*International Journal Of Engineering Research And Development*
*e- ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com*
*Volume 21, Issue 3 (March 2025), PP 231-235*

# Unsupervised and Supervised Machine Learning Algorithms for Cell Fault Detection and Classification in LTE Networks

N.R. Dalef[1], A.A. Karpuk[2]
[1]*Departmentof Telecommunication Systems, Faculty of Telecommunication*
[2]*Department of Telecommunication Network Software Systems, Faculty of Telecommunication*
*Belarusian State Academy of Communications, Minsk, BELARUS*
[2]*Corresponding Author: a_karpuk@mail.ru*

**ABSTRACT**
*The article provides an overview of the main methods for detecting and classifying cell faults in LTE cellular networks based on unsupervised and supervised machine learning algorithms. The sources of input data for machine learning algorithms are determined. Unsolved problems and open questions that need to be addressed when implementing intelligent means of detecting and classifying cell faults in self-organizing LTE cellular networks are identified.*
***Keywords**: Cellular Communications, LTE Networks, Cell Failures, Cell Degradation, Machine Learning*

---

Date of Submission: 15-03-2025 Date of acceptance: 31-03-2025

---

## I. INTRODUCTION

A Self-Organizing Network (SON) can be defined as an adaptive and autonomous network that is scalable, stable, and flexible enough to achieve the desired goals. The concept of SON in mobile networks can be divided into three main categories: Self-Configuration, Self-Optimization, and Self-Healing, which are usually referred to collectively as Self-x functions [1]. Self-configuration can be defined as all the configuration procedures required to ensure the operability of the network. These configuration parameters can be in the form of individual base station (BS) configuration parameters such as IP configuration, Neighbor Cell List (NCL) configuration, cell radio parameter configuration, or configurations that will apply to the entire network. Self-configuration is activated whenever a new BS is deployed in the system, but it can also be activated when the network changes (e.g. BS failure or service or network policy change).

After the network is properly configured, self-optimization functions are activated. The self-optimization phase can be defined as functions that continuously optimize BS and network parameters to ensure near-optimal performance. Self-optimization can perform optimization of backhaul, caching, coverage and throughput, antenna parameters, mobility, handover parameters, load balancing, resource optimization, energy efficiency, and SON function coordination. By continuously monitoring the system and using the measurements obtained, self-optimization functions can ensure that the set goals are met and the overall network performance is near-optimal.

Since no network is perfect, failures and crash can occur unexpectedly, and this is no exception in cellular networks. When a fault or disruption occurs for any reason (e.g. software or hardware failure), self-healing functions are activated. Self-healing functions must be able to not only detect a failure event, but also diagnose the failure (i.e. determine why it occurred) and initiate appropriate compensation mechanisms so that the network can return to proper operation. Self-healing in cellular networks includes fault detection, fault classification, and fault compensation management.

Current methods for solving self-organizing problems in cellular networks lack the adaptability and flexibility needed to become real solutions for LTE and 5G networks. Although mobile operators collect a huge amount of data from the network in the form of network measurements, control and management interactions, and even data from their subscribers, the current methods used to configure and optimize the network are quite rudimentary. It is clear that more intelligence is needed to use all the information already collected by operators and provide adaptable and flexible solutions to the network. With this in mind, several Machine Learning (ML) solutions are used in the context of SON to learn from the different types of data collected by operators. Thus, a SON system needs some intelligence to be able to perform all three functions.

Machine learning is the ability of systems to acquire and continuously improve their own knowledge by extracting patterns from raw data to solve real-world knowledge problems and make decisions that appear subjective and mimic human “cognitive” functions [2]. ML algorithms are divided into three categories: unsupervised learning, supervised learning, and reinforcement learning. Unsupervised learning has no supervision and is mainly used when the expected output is unknown, and then the system will have to learn on its own. Supervised learning requires a supervisor to train the system. This supervisor tells the system for each input what the expected output is, and then the system learns based on this guidance. Reinforcement learning works similarly to the unsupervised learning scenario where the system has to estimate the expected output on its own, but in this case a reward mechanism is applied so that the system is rewarded or penalized depending on the decision it makes. Reinforcement learning is used to solve problems such as cell fault compensation management and cell self-optimization.

