

MACHINE LEARNING FOR QOE PREDICTION AND ANOMALY DETECTION IN SELF-ORGANIZING MOBILE NETWORKING SYSTEMS

Chetana V. Murudkar* and Richard D. Gitlin

Innovation in Wireless Information Networking Lab (iWINLAB)
Department of Electrical Engineering, University of South Florida,
Tampa, Florida 33620, USA.

*Sprint Corp., USA.

ABSTRACT

Existing mobile networking systems lack the level of intelligence, scalability, and autonomous adaptability required to optimally enable next-generation networks like 5G and beyond, which are expected to be Self-Organizing Networks (SONs). It is anticipated that machine learning (ML) will be instrumental in designing future “x”G SON networks with their demanding Quality of Experience (QoE) requirements. This paper evaluates a methodology that uses supervised machine learning to predict the QoE level of the end user experiences and uses this information to detect anomalous behavior of dysfunctional network nodes (eNodeBs/base stations) in self-organizing mobile networks. An end-to-end network scenario is created using the network simulator ns-3, where end users interact with a remote host that is accessed over the Internet to run the most commonly used applications like file downloads and uploads and the resulting output is used as a dataset to implement ML algorithms for QoE prediction and eNodeB (eNB) anomaly detection. Three ML algorithms were implemented and compared to study their effectiveness and the scalability of the methodology. In the test network, an accuracy score greater than 99% is achieved using the ML algorithms. As suggested by the ns-3 simulation the use of ML for QoE prediction will help network operators understand end-user needs and identify network elements that are failing and need attention and recovery.

KEYWORDS

Machine learning, ns-3, QoE, SON

1. INTRODUCTION

There is little doubt that machine learning (ML) will be a foundation technology for next-generation wireless networks, such as 5G and beyond. Future wireless networks will be highly integrative and will create a paradigm shift that includes very high carrier frequencies with massive bandwidths, extremely high base station and device densities, dynamic “cell-less” networks, and unprecedented numbers of antennas (Massive MIMO), and these networks will have to possess ground-breaking levels of flexibility and intelligence [1]. This paper is directed towards demonstrating that the level of intelligence, complexity and autonomous adaptability required to build such networks can be achieved by implementing machine learning in combination with self-organizing networks (SON).

It is well-known that user experience is one of the most vital aspects of any industry or business domain. The occurrence of failures in a network element, such as a base station, may cause deterioration of this network element’s functions and/or service quality and will, in severe cases, lead to the complete unavailability of the respective network element [2]. Consequently, anomaly detection is crucial to minimize the effects of such failures on QoE of the network users. Another

important aspect to consider in the dawn of 5G is energy efficiency or green communications. Energy-efficient network planning strategies include networks designed to meet peak-hour traffic such that energy can be saved by partially switching off base stations when they have no active users or simply very low traffic [1]. This makes anomaly detection even more critical and machine learning can play an indispensable role in achieving high accuracy in detecting and validating dysfunctional network elements and distinguishing from an energy saving mode.

The authors in [3] use neighbor cell measurements and handover statistics to detect anomalies and outages that is based on the number of incoming handovers (inHO) measured on a per cell basis by neighboring cells. This approach monitors situations where the number of inHO becomes zero as a potential symptom of cell outage. The authors in [4] use a statistical-based anomaly detection scheme in 3G networks to find deviations between the collected traffic data and measured distribution. In [5] the *k-nearest neighbor* algorithm is used to detect and locate cell outages with key performance information that uses RSRP (reference signal received power) and SINR (signal to interference plus noise ratio) measurements collected during normal operations and radio link failures.

In [6], the authors use a hidden Markov model (HMM) to determine if a base station is healthy, degraded, crippled or catatonic. The measurements used are serving cell's reference signal received power (RSRP), reference signal received quality (RSRQ), and best neighbor cell's RSRP and RSRQ. In [7] minimization of drive tests (MDT) reports are used to gather data from a fault-free operating scenario to profile the behavior of the network. This approach exploits multidimensional scaling (MDS) techniques to reduce the complexity of data processing while retaining pertinent information to develop training models to reliably apply anomaly detection techniques. The performance of *k-nearest neighbor* and *local-outlier-factor* based anomaly detection algorithms was compared and it was found that a global anomaly detection model using *k-nearest neighbor* performed better than the *local-outlier-factor* based anomaly detector which adopts a local approach for classifying abnormal measurement.

