applied sciences

MDPI

Article

5G Technology: ML Hyperparameter Tuning Analysis for Subcarrier Spacing Prediction Model

Faris Syahmi Samidi, Nurul Asyikin Mohamed Radzi *, Kaiyisah Hanis Mohd Azmi, Norazizah Mohd Aripin and Nayli Adriana Azhar

The Institute of Power Engineering (IPE), Universiti Tenaga Nasional, Kajang 43000, Malaysia
* Correspondence: asyikin@uniten.edu.my

Abstract: Resource optimisation is critical because 5G is intended to be a major enabler and a leading infrastructure provider in the information and communication technology sector by supporting a wide range of upcoming services with varying requirements. Therefore, system improvisation techniques, such as machine learning (ML) and deep learning, must be applied to make the model customisable. Moreover, improvisation allows the prediction system to generate the most accurate outcomes and valuable insights from data whilst enabling effective decisions. In this study, we first provide a literature study on the applications of ML and a summary of the hyperparameters influencing the prediction capabilities of the ML models for the communication system. We demonstrate the behaviour of four ML models: k nearest neighbour, classification and regression trees, random forest and support vector machine. Then, we observe and elaborate on the suitable hyperparameter values for each model based on the accuracy in prediction performance. Based on our observation, the optimal hyperparameter setting for ML models is essential because it directly impacts the model's performance. Therefore, understanding how the ML models are expected to respond to the system utilised is critical.

Keywords: machine learning; hyperparameter tuning; 5G; resource allocation; resource management

Citation: Samidi, F.S.; Mohamed Radzi, N.A.; Mohd Azmi, K.H.; Mohd Aripin, N.; Azhar, N.A. 5G Technology: ML Hyperparameter Tuning Analysis for Subcarrier Spacing Prediction Model. *Appl. Sci.* **2022**, *12*, 8271. https://doi.org/10.3390/app12168271

Academic Editors: Jenhui Chen, Lei Wang and Zhiqun Hu

Received: 21 May 2022
Accepted: 18 July 2022
Published: 18 August 2022

1. Introduction

Today, machine learning (ML) is frequently used in intelligent systems to provide these systems with the ability to learn from problem-specific training data and to automate the construction of analytical models and the process of solving associated tasks [1]. ML techniques are widely utilised in advertising, recommendation systems, computer vision, natural language processing and user behaviour analytics. Given the wide range of applications and requirements, various ML methods have been developed to provide solutions for user needs and overcome various issues and challenges efficiently. Incorporating artificial intelligence into network infrastructure is one of the industry strategies for addressing the inherent complexities associated with the 5G network deployment. The expanding usage of wireless technology is concerned to possibly congest the airwaves that our devices utilise to communicate; thus, ML replaces conventional wireless technologies to decrease power consumption substantially and improve network performance [2]. In general, developing an efficient ML model is a complicated and time-consuming process that entails selecting the best algorithm and achieving the best model architecture by optimising its hyperparameters [3].

Hyperparameters are the variables that determine the network structure and how the network is trained. Each ML model has a unique set of hyperparameters, which require different settings when trained on different datasets. Hyperparameter tuning is a process of finding the best settings for the ML algorithm. The goal is to find the parameters that produce high accuracy and low error rate. In ML, many algorithms with different features

exist. These algorithms include logistic regression, linear regression, support vector machines (SVMs) and random forests (RFs). Each algorithm has its own optimisation method that can be used to optimise the model parameters. For example, in logistic regression a parameter is called lambda value can be optimised using multiple methods such as the gradient descent method, modified Newton–Raphson method or stochastic gradient descent method. Other intelligent algorithms, such as stacked denoising autoencoders, convolutional networks and classifiers based on sophisticated feature extraction techniques, have between 10 and 50 hyperparameters. The settings of hyperparameters depends on how the experimenter chooses to parametrise the model and how many hyperparameters are fixed at a reasonable default [4]. Published results are difficult to repeat and expand because of the difficulties in calibrating these models, resulting in challenges in optimising a loss function across a graph-structured configuration space, known as hyperparameter optimisation. A hyperparameter optimisation method must optimise across discrete and continuous variables whilst choosing which variables to optimise [4]. Therefore, frequently optimising the prediction model necessitates a thorough understanding of both ML methods and appropriate hyperparameter optimisation approaches. Although numerous automated optimisation strategies exist, their benefits and downsides vary when applied to different challenges, particularly those concerned with 5G technology. Thus, we anticipate that learning the model's behaviour and characteristics allows the excellent optimisation of processing power, time and accuracy [3].