## II. MAIN SECTION

### 2.1. Input Data for Machine Learning Algorithms

Each of the machine learning algorithms uses cell performance data as input, which forms an information model of the cell. Most algorithms for cell self-configuration, self-optimization, and self-healing require data on the cell and its neighboring cells. The most frequently used network configuration is with three-sector BSs, where each sector forms a separate cell. With this configuration, in a homogeneous network, each cell has 6 neighboring cells, and in a heterogeneous network, there are many more neighboring cells. The information model of each cell of the LTE standard network includes general cell characteristics: physical cell identifier (PCI); PCI list of neighboring cells; cell operating frequency range; cell BS geographic coordinates; antenna direction azimuth; antenna height above ground level; antenna type; transmitter radiated power limits; transmitter radiated power; antenna tilt angle limits in the vertical plane; antenna tilt angle in the vertical plane; type of buildings in the cell; estimated cell radius; the number of resource blocks available to the operator in a cell.

For each discrete moment in time, the information model of each cell of the LTE network includes: the number of subscriber stations (UE) served by the cell; the number of served UEs waiting for services to be provided; the number of UEs to which services are provided by the cell; the number of free resource blocks in the cell.

For each discrete moment in time for each UE served by the cell, the information model of the cell of the LTE network includes: UE identifier (phone number); the average power of received pilot (reference) signals from one resource block of the cell (RSRP); the power of the useful signal from the cell BS (RXLEV); the signal-to-noise ratio for the cell BS (SINR), the signal quality indicator (CQI). For each UE waiting for services to be provided, the information model of the cell of the LTE network, in addition to the parameters listed above, includes the service initiator flag (subscriber or handover) and the flow class identifier for the service (QCI). For each UE to which services are provided by the cell, the information model of the LTE network cell, in addition to the parameters listed above, includes: modulation code; code rate; throughput per resource block; number of resource blocks allocated for the provision of the service. When the service is terminated, the time of termination and the initiator of the termination (subscriber, handover or connection failure) are recorded in the information model of the cell.

In [3], a reference model of cell self-healing is proposed in which fault detection is responsible for identifying problematic cells that need to be repaired, including cells with service outage (cell failure detection) and cells with service degradation (cell degradation detection). A possible simple method for cell fault detection is to set threshold values for some key performance indicators (KPI). However, gradual degradation cannot be detected simply by threshold value, especially if proactive detection is performed. Therefore, it is necessary to develop algorithms that take into account the selected KPIs and use appropriate decision logic to determine whether a failure or degradation has occurred.

The values of the required KPIs can be obtained from the LTE cell information model. In [4], it is shown that an LTE BS can report 33 KPIs for each cell, including the number of users, connection attempts, data/signaling packet rate, resource occupancy, and radio channel quality. Each BS generates the values of these 33 KPIs for each of its cells every hour, the data of each BS for each cell per day contains 792 KPI values. Thus, when using any ML algorithm to solve the problems of self-healing of the LTE standard cellular network, it is necessary to select the values of which KPIs will be used in the algorithm. In addition, as shown in [5], to solve the problems of self-healing of a certain cell, retrospective and current values of the selected KPIs of the cell itself and neighboring cells can be used.

### 2.2. Unsupervised Machine Learning Algorithms for LTE Cell Fault Detection and Classification

The following main unsupervised ML algorithms can be used to solve the problems of cell fault detection and classification in LTE networks: *k*-means clustering algorithm [6]; *c*-means fuzzy clustering algorithm [7]; Kohonen self-organizing neural network (SOM) algorithm [8]; level of outlier (LOF) algorithm [9]; local outlier probability (LoOP) algorithm [10].