The authors in [8] study the degradation produced by cell outages in the neighboring cells and propose three methods. One of the methods analyzes the degradation produced by the cell outage in the neighboring cells based on KPI correlation using historical records for cell outages. The other two methods proposed are online methods where the first method is a correlation-based approach. This method calculates the correlation between the observed signal and a reference signal. The other method used is delta detection where a threshold is determined as a function of the Key Performance Indicators (KPIs) under normal circumstances. A sample measured under the cell outage is compared to this threshold in order to determine if a KPI degradation occurred.

While all of the above research approaches for anomaly detection have used different KPI's and measurements such as handover statistics, RSRP, RSRQ, number of connection drops, and number of connection failures, they lack the knowledge of the quality of experience observed by end-users. Quality of Experience is of crucial importance to end-consumers, network operators and any stakeholders involved in the service provisioning chain and is a dominant metric to be considered as the wireless communications networks shift from conventional network-centric paradigms to more user-centric approaches [9].

The recently introduced methodology, QoE-driven anomaly detection in self-organizing mobile networks using machine learning [10], implemented machine learning to learn and predict the quality of end-user experience that is further used for anomaly detection in self-organizing mobile networks. The metric used to determine the quality of the end-user is Quality of Experience (QoE), which is the overall acceptability of an application or service as perceived by the end-user [11]. Unlike QoS, QoE incorporates user-centric network decision mechanisms and processes such that it takes into account not just the technical aspects regarding a service but also incorporates any kind

of human-related quality-affecting factors reflecting the impact that the technical factors have on the user's quality perception [12]. The proposed system model [10] used a network simulator, a parametric QoE model and an optimized version of *decision tree* machine learning algorithm to demonstrate and evaluate the approach. This paper is an extension of that work where two other machine learning algorithms, *support vector machine (SVM)* and *k-nearest neighbors (k-NN)*, are implemented for QoE prediction to study the effectiveness and scalability of the proposed system model. This study evaluates and compares the performance of all three ML algorithms and analyzes their impact on the system model. The output of the machine learning model is further used for detecting dysfunctional network nodes (eNBs).

The structure of this paper is as follows: Section 2 briefly describes the system model [10]. In Section 3, *SVM* and *k-NN* algorithms implemented to train the machine learning model for QoE prediction are explained. Section 4 presents the results and observations obtained by studying the impact of both of these ML algorithms on the system model and also compares the performance of *SVM*, *k-NN* and *decision tree* for the dataset generated using the network simulator *ns-3* [13]. The paper ends with the concluding remarks in Section 5.

2. SYSTEM MODEL

A machine learning algorithm is an algorithm that is able to learn from data and make predictions of new data instances [14]. The system model described in Figure 1 [10] uses the LTE-EPC Network Simulator model of *ns-3* [13] to create a network scenario in order to generate representative data. ¹The simulation represents an end-to-end network communication where users run File Transfer Protocol (FTP) applications by interacting with a remote host accessible over the internet. The data obtained from the simulation serves as the input dataset for the machine learning model where a parametric QoE model and ML algorithms are implemented to predict QoE scores of end users that are further used to identify dysfunctional eNodeBs. The parametric QoE model for FTP services to generate the QoE scores ranging from 0 to 5 for the training set of the machine learning model to be used is given by the mean opinion score (1)

$$MOS_{FTP} = \begin{cases} 1 & u < u^- \\ b_1 \cdot \log_{10}(b_2 \cdot u) & u^- \leq u < u^+ \\ 5 & u^+ \leq u \end{cases} \quad (1)$$

where u represents the data rate of the correctly received data and the values of b_1 and b_2 coefficients are obtained from the upper rate (u^+) and lower rate (u^-) expectations for the service [10], [12], [15], [16]. The model is trained using a machine learning algorithm and QoE scores for all the users are predicted. The eNodeBs (eNBs) connected to all the users with poor QoE scores are identified and the mode is determined for each of these eNodeBs to find the QoE score that occurs most often. If the mode of the QoE scores that are computed using (1) of all the users connected to an eNB is less than or equal to one, then the eNB is declared as dysfunctional i.e. if most of the users connected to an eNB have poor QoE scores, then such an eNB is declared to be dysfunctional.

¹ The LTE-EPC simulation model of the *ns-3* simulator provides the interconnection of multiple UEs to the internet, via a radio access network of multiple eNodeBs connected to a single serving gateway-packet data network gateway node [13].