In 5G, a base station must process numerology assignment and resource allocation for each user according to the user's feedback. ML offers an advantage in providing rapid adaptation to assign the allocated time, space and frequency domain in the spectrum and satisfy the user's Quality of Service (QoS) requirement [5]. A base station also needs to decide on the optional waveform processing techniques, including windowing, filtering, guard utilisation, cyclic prefix utilisation and different operating subcarrier spacings. This resource optimisation is critical because 5G is intended to be a major enabler and a leading infrastructure provider in the information and communication technology sector by supporting a wide range of upcoming services with varying requirements. Given the network's rising complexity and the introduction of innovative use cases, such as autonomous cars, industrial automation, virtual reality, e-health and various intelligent applications, ML is projected to be critical in making the 5G vision a reality [6,7].

Given the relation between 5G and ML, this article focuses on one of the most critical parameters in the 5G system, which is the selection of numerology or 'subcarrier spacing'. Moreover, 5G numerology is a method of assigning the subcarriers in 5G frequency bands to different service types. These subcarriers are used for transmitting data and controlling information, as well as for synchronisation and signalling. Furthermore, 5G subcarrier spacing works by dividing a single carrier into multiple carriers, which are then transmitted at different frequencies. This way, we can achieve high speeds with little power consumption and an enhanced coverage area. We can also increase system capacity through this method, which helps to reduce the interference between neighbouring cells. Therefore, 5G networks must be able to choose the optimal subcarrier spacing that can be realised using ML to achieve the benefits. This finding emphasises the importance of acknowledging the ML model characteristics implemented into the wireless system.

The contributions of this study are as follows:

- A concise review of the recent ML algorithms used in the 5G network;
- Development of ML model training in RStudio to observe ML model accuracy based on the hyperparameter changes;
- A description of the working principles and hyperparameter tuning of four ML models: k nearest neighbour (KNN), classification and regression trees (CART), RF and SVM;
- Performance evaluation of the effects of hyperparameter tuning for all four ML models studied;
- An observation of the hyperparameter tuning effects on the cross-validation accuracy for the subcarrier spacing prediction models in the 5G system;

- The effects of the number of neighbours, maximum tree depths, number of randomly selected predictors and cost values on the accuracy of KNN, CART, RF and SVM, respectively.

The remainder of the paper is structured as follows. Section 2 presents the related work on ML in 5G networks. Section 3 discusses the possibility of tuning the ML model hyperparameters to predict the subcarrier spacing in the 5G network. Section 4 focuses on the performance comparison between the tuned ML models. The acronyms and notations used in this paper are shown in Table 1.

Table 1. Acronyms and definition.

Acronyms	Definition
5G	Fifth Generation
CART	Classification and Regression Trees
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
EDRP	Edit Distance with Real Penalty
FML	Fast Machine Learning
KNN	K-Nearest Neighbour
LCS	Longest Common Subsequence
MDT	Minimization of Drive Test
ML	Machine Learning
QoS	Quality of Service
RF	Random Forest
RNN	Recurrent Neural Network
SCOC	Small Cell Outage Detection
SDN	Software-Defined Networks
SVM	Support Vector Machine
VANET	Vehicular Ad Hoc Network

2. Literature Study

Since the introduction of the first generation (1G) in 1979, mobile communication systems have revolutionised various sectors, including social and education. However, as the number of sectors and applications that rely on these communication systems expand, the systems face challenges in meeting user demands regarding resources, particularly in QoS provisions and efficient spectrum utilisation. Therefore, planning resources is crucial and can be simplified using an ML algorithm to automate the challenging process. A study proposed by Meng et al. concentrates on base station beam selection using semionline learning, which considers the traffic environmental variations for locating beams [8]. Moreover, Tayyaba et al. [9] proposed a resource allocation policy framework using a deep neural network (DNN) for dynamic vehicular networks. The proposed framework was implemented in software-defined networks (SDNs), which demonstrated the performance of 5G system data centres and carrier networks. However, the DNN cannot compete in terms of time and precision in resource distribution.

ML and deep learning (DL) can also be used to detect irregularities in the Internet of Things (IoT) applications. The results of combining these strategies are better than those of using only one DL algorithm. In [10], several deployment and analysis approaches of the intrusion detection system (IDS) were investigated using the IoT architecture. The most advanced DL models available include recurrent neural network (RNN), convolutional neural network (CNN) and a combination of the two, with RNN and CNN as the most sophisticated systems involving a large number of tuning parameters and a long time to obtain proper optimisation [11].

Moreover, processes such as data gathering, integration and forecasting are the key components of a framework for resource management and network optimisation to ensure a proper prediction model in 5G. However, problems and challenges, such as extracting con-

texts from online data and simulating proactive optimisation still exist [12]. Therefore, fast ML (FML) was proposed as a method for adapting 5G vehicular ad hoc network (VANET) base stations [13]. Based on the results, FML outperforms the benchmark algorithms in terms of performance and adaptability. The findings of the online bandit learning imply that context awareness is critical in 5G situations. Creating a hybrid system for tracking individuals whilst increasing total network capacity may be important in the future [14].