The goal of clustering algorithms is to partition a set of KPI values into subsets or clusters, where KPI value points within one cluster are more similar to each other than to KPI value points in other clusters. The *k*-

means clustering algorithm seeks to minimize the standard deviation of the points in each cluster. At each iteration, the center of mass is recalculated for each cluster obtained at the previous step, then the KPI values are divided into clusters again in accordance with which of the new centers turned out to be closer according to the selected metric. The algorithm ends when no cluster changes occur at some iteration. The fuzzy *c*-means clustering algorithm, instead of an unambiguous answer to the question of which cluster a set of KPI values belongs to, determines the probability that a given set of KPI values belongs to one cluster or another.

Clustering algorithms can be used to solve the problem of detecting faulty and degrading cells and to solve the problem of classifying degrading cells. When solving the first problem, the set of selected KPI values is divided into three clusters: the first cluster contains sets of KPI values for a normally operating cell, the second cluster contains sets of KPI values for a degrading cell, and the third cluster contains sets of KPI values for a faulty cell. When solving the second problem, the set of KPI values for a degrading cell is divided into a number of clusters, each of which corresponds to one of the reasons for cell degradation. The reasons for cell degradation may include loss of radiated power, change in antenna tilt angle, cell overload due to insufficient resource blocks, occurrence of interference or noise emissions in the near, middle, or far zone. Currently, the issues of choosing KPIs for clustering algorithms, metrics for assessing the proximity of sets of KPI values, and classes for solving the problem of classifying degrading cells are open.

The Kohonen self-organizing neural network is a set of neurons located on a plane. Each neuron has coordinates and a weight vector, the dimension of which is equal to the number of KPIs selected to solve the problem of detecting faulty and degrading cells or solving the problem of classifying degrading cells. The number of neurons in the network is determined similarly to determining the number of clusters in clustering algorithms. The SON network training consists of three stages. At the first stage, the initial weights of neurons are initialized. The initial KPI values for each neuron can be set by random values, selected from specific real KPI values, or determined by a special algorithm that takes into account the entire set of input data. At the second stage, coarse training of neurons is performed. For each input set of KPI values, the value of the proximity function to the weight vector of each neuron is calculated and the best matching neuron (BMU) is determined. The weight vector of the BMU neuron is recalculated, neighboring neurons are determined from a sufficiently wide neighborhood and their weight vectors are recalculated. For a network with a small number of neurons, as in the case of a network for detecting faulty and degrading cells consisting of three neurons, the coarse training stage of neurons can be omitted. In the third stage, fine training of neurons is performed. The process is iterative in nature and stops when all the weight vectors of neurons remain unchanged at the next iteration. At the same time, at each iteration, the number of neighboring neurons considered for the BMU neuron decreases.

On the trained network, for each analyzed set of KPI values, a BMU neuron is defined, which provides a solution to the problem of detecting faulty and degrading cells or the problem of classifying degrading cells. At present, the issues of choosing a KPI for constructing neuron weight vectors, neurons for solving the problem of classifying degrading cells, the method of initializing the initial neuron weights, the function of the proximity of a set of KPI values to the neuron weight vector, the method of recalculating the neuron weight vectors, and the parameters of the fine-tuning algorithm for neurons have not been sufficiently studied. The LOF and LoOP algorithms can be used in self-training of unsupervised ML algorithms to level the influence of local outliers in the sets of KPI values used for self-training. The LOF algorithm calculates the densities of cluster points before adding the analyzed set of KPI values to the cluster and after adding this set of KPI values to the cluster, and then estimates the ratio of the obtained densities. If the coefficient of excess of the second density over the first is greater than the specified threshold, then the analyzed set of KPI values is considered a local outlier and is not taken into account when forming clusters. The LoOP algorithm is an improvement of the LOF algorithm and gives better results. This algorithm does not estimate the ratio of cluster point densities, but the probability that the analyzed set of KPI values is a local outlier with respect to the specified cluster. When using the LOF and LoOP algorithms, the questions of choosing the method for calculating the density of cluster points, the method for calculating the probability of obtaining a local outlier, the threshold value of the ratio of cluster point densities, and the threshold value of the probability of obtaining a local outlier for classifying the analyzed set of KPI values as a local outlier remain open.