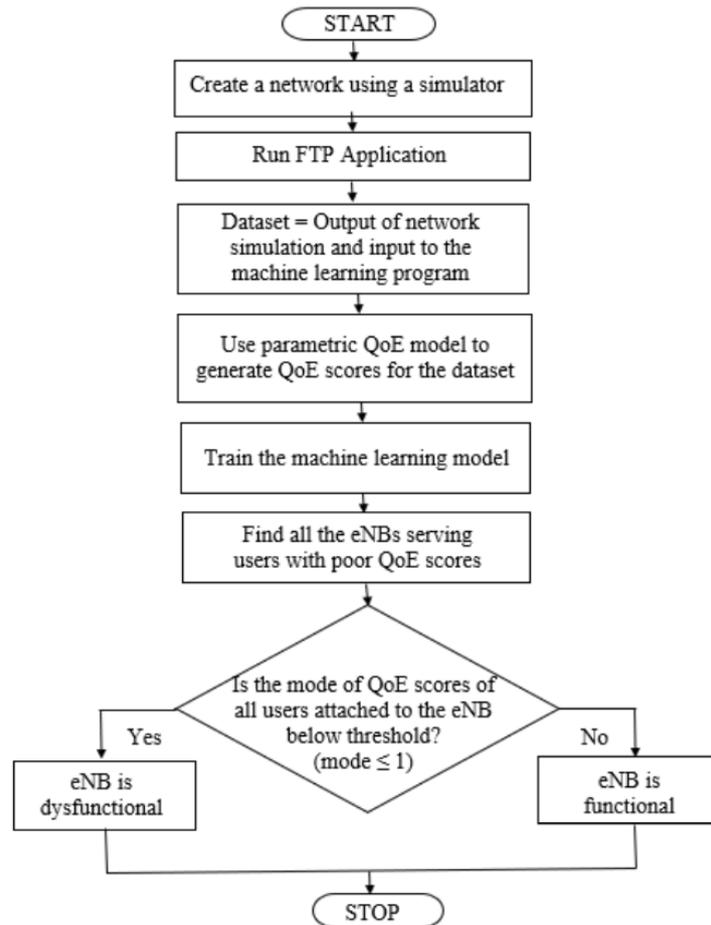


Figure 1. Flowchart describing the system model.

3. MACHINE LEARNING ALGORITHMS

A machine learning algorithm learns from experience, E , with respect to some tasks, T , and performance measure, P , to determine if the performance at tasks T , as measured by P , improves with experience E [14]. The type of task used that is applicable to the system model described in Figure 1 is regression. In a regression task, the ML algorithm is asked to output a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ [14]. A performance measure is a quantitative measure used to evaluate the abilities of an ML algorithm. The performance measure used here is the accuracy of the model in producing the correct output. The types of machine learning algorithms implemented in this research are supervised machine learning algorithms. These types of algorithms utilize a dataset containing features, where each example or data point is associated with a target [14]. In our recent work [10], the machine learning algorithm implemented was an optimized version of *decision tree*. This paper analyzes the performance of two other machine learning algorithms, *SVM* and *k-NN*, to study their impact on the system model.

3.1. Support Vector Machine Learning Algorithm

The first algorithm implemented to train the machine learning model is a *Support Vector Machine* algorithm. A *support vector machine (SVM)* constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks [17]. If sufficient separation is achieved by the hyperplane with the largest distance to the nearest training

samples of any class, the algorithm will generally be effective. The training samples that are the closest to the decision surface are called support vectors. The *SVM* algorithm finds the largest margin (i.e., “distance”) between the support vectors to obtain optimal decision regions. The type of *SVM* algorithm used in the proposed method is *SVM* regression which can be explained as follows [17], [18]: In *SVM* regression, the input vector \mathbf{x} is first mapped² onto an m -dimensional feature space using some fixed (nonlinear) mapping i.e. by using kernel functions, and then a linear model is constructed in this feature space to separate the training data points. The linear model in the feature space $f(\mathbf{x}, \omega)$ is given by

$$f(\mathbf{x}, \omega) = \sum_{j=1}^m \omega_j g_j(\mathbf{x}) + b \quad (2)$$

where $g_j(\mathbf{x}), j = 1, \dots, m$ denotes a set of nonlinear transformations and b is a bias term. A loss function [19] often used by an *SVM* to measure the quality of estimation is called the ε – insensitive loss function and is given below.

