MABASR - A Robust Wireless Interface Selection Policy for Heterogeneous Vehicular Networks

GEORGI NIKOLOV1, MICHAEL KUHN2, ALAN MCGIBNEY1, and BERND-LUDWIG WENNING1

1Georgi Nikolov, Alan McGibney, and Bernd-Ludwig Wenning are with the Nimbus Centre, Munster Technological University, Cork T12 Y275, Ireland.
2Department of Electrical Engineering and Information Technology, Darmstadt University of Applied Sciences, 64295 Darmstadt, Germany.
Corresponding author: Bernd-Ludwig Wenning (e-mail: berndludwig.wenning@mtu.ie).

InSecTT (https://www.insectt.eu/) has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey. The document reflects only the author’s view and the European Commission is not responsible for any use that may be made of the information it contains.

ABSTRACT Connectivity is rapidly becoming a core feature of modern vehicles to enable the provision of intelligent services that promote safer transport networks, real-time traffic infotainment systems and remote asset monitoring. As such, a reliable communications back-bone is required to connect vehicles that deliver real-time data to smart services deployed at cloud or edge architecture tiers. Hence, reliable uplink connectivity becomes a necessity. Next-generation vehicles will be equipped with multiple wireless interfaces, and require robust mechanisms for reliable and efficient management of such communication interfaces. In this context, the contribution of this article is a learning based approach for interface selection known as the Multi-Armed Bandit Adaptive Similarity-based Regressor (MABASR). MABASR takes advantage of the underlying linear relationship between channel quality parameters and uplink data rate to realise a robust interface selection policy. It is shown how this approach outperforms algorithms developed in prior work, achieving up to two orders of magnitude lower standard deviation of the obtained reward when trained on different data sets. Thus, higher reliability and less dependency on the structure of the training data are achieved. The approach is tested in mobile, static, and artificial static scenarios where severe network congestion is simulated. All data sets used for the evaluation are made publicly available.

INDEX TERMS Wireless Interface Selection, Vehicle-to-Infrastructure (V2I), Reinforcement Learning (RL)

I. INTRODUCTION

A STABLE, reliable end-to-end connection delivered via Vehicle-to-Infrastructure (V2I) communications backbone is needed to support advanced services for next-generation vehicles. These include remote monitoring, maintenance, navigation, infotainment, or value-added passenger services in the transportation sector. Many of these services not only require a reliable downlink connection to the vehicle, but also a reliable uplink to support the communication of critical sensor data and/or streaming services to edge and/or cloud services and systems. To achieve the desired reliability, vehicles may be equipped with multiple communication interfaces (e.g. Universal Mobile Telecommunications System (UMTS), Long Term Evolution (LTE), or 5G New Radio (NR)). Furthermore, informed intelligent decisions need to be made on which interface to use given the current context of operation and service requirements. One approach to achieve this is to provide an agent that learns the current state of the environment by performing actions (i.e. selecting a different interface) based on a specified policy to obtain rewards. In the context of the presented research, the notion of reward is described as the data rate which the agent achieves through its actions. The main objective of the agent is to maximize its reward. Prior work focused on designing an agent that has the ability to select an interface based on regression predictions [1]. The policy is based on the Linear Upper Confidence Bound (LinUCB) algorithm first considered in [2]. By enabling the agent to take unexpected network congestion into account, it was possible to enhance the interface selection policy. This resulted in the
development of the Modified LinUCB (ModLinUCB). The contribution of this paper is the introduction of MABASR, which provides further improvements: MABASR features an improved bandit inspired interface selection policy to search for linear structures in the data. It is demonstrated that by modifying the model knowledge criterion, the algorithm can be made more resilient against the influence of poor training data quality (structure) on overall performance.

The remainder of the paper is structured as follows: An overview of the related work is given in Section II, followed by a description of the challenges in Section III. Section IV gives an overview of the measurement and simulation methodology. Compression and correlation analysis of the measured data is carried out in Section V. The improved interface selection technique is presented in Section VI. Evaluation and results comparison are discussed in Section VII. Section VIII concludes the paper and provides an outlook to future work.

II. RELATED WORK
A. INTERFACE SELECTION APPROACHES
With the rise of technologies such as UMTS, Wireless Local Area Network (WLAN), LTE and 5G, it is not unusual for modern devices to be equipped with multiple radio interfaces. This has led to a significant number of academic and industry works targeting the interface selection problem.

For example, the authors of [3] present a dynamic decision mechanism for multihomed mobile hosts. The concept allows for policies for interface selection to be edited and exchanged by nodes and operators. A concept based on analytic hierarchy process and Grey relational analysis for the switching between UMTS and WLAN is proposed in [4]. For the selection of the most favorable network, a multi-attribute decision making algorithm is suggested in [5]. The issue with these approaches is that they require a lot of parameters (e.g. packet delay, packet jitter, etc.) in order to take a decision on which interface to use. Accessing these parameters is not always possible in practical settings as interface drivers often do not provide them. Similar issues arise in [6] where the authors present an extended attractor selection model (EASM). The requirement of broadcasting the cellular activity to each terminal makes this approach almost unusable in practice as it adds significant overhead. In contrast, the proposed solution only uses parameters widely accessible from every mobile device as context for the interface selection.

The authors of [7] propose a cloud-based network selection scheme where vehicles, assisted by a database implemented in the cloud, execute the interface selection task. One major disadvantage of this approach is the dependency of the system on a single centralized entity and its availability, creating a potential single point of failure.