### 2.3. Supervised Machine Learning Algorithms for LTE Cell Fault Detection and Classification

The following main supervised ML algorithms can be used to solve the problems of cell fault detection and classification in LTE networks: Naive Bayes algorithm [11]; *k*-nearest neighbors (*k*-NN) algorithm [12]; support vector machine (SVM) algorithm [13]; decision tree (DT) algorithm [14]; neural network (NN)-based algorithms [15]; hidden Markov model (HMM)-based algorithms [16].

In the Naive Bayes algorithm, KPIs are selected whose values will be used to detect or classify cell faults, the number of clusters is specified, and a training sample of KPI value sets is created. For each set of KPI values from the training sample, the cluster to which the set belongs is specified. Based on the training sample, the following quantities are calculated: *P(yk)* is the prior probability of a randomly selected set of KPI values belonging to cluster *yk*; *P(Xi|yk)* is the probable normalized value of the KPI with number *i* for cluster *yk*. For the set of KPI values to be classified, we denote the normalized value of the KPI with number *i* by *Yi*. We assign the considered set of KPI values to cluster *yk* for which the value $P(yk)\Pi(1-|Yi - P(Xi|yk)|)$ is maximum. More subtle decision rules can be used. The literature does not sufficiently study the issues of choosing KPIs for the problems of detecting or classifying cell faults, creating a training sample, calculating the values of *P(yk)* and *P(Xi|yk)*, choosing a decision rule when classifying sets of KPI values. The Naive Bayes algorithm shows good results if the KPIs used are independent of each other, which should be taken into account when choosing KPIs.

The *k*-NN algorithm calculates distances between the KPI set being classified and each KPI set from the training sample. The algorithm assigns the considered set of KPI values to the cluster to which most of its *k* nearest neighbors belongs. When using the *k*-NN algorithm to solve problems of detecting or classifying cell faults, it is necessary to select the KPIs used, determine the number *k*, and select a metric for calculating the distance between sets of KPI values.

In the SVM algorithm, a training sample of KPI value sets is used to construct linear or nonlinear functions that divide the entire set of possible KPI value sets into a given number of clusters. Each of the constructed functions defines a certain hyperplane in the KPI value space, and the KPI value sets from the training sample closest to this hyperplane are the support vectors of the hyperplane. When using the SVM algorithm to solve problems of detecting or classifying cell faults, it is necessary to select the KPIs used, determine the number of clusters, select the type of functions for constructing hyperplanes, and select support vectors from the training sample of KPI value sets for each hyperplane.

The DT algorithm is a method for representing decision rules in a hierarchical structure consisting of two types of elements – nodes and leaves. The nodes contain decision rules, and the sets of KPI values are checked for compliance with this rule based on the value of one KPI or a combination of values of several KPIs. In the simplest case, as a result of checking, the set of KPI value sets that fell into a node is divided into two subsets, one of which contains the sets of KPI values that satisfy the rule, and the other – those that do not satisfy the rule. Each of the subsets is considered as a new node. Its own decision rule is applied to each new node, and the procedure is repeated until the algorithm stop condition is reached. As a result, checking and splitting are not performed in the last node, and it is declared a leaf. A leaf corresponds to a cluster for each set of KPI values that fell into it. Unlike a node, a leaf does not contain a decision rule, but a subset of KPI value sets that satisfy all the rules of the branch that ends with this leaf. When using the DT algorithm to solve cell fault detection or classification problems, it is necessary to select the KPIs to be used, determine the number of leaves (clusters), determine the number of nodes, and construct decision rules for each node.

