Machine learning-based anomaly detection and root cause analysis in mobile networks

Miro-Markus Nikula
Tecnico Lisboa
Aalto university
Vantaa, Finland
miro-markus.nikula@aalto.fi

Abstract— **The evolution of mobile networks and the arrival of 5G technology have significantly increased network size and complexity leading to challenges in network management. This thesis addresses machine learning-based anomaly detection and root cause analysis in mobile networks while providing information of various related topics, such as 4G and 5G networks, quality of experience, machine learning methods and self-organising networks. The work focuses on the limitations of typical mobile network data, mainly the lack of labels and the need for expert knowledge and develops an unsupervised machine learning-based model that can automatically detect network anomalies and perform root cause analysis on unlabelled mobile network data.**

The proposed model combines DBSCAN and LSTM AE, and is trained and tested using real world 4G network data from a Portuguese telecom operator, NOS. The study shows promise in DBSCAN's ability to separate normal network traffic patterns from abnormal, and the ability of LSTM AE to learn the daily network KPI behaviour and to detect anomalies based on their reconstruction errors. The reconstruction errors also provide insight into the individual KPIs mostly contributing to the anomalies, thus the anomalies' root causes.

The research and findings in this thesis highlight the importance of self-healing in self-organising networks and how different machine learning models can be used to perform anomaly detection and root cause analysis. The obtained results show that anomalies in the available network KPIs do not always result in abnormal traffic patterns and vice versa. Consequently, it can be derived that DBSCAN based solely on traffic volumes is not an ideal method to separate normal network data from ab-normal, with the goal of finding network anomalies. In addition, the results underscore the importance of high-quality data in terms of sampling rate and the number of KPIs, as well as the importance of data analysis in finding patterns on different levels of mobile networks.

Keywords—Mobile network, Anomaly detection, Root cause analysis, Machine learning, DBSCAN, LSTM AE, SON, QoE)

I. INTRODUCTION

Each new generation of mobile networks has introduced numerous new technologies and enhancements to the previous ones. The first generation (1G) mobile networks introduced the first wireless cellular technology marking the transition from traditional landlines to voice only mobile networks. The second generation (2G) networks brought upon the transition from analogue to digital networks and introduced the Short Message Service (SMS). The third generation (3G) networks started to widely utilise packet switching, leading to higher data rates. The fourth generation (4G) networks enabled significant increases in data rates and introduced Voice over Internet Protocol (VoIP) service.

With the advent of the fifth generation (5G) mobile networks, the telecommunications industry is undergoing another major transformation. New mobile networks are characterised by their high-speed data transfer, low latency and reliable communications as well as their ability to simultaneously provide services to a vast number of devices and different applications. 5G networks will achieve these advancements through a myriad of revolutionary technologies and solutions. While the innovations in 5G networks will drastically reshape the landscape of mobile communications, these improvements will also increase the complexity of networks, making network management increasingly challenging.

Self-organising networks (SON) are designed to reduce the need for manual network management allowing for more effective and robust automatic management of cellular networks sometimes even in real-time [1]. One of the categories of SONs is called self-healing, to which network anomaly detection and root cause analysis (RCA) are integral parts of. Anomaly detection involves identifying unusual behaviour of the network and RCA is used to identify the underlying causes of the anomalies. There is a variety of different types of anomalies, varying from subtle, barely noticeable abnormal behaviours to anomalies severe enough to cause public safety hazards. Such is the case for instance with network-wide service outages, resulting in disconnected emergency calls. Consequently, effectively detecting anomalies and repairing their sources can result in better user experience and reduction of expenses for the network operator [1].

The development of anomaly detection and RCA methods is closely related to that of machine learning techniques, especially deep neural networks (DNN). Numerous automatic machine learning-based anomaly detectors in the context of 4G mobile networks have been developed throughout recent years. Most of them use simulation data or labelled anomaly datasets. However, a major problem with labelled mobile network datasets is their unavailability. Vast amounts of network data are continuously generated, but labelling it is an extremely laborious process often requiring the expertise of network engineers. The lack of

labelled data not only hinders anomaly detection but also poses significant challenges in RCA within mobile networks, which is a relatively scarcely studied subject. Moreover, existing solutions often depend on the insights from network experts during the system's setup phase.

The primary objective of this work is to design an unsupervised ma-chine learning-based model with the ability to automatically detect anomalies and to conduct RCA in unlabelled mobile network data, without re-quiring network expert input during the implementation stage. Specifically, the anomalies of interest are the ones resulting from network errors, and not for example from natural human behaviour. The model is developed through a process of experimenting with different models and lever-aging data analysis, particularly in identifying patterns and correlations within the data. Specifically, the final model consists of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and an Autoencoder (AE) incorporating Long Short-Term Memory (LSTM) layers. The model is designed to be used by network operators to enhance network performance and reliability. This work is centred around a dataset containing unlabelled real-world 4G network data provided by a Portuguese tele-com operator, NOS. This work aims to answer the following questions:

- How can anomaly detection and RCA be performed in the context of mobile networks?
- How can they be performed using only unlabelled data?
- How will the deployment of 5G networks change the landscape of anomaly detection and RCA in mobile networks?

The work is organised as follows: Chapter II provides fundamental information about essential themes of this work, such as mobile networks (especially 5G) and machine learning methods. It also presents the concepts of anomaly detection and RCA along with an overview of the existing research about them. In Chapter III, a description and an analysis of the dataset used for this research are given, and the methodologies chosen for anomaly detection and RCA are outlined, coupled with the thought process behind their selection. The results and the means to verify them are provided in Chapter IV along with a critical discussion of the results. Finally, the work is concluded and possibilities for future work are considered in Chapter V.

II. RELATED WORK

Anomaly detection specifically in the context of mobile networks has its fair share of literature from recent years. For example, [2] tries to capture the spatial and temporal nature of 4G Radio Access Network (RAN) data by using a model that combines the strengths of capturing spatial dependencies of Convolution Neural Networks (CNN) and capturing temporal dependencies of LSTMs. Then, together with AE-based feature selection, a decision tree model is used to classify cells into normal or anomalous classes. LSTMs are also utilised in [3] to capture temporal dependencies and combined with an AE, to detect anomalies based on the reconstruction error. Autoencoder is used also in [4], where the non-stationary nature of mobile data is considered with a Variational Autoencoder (VAE)-based model requiring significantly fewer computing resources while compromising only two percents in the F1-score when compared to OmniAnomaly. However, the comparison was done using only one dataset. A custom NN is designed in [5] to consider the effect of more than one fault happening simultaneously in 4G RAN. A more statistical approach is taken in [6], where the Chi-squared test is used to statistically detect anomalous network cells based on different traffic profiles (weekday, weekend, rural). Another method not relying on deep learning is experimented in [7], where a more user-centric approach is taken and a mapping between network KPIs and QoE is created. The authors use a Classification and Regression Tree (CART) to predict QoE score within base stations (BS). If the score is below a threshold, an anomaly is detected, indicating a dysfunctional base station.