In [8], an evolutionary game approach as well as a reinforcement learning methodology are proposed to solve the interface selection problem. While the first requires a centralized controller to function, the reinforcement learning algorithm does not have such a requirement and can be implemented into each device individually. The method is based on Q-learning and learns the estimated payoff from every interface available as time progresses. However, the strategy lacks the ability to deal with unexpected network congestion explicitly. It relies on a random variable which determines whether exploration or exploitation should be performed. In previous work [1] it was demonstrated that such algorithms are inferior to context-based approaches.

An Interface Manager (IM) for Unmanned Aerial Vehicles (UAVs) is proposed in [9]. The framework utilizes Decision Trees (DT) to predict which interface currently has most optimal transmission conditions. The parameters used for the prediction are quantity of bytes received, quantity of bytes lost, network throughput, the received signal strength indication (RSSI), and the signal to noise ratio (SNR). While the solution performs well, and does have low computational complexity, it needs frequent data transmissions to perform good estimation. While this is no issue in a multi-UAV scenario where multiple UAVs communicate with each other on a regular basis, a train or a car traveling in a rural environment can spend up to several hours without issuing any data transmissions. In [10] the authors propose a Deep Reinforcement Learning (DRL) interface selection scheme to optimize the transmission energy consumption, monetary cost, and distortion, while maintaining application Quality of Service (QoS) requirements in a smart health system. While DRL has certainly achieved great success in various tasks in recent years, this technique is not really applicable for our use case:

- As with other interface selection approaches mentioned previously, this scheme requires extensive knowledge of parameters that are not readily available in Commercial Off-The-Shelf (COST) devices (e.g. amounts of bits to be sent, data compression ratio, channel impulse response).

- Compared to a hospital environment, which rarely changes drastically, the channel characteristics for vehicle transportation systems can often change significantly and abruptly (often in the order of seconds). A DRL system usually utilizes a neural network in its core. A well known fact is that neural networks are slow learners in online setups and cannot adapt quickly to changing conditions [11].

B. MULTI-ARMED BANDIT ALGORITHMS
The potential applications of multi-armed bandit algorithms in the field of wireless networks have been discussed in [12] where an overview of the use of this type of algorithm is presented, also in the context of channel selection in a dynamic environment.

The authors of [13] develop a multiplayer multi-armed bandit game model to solve the multi-user channel selection problem. Their selection strategy consists of a calibrated forecasting approach and no-regret bandit learning. The same authors also developed an adversarial multiplayer multi-armed bandit game where players (agents) try to maximize
their average reward which is a function of the channel quality [14].

The authors of [15] present a bandit algorithm in a periodic setting suitable for modelling fixed wireless networks where certain patterns within certain periods emerge. However, in our case, the agent is highly mobile and this setting is not applicable.

A bandit algorithm using a reward function with nonlinear context is presented in [16]. The algorithm uses the same context for multiple arms. However, the implementation of networks based on different technologies can differ greatly, as is the case, for example, between LTE and UMTS. They are very different especially in the physical layer, which makes the two networks uncorrelated. This means the channel quality indicators for UMTS will not be applicable for the LTE data rate estimation and vice versa. Therefore, the aforementioned algorithm is not suitable for this interface selection problem.

C. MODLINUCB

The Linear Upper Confidence Bound (LinUCB) is proposed in [2]. It is a multi-armed bandit algorithm for context-enhanced web page recommendations. Every page is modelled as an arm. The strategy is to maximize total user clicks. Therefore, article selection is also adapted according to the click feedback and contextual information about the article and the user. The estimated mean number of user clicks per web page is modeled as a linear function of this information using ridge regression.

In [1], this mechanism was extended further, resulting in the Modified LinUCB (ModLinUCB) approach. In ModLinUCB, the applicability of the LinUCB approach to the interface selection problem was shown, where the arms are modelling two mobile channels. The reward is predicted from channel quality parameters using a ridge regression. In particular, Reference Signal Received Power (RSRP) and Reference Signal Received Quality (RSRQ) were identified as the most suitable channel quality parameters for LTE and RSSI and E_c/I_0 as the most suitable parameters for UMTS. All the parameters are scaled to [0, 1] intervals before they are used in the ridge regression. As more data becomes available, the confidence bound in LinUCB converges to a small number. In ModLinUCB, this was modified by adding a new parameter referred to as “additional confidence” to the interface selection policy which enables it to react to unexpected network congestion. If the actual throughput is much lower than expected, this parameter, which uses a Rectified Linear Unit (ReLU) function, modifies the confidence related to the affected interface to encourage the algorithm to explore the other interface and switch if a sufficient data rate is expected on that other interface. The inputs for the function are the threshold parameter γ and the difference between predicted and real uplink throughput, labelled as δ.

The additional confidence is described by the equation:

\[
C_i^t = ReLU(\delta_{1-i}, \gamma) = \begin{cases} 
\delta_{1-i} - \gamma & \text{for } \delta_{1-i} > \gamma \\
0 & \text{otherwise}
\end{cases}
\]

where \(i\) ∈ available interfaces. In the context of this research there are only two available interfaces to choose from: UMTS, represented by the number zero in the algorithm and LTE, which is represented by the number one. Hence, \(i\) can only be equal to either zero or one.

D. ASR

In [17], the Adaptive Similarity-based Regressor (ASR) was introduced, a combination of an online support vector regression with a novel similarity-based training and forgetting strategy. In ASR, new samples are first compressed and scaled using Principal Component Analysis (PCA). Next, they are compared to existing samples to identify the most similar sample. Depending on how similar the new sample is to the existing, one of two training approaches is used:

- If the similarity exceeds a given threshold, the most similar old sample will be forgotten and replaced with the new one.
- If the similarity remains below the given threshold, an iterative approach is used where for each iteration, one old sample is temporarily replaced by the new sample. Then the output is predicted for each of the samples in the temporary training set, and the mean absolute error between the predicted and the actual output is determined. The iteration which produced the lowest mean absolute error is the one that is kept, i.e. the old sample that was replaced by the new one in this iteration is forgotten and replaced permanently.