Moreover, ML techniques can be utilised to create a self-organising network, which can improve network performance, reduce network downtime and increase user experience over cellular networks. The Bayesian ML approach was proposed by Baz et al. [15] as a novel method for optimising the performance of SDNs. In this method, each node should detect the underlying stochastic process utilised by the controller to create the flow rules and predict the relevant rules for packets that do not meet the flow table. Furthermore, a small cell outage detection (SCOD) strategy was presented in [16], based on the partial key performance indicator (KPI) data from an extensive collection of Minimisation of Drive Test (MDT) reports. MDT reduction is an extensive collection of partial KPI data that can be utilised to compensate for outage users in a dynamic and dense deployment of the tiny cell environment learning technique.

However, resource limits in the existing edge ML architecture require self-organising network functions to prioritise inference workloads. In comparison, a different training process was performed in the central infrastructure without collaboration. This process involves the redundant transmission of massive training data from the edge to the central location, as well as a hyperparameter search with high space–time complexity, resulting in considerable signalling costs and delays. Farooq et al. aimed to address these issues by developing a system for edge intelligence. This system is a low-complexity and low-time framework that eliminates the transfer of huge training data from the edge to the central location and reduces the time required to discover optimal (or near-optimal) hyperparameters. The results suggest that the framework strikes an acceptable compromise between predicting the ideal hyperparameter configuration and minimising expenses. Managing this trade-off is essential in resource-constrained contexts, such as RAN nodes. The gain is also impacted by its operating frequency because of this framework's dynamic nature [17].

Isabona et al. summarised that many attempts were made in the literature to manage RF hyperparameter tuning challenges and improve predictive application performance. These attempts include the investigation on calculating the optimal number of RF trees in the literature and studies on how to best use the RF feature set size for robust regression analysis of distinct datasets. The researchers discovered that the ideal size is comparatively small if the dataset characteristics are associated and vice versa. The process of discovering and determining the optimum practicable values of hyperparameters for an ML model to achieve the intended, resulting modelling output is known as hyperparameter tuning or optimisation [18].

As we gain additional knowledge about ML technologies, we can efficiently add intelligence to the dynamic environment with unexpected network parameters. In a complicated time-varying environment, uncertainties can be avoided by employing parametric learning approaches that separately represent features based on the unique shape of training functions. ML algorithms may aid in identifying anomalies, faults and intrusions and in access control and authorisation [19]. The key works on ML approaches, along with their focus, contributions and main method, are listed chronologically in Table 2.

Table 2. Summary of literature study.

Author (Year)	Summary	Advantages	Limitations/Future Work
Meng et al. (2020) [8]	Three-dimensional semi-online learning problem of beam selection in a base station to cater vigorous traffic and environment change.	Acclimated to traffic by learning the association between the course of advent and the customary data.	Can be further improved by enabling smart entities.
Tayyaba et al. (2020) [9]	SDN-based vehicular networks for optimising resource allocations according to the changing demands and network dynamics in vehicular networks.	Proved the performance gain in the data centres and carrier networks by using SDN.	A performance deficiency shown by the DNN in terms of resource allocation time and accuracy.
Tahkkar et al. (2020) [10]	Discussed various IDS placement strategies and IDS analysis strategies in the IoT architecture.	Combined ML and DL techniques for detecting attacks in the IoT networks to ensure improved performance compared with the performance of individual DL algorithms.	Issues such as IDS administration, securing IDS communication, use of standardised datasets and building techniques for correlating alerts need to be addressed.
Ma et al. (2020) [12]	Proposed a framework involving data acquisition, integration and use of forecasting to drive resource management and network optimisation.	Made the enablers drive the proactive optimisation.	Challenges to complete the circle of making online data a reliable data source.
Fang et al. (2019) [19]	Introduced intelligence by exploring ML techniques to authenticate the complex time-varying environment under unknown network conditions and unpredictable dynamics, supporting radically new applications of 5G and beyond wireless networks.	Modelled the attributes independently by using the parametric learning methods on the basis of the specific form of training functions so that the uncertainties caused by the complex time-varying environment may be circumvented.	Utilisation of ML techniques for other security applications, such as anomaly/fault/intrusion detection, access control and authorisation, mainly because of their ability to provide continuous protection for legitimate communications in 5G and beyond networks.
Sim et al. (2018) [14]	Proposed FML for base station adaptability in 5G VANET.	Low complexity and a scalable online learning algorithm for mmWave base stations using FML.	Exploring a hybrid solution between tracking individual vehicles and increasing the overall network capacity.
Baz et al. (2018) [15]	Proposed a novel algorithm to improve the performance of SDNs by using the Bayesian ML.	Allowed each switch to infer the underlying stochastic process by which the controller generates the flow rules and to predict the suitable rules for those packets that have no match in the flow table.	The throughput of the standard OpenFlow protocol affecting the performance of the network from the observation of the decline rates of the throughput that is proportional to the traffic rates.
Qin et al. (2018) [16]	Proposed an SCOD algorithm using the ML approach based on partial KPI statistics that are a large-scale collection of MDT reports.	Fairly allocated resources to outage users for compensation, considering the dynamic and dense deployment of small cell environment learning approach based on partial KPI statistics.	Exploring from the domain of ML to create a self-organised network in the end-to-end intelligence of 5G networks.
Huang et al. (2017) [11]	Studied state-of-the-art DL models, including RNN, 3D CNN and a combination of CNN and RNN.	A multitask learning architecture using DL networks for mobile traffic forecasting to extract geographical and temporal traffic features.	Can be a solid benchmark for hybrid DL models.

