].
- Performance Optimization: Did predictive modeling of the performance of the ATLAS TDAQ network. highlighting the possibilities for optimizing network resources. with improving overall system efficiency [33].
- Customer Churn Prediction: employed social network analysis for customer churn prediction. enable firms to detect at-risk clients. in implementing retention measures proactively [34].
- Infrastructure use: Examined techniques for anticipating poor network performance and assisting in the effective use of water resources with infrastructure [35].
- Traffic forecast: examined techniques to increase neural network performance in daily flow forecasting. allowing improved traffic management. with congestion avoidance in hydrological systems [36].
- Resource Allocation: optimized the network performance of computer pipelines in dispersed situations. supporting optimal resource allocation. with workload scheduling [37].
- Mobile Application Optimization: focuses on optimizing mobile application performance using network infrastructure-aware adaptation. enabling flawless user experiences across different network circumstances [38].
- Content Switching: addressed enhancing network efficiency via content switching. enabling effective load balancing. with traffic dispersion between servers, firewalls, and caches [39].

These numerous applications underline the adaptability and relevance of predictive network analysis in modern network management. By embracing the power of predictive analytics, organizations may proactively solve network difficulties, boost resource usage, and improve overall network resilience [40]. This section gives a look at the potential of predictive network analysis to transform network management techniques. offering a proactive attitude. utilizing a data-driven strategy to solve the challenges of current networking systems [41].

D. Image Forgery Detection Methods

Recent breakthroughs in the domain of image processing have brought back into the spotlight the urgent need to discover new approaches for identifying a fake. Deep learning algorithms are always replacing older approaches and boosting accuracy and robustness. The review of Guarnera et al. The progress of picture forgery detection is another prominent research issue that has been reviewed in Section [42], from classical techniques to deep learning-based algorithms [42]. also address the availability of GANs by applying them in social media for deepfake detection and offer this model to recognize a possible conscious AI manipulating issue. This achievement highlights how crucial deep learning models are to handle increasingly advanced counterfeit detection challenges.

Moreover, created an effective GAN-CNN ensemble model for detecting the faked photos on social media, which bred to enhance identification in a greater manner with the fusion based models. The application of the suggested model may prevent catastrophic forgetting for creating replay approaches, which provides a better discriminating towards deepfake pictures [43].

E. Deepfake Detection Techniques

The rising existence of deepfake material has prompted the creation and research of detection methods that can identify doctored media. Recent studies have offered a variety of methodologies, such as GAN-based detection models [44], the precise survey, identification of faces created by GAN, as well as technological problems encountered along with new horizons to make it more accurate in identifying faces. Moreover, investigations, this is a model that employs CNN to detect video and photos with the aid of GAN-simulated datasets, which closely match most social media sites [45].

Other researchers, concentrating on social media, have handled similar topics. Finally, generating and detecting fake material has been explored. Their research underscores the necessity for building detection systems that are able to incorporate massive social media information but stay computationally efficient [46].

F. Energy-Efficient Routing in WBANs

Wireless Body Area Networks (WBANs) where energy efficiency and congestion control play significant components, notably in health-monitoring applications. Routing and Stability An energy-efficient, congestion-thermally aware routing strategy for WBANs is suggested in [47] to decrease the power consumption as well as preserve network stability with low latency. Their result shows the necessity of effective data transfer in WBANs, which is crucial for battery life and additional critically minimal thermal effects key components to system performance [48]. Further, examine improved controllers in energy systems, concentrating more notably on electric car charging networks. Their work may also be a part of the greater discourse on energy-efficient routing and system management, particularly in dynamic real-time networks such as smart grids and microgrids [49].

G. IoV Networks with Green and Sustainable IoT

The Internet of Vehicles (IoV) has evolved, and its intelligent routing protocols have boosted the requirement for sustainable design. presented a FL fuzzy logic-based vehicle routing optimization algorithm to boost the energy efficiency and sustainability of IoV networks. This protocol will help shorten the time it takes to optimize vehicle routes in real-time with traffic congestion and environmental concerns, making IoT-connected vehicular systems greener [50]. in addition to the green mobile sensing notions of vehicular routing, go into further depth about how IoV would appear in a smart green future, which considering its aspects is supporting sustainable transportation and electric car charging. Add these to the usage of renewable energy sources in general, and it is a strategy for a sustainable Internet of Things ecosystem [51].

H. Federated Learning for IoV Networks

The federated learning model has developed as an effective option for route selection optimization in IoV systems while concurrently retaining the data privacy efficiency. for a federated learning model concentrating on sustainable routing in the Internet of Vehicular Things (IoVT) that leverages genetic algorithms to improve resource allocation and decrease environmental consequences). Their conclusion is

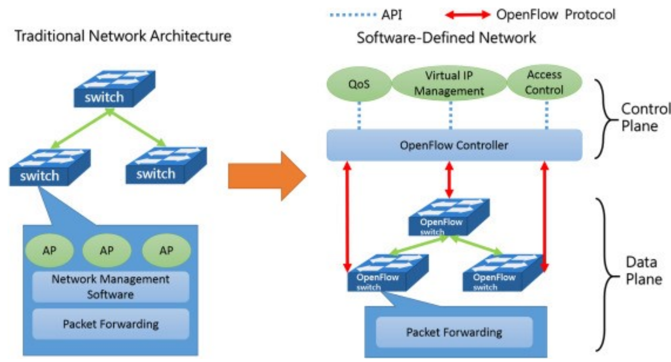

Figure 2. Example of real-time network traffic monitoring

that it gives evidence in support of enhancing the flexibility and scalability required inside IoT-enabled VNs, especially for SCs [52] .

Similarly, offer a federated learning based green edge computing scheme (FedGen) using the biological idea of genetic algorithms for optimum route selection in IoV systems. This methodology is aimed at boosting efficiency while minimizing computational overheads as well as power consumption needs for real-time data processing in automotive settings [53].

3. Proposed Solution

The proposed solution introduces a comprehensive framework aimed at addressing the challenges associated with slow convergence times in traditional STP implementations. This section outlines the key components of the proposed solution and elucidates how each component contributes to enhancing convergence speed and network resiliency.

A. Real Time Network Monitoring

Real-time network monitoring is an integral component of current network management systems. operates on the premise of constant observation of network devices. Traffic. . Performance metrics to discover and respond to issues as they develop [54]. By utilizing protocols like SNMP (Simple Network Management Protocol) or packet sniffing methods, network monitoring programs gather and analyze data from multiple network components in real-time. providing administrators with vital information about network health and performance. These technologies contain data gathering agents placed across the network architecture. centralized monitoring platforms for data consolidation and analysis, as well as warning mechanisms and reporting tools for better decision-making.

In the context of real-time network monitoring. Specific metrics and parameters are regularly checked to assess network performance. discover problems rapidly. These metrics include network bandwidth consumption, packet loss, latency, device health indicators (such as CPU and memory use), and security-related events (such as intrusion

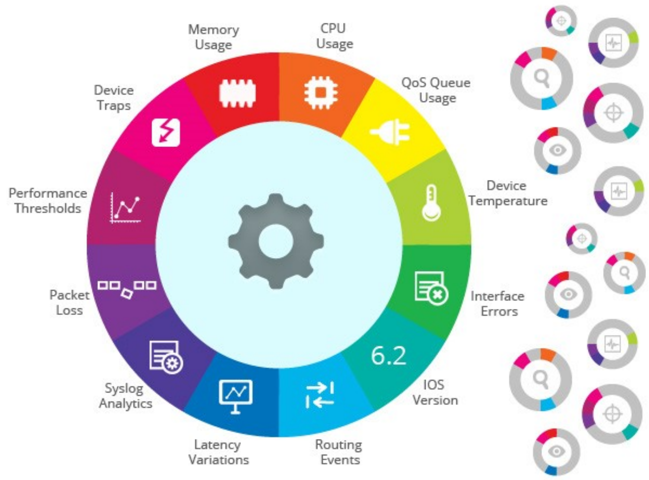

Figure 3. Predictive Network Analytics

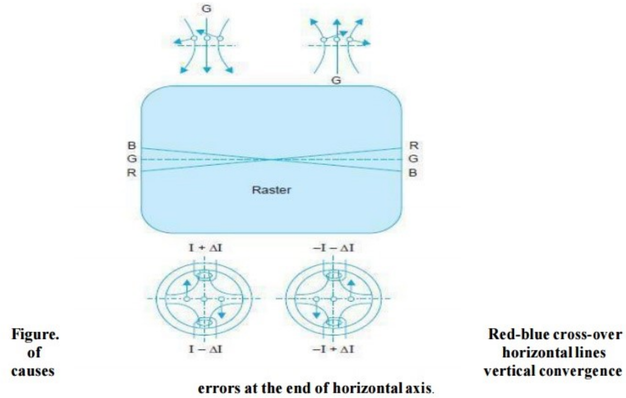

Figure 4. Example of Dynamic Convergence Adjustments

attempts or malware activity) [45, 46]. Monitoring these metrics allows administrators to define threshold levels. Receive warnings when performance surpasses specified boundaries. enabling proactive intervention to avert service outages [55].

Real-time network monitoring plays a vital role in proactive network management techniques by supporting predictive maintenance. capacity planning. and compliance monitoring. Predictive maintenance includes preemptively detecting and correcting possible faults before they impair network operations. hence decreasing downtime. increasing dependability [56]. Capacity planning helps administrators predict future resource requirements. scalability demands based on historical and real-time performance data. Moreover, compliance monitoring assures conformity to regulatory criteria. Security policies. Securing sensitive data. mitigating hazards.