Using this approach, ASR ensures that relevant model knowledge is not forgotten, as would be the case if simply the oldest samples were discarded. This prevents a loss of information if many similar samples are received, as would be the case if stationary conditions apply for some time.

III. PROBLEM DESCRIPTION

Although the ModLinUCB policy is able to react to unexpected network congestion and outperforms "follow the expert" policies in such scenarios [1], it suffers from one substantial drawback: If there is an underlying relationship between the channel quality parameters and the data rate, the ModLinUCB would not be able to take advantage of it. The way the algorithm is building its confidence about its prediction of interface \(i\) depends on the number of samples the algorithm has seen from this interface. The more samples the algorithm sees, the less it adjusts its parameters. This, combined with the fact that ModLinUCB learns data sequentially, creates a strong dependency on the structure of the training data. For example, if a training data set for an interface \(i\) contains 2000 samples, but the first 100 samples\(^1\) consist of

\(^1\)The numbers given here are for better understanding of the example. They do not represent the real size of the data sets obtained through this research.
a constant mobile network signal for a given interface $i$, then the algorithm would perform its main parameter adjustment at the beginning of the training set. Even if the rest of the training data contains useful information about the interface $i$, the algorithm would pay less attention to it, as it would have already converged and would consider the interface $i$ to have a constant network signal and constant data rate. On the other hand, if the first 100 samples of the interface are diverse and provide a more accurate understanding of the reward model, then the ModLinUCB would learn it and perform better.

To put this into perspective, the following hypothetical, albeit very realistic V2I scenario is constructed: The algorithm is running on a gateway on a train. The gateway is responsible for delivering IoT sensor information and video surveillance from the train to cloud services. For this purpose, the gateway manages two cellular connections (e.g., LTE and UMTS), but has no prior knowledge regarding the relationship between the channel quality parameters and the obtainable uplink data rate for each of them. In this online configuration, the ModLinUCB would be using its policy to learn the model of every cellular connection by sampling one sample at a time. As would be expected, the models, which are going to be learned by the algorithm during the initial exploration phase, will be strongly dependent on whether the train is sitting still at the train station (i.e., constant signal data), or moving on the track (i.e., diverse signal data). If the initial exploration phase occurs in stationary conditions, the model will only build on very limited knowledge and will not be able to capture and represent a mobility situation properly afterwards.

To address these shortcomings, one can take advantage of underlying patterns present in the data. That way, in the case of the train scenario, the solution should be less dependent on which sequential data (i.e., constant or dynamic signal data) is first fed to it during the training phase. The solution should not rely on the train to start moving in order to get dynamic signal data, as it will be able to ignore data points which do not fit a certain data pattern.

IV. MEASUREMENT AND SIMULATION METHODOLOGY

In order to investigate what kind of patterns exist between channel quality parameters and uplink rate, data which has already been acquired for the evaluation of the ModLinUCB [1] was utilised. The process of obtaining the data is described as follows: Four separate measurements in a mobile scenario and one measurement in a static scenario have been conducted. The first, second, and third measurement took place in a car, driving through the urban environment of Cork city centre, an area in the south of Ireland, on three different days with a different route for every measurement. The fourth measurement was done on a train, traveling between Cork and Mallow through a mostly rural environment.

The user equipment (UE) used was a gateway with two Sierra Wireless MC7455 modules, one with a UMTS subscriber identity module (SIM) and the other with an LTE SIM from the same operator. Generating user datagram protocol (UDP) traffic with Iperf [18], both mobile interfaces of the device were fully utilized with data rates as well as channel parameters being recorded. For the LTE link, received signal strength indicator (RSSI), signal-to-noise ratio (SNR), reference signal received power (RSRP), and reference signal received quality (RSRQ) were collected. For UMTS, RSSI, $E_c/I_0$ and received signal code power (RSCP) were acquired. The volume of the acquired measurements is as follows: First car measurement: 400 samples, second car measurement: 350 samples, third car measurement: 380 samples and for the train measurement: 200 samples. Due to hardware constraints the sample time was chosen to be five seconds.

The UE was on board a moving vehicle, passing through different cells. For each measurement, a different route was chosen. In this way it was ensured that the different data sets were taken under different conditions in order to test how robust the estimation works. A fifth static measurement was conducted in the main canteen area of a college campus (Cork Institute of Technology (CIT), Ireland) during a peak period of activity (lunch-time). The goal was to capture the uplink behaviour of both networks under heavy loads which are typical during this time of the day when a lot of users are using interactive services such as chat applications or mobile games on their devices. This measurement campaign produced a data set with the size of 1300 samples.

The results have shown that the uplink data rate of the LTE network can drop substantially without any indication provided by the measured channel quality parameters. To investigate these findings in more detail, a sample set with an artificial extension of the network performance degradation was created as illustrated in Section VII. The data rate was set to a reduced level between sample 190 and 300, using the output of a pseudo random generator ranging from 4 MBit/s to 6 Mbit/s to simulate prolonged severe network congestion. This simulation technique has already been used to successfully evaluate the performance of the ModLinUCB algorithm [1], therefore, to ensure fair comparison, it is applied here also.