Graph neural network (GNN)-based models have shown promise as general anomaly detection models. However, GNNs are a rather new technology, and have not been widely used in mobile anomaly detection publications. A recent paper published in 2023 by [8] uses a graph-based approach to detect anomalies from call detail records (CDR) in a supervised setting. The proposed method captures the spatial dependencies between different cells of a cellular network by using a Graph Convolution Neural (GCN) network. The GCN network is a CNN which considers the dependencies in a graph-structured cell network. The GCN module is combined with three LSTM modules. Each LSTM module operates on a different temporal scale with the goal of capturing hourly, daily and weekly dependencies of the data. Although applied only to a simple CDR dataset which does not include many typically used network Key Performance Indicators (KPI), the results are promising and encourage further research on GNN-based solutions for mobile anomaly detection.

It should be noted that many research papers are using a network simulator to acquire network data, since real world operator data can be difficult to come by. This is the case in [6], [2] — [5], [9] and [10]. Even if researchers are able to obtain labelled real-world data, it is usually very unbalanced, meaning that most of the time the network is functioning normally and anomalies occur rarely [10]. The imbalance between normal and abnormal data can make model training difficult [11]. In [10], this problem is tackled by using a generative adversarial network (GAN) to generate artificial data points similar to real-world data. In addition, data undersampling, where data points clearly far away from the time of appearance of anomalies are removed from training data, as well as penalised classification, where the classifier is additionally penalised for classifying an anomaly point as normal, are considered in [2]. Arguably the greatest challenge in real world network data is the lack of data labels, meaning the ground truth behind anomalies, for model training [12]. The lack of labels significantly limits the available approaches, especially supervised machine learning methods, and reduces the interpretability of the results. Therefore, “manually” labelling data points is used in some work. For example, in [13] data points with a Euclidean norm larger than the norm of standard deviations are labelled as anomalies. In [14], mobile network KPI dataset is labelled with user Quality of Experience (QoE) values extracted from user throughput and handover success rate. This approach based solely on a few KPIs to estimate QoE might turn out not to be the most accurate, considering that QoE estimation requires lots of information. The wide range of different applications with different

throughput requirements also restricts the use of fixed value thresholds.

The lack of labels mainly affects supervised learning methods, making unsupervised learning particularly useful in cases with large amounts of unlabelled data [1]. Clustering techniques are used in [15] and [16] to detect anomalies based on user activity from CDR data. Both use the same dataset with one-hour and 10-minute aggregation intervals respectively. Although anomaly detection from CDR using only user activity mainly provides information about users' behaviour and might not reveal faults in the network, these papers show the effectiveness of simple clustering algorithms (agglomerative and K-means) in finding simple anomalies in aggregated time series data.

Agglomerative clustering is used also for RCA in [2], where the authors try to group anomalies caused by the same problem into the same cluster. While this approach does not require data labels, it does require a network expert to determine the root cause label for each cluster. A similar, yet more complex solution is presented in [9], where the authors use a Self-Organising Map (SOM) to transform high-dimensional KPIs into a more simplified form. Next, Ward's hierarchical clustering is used to cluster the simplified data into clusters corresponding to different root causes. The authors ensure that the different clusters represent different faults by comparing each cluster's statistical properties with the Kolmogorov–Smirnov test. Network experts are needed to label the clusters with their root cause labels in the training phase, but in the exploitation phase the system compares the KPI statistics to those labelled in the training phase and is automatically able to determine the most probable root cause.

In addition to unsupervised clustering, in [2] a supervised decision tree approach is used to classify anomalies under different root cause labels generated by network experts, including too late handover and excessive antenna tilt. Decision tree structure is used also in [4], where anomalous data points are labelled with an anomaly score calculated from their KPIs. A boosting tree classifier then uses the list of anomaly scores to classify the anomalies to match one of six different root causes. The effect of multiple faults happening simultaneously is considered in [5] with the use of supervised NNs trained to detect specific root causes. However, only three root causes were considered, two of which, excessive antenna uptilt and downtilt, cannot happen simultaneously. Another supervised method, Naïve Bayes classifier is used in [6] to express the uncertain root causes in terms of probabilities.

Example root causes considered for example in [6] and [9] are excessive antenna downtilt or uptilt, coverage hole, inter-system interference, too late handover and cell outage. Examples of network KPIs used are reference signal received power, reference signal received quality, handover success rate, signal to interference plus noise ratio, distance between user and base station, and average user throughput. All discussed papers address anomaly detection and root causes analysis in 4G networks.

It is evident that there is a multitude of different approaches to conduct anomaly detection and RCA in the context of mobile networks. The choice of the approach depends on multiple factors including labels, the size and features of dataset, as well as available computational resources. Other considerations include the trade-off between missed anomalies and false alarms, whether proactive or reactive anomaly detection is required, whether the data is processed in sliding windows or incremental updates, and whether to use online or offline training [11].

III. Research material and methods

This chapter describes the implementation of a machine learning-based model to detect anomalies from unlabelled mobile network data and to automatically determine the anomalies' root causes. The used data is explained with data analysis and the specific steps to conduct anomaly detection and root cause analysis are explained under methodology.

A. Dataset description

The dataset used in this work includes mobile network KPIs collected during a period of over 6 months of network traffic from 25 Long-Term Evolution (LTE) base stations (BS) in the 4G radio access network of a Portuguese telecom operator, NOS. The data collected from one of the BSs is from the period of 13.9.2022 – 15.3.2023 and the data from the remaining 24 BSs is from 3.12.2022 – 6.6.2023. The data from the earlier period is from a BS called CEIRA_LTE_MCO001B2. The decisions regarding the system architecture and the used models are based on the data analysis and the observed results of this BS, even though the model parameters are set individually for each of the 25 BSs. Consequently, CEIRA_LTE_MCO001B2 can be viewed as a training dataset and will be referred to as the training BS. The KPIs along with their descriptions and units are presented in Table 1.

Table 1: Network KPIs.