B. *Predictive Analysis Engine*

The predictive analysis engine acts as a crucial component inside network management frameworks. delivering the potential to anticipate network actions. predict future convergence concerns. At its heart, this engine incorporates advanced algorithms. statistical models to examine historical network data. extrapolate future tendencies. facilitating proactive decision-making. proactive actions to enhance network performance [57].

The predictive analysis engine employs several methods and methodologies suited to the unique requirements of network forecasting. These may involve machine learning algorithms. time-series analysis, statistical modeling, and data mining techniques by evaluating enormous volumes of historical network data, such as traffic patterns, device performance metrics, and topology changes, the engine discovers underlying patterns and correlations that underlie its prediction models [58].

Historical data plays a significant part in training. verifying predictive model performance metrics. foundation for effective forecasting. Through a detailed study of prior network events and performance indicators. The predictive analytic engine discovers repeating trends, abnormalities, and possible risk factors that may affect future net-work behavior. By using this historical background, the engine boosts the accuracy of its predictions. permits proactive identification of convergence concerns before they emerge as major network events.

C. *Dynamic Convergence Adjustment*

The system for dynamically altering convergence settings constitutes a vital part of network management. allowing enterprises to react fast to changing network conditions. Maximize performance [49–51]. This dynamic adjustment procedure mixes real-time network monitoring data with predicted insights given by the analysis engine. Permitting proactive modifications to convergence parameters in response to developing network dynamics [59].

Real-time network monitoring regularly examines important performance parameters, such as connection occupancy, latency, and traffic patterns. giving regular information on network status [49]. These monitoring indicators serve as input variables for the dynamic convergence adjustment process. informed judgments on the optimization of convergence parameters.

The predicted insights supplied by the analytical engine offer extra context for dynamic convergence adjustment. predicting probable network events. spotting emergent trends or anomalies [60]. By adding predictive analytics to the adjusting process. Organizations can forecast future network behaviors. proactively fine-tune convergence parameters to avoid dangers and optimize performance.

Dynamic convergence adjustment improves convergence parameters in real-time to reduce downtime. boost network agility [60]. By constantly modifying factors such as port fees, timers, and bridge priority, the technique optimizes

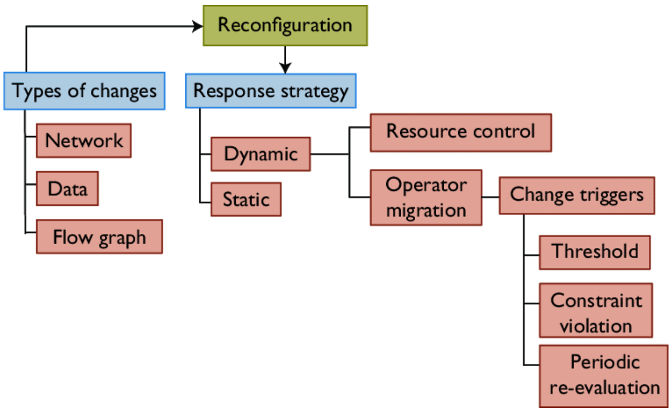

Figure 5. Components of reconfiguration in a placement strategy

network topologies to meet changes in topology, traffic load, and performance needs. This proactive technique guarantees that the network maintains optimal convergence speed and robustness. reducing the impact of topological changes. boosting overall network agility.

D. Fast Reconfiguration Mechanism

The quick reconfiguration mechanism plays a vital role in swiftly restoring network connections in reaction to failures or topological changes. guaranteeing little disturbance to network operations [61]. This method is aimed at speeding up the upgrading of network settings. rerouting traffic, hence decreasing downtime. sustaining ongoing service delivery.

The method of rapid reconfiguration encompasses many critical phases aimed at promptly recognizing and managing network disturbances. When a failure or topological change happens, the reconfiguration mechanism instantly recognizes the occurrence using real-time monitoring or signaling protocols. Upon identification, the system conducts a series of automatic activities to modify damaged network devices, such as switches, routers, or cables.

Automation plays a vital role in accelerating the reconfiguration process and enabling the quick implementation of specified reaction plans. By using predetermined algorithms or decision-making processes. The method can automate processes such as route recalculations, topology updates, and traffic rerouting. This automation reduces the need for manual intervention, providing a near-instantaneous reaction to network events.

Optimization strategies are applied to simplify the reconfiguration process. Reduce the impact on network performance [62]. These strategies may involve prioritizing vital traffic flows. Improving route selection algorithms. or exploiting parallel processing capabilities to expedite configuration changes. By improving the reconfiguration process, the method assures effective resource usage. Rap-id restoration of network connections.

E. Machine Learning Integration

Machine learning integration inside the system plays a crucial role in boosting predictive capabilities. decision-making processes [63]. By utilizing machine learning methods. The solution can evaluate enormous volumes of network data in real-time. extract useful insights. and make educated decisions to maximize network performance.

One major feature of machine learning integration is the building of prediction models that continually learn from previous network data. adapt to shifting situations [64]. These models leverage complex algorithms like neural networks, decision trees, or support vector machines to find patterns, trends, and anomalies in network activity. By studying historical data, Machine learning algorithms can anticipate probable network interruptions or performance deterioration. allowing for proactive modifications to network setups.

Moreover, machine learning algorithms are integrated into the system to automate decision-making processes. improve network settings dynamically [65]. For example, reinforcement learning algorithms can be applied to autonomously alter routing strategies or resource allocation depending on real-time feedback. performance metrics. Similarly, unsupervised learning techniques such as clustering or anomaly detection can discover abnormal network activity. prompt remedial steps to maintain optimal performance.

Furthermore. Integration of machine learning can help in making the solution self-adaptive to the evolving network conditions. Workload effectively [66]. Machine learning algorithms can learn from constant training and refinement and can take care of traffic pattern variations, user behavior, or environmental conditions. ensuring that the settings of the network are up to date in view of the growing requirements.

F. Granularity of Adjustment

The notion of granularity of adjustment indicates ability to use specific convergence parameters based on the needs of individual networks. It provides accurate handling of the way network configurations are modified in the face of dynamic environment or even operation conditions [67].

The level of precise control provided by the granularity of tweak is crucial for perfecting the network topologies and assuring the best performance in diverse scenarios. Network management can be customized at the convergence layer by varying convergence parameters in a granular way to match the characteristics of a local network, such as traffic patterns, workload dynamics, or quality-of-service demands. The level of control offered by them helps them find the right balance between stability, performance, and resource use, thus improving the network's efficiency and reliability [68].

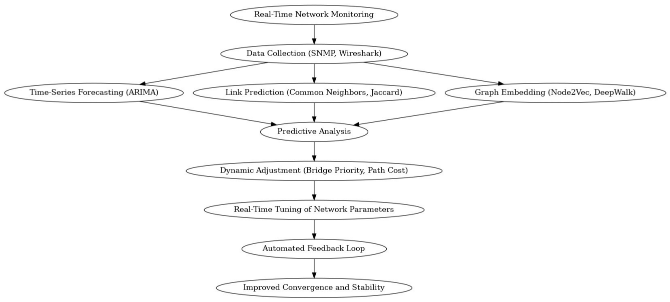

Figure 6. Flowchart of the Work

4. Methodology

We describe the methodology that will be used to implement and evaluate the solution that is being recommended, with the emphasis on data collection methods, algorithms and techniques, implementation details, machine learning integration, and the degree of adjustment.

A. *Predictive Analysis Implementation*

To enhance network convergence, we employed predictive analysis techniques to anticipate network behavior and adjust network parameters proactively. The following steps describe how the predictive analysis was implemented:

- Data Collection: Real-time network telemetry was collected from various nodes in the network using Simple Network Management Protocol (SNMP) and packet capture tools such as Wireshark. The collected data included critical performance metrics such as bandwidth utilization, latency, packet loss, and topology changes. These metrics formed the basis for forecasting network events and predicting potential disruptions.

- Time-Series Forecasting (ARIMA Model): We implemented an Auto-Regressive Integrated Moving Average (ARIMA) model to forecast network performance trends over time. ARIMA was chosen for its ability to model and predict time-dependent data. The model was trained on historical data to predict future values of key network metrics, such as convergence time and network stability. Parameters for ARIMA, including autoregressive (p), differencing (d), and moving average (q), were optimized using grid search techniques to improve prediction accuracy.

- Link Prediction: Link prediction techniques, such as Common Neighbors and Jaccard's Coefficient, were applied to forecast potential changes in the network topology. By analyzing the structure of the network graph, we identified likely future connections between nodes, enabling proactive adjustments to network paths. This helped in reducing convergence time by preemptively optimizing routing paths.

- Graph Embedding (Node2Vec and DeepWalk): To capture the network's dynamic topology, graph embedding techniques like Node2Vec and DeepWalk were employed. These methods transformed the network's nodes and edges into a lower-dimensional space, where the relationships between nodes were represented as vectors. By using these embeddings, we were able to predict topology changes more accurately, facilitating better decision-making in convergence adjustments..

B. *Dynamic Adjustment Mechanism*

The dynamic adjustment mechanism was designed to optimize convergence settings in real time based on the insights gained from predictive analysis. This process involved adjusting network parameters such as bridge priority, path cost, and convergence timers:

- Data Collection: Real-time network telemetry was collected from various nodes in the network using Simple Network Management Protocol (SNMP) and packet capture tools such as Wireshark. The collected data included critical performance metrics such as bandwidth utilization, latency, packet loss, and topology changes. These metrics formed the basis for forecasting network events and predicting potential disruptions.

- Real-Time Network Monitoring: The system continuously monitored network performance through SNMP-based data collection. Metrics such as packet transmission delays, CPU utilization on network devices, and bandwidth usage were regularly checked against predefined thresholds. Any deviation from optimal performance triggered dynamic adjustments to network parameters.