Table 2 shows that considerable reliable research on 5G overcomes various issues and challenges using ML. However, ML requires computational resources to train the models and process the incoming data. Therefore, we can reduce the time-consuming and computationally expensive efforts whilst finding an optimal combination of hyperparameters by understanding the hyperparameter tuning for each prediction model. This hypothesis is supported by a previous study [20], which showed that the sensitivity of the models differs from that of their hyperparameters. Therefore, parameter tuning can enhance model performance.

3. Experiment Setup and ML Tuning Parameters

ML, a subset of the artificial intelligent technology that learns patterns from empirical data, has been used in classification, regression and control applications. Wireless systems can accommodate varying traffic loads and data requirements in future cellular communication technologies by implementing ML as a subcarrier spacing prediction for 5G systems. In addition, ML implementation can aid in efficient spectrum utilisation, enabling abundant resources to cooperate with the exponential increase in user demands. Figure 1 shows the fundamental process of ML algorithms in which a dataset is supplied to the ML training platform. In our case, the dataset consists of packet size, data rate, total data size and numerology used. An additional optimisation is necessary if the accuracy does not seem promising after the ML model is built. This procedure is repeated until the method's accuracy converges as intended. The trained ML algorithm is then further evaluated on a new dataset to ensure that the system continues to produce accurate results.

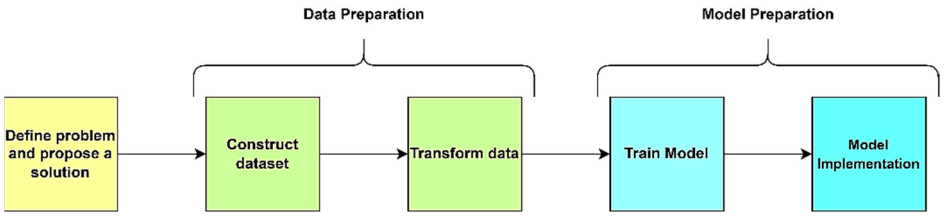

Figure 1. Fundamental process of machine learning implementation.

3.1. Setup Summary

Developing a rigorous test for assessing ML models is essential to ensure that the results obtained can be trusted and used to make selections amongst the tested models. Based on [21], the selection of ML models entails (1) selecting a hyperparameter metric to assess a model's skill, (2) establishing a baseline performance against which all model evaluations can be compared and (3) splitting the data into training and test sets using a resampling technique to simulate how the final model is used. Given the ML selection process, we can comprehensively assess the effects of hyperparameter tuning on the model's prediction accuracy using a single train-test split of the data to evaluate the model's performance quickly, as shown in Figure 1.

Four ML models, namely, KNN, CART, RF and SVM, are observed in our study because these models are the most common ML algorithms used for classifying. The grid search algorithm is used to iterate between the tuning value and observe the behaviour of the hyperparameter for each model. Grid search is a method for finding the best hyperparameters of Algorithm 1. It is used to find the parameters that optimise the objective function, thereby improving results. Grid search involves randomly varying each parameter and testing their effects on performance. The optimisation process can be repeated many times until convergence (i.e., when no further improvement occurs). This approach is a quick way to develop algorithms that can solve complex problems efficiently whilst minimising overfitting. Overall, the implementation and performance analysis of the four ML algorithms are conducted with the caret module in R using RStudio software and run on a laptop with the following specifications: Windows 10, Intel(R) Core(TM) i76500U CPU @ 2.50 GHz and 16 GB of RAM.