$$\mathcal{L}_\varepsilon(y, f(\mathbf{x}, \omega)) = \begin{cases} 0, & \text{if } |y - f(\mathbf{x}, \omega)| \leq \varepsilon \\ |y - f(\mathbf{x}, \omega)| - \varepsilon, & \text{otherwise} \end{cases} \quad (3)$$

The *SVM* performs linear regression in the high-dimension feature space using ε – insensitive loss and, at the same time, tries to reduce model complexity by minimizing $\|\omega\|^2$. This can be described by introducing (non-negative) slack variables ξ_i, ξ_i^* where $i = 1, \dots, n$, to measure the deviation of training samples outside the ε – insensitive zone. Thus, *SVM* regression is formulated as the minimization of the following function:

$$\min \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (4)$$

$$\text{subject to } \begin{cases} \mathbf{y}_i - f(\mathbf{x}_i, \omega) \leq \varepsilon + \xi_i^* \\ f(\mathbf{x}_i, \omega) - \mathbf{y}_i \leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0, i = 1, \dots, n \end{cases}$$

where C is a regularization parameter that determines the tradeoff between the model complexity and the degree to which deviations larger than ε are tolerated in optimization formulation, \mathbf{x}_i represents the input values, ω represents the weights, and \mathbf{y}_i represents the target values. This optimization problem can be transformed into the dual problem and its solution is given by

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) \quad (5)$$

subject to $0 \leq \alpha_i^* \leq C, 0 \leq \alpha_i \leq C$, where n is the number of support vectors, α_i is the dual variable, and the kernel function is given by

$$K(\mathbf{x}, \mathbf{x}_i) = \sum_{j=1}^m g_j(\mathbf{x}) g_j(\mathbf{x}_i) \quad (6)$$

SVM performance (estimation accuracy) depends on the optimized setting of meta-parameters C, ε and the kernel parameters.

² In SVM, the input space is transformed into a new feature space using kernel functions where it becomes easier to process the data such that it is linearly separable. Hard margin *SVM* works when data is completely linearly separable. But when we have errors (noise/outliers), we use soft margin *SVM* which uses slack variables (ξ).

3.2. *k*-Nearest Neighbor Machine Learning Algorithm

The second algorithm implemented to train the machine learning model is the *k*-nearest neighbor (*k*-NN) algorithm. The algorithm is explained as follows [17], [20]: The basic idea behind this algorithm is to base the estimation on a fixed number of observations *k* which are closest to the desired data point. A commonly used metric measure for distance is the Euclidean distance. Given $X \in \mathbb{R}^q$ and a set of samples $\{X_1, \dots, X_n\}$, for any fixed point $x \in \mathbb{R}^q$, it can be calculated how close each observation X_i is to x using the Euclidean distance $\|x\| = (x'x)^{\frac{1}{2}}$ where “'” denotes the vector transpose. This distance is given as

$$D_i = \|x - X_i\| = ((x - X_i)'(x - X_i))^{\frac{1}{2}} \quad (7)$$

The order statistics for the distances D_i are $0 \leq D_{(1)} \leq D_{(2)} \leq \dots \leq D_{(n)}$. The observations corresponding to these order statistics are the “nearest neighbors” of x . The observations ranked by the distances or “nearest neighbors”, are $\{X_{(1)}, X_{(2)}, X_{(3)}, \dots, X_{(n)}\}$. The k^{th} nearest neighbor of x is $X_{(k)}$. For a given k , let

$$R_x = \|X_{(k)} - x\| = D_{(k)} \quad (8)$$

denote the Euclidean distance between x and $X_{(k)}$. R_x is just the k^{th} order statistic on the distances D_i . In *k*-NN regression, the label³ assigned to a query point is computed based on the mean of the labels of its nearest neighbors. The weights used in the basic type of *k*-NN regression are uniform where each point in the local neighborhood contributes to the classification of a query point. In some cases, it can be beneficial to weigh points such that nearby points contribute more to the regression than points that are far away. The classic *k*-NN estimate is given as

$$\tilde{g}(x) = \frac{1}{k} \sum_{i=1}^k 1(\|x - X_i\| \leq R_x) y_i \quad (9)$$

This is the average value of y_i among the observations that are the k nearest neighbors of x . A smooth *k*-NN estimator is a weighted average of the k nearest neighbors and is given as

$$\tilde{g}(x) = \frac{\sum_{i=1}^k \omega\left(\frac{\|x - X_i\|}{R_x}\right) y_i}{\sum_{i=1}^k \omega\left(\frac{\|x - X_i\|}{R_x}\right)} \quad (10)$$

4. SIMULATION RESULTS AND OBSERVATIONS

The values of the primary parameters used to configure the network scenario created in the *ns-3* simulation are given in Table 1 [10].