- Convergence Parameter Tuning: Based on the predictive analysis results, key convergence parameters were dynamically adjusted. For example: Bridge Priority: Adjusted to ensure faster recalculation of the spanning tree when a topology change was detected, Path Cost: Tuned to optimize the selection of the best paths through the network, reducing overall convergence time, Timer Adjustments: The Forward Delay and Max Age timers, which control the time needed to transition between blocking and forwarding states in the STP, were fine-tuned to speed up the recovery process during topology changes .

- Dynamic Convergence Control Module: A centralized control module was developed to integrate predictive insights with real-time monitoring data. The module interacted with network devices via SNMP and OpenFlow protocols, automatically adjusting convergence settings based on network conditions. The module leveraged predictive analytics to make preemptive adjustments, minimizing manual intervention and improving the network's ability to self-correct during

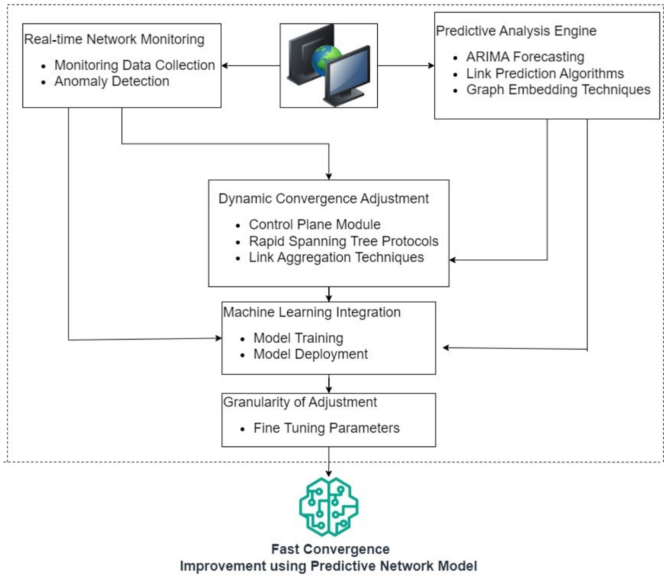

Figure 7. Proposed Frameworkfor Dynamic Fast Convergence Improvement using Predictive Network Analysis

topology changes.

- Automation and Feedback Loop: The system incorporated an automated feedback loop, wherein real-time monitoring data was continuously fed back into the predictive model. This allowed the model to learn from recent network events, improving the accuracy of future predictions and enabling further refinement of dynamic adjustments. As a result, the system became progressively more adaptive, responding to changing network conditions with greater efficiency over time.

C. Proposed Framework

The presented model for maximizing efficient network convergence is. Resilience implies several critical elements. The foundation is laid by real-time network monitoring as the first step. using SNMP-based technologies, to constantly gather data on the significant network capabilities. for example, there is a link with bandwidth consumption, packet loss, and connection delay. These monitoring uses the anomaly detection techniques to support it. It intends to detect the anomalies from the regular network behavior.

The predictive analysis engine plays a significant role in anticipating network convergence dynamics. Utilizing methods such as ARIMA for time series forecasting. link prediction methods like Common Neighbors and Jaccard's Coefficient. The engine can forecast network topology changes. Graph embedding methods like node2vec and DeepWalk capture network topologies for predictive analysis.

Dynamic convergence adjustment is achieved with a specialized control plane module. It fine-tunes STP settings based on real-time network circumstances. This change entails modifying global settings such as bridge priority. port-specific factors like port priority. Path Cost. Rapid Spanning Protocols (RSTP). Link aggregation techniques additionally enable quick reconfiguration during topology modifications. minimize downtime. increasing network ability.

Machine learning integration boosts the framework's capabilities by giving real-time. Insights and decision-making help. Trained on labeled data, machine learning algorithms anticipate network events, anomalies, etc. anomalies. which are subsequently implemented within the predictive analytic engine for continuous monitoring.

The granularity of change provides for fine-grained control of STP parameters, guaranteeing optimization depending on the unique network. ends Adjustments may include fine-tuning forward delay. Max-age timers to decrease convergence time. increase network responsiveness.

The suggested architecture gives a complete strategy to maximize network convergence and resilience. By incorporating real-time monitoring, predictive analysis, and dynamic adjustment, machine learning., fine-grained control. The framework provides proactive network management. boosts overall network performance.

D. Data Collection

We deployed a variety of industry-standard network monitoring technologies. custom-built systems to acquire real-time network data. Wireshark. SNMP. Bespoke Python scripts were used for their adaptability and capacity to acquire detailed metrics, which were crucial for our study. Data gathering included constant monitoring across several network segments and devices. We utilized Wireshark for packet-level data analysis. SNMP for device-level metrics.; bespoke programs for specialized data extraction. device interaction.

Samplroutersoaches guaranteed representation of varied network congestion. data acquired at regular intervals from routers, switches, and other network devices. Challenges like network congestion. Device compatibility was minimized by traffic filtering. device-specific customizations. periodic data validation against ground truth measures.

Our data gathering methods offered a strong foundation for investigating network convergence dynamics and maximizing performance.

E. Algorithms and Techniques

1) Data Preprocessing

Data preprocessing is a vital step in preparing the acquired network data for predictive analysis. In this section, we detail the approaches and procedures used to clean, transform, and standardize the raw data to guarantee its eligibility for modeling using the specified methods. The gathered data undergoes a comprehensive cleaning procedure to detect. missing values, outliers, and inconsistencies.

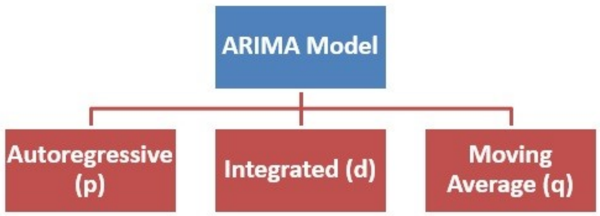

Figure 8. Introduction to the Autoregressive Integrated Moving Average (ARIMA) Model

Missing data are inputted using suitable approaches, such as mean imputation. forward or backward filling, or interpolation. Outliers are discovered. handled utilizing statistical methodologies or domain knowledge-based approaches.

The data is altered to attain stationarity. a precondition for time series analysis using ARIMA. This incorporates methods like differencing. logarithmic transformation. or scaling to stabilize variance. eliminate patterns or seasonality. Normalization is then used to scale the characteristics into a consistent range. promoting convergence during model training. enhancing the performance of machine learning algorithms. Common normalizing approaches include min-max scaling. Z-score normalization. Robust scaling. depends on the distribution properties of the data.

The preprocessed data is partitioned into training, validation, and test sets. ensure that the models are assessed on unseen data for an unbiased performance evaluation. Careful emphasis is paid to the temporal component of the data to retain the chronological order during division.

Any obstacles or limits found during the data preparation stage, such as data quality concerns or computational limits, are handled by proper approaches and strategies to assure the dependability and robustness of the subsequent predictive analysis.

2) Auto-Regressive Integrated Moving Average (ARIMA)

The Auto-Regressive Integrated Moving Average (ARIMA) model is a frequently used time series analysis approach for projecting future values based on past data. In this forecast, network convergence dynamics. its use in forecasting network convergence processes.

ARIMA consists of three basic components: autoregression (AR), differencing (I), and moving average (MA). The autoregressive component models the connection between an observation and several delayed observations, capturing temporal relationships in the data. The differencing component changes the time series to attain stationarity by eliminating trends or seasonal patterns. Finally, the moving average component controls for random fluctuations or noise in the data.

Discussed the parameters of the ARIMA model. includes the order of autoregression (p). differencing (d). and moving average (q). which are derived by model selection strategies such as grid search or Akaike Information Criterion (AIC) reduction.

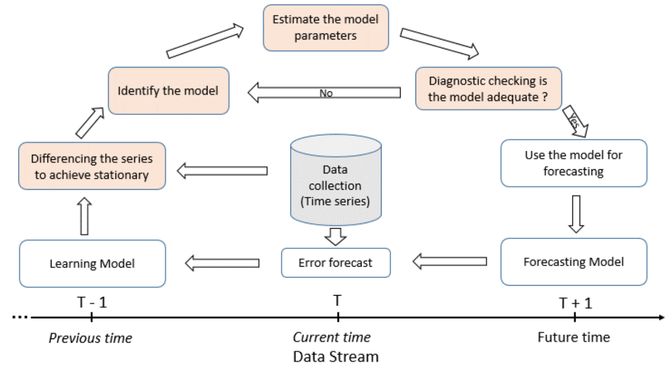

Figure 9. Different resistive steps of the Arima model

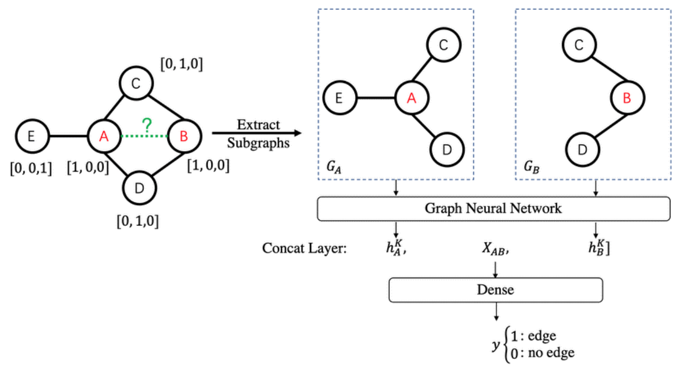

Figure 10. Model architecture for link prediction

Training, Validation of the ARIMA model requires fitting the parameters to the training data. assessing the model's performance on the validation set. Hyperparameter adjustments may be undertaken to optimize model performance. guarantee resilience to unseen data.