V. COMPRESSION AND CORRELATION ANALYSIS

With the analysis of the obtained data sets, this section aims to provide a justification why the input of the data rate prediction (the ASR) can be compressed and thus make the algorithm more efficient. Furthermore, this section shows that there is a certain linear relation between the transformed input data and the uplink rate, which can be used as prior knowledge by an interface selection policy.

A. COMPRESSION THROUGH PCA

Principal Component Analysis (PCA) is a widely used and established compression method that provides reliability in its results and is therefore chosen as compression method for this research. PCA is particularly useful for cases where

2 All data sets used in the course of this paper can be found in [19].
a strong correlation between the input features is present:

Table 1. Correlation matrix for the data set recorded on the commercial Irish LTE network.

|        | RSRP | RSRQ | SNR | RSSI | Rate |
|--------|------|------|-----|------|------|
| RSRP   | 1    | 0.56 | 0.82| 0.95 | 0.81 |
| RSRQ   | 0.56 | 1    | 0.85| 0.49 | 0.54 |
| SNR    | 0.82 | 0.85 | 1   | 0.71 | 0.67 |
| RSSI   | 0.95 | 0.49 | 0.74| 1    | 0.74 |
| Rate   | 0.81 | 0.54 | 0.67| 0.74 | 1    |

Table 2. Correlation matrix for the data set recorded on the commercial Irish UMTS network.

|        | ECIO | RSSI | RSCP | RATE |
|--------|------|------|------|------|
| ECIO   | 1    | 0.74 | 0.53 | 0.74 |
| RSSI   | 0.74 | 1    | 0.80 | 0.72 |
| RSCP   | 0.53 | 0.80 | 1    | 0.56 |
| RATE   | 0.74 | 0.72 | 0.56 | 1    |

As can be seen from Table 1 and 2, all parameters in the correlation matrix for both LTE and UMTS networks exhibit moderate to strong correlation ($r > 0.5$). This means that there is a lot of redundancy in the input and data can be compressed. The correlation matrix results from a concatenated data set which consists of all car measurements, as well as the train measurement. PCA typically works best with data that has zero mean and unit variance. However, scaling the data in this manner is not feasible as the mean and the variance of the input features can vary significantly depending on the network, mobility scenario, etc. A more practical approach is to scale all the input features to $[0, 1]$, since minimum and maximum for every channel quality indicator are defined in the 3GPP Standards [20, 21], and are used by every mobile network operator. Also the output (data rate) is scaled internally to $[0, 1]$.

### B. PEARSON CORRELATION ANALYSIS

After transforming the channel quality parameters of the LTE and UMTS networks into principal components, the results are presented in the following two tables:

Table 3. Principal components LTE network measurement.

|        | $\sigma^2$ proportion | $r$ with Data Rate |
|--------|------------------------|--------------------|
| Comp 1 | 0.8                    | 0.78               |
| Comp 2 | 0.166                  | 0.17               |
| Comp 3 | 0.026                  | 0.02               |
| Comp 4 | 0.008                  | -0.22              |

$\sigma^2$ indicates the proportion of variance described by every principal component, and $r$ is the Pearson correlation between the component and the data rate. Both tables show that Component 1, which has the largest proportion of $\sigma^2$, also has a strong linear correlation to the data rate. This indicates an underlying linear relationship between principal components and uplink data, which can be used as prior knowledge for the development of an improved interface selection policy, i.e. the policy can embed the identification of such a linear relationship in the data set and extract it by learning only samples (sequentially fed) in the training data set that increase the linear correlation. This way such policy can overcome the training data dependency problem outlined in Section III.

Moreover, as part of the work conducted in [22], data across five network operators from two countries were transformed from raw network parameters into principal components. A moderate to strong positive linear correlation in the range from 0.51 to 0.9 between the first principal component and the uplink data rate was discovered for all mobile networks.

### VI. MABASR

The outcomes of above analysis can be exploited by creating an interface selection policy that will maximize the linear correlation between input and output while learning new samples, i.e. data that does not comply with the policy’s goal can be ignored.

In this context the MABASR presents the following improvements over the ModLinUCB: The ridge regression is replaced by the ASR. The former is a linear model, and fails to capture the relationship between the channel quality parameters and the achievable uplink data rate to the same degree as the ASR. This is demonstrated through the results in [17]. Replacing the estimation algorithm also requires a new formula for the calculation of the confidence parameter (note that the calculation for the additional confidence remains unchanged). Unlike the ridge regression, where the parameters are updated without keeping any history of the learned samples, the nature of the ASR algorithm requires to keep all of its learned samples [23].

Although this process requires more memory, it can be exploited, so that a new confidence calculation is developed, based on the Pearson correlation coefficient $r_i$ between the input samples and the learned uplink data rate for a given interface $i$. One goal of the new policy is that the agent explores interfaces for which lower correlation between the input and output is observed. This can be done by making the confidence parameter $C_i$ somewhat inversely proportional to the correlation. This means that low correlation would result in low prediction confidence, which results in a large value for the confidence parameter $C_i$. As the algorithm has the goal of maximizing a predefined reward function, it will tend to prefer exploring interfaces with larger values of $C_i$.

3The correlation matrices are done on a data set which has a moving average filter applied to it. This process is described in Section VII.
way, it will build up knowledge about all available interfaces during the initial phase of the algorithm.

The next challenge is how such relation can be expressed mathematically. One way is to select a function that will take the correlation as input and produce a very large number, if it is close to zero. If the correlation is close to one the function’s result should be close to zero. A good candidate that fulfills these criteria is the logarithm function, more specifically the negative logarithm \(-\lg\) in the range \((0, 1]\). This function will take \(r\) as an argument and will deliver a number representing the confidence of the model (underlined in Algorithm 1). It is acknowledged that there may be other functions which could provide a better model for confidence. However, their further investigation is beyond the scope of the work presented in this paper and should be explored further as part of future work.