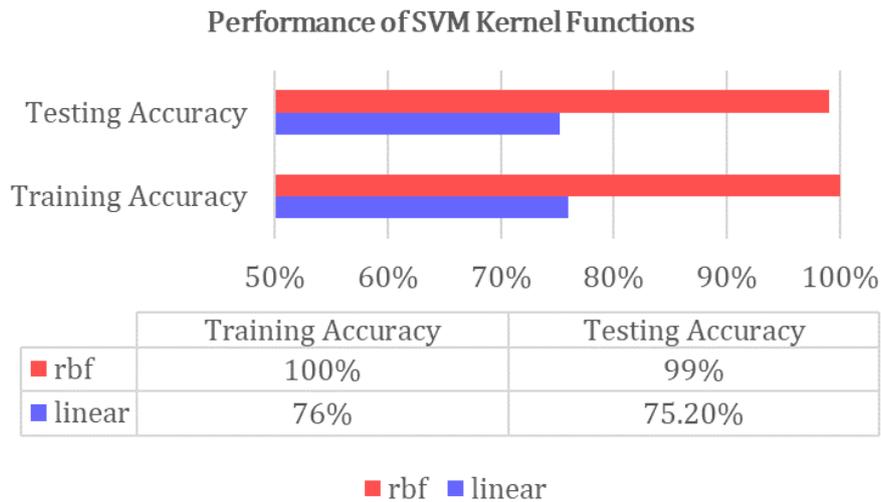
³ In supervised machine learning, the task of the ML model is to predict target values from labelled data. The input is referred to by terms such as independent variables or features. The output is referred to by terms such as dependent variables or target labels or target values.

Table 1. Network simulation configuration parameters

| Parameters | Value |
|-------------------------------------|--------|
| Number of users | 50 |
| Number of eNodeBs | 5 |
| eNodeB Bandwidth | 20 MHz |
| Transmit Power of functional eNB | 46 dBm |
| Transmit Power of dysfunctional eNB | 30 dBm |
| Application Type | FTP |

The output obtained from the *ns-3* simulation run is used as the input dataset for the machine learning model and the target values for the training set of the machine learning model are calculated using the parametric QoE model defined in (1). The *SVM* regression and *k-NN* regression algorithms are implemented using this dataset. The performance of *SVM*, *k-NN*, and *decision tree* [10] is evaluated to study their effectiveness and the scalability of the system model.

As previously mentioned in section III, *SVM* performance generally depends on the setting of meta-parameters C , ε and the kernel parameters. Two kernel functions linear and radial basis function (rbf) were used to test the performance. The training and testing accuracy for each of these kernel functions is given in Figure 2 that shows that the rbf function gives better accuracy for the dataset generated by the *ns-3* simulation.

Figure 2. Accuracy of the training and testing sets for *SVM* regression using linear and rbf kernel functions

Three additional computational parameters that affect the performance of *SVM* are C , epsilon and gamma. C is a regularization parameter that determines the tradeoff between the model complexity

and the degree to which deviations larger than epsilon are tolerated, epsilon specifies the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value, and gamma specifies how far the influence of a single training example reaches and is the inverse of the radius of influence of samples selected by the model as support vectors [17], [18]. Comparison done among these parameters to find the optimal value for each of these parameters for the dataset obtained in this work is illustrated in Figure 3. It is observed that the optimal values of these parameters for the dataset obtained from the *ns-3* simulation are $C = 5$, $\gamma = 0.001$, and $\epsilon = 0.01$.