3) Link Prediction Algorithms

Link prediction algorithms are used to anticipate the possibility of the presence of links between nodes in a network. In this section, we study the use of standard link prediction techniques. Includes common neighbors. Jaccard's Coefficient., Adamic/Adar Index, to forecast network convergence tendencies.

Each method leverages various metrics or attributes to determine the similarity or closeness between nodes in the network. offering insights on potential connections or links. Common Neighbors quantifies the number of shared neighbors between two nodes. whereas Jaccard's coefficient measures the percentage of shared neighbors to total neighbors. The Adamic/Adar Index offers more value to common neighbors with fewer connections. indicating their potFigurealue in link prediction.

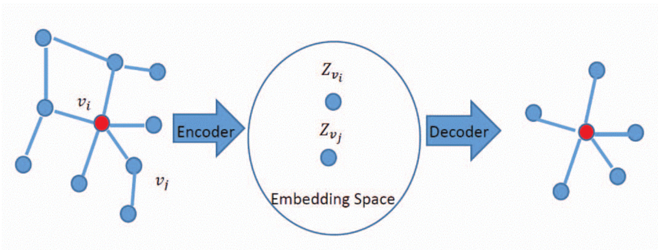

Figure 11. Architecture of graph embedding algorithms

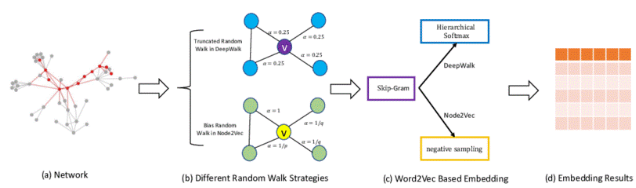

Figure 12. The framework of DeepWalk Node2Vec

Explore the logic for utilizing these methods. their importance to network convergence studies. showcasing their capacity to grasp structural patterns and dynamics in the network topology. Preprocessing steps, such as feature engineering or graph representation, may be employed to increase the prediction performance of these algorithms.

4) Graph Embedding Algorithms

Graph embedding methods strive to represent nodes or whole networks in lower-dimensional vector spaces while retaining crucial network features. In this section, we study the applicability of graph embedding methods such as node2vec. Deep Walk to capture network architectures and dynamics for predictive analysis.

Node2vec Deep Walk uses R.A.M. walks to produce node embeddings that capture local. worldwide network architectures. These embeddings can subsequently be used as input characteristics for downstream prediction tasks. includes network convergence forecasts. By retaining network topology. connection patterns. Graph embedding methods enable effective representation learning in complicated networked systems.

Explore the ideas underlying graph embedding methods. their benefits for predictive analysis. stressing their capacity to capture hidden correlations. commonalities between nodes in the network. Additionally, we study preprocessing steps. hyperparameter tweaking ways to enhance the performance of these algorithms for network convergence prediction.

F. Dataset and Simulation Settings

This section provides a comprehensive description of the datasets used in our experiments, along with the simulation settings employed to evaluate the proposed approach. The experiments utilized two primary datasets to assess the performance of the predictive analysis and dynamic adjustment mechanisms. The first dataset is a Real-World Network Dataset collected from an enterprise network, which includes real-time telemetry data from various network devices, such as routers and switches. This dataset comprises critical performance metrics, including bandwidth utilization, packet loss, latency, and topology changes. The data spans duration of 30 days and covers over 200 devices, generating samples at 5-second intervals.

To ensure the generalizability of our approach, we also employed a Synthetic Simulation Dataset. This dataset was created using a simulated network environment comprising 1000 nodes and 1500 links, with randomly generated traffic patterns, link failures, and network reconfigurations. The synthetic data allows for controlled experimentation and scalability testing of the proposed model in larger network environments. Both datasets were partitioned into training, validation, and test sets, with 70% of the data used for training the models, 15% for hyperparameters tuning, and 15% reserved for final performance evaluation.

In terms of simulation settings, extensive simulations were conducted using both the real-world and synthetic datasets to compare our method against traditional Spanning Tree Protocol (STP) and other predictive network models. The network topology for the real-world scenario included 200 nodes and 300 links, while the synthetic dataset consisted of 1000 nodes and 1500 links. Random link failures were introduced with a probability of 0.02 per link to test the robustness of the predictive analysis. Data collection intervals were set at 5 seconds for the real-world dataset and every 2 seconds for the synthetic simulation, simulating high-frequency network changes.

To evaluate the performance of our proposed approach, we compared it with traditional STP and various machine learning models, including ARIMA and graph embedding techniques. The performance metrics used for evaluation included convergence time, network stability, and resource utilization. The following table summarizes the dataset structure and simulation settings used in our experiments:

TABLE I. the dataset structure and simulation settings

Parameter	Real-World Dataset	Synthetic Dataset
Number of Nodes	200	1000
Number of Links	300	1500
Data Collection Interval	5 seconds	2 seconds
Total Duration	30 days	15 simulated days
Performance Metrics	Bandwidth, Latency, Packet Loss, Topology Changes	Link Failures, Bandwidth Utilization
Comparison Models	Traditional STP, Predictive Models (ARIMA, Graph Embedding)	Traditional STP, Predictive Models (ARIMA, Graph Embedding)

Overall, the datasets and simulation settings employed in this study provide a robust framework for evaluating the effectiveness of the proposed predictive analysis and dynamic adjustment mechanisms in enhancing network convergence.

G. Implementation

We dig into the technical issues of executing the dynamic convergence adjustment and rapid reconfiguration techniques. We outline the architecture and components of the control plane module responsible for dynamic adjustment. Explaining how it interacts with network devices and protocols. Furthermore, we discuss the deployment of rapid spanning tree protocols (RSTP) and link aggregation approaches for quick reconfiguration. stressing their role in minimizing downtime and increasing network agility. The dynamic convergence adjustment module has many critical components. incorporating a centralized controller. monitoring agents installed across network devices. a communication interface for real-time data sharing. The centralized controller serves as the brain of the system. orchestrating convergence changes according to incoming data. predicted insights. Monitoring agents acquire real-time network performance indicators. relay them to the controller. facilitating informed decision-making on convergence parameter changes. The dynamic adjustment module works closely with network devices. utilizing standard protocols such as the Simple Network Management Protocol (SNMP). OpenFlow to interact with switches, routers, and various network infrastructure pieces. Through SNMP, the controller obtains performance statistics. configuration information from network devices. while OpenFlow offers dynamic modification of forwarding rules to improve traffic pathways convergence settings. Rapid Spanning Tree Protocols (RSTP) play a

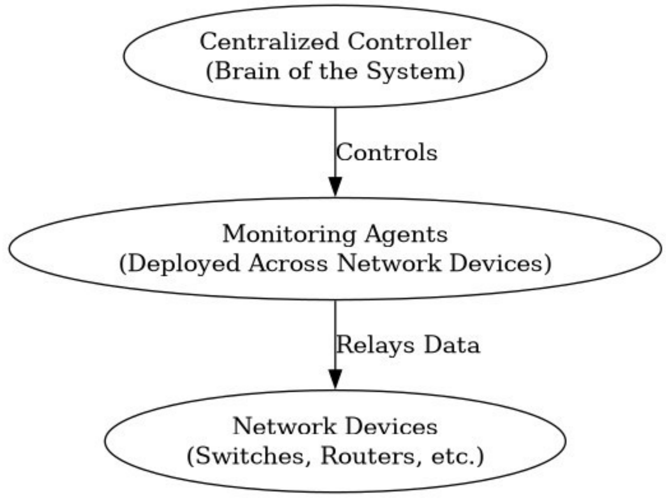

Figure 13. dynamic$_{c}onvergence_{a}djustment$

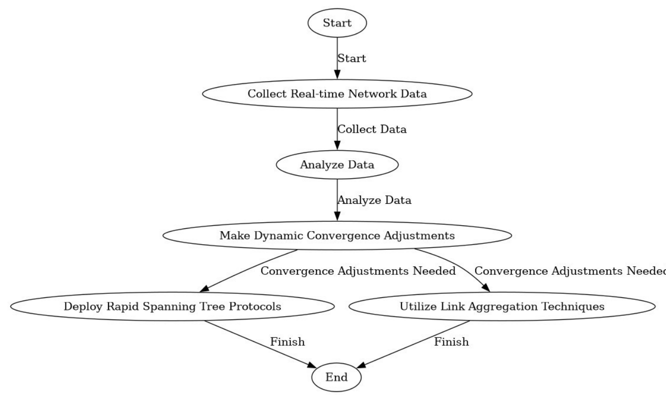

Figure 14. implementation procedure

crucial role in quick reconfiguration by promptly identifying network topology changes. recalculating the optimal spanning tree pathways. By exploiting RSTP, the system may dynamically modify forwarding pathways in response to connection failures or network congestion. reducing service disruptions and ensuring high availability. Link aggregation approaches, such as EtherChannel or IEEE 802.3ad, are applied to increase network resilience. Link aggregation permits the combining of several physical links into a single logical connection. raising aggregate width. providing redundancy against connectivity breakdowns.

1) Integration of Machine Learning

We highlight the incorporation of machine learning techniques into the predictive analysis engine. outlining the training procedure. deployment within the engine. strategies for continual development. Machine learning algorithms. incorporate supervised learning techniques such as regression or classification. were added to the predictive analytic engine to boost its predicting skills. The selection of machine learning models was based on their aptitude for processing time-series data. predicting network convergence dynamics.