The new confidence is directly related to the correlation of the samples in the ASR responsible for predicting the data rate for the given interface. In the case of the previously described railway station scenario, the confidence will remain very low (hence high \(C_i\)) even if the algorithm has already seen many samples from the interface. If these samples are all similar to each other or even the same, the correlation would remain low and so would the confidence. This will lead to the parameter \(C_i\) having large values for interfaces with low correlation and the policy would tend to select these interfaces in order to increase the correlation and the model knowledge.

In the beginning when the ASR has learned very few samples, the correlation for every interface could be zero or even sometimes less than zero. As shown in Tables 3 and 4 the correlation between the first principal component (used by ASR for prediction) and the obtained uplink data rate was discovered to be strictly positive. Therefore, such negative correlations are attributed to the fact that sometimes the noise in smaller sets of data can create the illusion of negative correlation which is not really present. As the logarithm is not defined at these values, it cannot accept them, since errors will be produced. Furthermore, initial tests of the concept showed that sometimes the correlation could become unrealistically large \((r > 0.999)\). Because of this reason, if \(r \leq 0\) or \(r \geq 0.999\), hence \(r_i \in \text{invalid value}\), it will be set to a very small positive number \(\epsilon\). That way the algorithm will know if the model knowledge of the ASR is poor. Invalid values of \(r\) occurred only when the ASR’s number of learned samples was small (typically from one to four).

Consequently, the confidence equation (also underlined in Algorithm 1) was further modified so that the number of learned samples \(n_i\) per interface \(i\) plays a role in the decision of the algorithm in the beginning:

\[
C_i = -\lg(r_i) - n_i. \tag{2}
\]

If the ASR responsible for predicting the data rate for a certain interface has learned the maximum allowed samples \(N_i\), then other interfaces with less learned samples should be preferred by the policy. To achieve this behaviour, the number of trained samples per interface \(n_i\) is subtracted from its confidence parameter \(C_i\). When the ASRs for every interface data rate prediction have learned their maximum number \(N_i\), the same number is subtracted from every interface’s confidence. Because the interface selection is done on the basis of a comparison between confidence values of the interfaces, this parameter stops playing a role in the decision process once it has the same value for every interface.

This research considers two managed interfaces - a primary LTE connection, backed by UMTS. In order to ensure that the algorithm only learns samples that contribute to its model knowledge, a new training function is added to the ASR that ensures that a sample is learned only if it increases the Pearson correlation \(r_i\) between input and output (i.e., the overall quality of the model knowledge). In previous work, it was found that the ASR predicts UMTS uplink data rates with higher accuracy when two principal components are used [17]. However, only the first principal component is used when calculating the input-output correlation. This is a configuration that results in good performance and simplified implementation at the same time. Applying these modifications ensures that the algorithm will search and find the underlying linear structure in the data, as the confidence depends on the relationship between the learned samples and not on their number alone.

Network congestion is a dynamic process that changes over time. For example, if a train leaves one train station and arrives at another, previous knowledge about the network congestion will not be applicable. Even in the case when the train is standing at the platform, network conditions can vary dramatically, due to network participants abruptly joining or leaving the network. Therefore, as with the ModLinUCB, the MABASR also needs to periodically flush old information relating to network congestion. The mechanism used for this is identical to the one used in the ModLinUCB policy: A counter is tuned that will clear the old entries of the additional confidence parameter \(C’\) by setting the parameter \(\delta = 0\) after a certain predefined value \(\beta\) is reached. To ensure fair comparison between the two algorithms, the value of \(\beta\) is kept identical to the one used in ModLinUCB and set to 10. Further reasoning behind the choice of this value can be found in [1]. Note that the value is not time dependent but count dependent. The counter will reset the errors after 10 samples regardless of the sample period.

VII. RESULTS AND DISCUSSION

This section provides an overview on how the proposed MABASR algorithm compares to ModLinUCB. A detailed overview of the measured data sets used for the evaluation of the algorithms has already been provided in Section IV. To measure the generalization ability of each algorithm, tests are structured as follows:

- **For the mobility scenario**: Training is performed on each of the data sets measured from a car followed by evaluation on a concatenated data set consisting of all car measurements except the one used for training.
Algorithm 1: The MABASR policy

Initialize ASR, IPCA instance for every interface, \( \beta, \gamma, \delta \)

while true do

    for interface \( i \) \( \in \) available interfaces do
    
    Obtain a channel quality parameter vector \( x_i \)
    Train IPCA\(_i\), with the new sample
    Compress sample \( x_i \) into \( u_i \) using IPCA\(_i\)
    ASR\(_i\) predicts mean data rate \( R_{pred,i} \) from \( u_i \):
    
    if \( r_i \) \( \in \) invalid value then
        \( r_i = \epsilon \)
        Calculate confidence parameter \( C_i \):
        \( C_i = -\log(r_i) - n_i \)
    
    if counter \( == \) \( \beta \) then
        \( \delta \) set for all interfaces to zero
        Calculate additional confidence \( C_i' \):
        \( C_i' = RReLU(\delta_1 - i; \gamma) \)
        Add \( C_i' \) to \( C_i \):
        \( C_i'' = C_i + C_i' \)
    
    \( p_{value,i} = R_{pred,i} + C_i'' \)
    
    if counter \( == \) \( \beta \) then
        \( \beta \) set counter to zero
    
    else
        increment counter by one
        Choose the interface \( i \) with the highest \( p_{value} \)
        Obtain the error \( \delta_i \) between \( R_{pred,i} \) and the observed data rate \( R_{actual,i} \) for the chosen interface \( i \)
    
    if Sample pair \((u_i, R_{actual,i})\) increases \( r_i \) then
        Train the ASR\(_i\) model for the selected interface
        Obtain the new correlation \( r_i \) for the selected interface after training

A mean value of the uplink data rates achieved across every permutation is calculated.