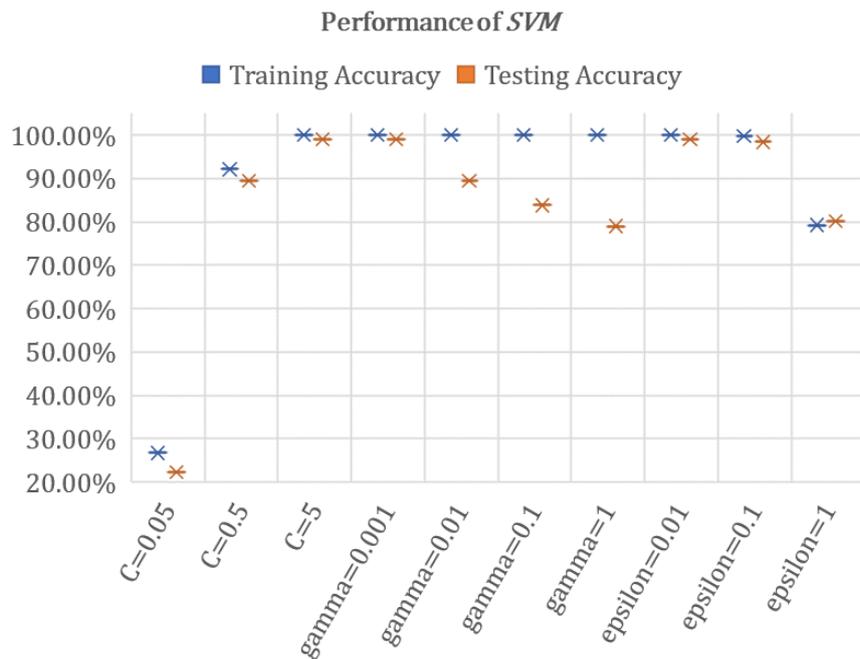


Figure 3. Accuracy scores for varying values of C, gamma and epsilon in SVM regression

The training and testing accuracies for *k-NN* regression for varying values of *k* are shown in Figure 4. It is observed that the most optimal value of *k* is 4 for the given dataset.

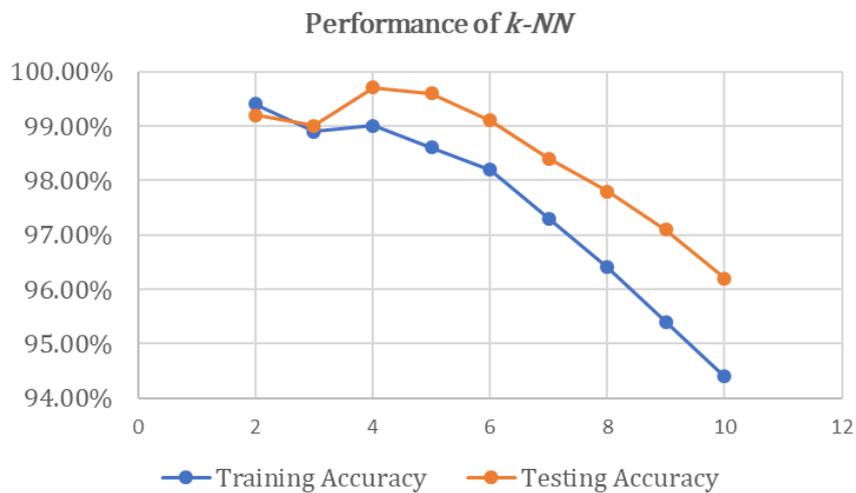


Figure 4. Accuracy of the training and testing sets for *k-NN* regression for varying values of *k*

The training and testing accuracies obtained [10] for *decision tree* regression for MSE and MAE criteria at varying values of maximum allowable depth are shown in Figure 5. It is observed that MSE at maximum depth value 3 gives the most optimum performance.

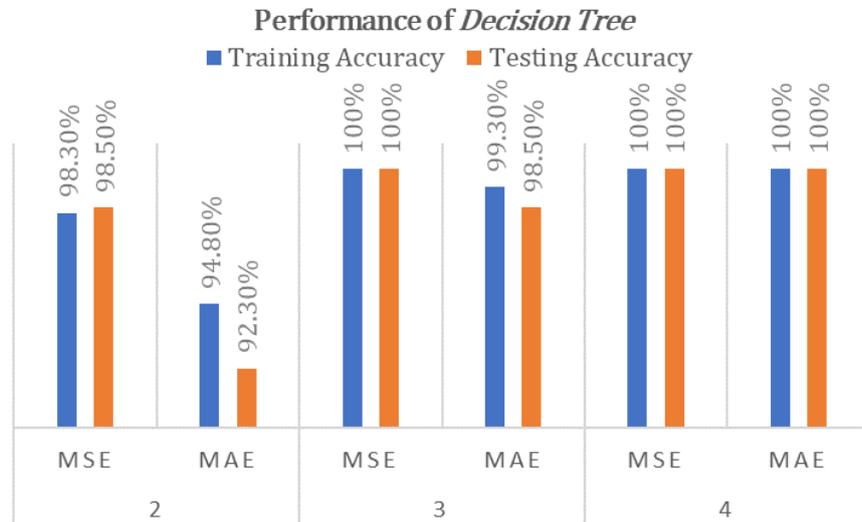


Figure 5. Accuracy of the training and testing sets for *decision tree* regression using MSE and MAE across varying values of maximum allowable depths

It is observed that for the dataset used in this work, accuracy of up to 99.5% is achieved using *SVM* regression, 99.4% is achieved using *k-NN* regression, and 100% is achieved using *decision tree* regression as shown in Figure 6.