Algorithm 1 Observing the accuracy of the model based on its hyperparameter changes

```
function caret;
    Input: Model type KNN, CART, SVM, RF
    Output: Subcarrier spacing prediction
    validation index = createDataPartition(dataset, p = 0.80);
    validation = dataset[-validation index];
    set.seed();
    if model = CART then
        return tunegrid = expand.grid(maxdepth));
    end
    if model = KNN then
        return tunegrid = expand.grid(k));
    end
    if model = SVM then
        return tunegrid = expand.grid(sigma, C);
    end
    if model = RF then
        return tunegrid = expand.grid(mtry);
    end
    fit.model = train(predict, data=dataset, model = model, tunegrid)
    results = resamples(list(model);
    summary(results);
    write.csv([["results"]]);
```

3.2. Brief Summary of the Selected ML Model

When employing ML models, tuned parameters are vital in obtaining high-accuracy outcomes. Each classifier has unique tuning steps and customised parameters; therefore, the users must understand the utilised ML model. This section offers a brief description of the ML models, with a focus on their working principles and hyperparameters.

(a) K-Nearest Neighbours

KNN algorithm relies on classification and clustering as its foundational processes. Rule induction, segmentation, anomaly detection and visualisation are examples of high-level algorithms that use these two processes. The underlying distance metric considerably impacts the model outcome for the clustering and classification techniques. Euclidean distance, longest common subsequence and edit distance with a real penalty are some of the distance metrics used in time series classification and clustering applications, as presented in Figure 2 [22]. Setting the hyperparameter right is critical because of the ability of the hyperparameter to improve the distance measurement accuracy. The main hyperparameter value for KNN is the k value, which indicates the count of the nearest neighbours. Setting this parameter to the lowest feasible value whilst considering the measure's quality may minimise the calculation time for certain elastic measures [23]. In addition to its ease of implementation, KNN has the benefit of working well with vast amounts of noisy training data [24]. However, a major drawback is that for every new instance, all the distances from K neighbours must be computed; this step consumes much computing time and resources [25]. Therefore, the k values must be established appropriately for error rate reduction [26].

(b) Classification and Regression Trees (CART)

Predictive ML techniques, such as the decision tree approach, are also widely utilised for classification and regression. The CART model, a variation in the decision tree approach, is represented by a binary tree with split rules at each root node. The split condition is applied to the root node, and the decision procedure is sequentially performed for each subroot node. Each root node is considered to have a single input variable, 'x', representing the variable and any splits on it to visualise the aforementioned scenario well. At the leaves, 'y' is an output variable that predicts the output. The categorical prediction uses the choice of entropy, whereas the continuous prediction relies on the sum of square errors. This method simplifies ML for novice users by providing an easy-to-understand and visible model. It also needs minimal data preprocessing and can handle categorical

and numerical data types. However, the CART model is an unstable paradigm and it may occasionally result in a complicated tree structure that is not sufficiently generalised [27]. The complexity parameter (cp) is utilised as a tuning parameter in the CART model to increase its performance and solve the issues mentioned. It penalises the tree if it has an excessive number of splits; in most cases, the default setting is set at 0.01. In particular, small trees are produced with the increase in the cp value. Figure 3 depicts the final tree generated using the CART model.

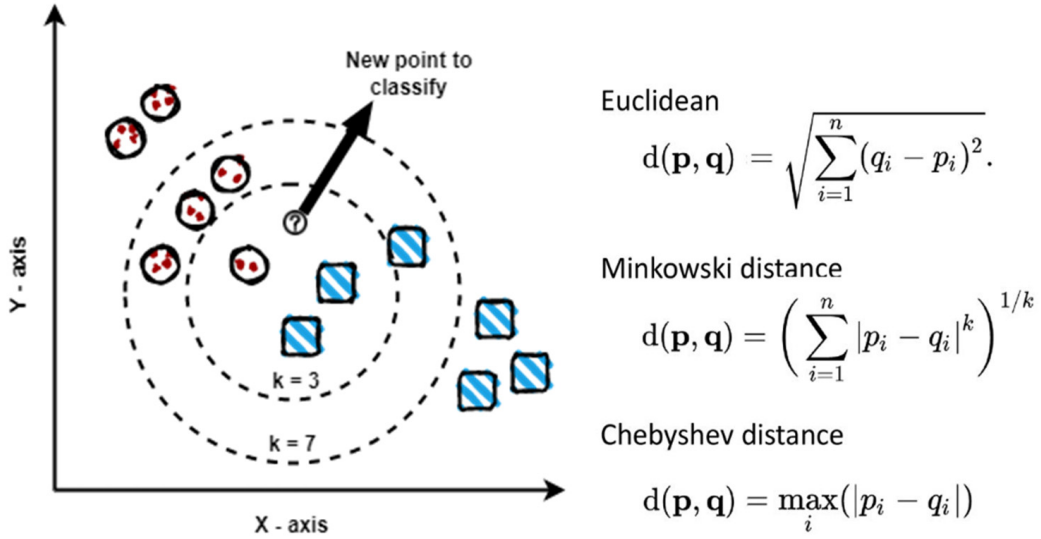

Figure 2. KNN model calculation and technique visualisation.