Abbreviation	Network KPI	Description	Unit
USERS	Average number of users	Average number of simultaneous UE.	-
PRB_DL	PRB DL average usage rate	Average physical resource block utilisation in downlink direction.	%
PRB_UL	PRB UL average usage rate	Average physical resource block utilisation in uplink direction.	%
HSR_INTRA	HSR intra frequency	Handover success rate of intra-frequency relations.	%
HSR_INTER	HSR inter frequency	Handover success rate of inter-frequency relations.	%
CELL_DL_TP	Cell DL average throughput	Average cell data throughput in downlink direction.	Kbps
CELL_UL_TP	Cell UL average throughput	Average cell data throughput in uplink direction.	Kbps
USER_DL_TP	User DL average throughput	Average user data throughput in downlink direction.	Kbps
USER_UL_TP	User UL average throughput	Average user data throughput in uplink direction.	Kbps
TVD	Traffic volume data	Traffic volume of data services.	MB
CA_TP	Carrier aggregation throughput	Total throughput using carrier aggregation.	Kbps
CQI	Average CQI	Average Channel Quality Indicator.	-
TIME_ADV	Average timing advance	Distance between UE and BS.	-

Individual data points in the dataset describe hourly aggregated network KPI-values. Aggregation here means the average over an hour in all the KPIs except for traffic volume data (*TVD*) and carrier aggregation throughput (*CA_TP*), where the values are the sums over an hour.

KPIs are calculated using data from multiple counters provided by network device vendors. The counters are located at the eNodeB base stations of the LTE network and are collecting data of the traffic going through BSs. Each BS consists of three 120-degree sectors, each containing three cells. Different cells represent different frequency bands. Most of the BSs use the 800, 1800 and 2100 MHz bands, but there are some exceptions, such as the 2600 MHz band being used. As a result, the dataset contains 25 (base stations) × 3 (sectors) × 3 (cells) × ~190 (days) × 24 (hours) > 1 000 000 data points. For better understanding the organisation of the dataset, a table of the cells based on sectors and frequency bands is provided in Table 2.

Table 2: Organisation of cells based on the sector and the frequency band.

	Sector 1	**Sector 2**	**Sector 3**
800 MHz	Cell 1	Cell 2	Cell 3
1800 MHz	Cell 4	Cell 5	Cell 6
2100 MHz	Cell 7	Cell 8	Cell 9

B. Data analysis

As with any data driven approach, a comprehensive data analysis is needed to understand the nature of data and the underlying patterns and dependencies between different features of it. Data visualisation, feature correlation and data grouping are methods that, when used together with domain knowledge, can help discover useful insights. These insights can further be used for first, understanding data, and secondly, to help detect anomalies.

Pearson correlation coefficients [17] between different KPIs in different sectors of a BS are calculated according to:

$$r = \frac{COV(X,Y)}{\sigma_X \sigma_Y}, \quad (1)$$

where X and Y are time series of different KPIs, $COV(X,Y)$ is the covariance between X and Y, σ_X is the standard deviation of X, and σ_Y is the standard deviation of Y. It is noteworthy that KPIs can correlate to each other differently in different sectors. For example, there is a correlation of 0.4 between *USERS* and *TVD* in sector 2, whereas the correlation is 0.8 between the same KPIs in sector 3 of the training BS. This means that sector 2 has more factors affecting traffic volume than just the number of users, such as for example the nature of data services used. Correlations between KPIs of sector 1 are also unique. Correlation analysis can be helpful in finding the relationships between different KPIs, which can be useful especially when performing RCA.

In addition to differences in correlations, different sectors can exhibit distinct characteristics such as variations in traffic patterns. These distinctions are a result of sectors facing towards different geographical regions. Several studies have investigated the characteristic behaviours based on geographical location. For example, [18] identifies five different traffic profiles in the urban environment: residential, office, transportation, entertainment and comprehensive (for the areas with mixed profiles). Typically, the geographical areas do not fully correspond to a single profile, but rather contain characteristics of multiple profiles.

Inspecting traffic volume patterns show a dominant profile of a residential area in sector 1, whereas sector 3 shows a dominant profile of an office area. Sector 2 on the other hand, does not fully match any of the known traffic profiles, and therefore most likely consists of a mix of several of them. Notably, some sectors have completely different behaviours on working days and during weekends. Especially, the office area in sector 3 displays lots of variance between working days and weekends. As the sector is oriented towards an office area, the weekdays have a traffic peak during the day, whereas on weekends the traffic pattern resembles more that of a residential area with the peak occurring closer to midnight. While the patterns between different sectors may vary, the three cells within a sector usually exhibit similar patterns. This is based on the fact that even though cells are using different frequency bands, they are still facing the same direction, and are therefore affected by the same traffic patterns. Even though the traffic patterns seem to be relatively similar, there are some differences between cells, including those in traffic volume, which can stem from the network's configuration. Various frequencies, for instance, might be allocated to different types of services and ranges. However, grouping the frequency bands within each sector simplifies matters by reducing the number of variables, making the model outlined in the subsequent sections more manageable.

The key takeaways from the data analysis are the factors that significantly affect the design of the developed model. To summarise the data analysis part, the dataset can be condensed to a sector-level analysis instead of a more granular cell-level examination. Reducing the dataset any further is not feasible, given the differences in traffic patterns and KPI correlations between different sectors. Finally, the data displays large variations in patterns between weekdays and weekends.

C. Methodology: Anomaly detection and RCA

This subsection presents the main idea and initial considerations behind the developed model and describes the individual steps that need to be taken to build it.

1) Initial considerations

The anomaly detection and root cause analysis system for unlabelled mobile network data consists of data preprocessing and unsupervised learning methods, DBSCAN [19] and LSTM autoencoder [20]. The system pipeline is depicted in Figure 1.

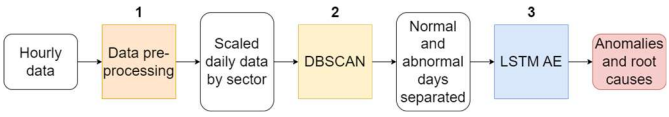

Figure 1: Anomaly detection and RCA system pipeline for unlabelled mobile network data.

The core idea is to train an LSTM autoencoder model to identify normal network behaviour. By using a labelled dataset, where normal traffic is already separated from anomalies, the training process would be very straightforward, but as the

dataset in use is unlabelled, an alternative approach of separating normal data from anomalous data is required.

DBSCAN is used to separate full days (24 hours) of network data into normal and abnormal clusters, resulting in a set of days containing only typical traffic patterns. The days where the traffic pattern is atypical are considered abnormal. The abnormality in the traffic pattern can be the result of network issues, but it can also be caused by natural human behaviour, such as holidays or events, which are outside the scope of this work. For this reason, DBSCAN is not used as the final anomaly detector, but rather a step to obtain a set of normal days, which are then used to train the LSTM AE.