In NN-based algorithms, a neural network is created to classify a set of KPI values used, in the input layer of which each KPI used corresponds to its own neuron, and in the output layer, each neuron corresponds to one of the cell states to be classified. The number of hidden layers and the number of neurons in each hidden layer depend on the type of neural network and can be determined at the network training stage. The most well-known type of neural networks are feedforward networks (FFNNs), which have one or more hidden layers in which each neuron receives input from the previous layer and applies an activation rule to obtain output. Recurrent neural networks (RNNs) are designed to process sequential data where the order of the input data matters. Unlike FFNNs, RNNs have loops in their architecture that allow them to maintain an internal state or memory, which allows them to process sequences of varying lengths. LSTM (Long Short-Term Memory) is a type of recurrent neural network architecture that is well suited for modeling sequential data. LSTM networks are particularly effective at detecting long-term dependencies and can be used to predict outcomes based on past events. The LSTM architecture includes memory cells that can selectively forget or retain information over time, allowing the network to capture both short-term and long-term patterns in input data.

When training a network based on a set of KPI values from a training sample, the transition probabilities from each neuron of the input and hidden layers to each neuron of the next layer (for FFNN networks) and to each neuron of the next and previous layers (for RNN and LSTM networks) are determined, as well as the activation rules for neurons of each layer. When using NN-based algorithms to solve problems of detecting or classifying cell faults, the questions of choosing the KPIs used, the number of cell states, the type of neural network, the number of hidden words in the network, and the network training algorithm are open.

A Hidden Markov Model is a type of statistical model that can be used to predict the future state of a system based on its past and present states. For the purposes of cell fault detection or classification, the system is the LTE cell itself. The states in this system are the cell states that correspond to the clusters in the algorithms discussed above. An HMM-based algorithm can model the probability that a cell will degrade over time. This information can be used to predict when a cell is likely to degrade and take steps to prevent or mitigate degradation. An HMM-based algorithm can model the probability of cell degradation in a particular location. This information can be used to identify areas where LTE cell degradation is likely to occur and take steps to improve the performance of LTE networks in these areas. An HMM-based algorithm can model the probability of various causes influencing cell degradation. When training HMM-based algorithms, the probabilities of cell transitions from one state to another are calculated based on the current set of values of the KPIs used. Several HMM learning algorithms are known, of which the Viterbi algorithm or the Baum-Welch algorithm can be used to solve problems of detecting or classifying cell faults. When using HMM-based algorithms to solve problems of detecting or classifying cell faults, the questions of choosing the KPIs to use, choosing the number of cell states, and choosing the HMM learning algorithm are open.

## III. RESULTS

Unsupervised and supervised machine learning algorithms can be used to solve problems of cell fault detection or classification in LTE cellular networks. When solving the problem of cell fault detection, each cell is classified as healthy, degrading, or faulty (disabled). When solving the problem of cell fault classification, the most probable cause of degradation is determined for each degrading cell, which may include loss of radiated power, change in antenna tilt angle, cell overload due to insufficient resource blocks, the appearance of interference or noise emissions in the near, medium, or far zone. For each machine learning algorithm, it is necessary to determine the composition of KPIs, the values of which will be used by the algorithm to detect or classify cell faults. Due to the peculiarities of machine learning algorithms, the selected KPIs for different algorithms can be different. Unsupervised machine learning algorithms are trained on sets of values of selected KPIs, the classification of which is unknown. Supervised machine learning algorithms are trained on sets of selected KPI values for which the cell state is known, and for a degrading cell, the cause of cell degradation is known. All machine learning algorithms require selection of methods for configuring the algorithm parameters during its training. Finally, the question remains open as to which machine learning algorithms are preferable to use in each specific LTE cellular network to solve problems of cell fault detection or classification.