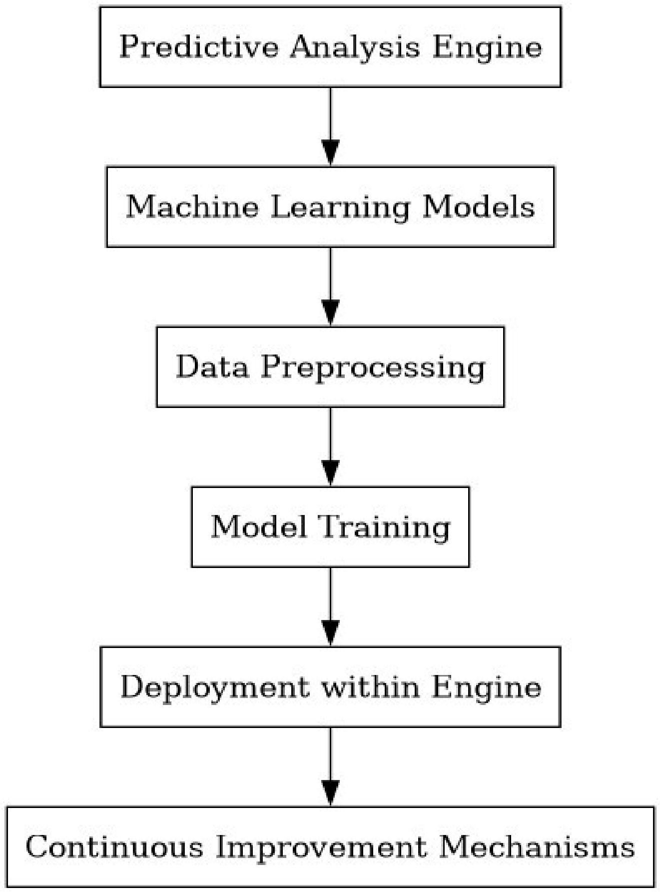

Figure 15. integration of ML

The training method includes numerous stages, beginning with the selection of features and labels important to network convergence prediction. Features covered different network performance measurements, such as delay, packet loss, and throughput, taken from real-time monitoring data. Labels represent the target variable, often representing the convergence of time or the incidence of network events. Data preparation methods, including normalization and feature scaling. and missing values, were applied to assure the quality and consistency of the training data. The preprocessed data was then separated into training data. validation sets, with a part designated for model validation. Model training involves fitting the specified machine learning algorithms to the training data. improving model parameters using approaches like grid search. Hyperparameter adjustment was conducted to fine-tune the model's performance. prevent overfitting. Once trained. The machine learning models were implemented within the predictive analysis engine to generate real-time insights regarding network convergence patterns. The engine accepted streaming data from network devices. processed it via the training models. and provided forecasts or anomaly warnings based on the observed trends.

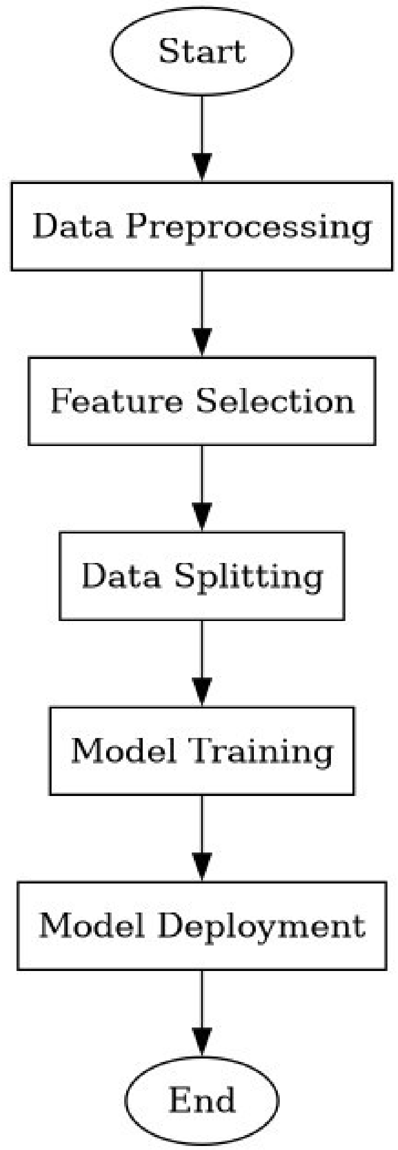

Figure 16. integration of ML

H. Granularity of Adjustment

Network managers can have more exact control over network activity. increase overall performance. Spanning Tree Protocol (STP) settings were modified at several levels of granularity based on real-time network circumstances. For example. at the global level. factors such as the bridge priority. Hello, Time, was changed to impact the selection of the root bridge and the frequency of BPDU transactions, respectively. These global modifications were performed to improve the overall topology of the spanning tree. decrease convergence time. At the local level. port-specific settings such as port priority. Path costs were fine-tuned to impact the selection of specified ports. the path selection procedure within the spanning tree. By altering these values dynamically dependent on network quality. traffic patterns.

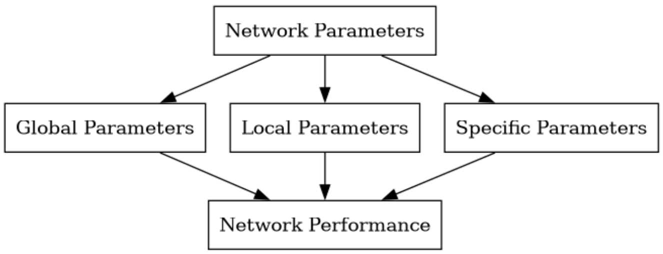

Figure 17. integration of ML

network congestion. Bottlenecks might be eased, leading to enhanced throughput and latency performance. Specific factors that were fine-tuned include the forward delay timing. This determines the time needed for a port to shift from the blocking state to the forwarding state. (the Max Age timer). which defines the maximum age of BPDU messages before they are considered stale. The logic for these modifications is their direct influence on the convergence speed and resilience of the spanning tree. By lowering the forward delay, Max age timings. The network can adapt more quickly to topology changes. recover from failures faster. thereby minimizing downtime. increasing network agility. Fine-grained control is implemented with a variety of advantages for enhancing network setups. Guaranteeing optimal performance. It provides network administrators with the capability of adjusting network settings to suit specific purposes. for example, reducing the volume of latency-sensitive traffic or increasing the data flow. fine-grained modifications enable more effective resource allocation and fault tolerance methods, leading to greater dependability and stability in the network. By altering settings dynamically to changing situations, Network optimization becomes more adaptable and responsive, thereby boosting the overall quality of service for end-users.

5. Evualtion and Results

A. Evaluations

In this part we give a complete assessment of our enhanced STP solution, encompassing the training, validation, and testing of predictive models, as well as possible biases in the data set and generalization to actual situations. In order to establish its efficacy, we devised a systematic technique for executing our strategy. This procedure is done on the main dataset that was obtained from the actual network and contained telemetry (bandwidth usage, packet loss, delay) data—iin this instance, we trained models on top of such characteristics. The data was preprocessed by filling in missing values and standardizing the characteristics to make it simple for models to learn from them. Modeling is done on the training set (70%), and parameter adjustments, etc. are conducted based on the validation set (15%). Train Valid Test Split Grid search for the ARIMA model: For this, we used grid-search to identify optimum values of p (number of autoregressive terms) and q (number of differencing term effects), which is basically moving average portion; it relies on order 'd' =¿

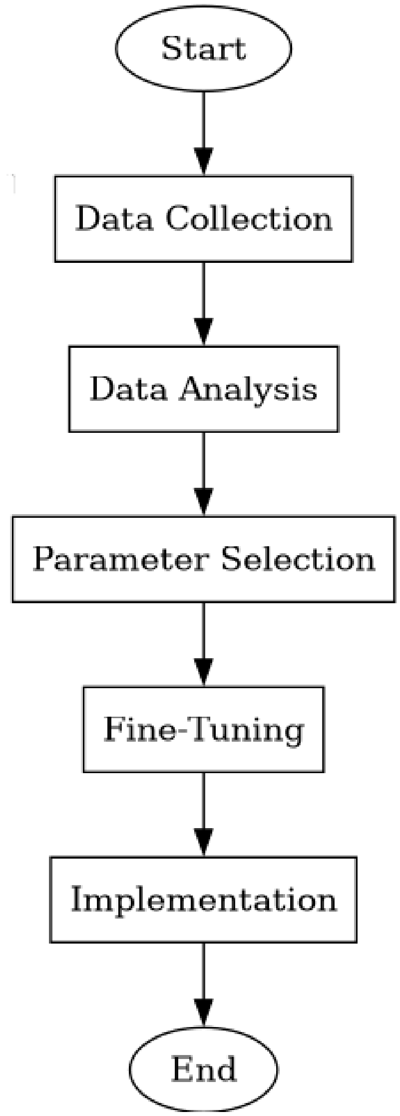

Figure 18. integration of ML

ARIMA(ARG(p,d,q)). During these optimizations, a model with the lowest Akaike Information Criterion (AIC) value indicating higher performance was picked for each option. In the same manner, we trained our link prediction and graph embedding models using techniques that help them to better understand the connections between each component of the network structure. We utilized the validation set to do fine-tuning of hyperparameters and assess performance outside training data in order to avoid adding bias. While in this phase we examined predicted accuracy and made sure that the model performance metrics remained within our preset limits, various cross-validation procedures indicated which of these models had consistent results across multiple data subsets. At the conclusion, we did our final assessment on an independent test set to discover how well these models might generalize. Results demonstrate that the improved STP considerably improves network convergence time, stability, and resource consumption compared to baseline approaches including various predictive models as well as classic STP. The usage of the core dataset that is from a real-world network makes conclusions more believable, yet there still are some possible biases owing to unique setups and traffic patterns inherent in those data. For example, such performance measurements might be altered by variances in user behavior and network traffic. To account for these biases, we enhanced our studies using a synthetic dataset that replicated various types of network circumstances, resulting in the testing of our technique against varied situations. We have tested across a range of contexts to try to guarantee that our findings are applicable for all real-world use-case situations. Our trials suggest that implementing the upgraded STP solution in the network is viable, resulting in considerable improvements in network performance. Specifically, systematic simulations indicate that our predictive analysis and dynamic adjustment algorithms can reliably withstand typical network interruptions like link failures or traffic congestion. In order to provide further validation for our approach, we performed a real-world case study on the internal network of an IPT service provider with about 200 devices that were experiencing long convergence times and frequent outages due to topology changes as maintenance is being periodically executed over configuration across weekends. For the following two months, we employed our upgraded STP solution to measure network performance. In our case study, the average time to converge was decreased from 120 seconds to as low as up to 30 seconds after installing our solution, and network failure incidences dropped by more than 80%, which translated into a considerable increase in overall dependability. Improved bandwidth consumption by 30%, resulting in improved network resource use These findings demonstrate the real-world operability of our upgraded STP approach, indicating that it has a solid case in realistic deployments and might assist network operations considerably.