In addition to anomaly detection, the LSTM AE model is used also in automatic RCA. The KPI-wise reconstruction errors generated by the AE, are used to indicate the most probable root causes of each anomaly. For clarity, the term abnormal day is used to refer to the atypical days from DBSCAN, and the term anomalous day is used to refer to the days flagged by LSTM AE.

The work has been conducted using the Python programming language and its libraries. A list of the main libraries and software along with their versions is available in Table 3.

Table 3: Main libraries used.

Software	Version	Usage
Python	3.11.2	Programming language
Pandas	1.5.3	Data structure and data processing
Keras	2.12.0	LSTM AE
Tensorflow	2.12.0	Used in Keras
Scikit-learn	1.2.2	DBSCAN, NearestNeighbors and StandardScaler
Numpy	1.23.5	Mathematical operations and data structures
Kneed	0.8.2	Locating the knee point for DBSCAN
Seaborn	0.12.2	Visualisation
Matplotlib	3.7.1	Visualisation

2) Data preprocessing

Data preprocessing is an essential part of any machine learning pipeline to guarantee that the data used is clean and consistent for training models. Outlier removal is often a step performed in the preprocessing part, but with anomaly detection, outliers are the points of interest and therefore they should not be removed. The preprocessing of the data consists of the following three steps:

1. Data sampling:

   The design choices for data sampling are based on the data analysis part. Each base station is divided into three sectors. A separate dataset is created for each sector where the hourly KPIs are now aggregated from the three cells within the sector. The hourly data is divided into individual days: 24 hours from 01:00-24:00.

2. Data filtering:

   In the filtering part, if a day is not full, for example, if it is missing some hours of data, it will be filtered out because missing hours proved to have a large effect in the performance of the autoencoder. The resulting dataset contains only full days of mobile network KPIs, meaning that they include 24 hourly values of each KPI.

3. Data scaling:

   In the scaling phase each KPI is scaled individually using standard scaling [21], where the scaled value for each data point is calculated using:

$$z = \frac{x - \mu}{\sigma}, \quad (2)$$

   where x is the original value, μ is the mean and σ is the standard deviation of the KPI. As a result, each KPI has zero mean and standard deviation of 1 across all the datasets. Scaling data is important as different KPIs have values in highly different magnitudes.

3) DBSCAN

In most of the related work, as discussed in Chapter II, there is already a dataset containing only normal samples, which can directly be used to train the AE. However, since the data used in this work consists of both normal and abnormal days within the same dataset, it is essential to obtain the set of normal days by separating it from the abnormal one before training the AE.

K-Means clustering is a very commonly used clustering algorithm useful in many problems, but as the abnormal days can be very different from each other, K-Means would not place them in the same cluster, resulting potentially in multiple clusters that are difficult to interpret. DBSCAN however, places outliers into the same cluster even if they are not very similar to each other. The result is a clear distinction between outliers and normal data points, which is exactly what is needed in order to train the AE.

DBSCAN was selected due to this property as well as the fact that it is a clustering algorithm that does not require the number of clusters predefined. Instead, it creates a cluster and assigns a data point and its neighbours into the cluster if the data point has at least a certain number of neighbours (*MinPts*) within a certain radius (*Eps*). If it does not, the data point is considered an outlier, or as in this case, abnormal. The pseudo code of the DBSCAN algorithm explaining the clustering process in more detail can be found in Algorithm 1. *MinPts* and *Eps* are defined by the user before clustering.

In this work, the separation of normal days from abnormal ones is based on a single KPI, traffic volume data (*TVD*). *TVD* was chosen for two reasons. Firstly, it forms a consistent pattern in most of the days and these patterns have been studied a great deal in the literature as characteristic traffic patterns of different geographical areas. To ensure the accuracy of the clustering, the patterns of the days considered normal, are compared to those from the literature by using correlation analysis and visual inspection.

```
DBSCAN (W, MinPts,Eps)
Input: a data set W, MinPts
Output: arbitrary shape clusters
for each data point p ∈ W do
    if p is not mark as 'seen' then
        Mark p as 'seen'
        Find neighborhood of data point p, NeighborPts
        if NeighborPts < minimum points then
            Mark data point as a noise
            else
                clusterid=clusterid+1
            end
        end
        for all q ∈ NeighborPts do
            Mark data points q as seen
            Find neighborhood of data point q, NeighborPts
            if NeighborPts > minimum points then
                Give data point q a clusterid
            end
        end
    end
end
```

Algorithm 1: Pseudo code of the DBSCAN algorithm [22].

Secondly, when experimenting DBSCAN with multiple KPIs in comparison to only *TVD*, a considerably larger portion of data was labelled abnormal. An explanation for this phenomenon could be the curse of dimensionality, which increases the distance between individual data points. It is further confirmed in [23], that high dimensional data makes DBSCAN unstable. In addition, it is stated by [10] that anomalies in mobile networks are rare. Therefore, to avoid classifying a significant amount of the normal data as abnormal, only *TVD* was used. As a result, the rest of the KPIs are not considered in the clustering phase.

The clustering is performed separately for weekdays and weekends because, as seen in the data analysis part in Section III.B, there might be considerable variation between the two. In the case that the clustering was done for weekdays and weekends combined, the normal days could be placed in one or two clusters depending on how similar the traffic patterns of weekdays and weekends are to each other. By clustering weekdays and weekends separately, the system is made sure to find a single cluster containing the normal days and therefore the interpretability of the results increases.

Selecting the values for parameters *MinPts* and *Eps* plays a pivotal role in the clustering result. The selection of parameters was performed by using existing practices found in the literature as well as experimenting with different values. The value of *MinPts* determines how many data points are required to form a cluster. A very large value can result in absence of any meaningful clusters whereas the value of 1 leads to every data point forming its own cluster. An appropriate value was determined through a process of experimentation with the training BS, as well as correlation analysis elaborated upon in Section IV.A. In addition, the value of *MinPts* commonly depends on the number of features [24]. Specifically, the value of *MinPts* was set to the total number of hours in a day, which is also the number of features, totalling 24. The input data for the clustering is of shape $n \times 1 \times 24$, where n is the total number of days to be clustered, and therefore, the number of features is 24.

