Survey on Random Forest Algorithm for Detecting Failures in Computer Networks

Nikhilesh Raut; Kartik Rokade; Deepa Barethiya
Department of Master in Computer Application, GHRCEM,
Nagpur, India

Abstract:
Computer networks are indispensable in today's digital infrastructure, and hence reliability and fault tolerance are imperative for them. Failure in computer networks can cause widespread disruptions, with attendant financial and operational losses. To improve fault detection, machine learning methods like the Random Forest algorithm have found extensive application due to their resilience and accuracy.

This study explores the use of the Random Forest algorithm for failure detection in computer networks, examining its efficacy in network anomaly classification, failure prediction, and minimizing false alarms. Through the application of ensemble learning, Random Forest improves the accuracy of prediction and offers insightful information regarding network health. Comparative analysis with other machine learning algorithms, including Support Vector Machines and Neural Networks, accentuates the strengths and weaknesses of Random Forest in network failure detection. Research directions for the future involve combining Random Forest with deep learning models and enhancing real-time fault diagnosis systems.

Keywords:
Random Forest Algorithm, Machine Learning,
Computer Network Failures, Network Anomaly Detection, Predictive Analytics.

I.Introduction

The growing dependence on computer networks across different sectors has heightened the need for effective failure detection systems. Failures in networks can result in service loss, security threats, and cost expenditures. Conventional failure detection techniques are based on rule-based systems that need to be manually configured extensively and regularly updated [8]. These traditional approaches fail to handle dynamic network environments, resulting in high false-positive rates and inefficiency in large-scale networks. With the development of machine learning and artificial intelligence, new methods like ensemble learning have been developed to improve network monitoring and failure detection.

Random Forest, a supervised learning algorithm, has attracted widespread attention due to its accuracy and robustness in classification. Random Forest is an ensemble learning technique that constructs a group of decision trees and combines their predictions to achieve higher accuracy and prevent overfitting [1]. It helps achieve superior generalization to multiple network environments and thus remains an effective means to identify failures and anomalies in real-time [14]. When compared with both conventional statistical and one-machine models, Random Forest yields improved accuracy and stability with greater confidence to effectively identify failure [8]. This study investigates the effectiveness of the Random Forest algorithm in failure detection in computer networks through its applications in network anomaly detection, fault classification, and predictive maintenance. The survey also analyses the strengths and weaknesses of employing Random Forest in failure detection systems and possible enhancements through hybrid machine learning approaches [3]. By

DOI: https://doi.org/10.5281/zenodo.15571823

examining existing research, methodologies, and technological advancements, this paper aims to provide insights into how Random Forest can enhance network reliability and security. Future research directions are also discussed, focusing on integrating deep learning models and cloud-based architectures to further improve network failure detection mechanisms.

II.Literature Review

There are some studies which emphasized the importance of machine learning for network failure detection. Breiman (2001) presented the Random Forest algorithm and its ability to eliminate overfitting and enhance the accuracy of classification. In another study, investigated Random Forest for the purpose of detecting network traffic anomalies [8]. The conclusion drawn from the survey was that Random Forest considerably lowers false positives as opposed to rule-based methods. In addition, they highlighted the model's potential to learn novel threats through retraining on fresh datasets.

Xu et al. (2022) did an empirical comparison of fault classification in network systems using Random Forest. They made a comparison between the performance of the model with other machine learning algorithms like Support Vector Machines (SVM) and k-Nearest Neighbours (k-NN) and found Random Forest to perform better in accuracy and computation speed. In the same vein, [14] compared network failure detection models and concluded that Random Forest performed better than linear classifiers in detecting intricate network problems. Their research further observed that hyperparameter optimization, including varying the number of trees and feature selection, was vital in improving detection performance. Scientists have also explored the combination of Random Forest with deep learning techniques. Chen et al. (2021) suggested a hybrid model integrating Random Forest with Long Short-Term Memory (LSTM) networks to enhance predictive maintenance in massive networks [3]. The hybrid model improved failure prediction accuracy by utilizing LSTM's sequential data processing and Random Forest's capacity to process structured data. Furthermore, investigated the influence of feature engineering on Random Forest performance [6]. They discovered that deep learning-based feature optimization substantially decreased false negatives and enhanced real-time failure detection.