## REFERENCES


[1]. A Survey of Self Organization in Future Cellular Networks / O.G. Aliu [et al.] // IEEE Communications Surveys Tutorials. – Vol. 15. –2013. – P. 336-361.

[2]. A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks / Klaine P.V. [et al.] // IEEE Communications Surveys Tutorials. – Vol. 19, No 4. – 2017. – P. 2392–2431.

[3]. Barco, R. A Unified Framework for Self-healing in Wireless Networks / R. Barco, P. Lazaro, P. Munoz // IEEE Communications Magazine. – 2012. – Vol. 50, No 12. – P. 134–142.

[4]. Aoki, S. Few-Shot Learning and Self-Training for eNodeB Log Analysis for Service-Level Assurance in LTE Networks / S. Aoki, K. Shiomoto, C.L. Eng // IEEE Transactions on Network and Service Management. – 2020. – Vol. 17, Issue 4. – P. 2077–2089.

[5]. Asghar, M. Z. Cell Degradation Detection based on an Inter-Cell Approach / M. Z. Asghar [et. al.] // International Journal of Digital Content Technology and Its Applications. – 2017. – Vol. 11, No 1. – P. 25–33.

[6]. Bascasov, A. Using Data Mining for Mobile Communication Clustering and Characterization / A. Bascasov, C. Cernazanu, M. Marcu // Applied Computational Intelligence and Informatics (SACI), 2013 IEEE 8thInternational Symposium. – May 2013. – P. 41–46.

[7]. Velmurugan, T. Performance Based Analysis Between k-Means and Fuzzy c-Means Clustering Algorithms for Connection Oriented Telecom-Munication Data / T. Velmurugan // Applied Soft Computing. – 2014. – Vol. 19. – P.134–146.

[8]. Hashmi, U. S. Enabling Proactive Self-healing by Data Mining Network Failure Logs / U. S. Hashmi, A. Darbandi, A. Imran // 2017 International Conference on Computing, Networking and Communications (ICNC) . – Jan. 2017. – P. 511–517.

[9]. LOF: Identifying Density-based Local Outliers / M. Breuning [et al.] // SIGMOD Record. – Vol. 29, no 2. – 2000. – P. 93–104.

[10]. LoOP: Local Outlier Probabilities / H. P. Kriegel [et al.] // 18th ACM Conference on Information and Knowledge Management, ser. CIKM ’09. New York, NY, USA. – 2009. – P. 1649–1652.

[11]. Contextualized Indicators for Online Failure Diagnosis in Cellular Networks / S. Fortes [et al.] // Computer Networks. – 2015. – Vol. 82. – P. 96–113.

[12]. A Cell Outage Management Framework for Dense Heterogeneous Networks / O. Onireti [et al.] // IEEE Transactions on Vehicular Technology. – 2015. – Vol. 65, No 4. – P. 2097–2113.

[13]. A Learning-Based Approach for Autonomous Outage Detection and Coverage Optimization / A. Zoha [et al.] // Transactions on Emerging Telecommunications Technologies. – 2016. – Vol. 27, No 3. – P. 439–450.

[14]. Asghar, M. Z. Artificial Intelligence Enabled Self-Healing for Mobile Network Automation / M. Z. Asghar, F. Ahmed, J. Hamalainen // 2021 IEEE Globecom Workshops (GC Wkshps). – 2021. – P. 1–6.

[15]. Oguz, H. T. Cell Outage Detection in LTE-A Cellular Systems Using Neural Networks / H. T. Oguz, A. Kalaycioglu // Commun. Fac. Sci. Univ. Ank. SeriesA2-A3. – 2018. – Vol.60, No 1. – P. 31–40.

[16]. Alias, M. Efficient Cell Outage Detection in 5G Hetnets using Hidden Markov Model / M. Alias, N. Saxena, A. Roy // IEEE Communications Letters. – 2016. – Vol. 20, No 3. – P. 562–565.