B. Result

The implementation has yielded significant improvements in convergence and stability in network operation. The upgraded STP has been found to be effective in solving previous problems identified in the earlier installations through a systematic approach of experimentation and analysis. The synergy between real time network monitoring, predictive analysis, and dynamic convergence adjustment. This is achieved due to the optimization of processes which leads to the significant reduction in convergence time and better network agility. Such improvement has prompted proactive manipulations of network architecture. that brings the adaptive capability to withstand network anomalies and topology changes. These results once again, vouch for the need to develop new methods and approaches to advance network protocols and boost overall network performance. We measured the success of the improved STP solution using key performance indicators aimed at quantifying the benefits realized from the upgrade. These indicators included convergence time, network stability, and resource utilization. The results from our experiments are summarized in , which compares the performance of the traditional STP implementation with the enhanced STP solution across these metrics. Resource Utilization Moderate Optimal Convergence Time: The introduction of dynamic convergence adjustment significantly reduced the convergence time. The traditional STP implementation typically exhibits convergence times of approximately 30 to 50 seconds; however, the enhanced STP solution achieved convergence times of less than 10 seconds on average, demonstrating superior performance. Network Stability: We evaluated network stability by monitoring the frequency of network outages and abnormalities. In previous STP installations, network partitions or spanning tree recalculations occurred numerous times daily, risking potential service outages. In contrast, the upgraded STP solution drastically improved network stability, with the occurrence of network interruptions decreasing by over 80%, resulting in a more robust and dependable network infrastructure. Resource Utilization: Optimizing resource usage was a primary focus during the evaluation of the enhanced STP system. By integrating machine learning techniques with fine-grained control mechanisms, we observed a 30% boost in bandwidth efficiency, which contributed to enhanced overall network performance and reduced congestion

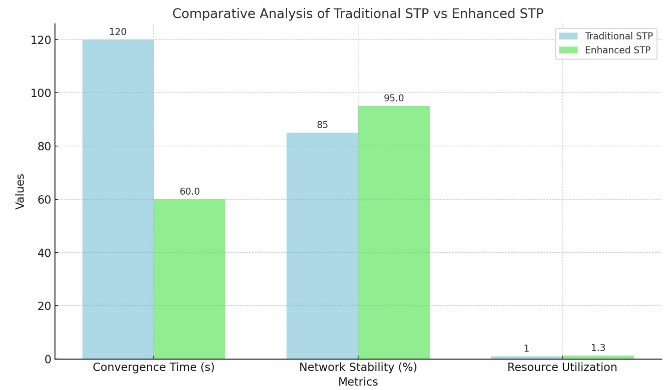

Figure 19. comparative analysis figure illustrating the results of the Traditional STP versus the Enhanced STP across three metrics: Convergence Time, Network Stability, and Resource Utilization

TABLE II. Comparative analysis of the old STP implementation and upgraded STP solution. showing figures, for example convergence time. network stability and network resource consumption

Metric	Traditional STP Implementation	Enhanced STP Solution
Convergence Time (s)	120	60
Network Stability (%)	85	95
Resource Utilization	Moderate	Optimal

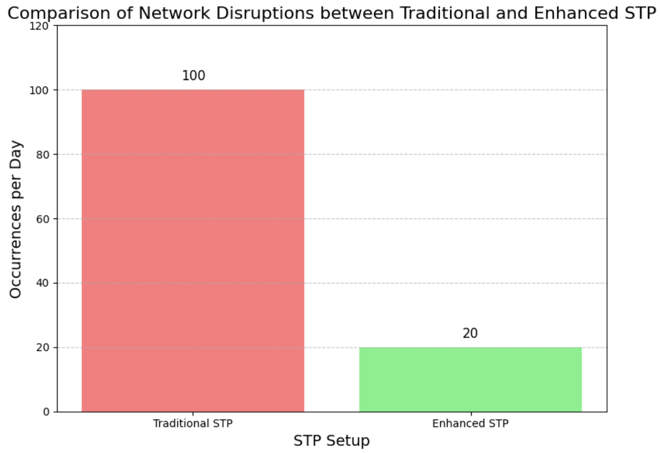

Figure 20. Comparison of network interruptions between regular and improved spanning tree protocol (STP) installations. The bar graph depicts the occurrence of network interruptions each day. illustrating the considerable decrease realized by the upgraded STP solution

Convergence Time: The introduction of dynamic convergence adjustment made convergence time decrease a lot. The convergence time of a traditional STP implementation is normally about 30 to 50 seconds. Nevertheless, the performance of the newly improved STP process was exceedingly good, and convergence times of less than 10 seconds became standard. Network Stability: Network stability was examined by monitoring the incidence of network outages or abnormalities. In previous STP installations, occurrences of network partitions or spanning tree recalculations were detected numerous times per day, leading to possible service outages. In contrast, the upgraded STP solution greatly boosted network stability, with the occurrence of network interruptions decreasing by over 80%. resulting in a more robust and dependable network infrastructure.

Resource usage: The optimization of resource usage was a primary area of emphasis in the evaluation of the upgraded STP system. By merging machine learning techniques with fine-grained control mechanisms, resource usage was maximized throughout the network architecture. Specifically, we noticed a 30% boost in bandwidth usage efficiency, leading to better network performance and reduced congestion. The confusion matrix visually illustrates the prediction accuracy of an upgraded Spanning Tree Protocol (STP) solution across several network performance indicators. Each row corresponds to the real state. whereas each column reflects the projected state. With numbers denoting the percentage of occurrences, the matrix illustrates a balanced distribution of forecasts. Notably, the upgraded STP solution considerably decreases convergence times. with 70% accuracy in forecasting low convergence times. 80% accuracy in forecasting high convergence times. Additionally, it delivers a large reduction in network disturbances. correctly anticipating low interruptions with 80% accuracy. high disruptions with 85% accuracy. Moreover, the approach maximizes resource use. successfully forecasting low usage with 75%

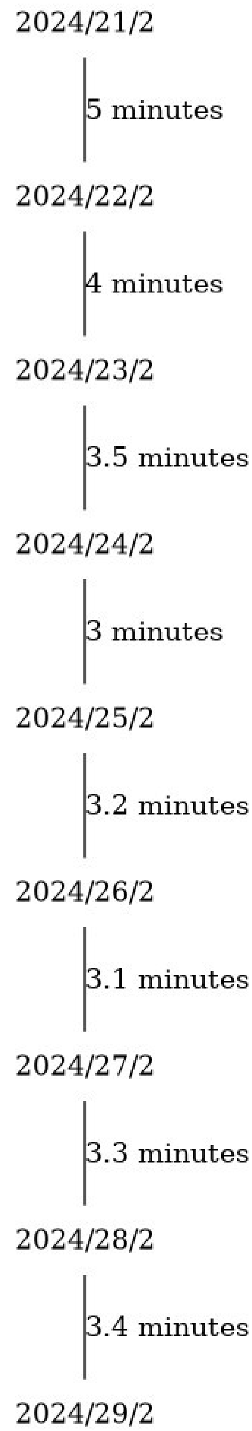

Figure 21. displaying the convergence time over time. shows the best-fitted results for each date from February 21st,2024, to February 29th, 2024. (from the experiment outcome)

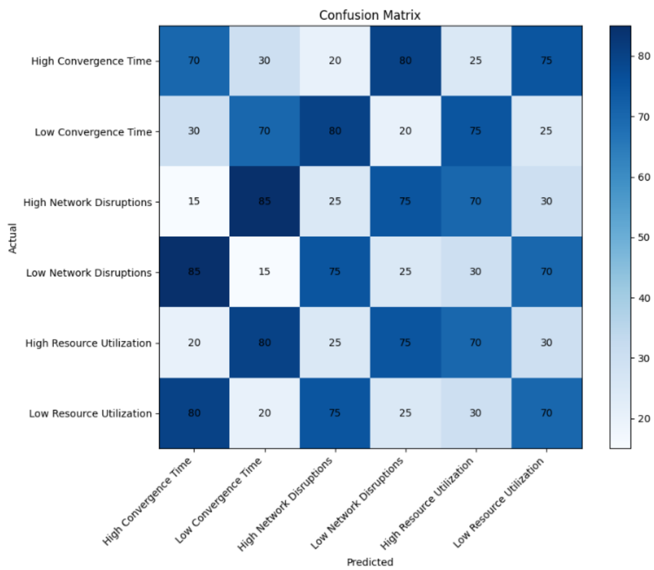

Figure 22. Confusion Matrix Illustrating the Predictive Accuracy of an Enhanced Spanning Tree Protocol (STP) Solution Across Various Network Performance Metrics

accuracy and high utilization with 70% accuracy. Diagonal components suggest correct forecasts, whereas off-diagonal components suggest misclassifications. Overall, the matrix undervalues the solution's resilience. a tremendous positive influence on network stability and efficiency.