In spite of its merits, the Random Forest method has some drawbacks, emphasized that a large number of decision trees demand huge computational resources, which may be a constraint for real-time application [12]. In addition, contended that although Random Forest is extremely efficient in classification problems, it might not always be interpretable, since single decision paths in the ensemble are not easily interpretable [13]. Future research should emphasize combining explainable AI methods to improve model transparency. Another research area concerns hybrid machine learning methods. Researchers have attempted to integrate Random Forest with reinforcement learning and deep neural networks to enhance its adaptability and decision-making ability [8]. The methodologies have been promising in reducing false positives without compromising detection rates. Additionally, proposed that implementing Random Forest-based failure detection models on cloud platforms can improve scalability and efficiency to be applicable to large-scale network infrastructures [3].

In general, the literature confirms the efficacy of Random Forest in detecting network failures. Its ensemble learning strategy offers a high degree of accuracy and reliability, and thus it is a necessary component for contemporary network monitoring systems. Yet, overcoming computational issues, enhancing interpretability, and investigating hybrid approaches will be paramount for further development in this area.

DOI: https://doi.org/10.5281/zenodo.15571823

iii. Methodology

The research method used in this survey is a systematic analysis of the literature on the use of the Random Forest algorithm for network failure detection. This entails examination of relevant academic work, experimental research, and industry documents for assessing the performance of Random Forest in failure detection in computer networks. The survey is carried out in a systematic way with steps including data gathering, preprocessing, implementation of the model, assessment, and comparative evaluation with other machine learning approaches.

The initial step is data collection from various sources, such as publicly available network datasets, academic papers, and actual network logs. Datasets like the KDD Cup 99, UNSW-NB15, and CICIDS2017 are popularly utilized in network anomaly detection studies [8]. The datasets include labelled instances of normal and abnormal network behaviours, thus being applicable for training and testing the Random Forest model.

After data has been gathered, preprocessing is carried out to remove noise and organize the dataset into a form ready for analysis. This involves the management of missing values, the normalization of numeric features, categorical variable encoding, and feature selection. Good feature engineering is critical in enhancing the performance of the model, as established that optimized feature selection improves the accuracy of detection [4].

The second stage entails the application of the Random Forest algorithm in Python and packages like Scikit-learn and TensorFlow.

The model is trained on a sample of the dataset, where a collection of decision trees is built utilizing bootstrap sampling. The prediction is made through majority voting among the trees to achieve robustness and generalizability [14].

Model assessment is conducted utilizing standard metrics including accuracy, precision, recall, and F1-score. Cross-validation techniques like k-fold validation are used to evaluate the reliability of machine models in varying conditions. Random Forest performance is subsequently compared with other machine learning models, for example, SVM, k-NN, and deep learning models like CNNs and LSTMs, shows that Random Forest outperforms consistently in both accuracy and computational costs [3] [8].

Lastly, the model's challenges and limitations are examined, such as computational resource needs and interpretability concerns. Hybrid models and cloud-based implementations for potential improvements are explored, and implications for future survey and real-world applications are offered.

iv. Key Technologies

Random Forest for network failure detection employs various cutting-edge technologies, such as machine learning frameworks, big data processing tools, cloud computing platforms, and edge computing solutions.

Machine Learning Frameworks: Scikit-learn, TensorFlow, and PyTorch offer necessary tools for implementing and optimizing Random Forest models with reliable failure detection [8].

Big Data Processing Tools: Apache Spark and Hadoop process large-scale network traffic data effectively, facilitating high-speed feature extraction and real-time anomaly detection [8].

Cloud Computing Platforms: AWS, Google Cloud, and Microsoft Azure facilitate scalable deployment of failure detection models, enhancing accessibility and efficiency in large network environments [3].

Edge Computing Solutions: Edge computing platforms, like AWS Greengrass and Azure IoT Edge, support real-time detection of failures at the network edge, cutting down latency and bandwidth usage [13].

DOI: https://doi.org/10.5281/zenodo.15571823

v. Future Scope

The application of Random Forest in network failure detection will continue to grow with the push from emerging trends in artificial intelligence (AI), cloud computing, and cybersecurity. With the increasing complexity of network infrastructures from the rollout of 5G, IoT, and software-defined networking (SDN), there will be greater demand for scalable, efficient, and real-time failure detection technologies [14].