Selecting the value for *Eps* is slightly more complex. The value was set to be the knee point of the sorted distances to the 5th nearest neighbour of each data point. This type of approach of using the knee point is common in the DBSCAN literature and is used for example in [24]. The 5th nearest neighbour was selected via trial and error and it proved to be a good compromise between a too short radius where significantly more days would be considered abnormal, and a too long radius where next to no abnormalities were found. The distance metric for the nearest neighbours is Euclidean distance and the knee point is calculated using the KneeLocator function from Python's Kneed package. The knee point should be the point where a curve of sorted values experiences a significant change. The knee point, which is also the *Eps* value used for clustering the weekdays of sector 1 of the training BS is 3.33.

Resulting from the DBSCAN clustering is a set of labels, which in this work only contained two different values, normal and abnormal. The DBSCAN algorithm used in this work is part of Sklearn's clustering package and it assigns the label -1 to the abnormal cluster. An example of a normal day in contrast to abnormal day is displayed in Figure 2

. It is noteworthy that the traffic pattern of the normal day resembles closely the average pattern of sector 1, and the pattern of the abnormal day seems to be unexpected, considering the dominant residential traffic profile in sector 1.

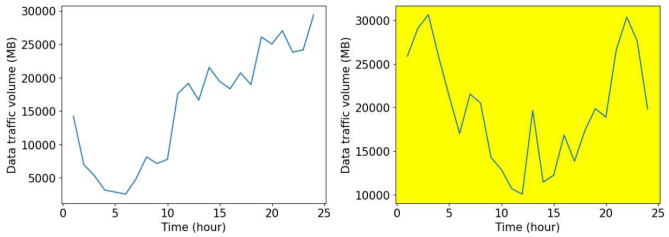

Figure 2: Normal day (a) and abnormal day (b) in sector 1 in the training BS based on *TVD*.

4) LSTM AE

Autoencoders consist of two parts, an encoder and a decoder. They can be used to learn the normal behaviour of mobile networks by letting them learn how to first encode the original normal data into a lower-dimensional feature space and then how to decode the lower-dimensional representation to match the original data as well as possible.

Following is the general definition of an autoencoder by [3]. Let
$X = R^D$ be the input space and F the feature space. An encoder is a function $\phi : X \to F$, that encodes inputs into the feature space F, while a decoder is a function $\psi : F \to X$, that tries to reconstruct the original input from the encoded representation in the feature space F. An autoencoder $\Phi_{AE} : \phi \circ \psi$, can be denoted as $\Phi_{AE}(x(n)) = \hat{x}(n)$, where $x(n)$ is the original sequence and $\hat{x}(n)$ is the reconstructed sequence. The difference between the original and the reconstructed sequences is called the reconstruction error [3]. The reconstruction error can be used to detect an anomaly as it should be low for normal data and high for anomalous data, but in this work, it is also used to determine the anomalies' root causes.

AEs which consider the temporal nature of the data (such as AEs with LSTM layers), are commonly used in the related work.

In this work, LSTM-layers are included both in the encoder and the decoder, because the AE is designed to learn the behaviour of individual days, which consist of consecutive time steps. LSTM layers are especially suitable for capturing the temporal dependency in sequential data, while having a "longer memory" than traditional Recurrent Neural Networks (RNN) and solving the vanishing gradient problem [25]. The architecture of the LSTM AE neural network can be seen in Table 4 and the three main parts of it are explained below:

Table 4: LSTM AE model architecture.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 64)	18432
repeat_vector (RepeatVector)	(None, 24, 64)	0
lstm_1 (LSTM)	(None, 24, 64)	33024
time_distributed (TimeDistributed)	(None, 24, 7)	445
Total params	51911	
Trainable params	51911	
Non-trainable params	0	

- Encoder:

  A single LSTM layer consisting of 64 units. The number of layers and their sizes typically depend on the complexity and the amount of data available for training. The goal here is to learn the daily pattern of different mobile network KPIs. Even though there can be multiple different KPIs behaving differently throughout the day, the patterns of individual KPIs especially within a single sector are not very complex. Therefore, a single layer with 64 units proved to be sufficient. The activation function used is Rectified Linear Unit (ReLU) in order to include non-linearity in the network. The input shape for this layer is (24, *n_KPIs*), where 24 is the number of hours in a day and *n_KPIs* is the number of different KPIs in the data.

- RepeatVector:

  The RepeatVector layer is the connection between the encoder and the decoder, and it modifies the shape of the output of the encoder to be inputted to the decoder. The parameter used for the layer is 24, which means that the encoded representation is applied for each timestep of the following decoder.

- Decoder:

  The decoder is responsible for reconstructing the encoded data to match the original input. It consists of a single LSTM-layer and a time distributed dense-layer. Just like the LSTM-layer in the encoder, the LSTM-layer of the decoder has 64 units and uses ReLU as the activation function. The network is a sequence-to-sequence model, and therefore it needs to output a sequence. Therefore, the layer has the *return_sequences*-parameter set to True. Lastly, the decoder has a TimeDistributed Dense layer with the parameter of *n_KPIs* to match the dimensionality of the original input data.

The model is compiled using the Adaptive Moment Estimation (Adam) optimiser and Mean Squared Error (MSE) is used as the loss function. Adam optimiser is a widely used one for training neural networks in various domains and it proved to converge significantly faster than for example Stochastic Gradient Descent (SGD) for the used dataset. The learning rate used is 0.001, which is the default for Adam optimiser. As the name suggests, MSE calculates the mean of the square of the errors. By squaring the errors, MSE amplifies larger errors, causing anomalies to become more prominent and easily identifiable. The MSEs [26] are calculated using:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 , \quad (3)$$

where n is the number of samples in the dataset, y_i is the actual value of the i-th sample of the target variable and $\hat{y}_i$ is the predicted value. To construct the LSTM-AE network, Keras was used, which is an API built on top of the TensorFlow-platform and is very straightforward to use.

The normal days identified by the DBSCAN clustering are divided into training and test sets with 90 and 10 % of the normal days, respectively. The training set is used to train the LSTM AE and the test set is used to observe its results. Because different sectors of different BSs can be very different from each other, the model is trained individually for each of the three sectors of every BS. The training dataset for each sector of each base station is of shape (*n_days*, 24, 7), where *n_days* is the number of days in the sector, 24 is the number of hours in a day and 7 is the number of potential root cause KPIs in the dataset. The model iterates through the training dataset 30 times, by setting the *epochs*-parameter of the fit-method to 30. This number of iterations was observed to be enough for the model to converge while keeping the computation time relatively low and avoiding overfitting. The convergence criterion was to monitor the reduction in the reconstruction error and to stop when it stabilised. A similar criterion is used for example in [27].