- **For the static scenario:** Training is performed again on each data set obtained from the mobility measurement. Evaluation is done on the canteen measurements. The same approach as described above is used for calculating the mean achieved data rate of each algorithm. No data sets recorded in the canteen are used for training, since no significant variation in the channel quality parameters is observed as the measurement equipment was stationary.

- **For the artificial static scenario:** The static measurement is modified to have artificial network congestion as described in Section IV. Exactly the same evaluation methodology is used as for the static scenario.

A mean achieved reward \( \mu \) is calculated from the achieved rewards obtained from the three scenarios. For the static and the static artificial scenario, a standard deviation \( \sigma \) is also calculated in order to measure the reliability of the algorithm. A smaller \( \sigma \) indicates consistent performance of the algorithm resulting in less dependency on the training data. In the case of the mobility scenario, no \( \sigma \) is calculated as the evaluation sets are different.

During the fine tuning process, the threshold \( \gamma \), above which the additional confidence parameter is nonzero, was found to deliver the best results for \( \gamma = 2 \) MBit/s in case of the MABASR, but with \( \gamma = 5 \) MBit/s the ModLinUCB exhibits optimal performance\(^4\). Therefore, both algorithms are tested and compared utilising these optimal values.

A moving average filter with window size of ten was applied to the data to reduce the amount of random switching between LTE and UMTS due to noise. Reducing the noise also improves prediction performance. Random fluctuations in the data rate can only degrade the Quality of Service (QoS) of an interface selection policy, forcing it to switch back and forth between the two interfaces. In the case of a running TCP connection, such switching will introduce a huge delay in the communication, because connections have to rebuild every time when there is a switch between LTE and UMTS. Therefore, sporadic changes in the interfaces have to be reduced to a minimum, while still being able to detect degradation in the data rate.

Combining the moving average filter with the MABASR improves its performance even further, as it filters out noise, and makes the discovery of the underlying linear structure of the data easier. Coincidentally, applying the moving average filter to the MABASR has another positive effect on the performance of the algorithm. Namely, it substantially reduces the amount of false decisions performed by the policy. In the context of this work, a false decision is defined as the algorithm selecting the interface with actual lower data rate resulting from a false prediction of the ASR. Table 5 presents a summary of the achieved improvements.

| Scenario         | Averaging | No Averaging |
|------------------|-----------|--------------|
| Car Set 1 – Mobility | 9.6%      | 17.6%        |
| Car Set 2 – Mobility | 7.4%      | 18.3%        |
| Car Set 3 – Mobility | 7.5%      | 19.3%        |
| Car Set 1 – Static   | 0.6%      | 4.9%         |
| Car Set 2 – Static   | 0.6%      | 4.6%         |
| Car Set 3 – Static   | 0.0%      | 4.9%         |
| Train set – Static   | 0.6%      | 4.3%         |
| Car Set 1 – Artificial Static | 2%      | 9.1%         |
| Car Set 2 – Artificial Static | 1.3% | 5.6%        |
| Car Set 3 – Artificial Static | 1.4%   | 6%          |
| Train set – Artificial Static | 1.8% | 5.3%        |

From Table 6 it can be seen that the best configuration of MABASR at \( \gamma = 2 \) MBit/s outperforms the best configuration of ModLinUCB at \( \gamma = 5 \) MBit/s in every key test (emphasised in bold in Table 6). It achieves better performance.

\(^4\)The parameter \( \gamma \) could also be auto-tuned using an approach such as Bayesian Optimization [24]. However, in this case, the problem complexity makes it suitable to allow for manual adjustment of this parameter as it requires very little time while resulting in sufficient performance.
average reward $\mu$ for the mobility scenario - 25.409 MBit/s versus 24.996 MBit/s. It has two orders of magnitude lower standard deviation when on the static scenario, and achieves better reward and lower standard deviation $\sigma$ on the artificial static scenario (20.0375 and 0.013 MBit/s compared to 18.092 and 3.006 MBit/s for the ModLinUCB).

These results further confirm the theory that ModLinUCB is much more dependent on the training data than MABASR. An example of that is comparing the results of the artificial static scenario where the ModLinUCB performs well when trained on the second car measurement (cf. Table 6), achieving an average reward of 20.068 MBit/s, but completely fails to learn the reward model when trained on the train measurement (cf. Figure 1). The achieved reward is then only 13.66 MBit/s. During the training phase, the algorithm never switches to LTE and does not learn its properties (cf. Figure 3).