Figure 6. ML algorithm performance comparison

It is observed that while *decision tree* and *k-NN* models are easy to understand and implement, the complexity of *SVM* is higher. A limitation of *k-NN* is that it is sensitive to localized data where localized anomalies can affect outcomes significantly. *Decision tree* has a high probability of overfitting and needs pruning for larger datasets. Subsequent to QoE prediction, all the users with poor QoE are found and the set of eNBs that served these users are isolated. If the maximum number of users served by a particular eNB have a QoE score less than or equal to one, such an eNB is

declared to be dysfunctional but if the maximum number of users served by a particular eNB have a QoE score above 1, then the eNB is declared to be functional.

5. CONCLUSIONS

This paper evaluates the performance of three ML algorithms used in a system model that uses ML algorithms to learn and predict the end-user experience and is able to detect dysfunctional eNodeBs in the network. Three ML algorithms were implemented and compared to study their effectiveness and the scalability of the system model. The ML algorithms *SVM* regression, *k-NN* regression, and *decision tree* regression were implemented to train a machine learning model used for QoE prediction that is further used for anomaly detection in SON networks. It was observed that high accuracy ($\geq 99\%$) can be achieved for QoE prediction and anomaly detection using all three ML algorithms for the dataset obtained from the *ns-3* simulation performed. *Decision tree* regression performed slightly better than *SVM* and *k-NN* regression, since the training and testing accuracy for the decision tree regression was better than the other two algorithms. However, *decision tree* has a high probability of overfitting and needs pruning for larger datasets. Hence, in case of overfitting, *SVM* regression and *k-NN* regression can serve as good alternatives for the *decision tree* regression machine learning algorithm for QoE-driven anomaly detection in SON networks. This paper demonstrates the potential for incorporating machine learning in next-generation networks for anomaly detection and suggests that the observed effectiveness and scalability of the proposed system model should be evaluated in actual networks with physically built hardware and actual users in the real-world environment.

REFERENCES

- [1] Jeffrey G. Andrews, Stefano Buzzi, Wan Choi, Stephen V. Hanly, Angel Lozano, Anthony C. K. Soong, and Jianzhong Charlie Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, volume: 32, no. 6, June 2014.
- [2] 3GPP TS 32.111-1, "Fault Management; Part 1: 3G fault management requirements," v14.0.0, March 2017.
- [3] I. de-la-Bandera, R. Barco, P. Muñoz, and I. Serrano, "Cell Outage Detection Based on Handover Statistics," *IEEE Communications Letters*, Volume: 19, Issue: 7, Pages: 1189 – 1192, 2015.
- [4] A. D'Alconzo, A. Coluccia, F. Ricciato, and P. Romirer-Maierhofer, "A distribution-based approach to anomaly detection and application to 3G mobile traffic," *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, Pages: 1 – 8, 2009.
- [5] Wenqian Xue, Mugen Peng, Yu Ma, and Hengzhi Zhang, "Classification-based Approach for Cell Outage Detection in Self-healing Heterogeneous Networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, Pages: 2822 – 2826, 2014.
- [6] Multazamah Alias, Navrati Saxena, and Abhishek Roy, "Efficient Cell Outage Detection in 5G HetNets Using Hidden Markov Model," *IEEE Communications Letters*, Volume: 20, Issue: 3, Pages: 562 – 565, 2016.
- [7] Oluwakayode Onireti, Ahmed Zoha, Jessica Moysen, Ali Imran, Lorenza Giupponi, Muhammad Ali Imran, Adnan Abu-Dayya, "A Cell Outage Management Framework for Dense Heterogeneous Networks," *IEEE Transactions on Vehicular Technology*, Volume: 65, Issue: 4, Pages: 2097 – 2113, 2016.