As data was separated in the clustering phase only by using the *TVD*-KPI, detecting anomalies on that KPI would point out mostly the same abnormal days as the DBSCAN, and therefore, would not add much value. Instead, anomaly detection is performed using all the KPIs besides *TVD* to detect anomalies rooting from network errors.

A day is considered an anomaly if its reconstruction error is higher than the 95th percentile of the reconstruction errors of the training set. In the majority of the work where AEs are used, the threshold is equal to the highest reconstruction error of normal data, but it should also be considered that these works utilise a labelled dataset. In this work however, the data was mixed in the beginning, and the normal set was only acquired by DBSCAN clustering. The division between normal and abnormal days is based on data density and not the ground truth, and therefore might not be definite. For this reason, the 95th percentile is used to account for any high reconstruction errors that could be present within the normal days, while not being low enough to cause misclassifications of normal days leading into false alarms.

Instead of computing a single reconstruction error for a day as is done in the anomaly detection, the reconstruction error can also be computed individually for each KPI of the anomalous days. The KPI-wise errors are then used to help determine the root cause of the anomaly in the sense that the KPIs with the highest errors are the potential root causes. In this work, if an individual KPI has an MSE larger than the mean of all the KPI-wise errors, it is considered a potential root cause. The rationale

behind the definition of a root cause is based on trying to find root causes that increase the total MSE. Also, setting definite thresholds for the MSE could be problematic, considering the variance between different sectors.

Consequently, the set of KPIs does not include all KPIs in the data, but only the potential root causes of the anomaly. For instance, most throughput KPIs (*USER_DL_TP, USER_UL_TP, CELL_DL_TP, CELL_UL_TP)* can be considered more of a consequence of *TVD* rather than the cause. As the analysis is done on the sector level and data is hourly aggregated, Timing advance (*TIME_ADV*) only tells the average distance of users to the base station, and therefore would not reveal very interesting root causes. The number if the KPIs used in the system is eight. The list of KPIs and the part of the system where they are used is presented in Table 5.

Table 5: Final set of KPIs and where they are used in.

Abbreviation	KPI	Used in
TVD	Traffic volume data.	DBSCAN
USERS	Average number of users.	LSTM AE
PRB_DL	PRB DL average usage rate.	LSTM AE
PRB_UL	PRB UL average usage rate.	LSTM AE
HSR_INTRA	HSR intra frequency.	LSTM AE
HSR_INTER	HSR inter frequency.	LSTM AE
CQI	Average CQI.	LSTM AE
CA_TP	Carrier aggregation throughput.	LSTM AE

IV. RESULTS

This chapter presents and discusses the outcomes achieved by applying the developed system on the dataset featured in Section III.A. This chapter is divided into six subsections. It starts with the specific results from using DBSCAN to differentiate normal from abnormal days. Following this, the effectiveness of LSTM AE in anomaly detection and RCA is presented, along with a section devoted to validating these outcomes. An analysis of the overall findings, including the number and nature of the finally identi-fied anomalies is then provided. The final part critically considers the choices made, as well as alternative approaches.

A. Distinguishing normal from abnormal days

The performance of machine learning models is usually measured by comparing the real values of the target variable to the ones generated by the model. However, the original dataset does not have any labels or ground truth indicating the real anomalies, therefore comparing results in terms of precision, F1-score or any other commonly used metric is not possible. Instead, some idea of the correctness of the DBSCAN clustering can be gained by comparing results with the correlations of each day to the average day. Here the TVD of each weekday (Mon – Fri) is compared to the average TVD of all weekdays, while weekends (Sat – Sun) are compared to the average weekend. The correlation between an individual day and the average day is high if they have a similar shape. As the average traffic patterns resemble the ones identified in the existing literature and the correlations can reveal the days that do not follow the pattern, this is a way to verify the validity of the clustering.

So, why are the correlations not used to separate the abnormal days from the normal days in the first place? Instead, why is the computationally heavier DBSCAN used? The reason for this is that different sectors behave differently, and the daily pattern might vary more in certain sectors than in others. This makes it difficult to set a threshold for the correlation of a day to be considered normal. By using the density based DBSCAN, there is no need to set any thresholds.

The clustering of days of sector 1 demonstrates promising results, in the sense that the abnormal days from the clustering also have the lowest correlations. This was also further noticed in the rest of the data for different BSs and sectors, in the cases when the dominant traffic patterns were clear and resembled the ones mentioned in Section III.B. On the other hand, when the dominant traffic pattern was not clear and the individual days displayed varied patterns, the clustering and verifying its results were more difficult. This resulted in clustering outcomes where the normal cluster included days with lower correlation than the abnormal cluster. This would suggest that either the clustering, the verification method, or most likely both, may not be optimal for all situations.

The fact that data is sampled by the hour means that there can be significant variation in the amount of traffic even between consecutive hours. Also, if the traffic pattern of the sector is mixed and there is not one dominant pattern, it could mean that the average day is not descriptive of all days. These are challenging factors for DBSCAN but could be overcome by more accurate and uniform data.

Nevertheless, the DBSCAN clustering shows promise in being applied to an anomaly detection system for unlabelled data, not as the final anomaly detector, but rather a preprocessing step. After performing this step, the user should have at least some idea of what the normal data could be, or where the anomalies are more likely to be located in.

B. Anomaly detection

Let us inspect the results of the LSTM AE in sector 1 of base station VIANA_SUL_LTE_NVC103B3. Out of 11 abnormal days, there is a single abnormal day, which has a reconstruction error much larger than the other abnormalities. Since the AE is able to reconstruct the other abnormal days with an error even lower than some of the normal days, these days could be the ones where the abnormal traffic pattern is due to natural human behaviour. The abnormal day, with an MSE over 10, can be considered an anomaly deriving from a problem in the network.

The abnormal day was also classified as an anomaly by the LSTM AE, which calculated the 95th percentile of training MSEs to be 1.76. This means that out of all the 11 abnormal days from DBSCAN, only one is an actual anomaly deriving from unexpected behaviour in the network and is detected due to unpredictability in the remaining KPIs, not TVD.

In sector 1, there is no clear distinction in the MSEs among training, test and abnormal sets. A similar behaviour was inspected also in different sectors of other BSs. In addition to the anomaly, there are multiple days in the normal training set which also have a large reconstruction error. One of them even almost as large as the anomaly. This would strongly indicate that there exist anomalies also within the normal days and that

network errors do not always display abnormal traffic volume pattern. These anomalies do not display abnormal behaviour in the TVD-KPI but seem to show abnormal behaviour in some other KPIs regardless.