Table 6. Achieved reward for the ModLinUCB and the MABASR evaluated with two $\gamma$ values. Rewards and $\gamma$ are in MBit/s.

| Scenario               | MABASR $\gamma = 2$ | MABASR $\gamma = 5$ | ModLinUCB $\gamma = 2$ | ModLinUCB $\gamma = 5$ |
|------------------------|---------------------|---------------------|-------------------------|-------------------------|
| Car Set 1 – Mobility   | 25.742              | 25.94               | 25.53                   | 25.65                   |
| Car Set 2 – Mobility   | 26.552              | 26.501              | 25.22                   | 26.622                  |
| Car Set 3 – Mobility   | 23.934              | 23.877              | 21.915                  | 22.717                  |
| Car Set 1 – Static     | 20.841              | 20.841              | 20.372                  | 20.994                  |
| Car Set 2 – Static     | 20.841              | 20.841              | 18.764                  | 20.948                  |
| Car Set 3 – Static     | 20.841              | 20.841              | 17.629                  | 19.644                  |
| Train set – Static     | 20.841              | 20.841              | 13.66                   | 13.66                   |
| Car Set 1 – Artificial Static | 20.032          | 19.812              | 19.667                  | 19.847                  |
| Car Set 2 – Artificial Static | 20.032          | 20.037              | 18.097                  | 20.088                  |
| Car Set 3 – Artificial Static | 20.057          | 20.024              | 17.124                  | 18.795                  |
| Train set – Artificial Static | 20.058          | 20.012              | 13.66                   | 13.66                   |
| Mobility $\mu$         | 20.841              | 20.841              | 17.606                  | 18.059                  |
| Static $\sigma$        | 0                   | 0                   | 2.801                   | 3.381                   |
| Artificial Static $\mu$ | 20.037              | 19.978              | 17.137                  | 18.092                  |
| Artificial Static $\sigma$ | 0.013             | 0.111               | 2.543                   | 3.006                   |

Figure 1. ModLinUCB trained on train measurement - Artificial static scenario, moving average filter applied.

Therefore, during testing time, the algorithm selects only the UMTS interface. A possible explanation for the behaviour can be the filtering of noise in the data. The algorithm will first select one interface and explore it until it is confident enough about its reward model, or the reward drops. The randomness of the noise and the sudden change in data rates can cause the algorithm to randomly switch to a different interface when both interfaces have similar channel quality (cf. Figure 2). Thus, the ModLinUCB can explore both reward models by chance. If the noise is removed, the algorithm will switch to a different interface only if the data rate is degrading for a longer period of time.

Figure 2. ModLinUCB training process on train measurement - No moving average filter applied.

When the ModLinUCB is trained on the averaged train set, this phenomenon does not occur (cf. Figure 3). Hence, the policy never learns about the reward model of LTE and thinks its reward is always equal to 0 MBit/s. It will switch to it, only in the case of severe degradation of the UMTS link. In contrast, the MABASR is able to explore both interfaces due to its improved policy and learns a better representation of the reward model (cf. Figure 4).

This allows it to consistently achieve a higher uplink data rate (cf. Figure 5 and Table 5).

Moreover, the MABASR is compared to the REXP 3 Algorithm [25] - a bandit approach developed for non-stationary rewards. This bandit approach is used as a baseline comparison in previous work [1]. The results clearly illustrate the superiority of our approach relative to a generic bandit framework. When trained on the first car measurement, the MABASR achieves 25.742 MBit/s, while the
The REXP 3 algorithm achieved only 21.647 MBit/s. The results of the static and artificial static scenarios are similar, where the MABASR achieves 20.841 MBit/s and 20.032 MBit/s, compared to REXP 3 which only reached 18.518 MBit/s and 17.911 MBit/s respectively.

**Figure 4.** MABASR training process on train measurement - moving average filter applied.

**Figure 5.** MABASR trained on train measurement - Artificial static scenario, moving average filter applied.

**VIII. CONCLUSION AND FUTURE WORK**

This paper presented MABASR - a robust algorithm for wireless interface selection. The MABASR aims to maximize the linear correlation between input and output and forces the algorithm to search for a linear model in the data it is learning. Data that does not fit this linear model is simply ignored. The policy is developed based on the results presented in Section V, from which a moderate to strong linear correlation between the data rate and the first principal component was observed. Although this relationship cannot be transferred to every single cellular network in existence, the results clearly show that embedding this knowledge in the objective of the policy as prior information leads to significant increase in the reliability of the MABASR interface selection algorithm, resulting in an improved robust performance across different structures of the different training sets. Future work will include tests on the 5G NR network, expanding the MABASR’s policy to select between UMTS, LTE, and 5G.

**ACKNOWLEDGMENT**

The authors of this work would like to thank Klas Telecom for providing the necessary equipment and thus making this research possible.

**References**

[1] G. Nikolov, M. Kuhn, and B.-L. Wenning. A contextual bandit approach to the interface selection problem. In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pages 1707–1714, 2019.

[2] Lihong Li, Wei Chu, John Langford, and Robert Schapire. A contextual-bandit approach to personalized news article recommendation. Raleigh NC USA, April 26-30, 2010.

[3] Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, and Jaakko Laine. Dynamic network interface selection in multimoded mobile hosts. Proceedings of the 36th Hawaii International Conference on System Sciences, March 2002.

[4] Qingyang Song and Abbas Jamalipour. Network selection in an integrated wireless lan and umts environment using mathematical modeling and computing techniques. IEEE Wireless Communications, June 2005.

[5] Farooq Bari and Victor C. M. Leung. Automated network selection in a heterogeneous wireless network environment. IEEE Network, January/February 2007.

[6] Daxin Tian, Jianzhao Zhou, Yunpeng Wang, Yingrong Lu, Haiying Xia, and Zhenguo Yi. A dynamic and self-adaptive network selection method for multimode communications in heterogeneous vehicular telematics. IEEE Transactions on Intelligent Transportation Systems, December 2015.

[7] Ke Xu, Kuang-Ching Wang, Rahul Amin, Jim Martin, and Ryan Izard Student Member. A fast cloud-based network selection scheme using coalition formation games in vehicular networks. IEEE Transactions on Vehicular Technology, November 2015.

[8] Dusit Niyato and Ekram Hossain. Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach. IEEE Transactions on Vehicular Technology, May 2009.