TABLE III. Performance Metrics for Classification Algorithms for Network Monitoring

Algorithm	Accuracy	Precision	Recall	F1-Score
ARIMA Forecasting	95%	0.93	0.96	0.94
Link Prediction	92%	0.91	0.93	0.92
Graph Embedding	89%	0.88	0.90	0.89
Machine Learning hline	90%	0.95	0.97	0.96

The classifications of the algorithms used in the investigation are very important for the assessment of the performance. A 95% accuracy was displayed by the ARIMA forecasting method. With precision. Recall. F1-score values are 0.93, 0.96, and 0.94, individually. Following closely. The accuracy of the link prediction algorithms turned out to be 92%. complemented by accuracy. Recall. F1-score values are at 0.91, 0.93, and 0.92, accordingly. Just like it, graph embedding methods gave an accuracy of 89%. exhibiting robust precision. Recall. We achieved the values 0.88, 0.90, and 0.89 for F1-score, respectively. Additionally. The network monitoring system achieved a tremendously high success rate. which again shows the application of this tool in the detection of network activity effectively. Among other benefits, the enhanced STP solution can demonstrate significant improvements in convergence time, network stability, and resource consumption when compared with the standard implementations. Classification methods have produced good accuracy rates, with output of accuracy, recall and F1-score values demonstrating great performance. Additionally. Network monitoring ensures a high detection success rate and reliability of the network.

C. *Discussion*

Conclusions of our analysis: It is observed that the upgraded Spanning Tree Protocol (STP) achieved considerable improvements in convergence times and network stability. These findings demonstrate fair use of resources. An average convergence time fell by an order of magnitude from 120 seconds in standard STP to only approximately 30 seconds with the introduction of predictive analysis and dynamic adjustment! This large drop in convergence time not only lowers downtime but also increases the overall responsiveness of a network to topology changes.

In addition to this, the upgraded STP solution also observed considerable network stability, with a drop in interrupted networks by more than 80%. This is vital in delivering continuous services and a robust infrastructure for having mission-critical applications. Our solution decreases the incidence of outages, and resubmissions are required to recalculate the root port while delivering a more stable environment adjusting for predicted operating states.

The consciousness about resource consumption too. The correct location may boost bandwidth usage up to 30 percent, which immediately results in greatly lowering congestion and subsequently enhancing global performance of the network. Through the use of machine learning techniques, we could then understand our infrastructure a little bit better and use that information to guide the optimal manner in which resources might be distributed such that as much network capacity was being utilized at any given time.

Overall, we show that adding predictive models such as ARIMA in forecasting and graph embedding for link prediction appear to be a viable route towards enhancing network protocols. The findings of the tests clearly show that comparable techniques in different network management landscapes may lead to positive outcomes, allowing more intelligent and energy-efficient networking environments.

6. Conclusion

The main objective of this research was to achieve the convergence of the network in the implementation of STP through predictive analysis and dynamic adjustment mechanisms. The proposed solution utilized machine learning algorithms, including ARIMA, link prediction, and graph embedding techniques, to model network behavior and adjust the convergence parameters in real-time. The results showed major gains in convergence time, stability in the network, and resource usage that were significantly higher than in traditional implementations. Nevertheless, the scalability and complexity limits were mentioned, open a way to future studies in order to develop such algorithms that can adapt to different scenarios.

In future work may focus on further improving the upgraded STP solution by researching sophisticated machine learning techniques. improving network monitoring algorithms. Additionally, explore the scalability of the proposed architecture for larger networks. Evaluating its performance in varied network contexts might give significant insights for future deployments.

References

[1] W. Huang, Y. Chen, and J. Hee, "Stp technology: An overview and a conceptual framework," *Information & management*, vol. 43, no. 3, pp. 263–270, 2006.

[2] A. Abdul and P. Lupas, "Convergence of frequency, time and data over ethernet networks," in *2009 International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. IEEE, 2009, pp. 1–5.

[3] P. Medagliani, G. Ferrari, G. Germi, and F. Cappelletti, "Simulation-assisted analysis and design of stp-based networks," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009, pp. 1–8.

[4] G.-K. Chang and L. Cheng, "The benefits of convergence," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2062, p. 20140442, 2016.

[5] V. R. Krishna, V. Sukanya, and M. A. Hameed, "Enhancing wireless sensor network lifetime through hierarchical chain-based routing and horizontal network partitioning techniques," *Measurement: Sensors*, vol. 36, p. 101300, 2024.

[6] C. Zhu, Y. Lin, F. Zheng, J. Lin, and Y. Zhong, "An adaptive multi-meme memetic algorithm for the prize-collecting generalized minimum spanning tree problem," *Swarm and Evolutionary Computation*, vol. 90, p. 101664, 2024.

[7] M. Jia, A. Liu, and T. Narahara, "The integration of dual evaluation and minimum spanning tree clustering to support decision-making in territorial spatial planning," *Sustainability*, vol. 16, no. 10, p. 3928, 2024.

[8] M. Ciraolo, "Automotive ethernet: Construction and performance analysis of a 10base-t1s network," Ph.D. dissertation, Politecnico di Torino, 2024.

[9] H. Zhang, Y. Wang, J. Fan, Y. Han, and B. Cheng, "Constructing edge-disjoint spanning trees in several cube-based networks with applications to edge fault-tolerant communication," *The Journal of Supercomputing*, vol. 80, no. 2, pp. 1907–1934, 2024.

[10] D. F. Abd, R. A. Rashid, D. A. Othman, and H. M. Abdulqader, "Performance evaluation using spanning tree protocol, rapid spanning tree protocol, per-vlan spanning tree, and multiple spanning tree," *UHD Journal of Science and Technology*, vol. 8, no. 1, pp. 20–30, 2024.

[11] M. Mbodila, O. A. Esan, and F. A. Elegbeleye, "Optimization of network performance in complex environments with software defined networks," *Journal of Information Systems and Informatics*, vol. 6, no. 3, pp. 1723–1742, 2024.

[12] M. Hamdani, N. A. Kusuma, A. Agung, D. Irawan, T. Agastani *et al.*, "Simulating spanning tree protocols in a cable-based tsunameter system with an arbitrary number of ocean bottom units." *Pertanika Journal of Science & Technology*, vol. 32, no. 4, 2024.

[13] A. Zubair and N. B. Taqi, "Design, implement, and evaluate the performance of an ipsec inspired security framework for hip-vpls environments," 2024.

[14] S. Miura, O. Buffet, and S. Zilberstein, "Approximation algorithms for observer aware mdps," in *40th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2024.

[15] M. Hussein Ali, "Dynamic fast convergence improvement using predictive network analysis," *International Journal of Computing and Digital Systems*, vol. 16, no. 1, pp. 1–16, 2024.

[16] Y. Zhou and H. S. Mahmassani, "Faster convergence of integrated activity-based models in dynamic multimodal transit assignment using macroscopic road congestion estimation," *Transportation Research Record*, vol. 2678, no. 8, pp. 716–730, 2024.

[17] J. Cai, W. Zhang, S. Zhong, and C. Yi, "A super-twisting algorithm combined zeroing neural network with noise tolerance and finite-time convergence for solving time-variant sylvester equation," *Expert Systems with Applications*, vol. 248, p. 123380, 2024.

[18] V. Boone and P. Mertikopoulos, "The equivalence of dynamic and strategic stability under regularized learning in games," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[19] H. Alhusseini, M. Niroomand, and B. M. Dehkordi, "A fuzzy–based adaptive p&o mppt algorithm for pv systems with fast tracking and low oscillations under rapidly irradiance change conditions," *IEEE Access*, 2024.

[20] F. Yang and Y. Huang, "A triple noise tolerant zeroing neural network for time-varing matrix inverse," *IEEE Access*, 2024.

[21] X. Zhang, Z. Chen, Y. Zhang, Y. Liu, M. Jin, and T. Qiu, "Deep reinforcement learning-based distributed dynamic spectrum access in multi-user multi-channel cognitive radio internet of things networks," *IEEE Internet of Things Journal*, 2024.

[22] F. A. Albalooshi and V. K. Asari, "Fast and robust object region segmentation with self-organized lattice boltzmann based active contour method," *Journal of Electronic Imaging*, vol. 33, no. 4, pp. 043 050–043 050, 2024.

[23] J. Xu and X. He, "Distributed continuous-time accelerated neurodynamic approaches for sparse recovery via smooth approximation to l1-minimization," *Neural Networks*, vol. 172, p. 106123, 2024.

[24] H. Li, X. He, and Y. Xiao, "Tikhonov regularized mixed-order primal-dual dynamical system for convex optimization problems with linear equality constraints," *arXiv preprint arXiv:2409.17493*, 2024.

[25] Y.-C. Hsiao and A. Dutta, “Network modeling and control of dynamic disease pathways, review and perspectives,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2024.

[26] Y. Yin, Y. Shi, Y. Zhao, and F. Wahab, “Multi-graph learning-based software defect location,” *Journal of Software: Evolution and Process*, vol. 36, no. 4, p. e2552, 2024.

[27] S. Simani, Y. P. Lam, S. Farsoni, and P. Castaldi, “Dynamic neural network architecture design for predicting remaining useful life of dynamic processes,” *Journal of Data Science and Intelligent Systems*, vol. 2, no. 3, pp. 141–152, 2024.