When it comes to the performance of the LSTM AE, considering that most of the days, be it in the train, test or even in the abnormal set, are within the 95 percentile of the training MSEs (1.76), it seems that the LSTM AE is able to quite accurately reconstruct the data. What is considered a small MSE depends on the data scale, but in this case, after standard scaling, most of the KPIs have a range of values of about 10 units. Comparing the MSE of 1.76 to that range, the error is not particularly high. This would indicate that the LSTM AE is learning the patterns of the majority of the days and is able to reconstruct them. In addition, the few days with significantly larger MSEs indicate that the LSTM AE is not able to reconstruct all the days, making it suitable for anomaly detection, as it is sensitive to data patterns that differ significantly from the typical or expected patterns.

C. *Root cause analysis*

As described in Section III.C.4), the LSTM AE can be used for RCA by inspecting the reconstruction errors computed individually for each KPI. In the case of the previously detected anomaly from sector 1, it seems that the anomaly was caused by a problem in handovers, as was indicated by the high MSE of both inter frequency (HSR_INTER) as well as intra frequency (HSR_INTRA) handover success rates.

When comparing the scaled HSR_INTER pattern of all abnormal days, it can be noted that all days are quite similar to each other except for one. A single abnormal day has lots of very small values during the night and morning, and is in fact, the day that was detected as an anomaly. Because the KPI in question is HSR_INTER, which has a range of 0 – 100 before scaling, it seems like the success rates of inter frequency handovers have been zero across the period of 00:00 – 10:00. This is a serious network problem, as it means that calls or data sessions have not successfully been transferred from one cell to another from a different frequency. This leads to dropped calls as well as disruptions in data streams. Another option is that handovers were fine, but some of the counters that keep track of the KPI have been inoperable. Nevertheless, in both scenarios the network operator should be alerted so that adequate actions can be taken to fix the issue.

When inspecting the reconstructed HSR_INTERs of all abnormal days, it evident that the AE was able to reconstruct most of the HSR_INTER curves to closely match the original ones, but the reconstructed anomalous day is very different from the original. This is most likely due to the training set not containing enough samples like the anomaly that would have allowed the LSTM AE to learn the reconstruction of similar data. Instead, it seems that the AE tried applying a similar shape to the anomaly as to the other abnormal days. This led to a large reconstruction error and the detection of the anomaly.

V. CONCLUSIONS AND FUTURE WORK

The primary objective of this thesis was to develop an automatic machine learning-based system able to detect anomalies and conduct RCA in unlabelled mobile network data. The thesis also aimed to answer questions about general anomaly detection and RCA in mobile networks, performing them using only unlabelled data as well as the effect of 5G network deployment to the mobile anomaly detection and RCA landscape.

This thesis provided a comprehensive overview of 4G and 5G networks, QoE, machine learning methods, SONs and the principles of anomaly detection and RCA, along with a discussion of related work. A system was developed to detect anomalies and to determine their root causes in unlabelled 4G network data. The system (described in Chapter III) could work as a part of the self-healing function of a self-organising network to reduce the problems and downtime of the network which is expected to lead to an increase in the objective QoE of the network users.

Data analysis was performed to find different features of the data, including similarities between different cells within sectors, differences between sectors of the same BS as well as weekend and weekday traffic patterns. These features were used to determine some of the design choices of the system. The system preprocesses the data, including splitting it into days, removing days with missing data points and scaling the KPIs. It uses DBSCAN clustering to acquire a set of normal days from both weekdays and weekends of each sector, which are then used for training of a sector-wise LSTM AE model which can be used for day-wise anomaly detection and RCA.

The results of applying the system to the available dataset and the manually created artificial data revealed promising outcomes. Specifically, DBSCAN demonstrated proficiency in being able to differentiate between normal and abnormal network traffic patterns of 24-hour time series sequences. Meanwhile, LSTM AE showcased its ability to detect anomalous days of various types based on their reconstruction errors. In addition, it could distinguish the individual KPIs mostly contributing to the anomalies, paving way for automatic RCA.

To summarise the answers for the research questions presented in Chapter I, anomaly detection and RCA can be performed in various ways depending on the nature of the data. Traditionally used statistical and threshold-based methods often relying on expert input may be enough to address the needs of simple and smaller networks, but as networks increase in size and complexity, there is a growing trend towards machine learning-based models such as random forests and DNNs.

With mobile network data often being unlabelled, the set of suitable machine-learning methods is narrowed down. Due to their independence of labelled data, unsupervised clustering and AE-based models have been at the centre of many anomaly detectors throughout the recent years. Both can be used also for RCA, but clustering-based models tend to depend on expert input during the setup of the model.

It is important to note that although this work uses a 4G dataset, the proposed model is not limited to any specific network technology or a set of KPIs (apart from *TVD*). The

model works purely from the basis of network KPIs and therefore, in principle, could be applied to any network that generates them, including 5G. However, in the upcoming years, the increasing complexity and the number of features while moving towards non-standalone 5G networks, will set new requirements for mobile network anomaly detection. With different application scenarios, frequency bands and network slicing, a wide range of normal behaviours can be expected even within the same geographical region. Therefore, either multiple different models monitoring different parts of the network or dynamic models capable of adjusting their behaviour are needed. Fortunately, the decentralised MEC architecture of 5G will bring computing power to the edges of the network, therefore enabling real-time data processing and robust anomaly detection close to where data is generated. Real-time anomaly detection is crucial especially for URLLC applications, underlining the potential necessity for proactive anomaly detection. In addition, the sheer volume of data projected to be transferred by 5G networks makes detecting, and more importantly, reacting to every single anomaly infeasible. For this reason, not all the anomalies should be causes of alarms, but network operators should rather concentrate on narrowing down to a set of specific anomalies. Such as the ones directly impacting QoE of the users, and for which RCA can be performed and the root causes addressed.

Further aspirations about improvements of the developed model will focus on experimenting with more granular data containing more KPIs and at least some anomaly and root cause labels. This will support the fine-tuning of the model parameters, the development of a more reactive model, outputting of more specific root causes as well as verifying the results more reliably. Using GANs to generate new artificial data to increase the amount of labelled data or to characterise more versatile network conditions is also an interesting prospect. Another important topic that was not included in this thesis is the computational requirements of the proposed model. In future, the computational overhead of the developed model should be compared to other models to determine whether it is feasible to deploy the model in a distributed network architecture, for example in 5G small cells, or whether a centralised setting is more suitable. Finally, deployment of the model in an operating mobile network and evaluating its impact to the operator and users, for example in terms of operational efficiency and QoE, would serve as the ultimate test.