- [8] Isabel de la Bandera, Pablo Muñoz, Inmaculada Serrano, and Raquel Barco, "Improving Cell Outage Management Through Data Analysis," IEEE Wireless Communications, Volume:24, Issue: 4, Page s: 113 – 119, 2017
- [9] Eirini Liotou, Dimitris Tsolkas, Nikos Passas, and Lazaros Merakos, "Quality of Experience Management in Mobile Cellular Networks: Key Issues and Design Challenges," IEEE Communications Magazine, volume: 53, issue: 7, 2015.
- [10] Chetana V. Murudkar, Richard D. Gitlin, "QoE-driven Anomaly Detection in Self-Organizing Mobile Networks using Machine Learning," IEEE Wireless Telecommunications Symposium (WTS), April 2019, accepted - to be published.
- [11] ITU-T Recommendation P.10/G.100 Amendment 2, "Vocabulary for performance and quality of service," July 2008.
- [12] Eirini Liotou, Dimitris Tsolkas, Nikos Passas, Lazaros Merakos, "A Roadmap on QoE Metrics and Models," 23rd International Conference on Telecommunications (ICT), 2016.
- [13] ns-3 [online]. Available: <https://www.nsnam.org/>
- [14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.
- [15] Dimitris Tsolkas, Eirini Liotou, Nikos Passas, and Lazaros Merakos, "A Survey on Parametric QoE Estimation for Popular Services," Journal of network and computer applications, volume: 77, pages: 1-17, January 2017.
- [16] Srisakul Thakolsri, Shoaib Khan, Eckehard Steinbach, Wolfgang Kellerer, "QoE-Driven Cross-Layer Optimization for High Speed Downlink Packet Access," Journal of Communications, volume: 4, no. 9, pp. 669–680, Oct. 2009.
- [17] Sci-kit learn [online]. Available: <http://scikit-learn.org/stable/#>
- [18] Support Vector Machine Regression. [Online]. Available: <http://kernelsvm.tripod.com/>
- [19] Sergios Theodoridis, Machine Learning, Academic Press, 2015
- [20] Bruce E. Hansen, Nearest Neighbor Methods. [Online]. Available: <https://www.ssc.wisc.edu/~bhansen/718/NonParametrics10.pdf>
- [21] Paulo Valente Klaine, Muhammad Ali Imran, Oluwakayode Onireti, Richard Demo Souza, "A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks," IEEE Communications Surveys & Tutorials, volume: 19, issue: 4, 2017.

Authors

Chetana V. Murudkar is pursuing a Ph.D. in Electrical Engineering at University of South Florida under the supervision of Dr. Richard D. Gitlin. She is an RF Engineer at Sprint Corporation and her responsibilities involve design, deployment, performance monitoring, and optimization of Sprint's multi-technology, multi-band, and multi-vendor wireless communications mobile network. Her past work experience includes working with AT&T Labs and Ericsson. She has received an MS degree in Telecommunications Engineering from Southern Methodist University and a bachelor's degree in Electronics and Telecommunications Engineering from University of Mumbai.



Richard D. Gitlin is a State of Florida 21st Century World Class Scholar, Distinguished University Professor, and the Agere Systems Chaired Distinguished Professor of Electrical Engineering at the University of South Florida. He has 50 years of leadership in the communications industry and in academia and he has a record of significant research contributions that have been sustained and prolific over several decades. Dr. Gitlin is an elected member of the National Academy of Engineering (NAE), a Fellow of the IEEE, a Bell Laboratories Fellow, a Charter Fellow of the National Academy of Inventors (NAI), and a member of the Florida Inventors Hall of Fame (2017). He is also a co-recipient of the 2005 Thomas Alva Edison Patent Award and the IEEE S.O. Rice prize (1995), co-authored a communications text, published more than 170 papers, including 3 prize-winning papers, and holds 65 patents. After receiving his doctorate at Columbia University in 1969, he joined Bell Laboratories, where he worked for 32-years performing and leading pioneering research and development in digital communications, broadband networking, and wireless systems including: co-invention of DSL (Digital Subscriber Line), multicode CDMA (3/4G wireless), and pioneering the use of smart antennas (“MIMO”) for wireless systems. At his retirement, Dr. Gitlin was Senior VP for Communications and Networking Research at Bell Labs, a multi-national research organization with over 500 professionals. After retiring from Lucent, he was visiting professor of Electrical Engineering at Columbia University, and later he was Chief Technology Officer of Hammerhead Systems, a venture funded networking company in Silicon Valley. He joined USF in 2008 where his research is on wireless cyberphysical systems that advance minimally invasive surgery and cardiology and on addressing fundamental technical challenges in 5G/6G wireless systems.