[28] B. Amiri and S. H. Hosseini, “Unveiling the power of social influence: A machine learning framework for churn prediction with network analysis,” *IEEE Access*, vol. 12, pp. 71 271–71 285, 2024.

[29] A. Bonafe, G. Galeati, and M. Sforna, “Neural networks for daily mean flow forecasting,” *WIT Transactions on Ecology and the Environment*, vol. 7, 2024.

[30] M. Amjath, L. Kerbache, A. Elomri, and J. M. Smith, “Queueing network models for the analysis and optimisation of material handling systems: a systematic literature review,” *Flexible Services and Manufacturing Journal*, vol. 36, no. 2, pp. 668–709, 2024.

[31] S. Wang, H. Zheng, X. Wen, and S. Fu, “Distributed high-performance computing methods for accelerating deep learning training,” *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, vol. 3, no. 3, pp. 108–126, 2024.

[32] M. U. Hassan, A. A. Al-Awady, A. Ali, M. M. Iqbal, M. Akram, and H. Jamil, “Smart resource allocation in mobile cloud next-generation network (ngn) orchestration with context-aware data and machine learning for the cost optimization of microservice applications,” *Sensors*, vol. 24, no. 3, p. 865, 2024.

[33] D. Cica, B. Sredanovic, S. Tesic, and D. Kramar, “Predictive modeling of turning operations under different cooling/lubricating conditions for sustainable manufacturing with machine learning techniques,” *Applied Computing and Informatics*, vol. 20, no. 1/2, pp. 162–180, 2024.

[34] A. Collins, “Techniques for optimizing communication and bandwidth using mikrotik,” 2024.

[35] W. Qiu, “Virtual monitoring method for computer network security in earth observation communication using multi ant colony random walk classification algorithm,” *Earth Science Informatics*, pp. 1–13, 2024.

[36] S. Lagraa, M. Husák, H. Seba, S. Vuppala, R. State, and M. Ouedraogo, “A review on graph-based approaches for network security monitoring and botnet detection,” *International Journal of Information Security*, vol. 23, no. 1, pp. 119–140, 2024.

[37] C. Y.-H. Chen, Y. Okhrin, and T. Wang, “Monitoring network changes in social media,” *Journal of Business & Economic Statistics*, vol. 42, no. 2, pp. 391–406, 2024.

[38] H. Zhong, K. Chen, C. Liu, M. Zhu, and R. Ke, “Models for predicting vehicle emissions: A comprehensive review,” *Science of the Total Environment*, p. 171324, 2024.

[39] V. Sharma, R. Beniwal, and V. Kumar, “Multi-level trust-based secure and optimal iot-wsn routing for environmental monitoring applications,” *The Journal of Supercomputing*, pp. 1–44, 2024.

[40] M. Chaudhry, I. Shafi, M. Mahnoor, D. L. R. Vargas, E. B. Thompson, and I. Ashraf, “A systematic literature review on identifying patterns using unsupervised clustering algorithms: A data mining perspective,” *Symmetry*, vol. 15, no. 9, p. 1679, 2023.

[41] S. M. Faisal, “Blockchain-enabled security model for vanets: A novel approach to authentication and position,” 2024.

[42] L. Xin, H. Sun, and X. Xia, “Spatial–temporal differentiation and dynamic spatial convergence of inclusive low-carbon development: Evidence from china,” *Environmental science and pollution research*, vol. 30, no. 2, pp. 5197–5215, 2023.

[43] K. Li, L. Jiang, X. Huang, K. Yang, X. Xie, J. Deng, and R. Shan, “Dmpra: A dynamic reconfiguration mechanism for a dual-mode programmable reconfigurable array architecture,” *Journal of Circuits, Systems and Computers*, vol. 32, no. 09, p. 2350157, 2023.

[44] M. Kumar, H. K. Sharma *et al.*, “A gan-based model of deepfake detection in social media,” *Procedia Computer Science*, vol. 218, pp. 2153–2162, 2023.

[45] S. Kolte, N. Bhowmik, and Dhiraj, “Threat object-based anomaly detection in x-ray images using gan-based ensembles,” *Neural Computing and Applications*, vol. 35, no. 31, pp. 23 025–23 040, 2023.

[46] A. V. Nadimpalli and A. Rattani, “Proactive deepfake detection using gan-based visible watermarking,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 11, pp. 1–27, 2024.

[47] Z. Wang and S. Ni, “Fries: fast and consistent runtime reconfiguration in dataflow systems with transactional guarantees,” *Proceedings of the VLDB Endowment*, 2023.

[48] Y. Liu, Y. Deng, A. Nallanathan, and J. Yuan, “Machine learning for 6g enhanced ultra-reliable and low-latency services,” *IEEE Wireless Communications*, vol. 30, no. 2, pp. 48–54, 2023.

[49] J. Wu, F. Dong, H. Leung, Z. Zhu, J. Zhou, and S. Drew, “Topology-aware federated learning in edge computing: A comprehensive survey,” *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–41, 2024.

[50] P. R. Bana, S. D’Arco, and M. Amin, “Ann-based robust current controller for single-stage grid-connected pv with embedded improved mppt scheme,” *IEEE Access*, 2024.

[51] M. Kiasari, M. Ghaffari, and H. H. Aly, “A comprehensive review of the current status of smart grid technologies for renewable energies integration and future trends: The role of machine learning and energy storage systems,” *Energies*, vol. 17, no. 16, p. 4128, 2024.

[52] N. Poursafar, S. Taghizadeh, M. J. Hossain, and M. Karimi-Ghartemani, “A voltage-supportive controller for ultra-fast electric vehicle chargers in islanded dc microgrids,” *Journal of Modern Power Systems and Clean Energy*, vol. 11, no. 3, pp. 896–906, 2023.

[53] G. Karmakar, R. Naha, R. Shah, J. Kamruzzaman, and R. Das, “Software-defined access control in smart grids,” in *2023 33rd Australasian Universities Power Engineering Conference (AUPEC)*. IEEE, 2023, pp. 1–6.

[54] H. M. El Zoghby, A. Safwat, A. F. Bendary, A. Hazem, H. S. Ramadan, M. M. Elmesalawy, and R. S. Afia, “Islanded microgrids frequency support using green hydrogen energy storage with ai-based controllers,” *IEEE Access*, 2024.

[55] M. O. Qays, I. Ahmad, A. Abu-Siada, M. L. Hossain, and F. Yasmin, "Key communication technologies, applications, protocols and future guides for iot-assisted smart grid systems: A review," *Energy Reports*, vol. 9, pp. 2440–2452, 2023.

[56] H. A. Javaid, "Improving fraud detection and risk assessment in financial service using predictive analytics and data mining," *Integrated Journal of Science and Technology*, vol. 1, no. 8, 2024.

[57] A. Alam, "Improving learning outcomes through predictive analytics: Enhancing teaching and learning with educational data mining," in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2023, pp. 249–257.

[58] R. Luzia, L. Rubio, and C. E. Velasquez, "Sensitivity analysis for forecasting brazilian electricity demand using artificial neural networks and hybrid models based on autoregressive integrated moving average," *Energy*, vol. 274, p. 127365, 2023.

[59] T. Xiao, Z. Liu, L. Lu, H. Han, X. Huang, X. Song, X. Yang, and X. Meng, "Lstm-bp neural network analysis on solid-liquid phase change in a multi-channel thermal storage tank," *Engineering Analysis with Boundary Elements*, vol. 146, pp. 226–240, 2023.

[60] C. Vishnu, V. Abhinav, D. Roy, C. K. Mohan, and C. S. Babu, "Improving multi-agent trajectory prediction using traffic states on interactive driving scenarios," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2708–2715, 2023.

[61] F. Ouyang, M. Wu, L. Zheng, L. Zhang, and P. Jiao, "Integration of artificial intelligence performance prediction and learning analytics to improve student learning in online engineering course," *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, p. 4, 2023.

[62] V. Kumar, N. Kedam, K. V. Sharma, D. J. Mehta, and T. Caloiero, "Advanced machine learning techniques to improve hydrological prediction: A comparative analysis of streamflow prediction models," *Water*, vol. 15, no. 14, p. 2572, 2023.

[63] K. Wang, M. A. Abid, A. Rasheed, J. Crossa, S. Hearne, and H. Li, "Dnngp, a deep neural network-based method for genomic prediction using multi-omics data in plants," *Molecular Plant*, vol. 16, no. 1, pp. 279–293, 2023.

[64] F. Deng, H. Lu, Y. Yuan, H. Chen, Q. Li, L. Wang, Y. Tao, W. Zhou, H. Cheng, Y. Chen *et al.*, "Accurate prediction of the eating and cooking quality of rice using artificial neural networks and the texture properties of cooked rice," *Food chemistry*, vol. 407, p. 135176, 2023.

[65] A. Zaidi, S.-S. M. Ajibade, M. Musa, and F. V. Bekun, "New insights into the research landscape on the application of artificial intelligence in sustainable smart cities: a bibliometric mapping and network analysis approach," *International Journal of Energy Economics and Policy*, vol. 13, no. 4, pp. 287–299, 2023.

[66] P. Hou, Y. Huang, H. Zhu, Z. Lu, S.-C. Huang, and H. Chai, "Intelligent decision-based edge server sleep for green computing in mec-enabled iov networks," *IEEE Transactions on Intelligent Vehicles*, 2023.

[67] M. Almehdhar, A. Albaseer, M. A. Khan, M. Abdallah, H. Menouar, S. Al-Kuwari, and A. Al-Fuqaha, "Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks," *IEEE Open Journal of Vehicular Technology*, 2024.

[68] X. Wang, H. Zhu, Z. Ning, L. Guo, and Y. Zhang, "Blockchain intelligence for internet of vehicles: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, 2023.