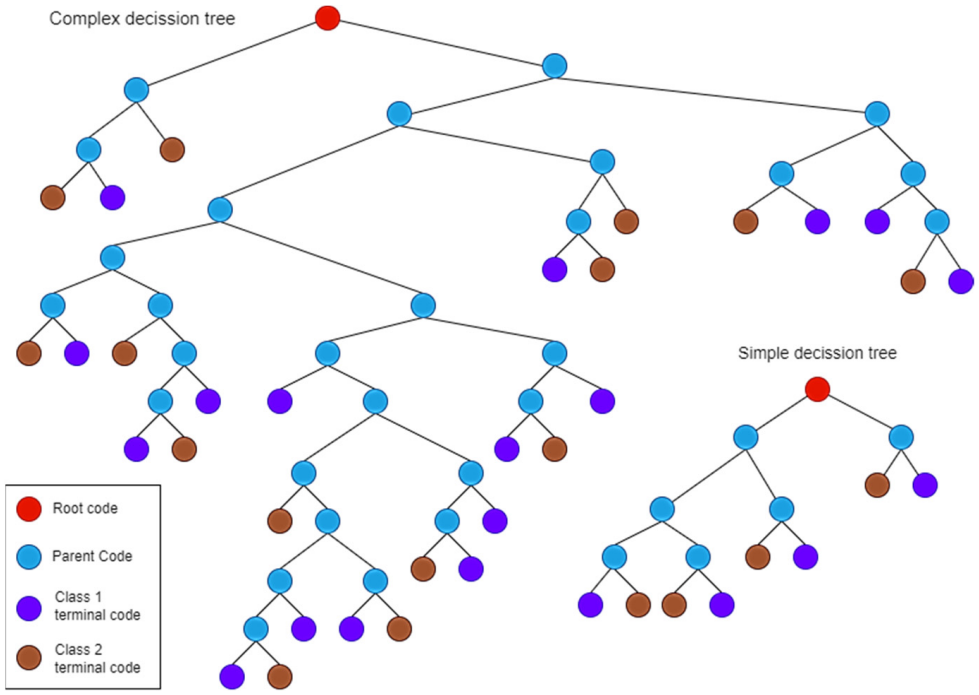

Figure 3. The effect of different complexity parameters on the decision tree model.

(c) Random Forest (RF)

An ensemble learning approach known as RFs or random decision forests can also be utilised to classify and predict data. RF provides strong ensemble classifiers with a capacity to generate a large number of trees using random bootstrapped samples of the training dataset, as demonstrated in Figure 4. Two parameters are modified in RF, namely,

the number of trees (ntree) and the number of variables or features (mtry) [28]. The ntree parameter allows experimentation with the number of trees based on the data size and type. However, using more than the required number of trees may be unnecessary because it may not negatively impact the model [29]. The mtry parameter refers to the number of variables randomly chosen as potential candidates at each split. The default value of mtry is usually the number of predictor variables, as shown in [30]. The RF model is more time consuming and difficult to understand and analyse than the decision tree because it creates many trees [31]. In this study, we evaluate a wide range of values for ntree and mtry to determine the optimum values for the best RF model performance.