To conclude, machine learning-based anomaly detection and root cause analysis are complex topics with use cases not only in mobile networks but in other fields as well. The design choices of the models are affected by the quality of the data, whether it is labelled or not, specific goals and the desired accuracy as well as available domain knowledge. The development of automatic anomaly detectors and root cause analysis solutions will be crucial activities moving forward towards fully self-organising mobile networks in 5G and beyond.

REFERENCES

[1] P. V. Klaine, M. A. Imran, O. Onireti and R. D. Souza, "A Survey of Machine Learning Techniques Applied to Self Organizing Cellular Networks," *IEEE Communications Surveys & Tutorials,* vol. 19, pp. 2392-2431, 2017.

[2] W. Zhang, R. Ford, J. Cho, C. J. Zhang, Y. Zhang and D. Raychaudhuri, "Self-organizing cellular radio access network with deep learning," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, 2019.

[3] H. D. Trinh, E. Zeydan, L. Giupponi and P. Dini, "Detecting Mobile Traffic Anomalies Through Physical Control Channel Fingerprinting: A Deep Semi-Supervised Approach," *IEEE Access,* vol. 7, pp. 152187-152201, 2019.

[4] Y. Yuan, J. Yang, R. Duan, I. Chih-Lin and J. Huang, "Anomaly Detection and Root Cause Analysis Enabled by Artificial Intelligence," in *2020 IEEE Globecom Workshops (GC Wkshps)*, Taipei, Taiwan, 2020.

[5] K.-F. Chen, C.-H. Lin and T.-S. Lee, "Deep Learning-Based Multi-Fault Diagnosis for Self-Organizing Networks," in *ICC 2021 - IEEE International Conference on Communications*, Montreal, Canada, 2021.

[6] H. Mfula and J. K. Nurminen, "Adaptive Root Cause Analysis for Self-Healing in 5G Networks," in *2017 International Conference on High Performance Computing & Simulation (HPCS)*, Genoa, Italy, 2017.

[7] C. V. Murudkar and R. D. Gitlin, "QoE-driven anomaly detection in self-organizing mobile networks using machine learning," in *2019 Wireless Telecommunications Symposium (WTS)*, New York, USA, 2019.

[8] G. Y. Z. Q. Z. Y. Chen G, “A Novel Cellular Network Traffic Prediction Algorithm Based on Graph Convolution Neural Networks and Long Short-Term Memory through Extraction of Spatial-Temporal Characteristics.,” *Processes,* vol. 11, no. 8, 2023.

[9] A. Gómez-Andrades, P. Muñoz, I. Serrano and R. Barco, "Automatic Root Cause Analysis for LTE Networks Based on Unsupervised Techniques," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 4, pp. 2369-2386, 2016.

[10] T. Zhang, K. Zhu and D. Niyato, "A Generative Adversarial Learning-Based Approach for Cell Outage Detection in Self-Organizing Cellular Networks,"

IEEE Wireless Communications Letters, vol. 9, no. 2, pp. 171-174, 2020.

[11] K. Choi, J. Yi, C. Park and S. Yoon, "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines," *IEEE Access,* vol. 9, pp. 120043-120065, 2021.

[12] Z. Chen, D. Chen, X. Zhang, Z. Yuan and X. Cheng, "Learning graph structures with transformer for multivariate time series," *IEEE internet of things journal,* vol. 9, no. 12, pp. 9179-9189, 2022.

[13] B. Hussain, Q. Du, S. Zhang, A. Imran and M. A. Imran, "Mobile Edge Computing-Based Data-Driven Deep Learning Framework for Anomaly Detection," *IEEE Access,* vol. 7, pp. 137656-137667, 2019.

[14] U. S. Hashmi, A. Rudrapatna, Z. Zhao, M. Rozwadowski, J. Kang, R. Wuppalapati and A. Imran, "Towards Real-Time User QoE Assessment via Machine Learning on LTE Network Data," in *IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, Honolulu, USA, 2019.

[15] M. S. Parwez, D. B. Rawat and M. Garuba, "Big Data Analytics for User-Activity Analysis and User-Anomaly Detection in Mobile Wireless Network," *IEEE Transactions on Industrial Informatics,* vol. 13, no. 4, pp. 2058-2065, 2017.

[16] K. Sultan, H. Ali and Z. Zhang, "Call Detail Records Driven Anomaly Detection and Traffic Prediction in Mobile Cellular Networks," *IEEE Access,* vol. 6, pp. 41728-41737, 2018.

[17] Y. Liu, Y. Mu, K. Chen, Y. Li and J. Guo, "Daily Activity Feature Selection in Smart Homes Based on Pearson Correlation Coefficient.," *Neural processing letters,* vol. 51, p. 1771–1787, 7 Jan 2020.

[18] F. Xu, Y. Li, H. Wang, P. Zhang and D. Jin, "Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment," *IEEE/ACM transactions on networking a joint publication of the IEEE Communications Society, the IEEE Computer Society, and the ACM with its Special Interest Group on Data Communication.,* vol. 25, no. 2, pp. 1147-1161, 2017.

[19] S. Learn, "Scikit Learn," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html. [Accessed 18 June 2023].

[20] F. Chollet, "Building autoencoders in Keras," The Keras blog, 14 May 2016. [Online]. Available: https://blog.keras.io/building-autoencoders-in-keras.html. [Accessed 18 June 2023].

[21] Scikit-learn, "sklearn.preprocessing.StandardScaler," Scikit-learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. [Accessed 22 Sep 2023].

[22] D. K. Kotary and S. J. Nanda, "A Distributed Neighbourhood DBSCAN Algorithm for Effective Data Clustering in Wireless Sensor Networks.," *Wireless Personal Communications,* vol. 121, p. 2545–2568, 2021.

[23] Q. Xianting and W. Pan, "A Density-Based Clustering Algorithm for High-Dimensional Data with Feature Selection," in *2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, Wuhan, China, 2016.

[24] K. K. Nisa, H. A. Andrianto and R. Mardhiyyah, "Hotspot clustering using DBSCAN algorithm and shiny web framework," in *2014 International Conference on Advanced Computer Science and Information System*, Jakarta, Indonesia, 2014.

[25] P. Le and W. Zuidema, "Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs," *arXiv preprint,* 2016.

[26] Scikit-learn, "Metrics and scoring: quantifying the quality of predictions," Scikit-learn, [Online]. Available: Metrics and scoring: quantifying the quality of predictions. [Accessed 22 September 2023].

[27] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, Halifax, Canada, 2017.