[9] Laura Michaeila Batista Ribeiro, Ivan Muller, and Leandro Buss Becker. Communication interface manager for improving performance of heterogeneous uav networks. Sensors, 21(13), 2021.

[10] Zina Chikrbine, Alaa Awad Abbadi, Amr Mohamed, Aiman Erbad, and Mohsen Guizani. Deep reinforcement learning for network selection over heterogeneous health systems. IEEE Transactions on Network Science and Engineering, 9(1):258–270, 2022.

[11] Neural Networks: A Comprehensive Foundation, chapter Dynamically Driven Recurrent Networks. Pearson Education, Inc., 2005.

[12] Chunxiao Jiang, Haijun Zhang, Yong Ren, Zhu Han, Kwang-Cheng Chen, and Lajos Hanzo. Machine learning paradigms for next generation wireless networks. IEEE Wireless Communications, April 2018.

[13] Setareh Maghsudi and Slawomir Stanczak. Channel selection for network-assisted d2d communication via no-regret bandit learning with calibrated forecasting. IEEE Transactions on Wireless Communications, March 2015.

[14] Setareh Maghsudi and Slawomir Stanczak. Joint channel selection and power control in infrastructureless wireless networks: A multiplayer multi-armed bandit framework. IEEE Transactions on Vehicular Technology, April 2015.

[15] Oh Shunhao, Meetoo Appavoo Anuja, and Gilbert Seth. Periodic bandits and wireless network selection. 46th International Colloquium on Automata, Languages and Programming (ICALP 2019), April 2019.

[16] Pranav Sakulkar and Bhaskar Krishnamachari. Stochastic contextual bandits with known reward functions. CoRR, abs/1605.00176, 2016.

[17] G. Nikolov, M. Kuhn, A. McGibney, and B.-L. Wenning. ASR - adaptive similarity-based regressor for uplink data rate estimation in mobile networks. IEEE Journal on Selected Areas in Communications, 38(10):2284–2294, 2020.

[18] Iperf. https://iperf.fr/. [Online; last accessed 15.03.21].

[19] Mobile Network Measurements. https://github.com/Georgi917/Mobile-Network-Measurements. [Online; last accessed 15.03.21].

---

\[\text{The REXP 3 algorithm is a purely online approach. Moreover, it had to periodically restart itself, to account for the non-stationarity of the data. As such, there is not much sense to pre-train the algorithm, and thus the algorithm learns directly from the test set} \]
[20] 3GPP TS 136-133 evolved universal terrestrial radio access (E-UTRA); requirements for support of radio resource management (version 8.2.0 release 8).
[21] 3GPP TS 36.214 evolved universal terrestrial radio access (E-UTRA); physical layer measurements (version 10.1.0 release 10).
[22] Georgi Nikolov. Adaptive Uplink Data Rate Prediction and Network Selection in a Vehicular Mobile Channel. PhD thesis, Munster Technological University, 2021. Section 7.2.3 - Principal Component Analysis.
[23] Junshui Ma, James Theiler, and Simon Perkins. Accurate online support vector regression. Neural Computation 15(11):2683-703, November 2003.
[24] https://github.com/fmfn/BayesianOptimization. [Online; last accessed 27.01.2020].
[25] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Optimal exploration-exploitation in a multi-armed-bandit problem with non-stationary rewards. May 15, 2014.

GEORGI NIKOLOV Georgi Nikolov received his M.Sc.(2018) in electrical engineering in communication technology from the University of Applied Sciences Darmstadt (h_da), Germany. He was awarded a PhD in 2021 from the Munster Technological University. His research focused on applying Machine Learning techniques to uplink data rate prediction in mobile networks enhancing Vehicle-to-Infrastructure (V2I) communication. He is currently working as Data/Machine Learning Engineer at h.a.I.m. Elektronik GmbH.

MICHAEL KUHN Michael Kuhn received the Dipl.-Ing. degree in electrical engineering in communication technology from the Technical University Darmstadt (TUD), Darmstadt, Germany, in 1998 and the Doctorate degree from Saarland University, Saarbruecken, Germany in 2002. He is a professor of mobile communication with University of Applied Sciences (h_da), Darmstadt, Germany. Before he joined h_da in 2006, he was head of the R&D department with Wavecom Electronic AG (Switzerland) where he was in charge of hardware and software developments for software defined radios. His research focuses on cooperative wireless communication, AI in communication systems, and software defined radios.

ALAN MCGIBNEY received his PhD in Electronic Engineering from the Cork Institute of Technology, Ireland in 2008. His research interests are in the area of Internet of Things (IoT) and Cyber Physical Systems. He is currently a group lead for IoT systems and user interaction at the Nimbus Centre, Munster Technological University with a particular focus in the areas of distributed optimisation, tool and service development, software architectures and digital platforms. He leads and coordinates a number of EU funded and national projects in this area and is an SFI funded investigator as part for the CONFIRM centre for smart manufacturing contributing to industrial IoT and wireless communications research activities.

BERND-LUDWIG WENNING Bernd-Ludwig Wenning holds Dipl.-Ing. and Dr.-Ing. degrees in Electrical Engineering and Information Technology from the University of Bremen, Germany. His research interests are mobile and wireless communication networks, protocols and Quality of Service. From 2002 to 2012, he was a researcher at the Communication Networks group, University of Bremen, contributing to several nationally (German) funded projects. In 2012, he joined the Nimbus Centre at Cork Institute of Technology (now Munster Technological University), Ireland. Since then, he has been involved in several nationally (Irish) and EU funded projects. He is the Nimbus Centre’s principal investigator in the ECSEL funded project InSecTT. Throughout his research career, he has authored or co-authored more than 50 publications.