Integration with Deep Learning: Although RF is very effective, combining it with deep learning models like CNNs and RNNs can improve predictive accuracy even further. Hybrid RF-Deep Learning models have been found to be effective in lowering false positive rates and enhancing anomaly classification in large networks [11]. Future studies should aim to optimize computational efficiency while using deep learning architectures [7].

Cloud-Based and Edge Computing Deployments: Implementing RF on cloud and edge computing setups can offer scalable failure detection mechanisms for real-time monitoring. Edge AI-powered RF models can make local decisions on network anomalies, minimizing latency and enhancing response times [10]. Cloud implementations of RF can manage high-scale traffic examination, which makes them appropriate for worldwide network infrastructures [2].

Self-Healing Networks: Future self-healing networks will use AI and ML algorithms that automate the network self-healing process without any manual intervention. RF models together with reinforcement learning (RL) mechanisms can facilitate adaptive network management such that systems learn from past failures and adjust configuration proactively to avoid outages [14].

Advanced Cybersecurity Applications: Cyber-attacks are constantly changing, requiring sophisticated intrusion detection and prevention systems. RF-based anomaly detection models combined with blockchain and zero-trust security frameworks can improve cybersecurity resilience, minimizing the effects of network attacks [8].

Future Research Directions:

To further improve RF's use in network failure detection, survey needs to address:

Decreasing computational complexity to enhance real-time performance [5].

Automated feature selection methods for model accuracy optimization [15].

Federated learning methods to train RF models over distributed data with privacy preservation [4].

The ongoing development of RF-based network failure detection will be key to determining the future of intelligent, secure, and self-management network systems.

vi. Conclusion

This study ventured into the Random Forest algorithm as a powerful solution to network failure detection, where it emphasized strengths, weaknesses, and areas for further survey [1]. RF's ensemble learning makes its classification accurate, stable, and scalable, with it being the go-to in real-time failure detection [13]. Still, computational limitations are present, prompting survey in the area of hybrid models and cloud deployment. Future development must emphasize the combination of RF with deep learning, the optimization of its computational efficiency, and the use of autonomous failure management in 5G, IoT, and SDN-based systems [14].

vii. References

1. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

2. Chandra, R., & Bose, M. (2022). Automated troubleshooting using AI. IEEE Network, 36(4), 99-107.

DOI: https://doi.org/10.5281/zenodo.15571823

3. Chen, X., Zhang, Y., & Li, K. (2021). AI-based network monitoring for IoT environments. Future Internet, 13(6), 156.

4. Cheng, L., & Yao, K. (2023). Federated learning in network failure detection. Computers & Security, 91, 102013.

5. Gupta, R., & Sharma, P. (2022). Predictive maintenance in cloud networks using machine learning. Journal of Network and Computer Applications, 203, 103345.

6. Jones, T., & Brown, S. (2022). Network self-healing techniques with AI. Journal of Artificial Intelligence Research, 76, 245-267.

7. Kim, Y., & Park, H. (2019). An overview of ML in cybersecurity. ACM Computing Surveys, 51(4), 79.

8. Liu, C., & Wang, H. (2021). Machine learning for network self-healing. IEEE Transactions on Network Science, 18(3), 678-690.

9. Luo, M., & Liu, Z. (2018). A hybrid approach to network security using machine learning. Security and Privacy, 1(3), e45.

10. Patel, R., & Singh, T. (2021). SDN-based failure detection using RF. Journal of Networks, 36(2), 178-195.

11. Roberts, M., & Thomas, N. (2020). IoT-based network monitoring with ML. Sensors, 20(14), 3987.

12. Smith, J., & Kumar, A. (2019). Anomaly detection in software-defined networking. Computer Communications, 144, 12-25.

13. Tan, H., Liu, J., & Hu, Y. (2020). Machine learning-based network failure detection. IEEE Transactions on Network and Service Management, 17(3), 1123-1134.

14. Zhang, J., & Wei, H. (2021). AI-driven failure prediction in 5G networks. IEEE Transactions on Mobile Computing, 20(5), 1450-1462.

15. Zhao, M., & Li, B. (2023). Future perspectives on AI-driven networking. IEEE Communications, 29(1), 56-74.

DOI: https://doi.org/10.5281/zenodo.15571823