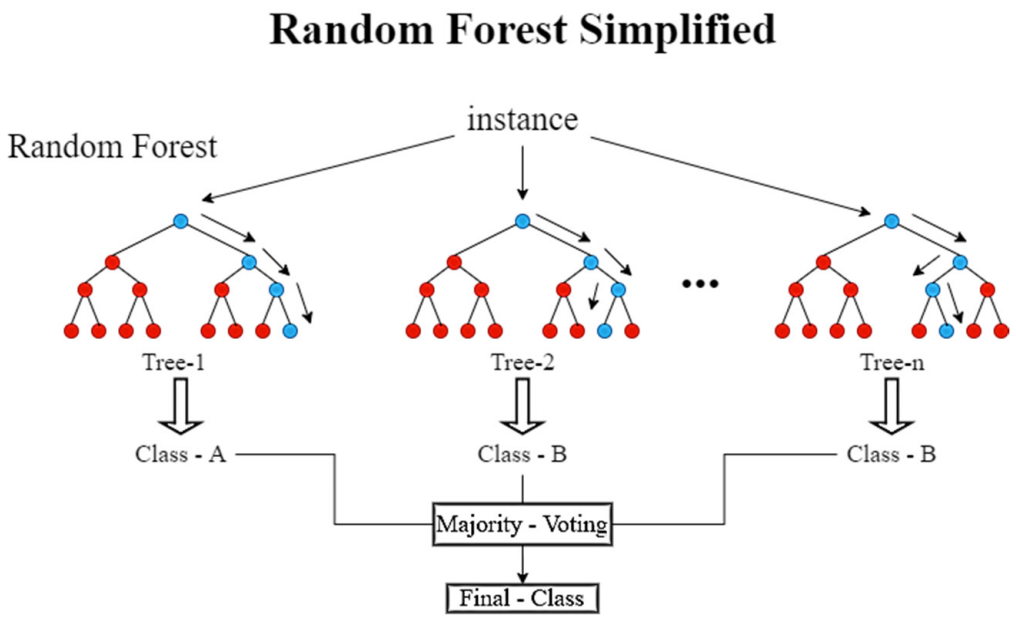

Figure 4. Random forest model.

(d) Support Vector Machine

SVM models use a hyperplane with the greatest margin to categorise between classes. The hyperplane is defined by the support vectors; SVMs successfully handle a wide range of ML problems that are neither linear nor separable [32]. The regularisation parameters, such as cost and sigma, must be determined before training an SVM classifier to utilise the model fully [33,34]. The cost parameter allows the alteration of the training data stiffness by determining the misclassification limit for the nonseparable training data. In addition, SVM uses sigma in smoothing the class dividing hyperplane in the Gaussian RBF kernel. The class dividing hyperplane's form may influence the classification accuracy results when the cost parameter value is increased, whereas decreasing the value can lead to an overfitted model [35]. The impact of sigma and cost is shown in Figure 5 [36]. The SVM performance improves when the number of dimensional spaces is larger than the entire sample set size, resulting in an excellent option for working with high-dimensional data [37]. Including subset training points in the decision function also enhances this model memory. However, SVM has a downside, that is, the prediction task becomes substantially computationally expensive when the dataset is extensive, resulting in SVM's long training time [38]. Furthermore, the performance of the algorithm also suffers if the dataset contains many overlapping classes [39]. Therefore, assessing these issues concerning our dataset is interesting. In particular, various values for both parameters are explored in this study to determine the ideal SVM model.

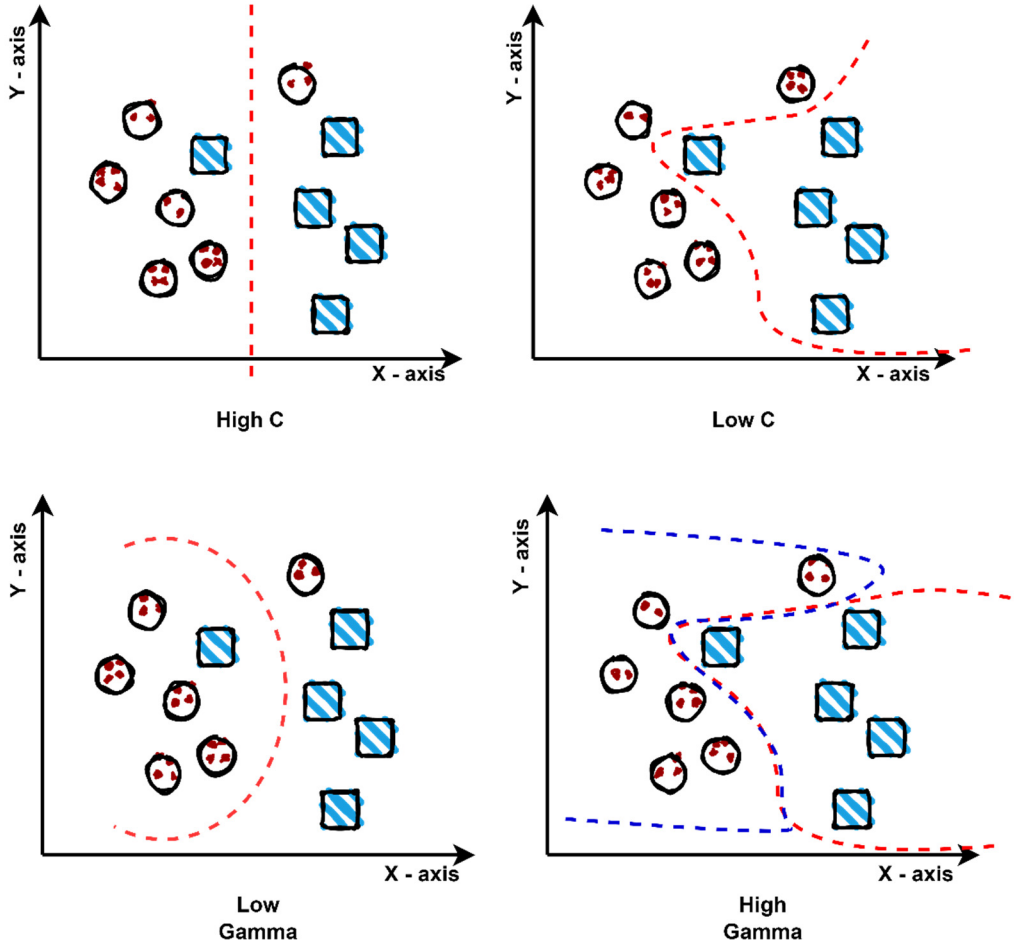

Figure 5. The effect of the SVM parameter setting on the hyperplane.

4. Parameter Tuning Results

In this section, we discuss the effects of tuning the hyperparameter on each of the models presented in Section 3. Each model is trained using the same dataset and given a range of parameter tuning values to be evaluated. The hyperparameters discussed in this section are the k values for KNN, cp for CART, ntree and mtry for RF and sigma and cost for SVM.

(a) KNN—Accuracy versus the number of neighbours

The KNN classifier uses the class properties of its k closest neighbours to categorise the predictors. Therefore, the k value is a critical factor in the KNN algorithm's performance and serves as a primary tuning parameter. In this study, we tested a range of k values from 1 to 50 to determine the best value for the KNN classifier parameter. This value range was chosen because it was not excessively large enough to cause considerable computation time. Then, the effects of varying the k values (or the number of neighbours) on the KNN model's accuracy can be observed. Therefore, the link between the number of neighbours and the accuracy of the KNN model can be observed by manipulating different k values. As seen in Figure 6, an initial 100% accuracy was reported between k values 1 and 20. However, the accuracy dropped considerably to ~90% between the k values 20 and 30. Then, an uneven plateau was detected. The link between the size of neighbours and the classification error rate is not uniform, as shown by these imbalanced binary class issues. The results indicate that the model becomes too specific and fails to generalise well when the k values are set too low. Furthermore, the model tends to be highly sensitive to noise, resulting in an overfitting model. We can conclude from Figure 6 that if the k values chosen are too large, the model becomes too generalised and fails to predict the data points accurately in

both the training and test sets, resulting in underfitting. However, small datasets may not always benefit from a high k value. A high k value is not always the best choice for a large dataset because its results are not always better than the results of a low k value [40].

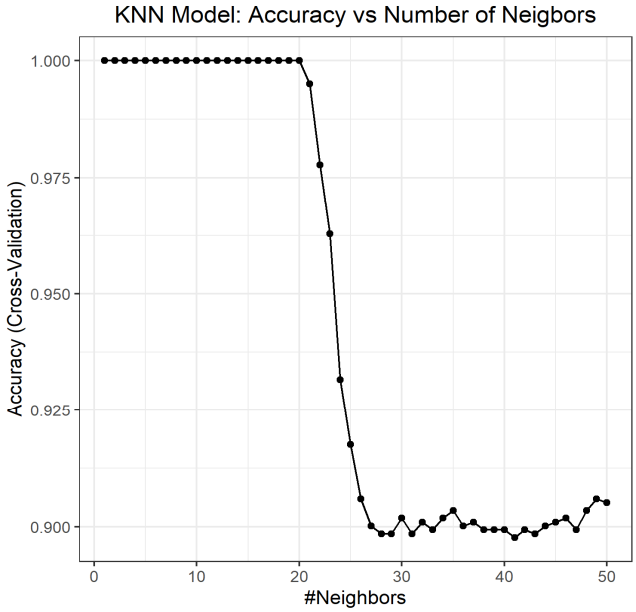

Figure 6. KNN model accuracy versus the number of neighbours.

(b) Classification and Regression Trees—Accuracy versus Max Tree Depth

As mentioned in Section 3, the CART model uses split conditions that are applied to the root node, and the decision procedure is repeated sequentially for each subroot node. The term max depth refers to the maximum depth to which the tree can grow. The larger the tree is allowed to grow further into the data, the more intricate the model becomes. As stated in [41,42], small trees are preferred as they are simple to print and display to subject matter experts. They also have a low likelihood of overfitting the dataset. A range of max depth values from 1 to 10 was chosen as our initial observation parameter to study the behaviour of the CART model. Based on our observation, our CART subcarrier prediction model in Figure 7 depicts an exponential increase in accuracy for a max depth range of 1 to 3. It starts to plateau at 100% accuracy as the max depth increases. The results indicate that the CART model prefers trees with a small number of nodes. Although increasing the max depth yields high accuracy cross-validation results, it overfits the training data and fails to capture important patterns as expected. A similar result is shown in [43] where the CART model easily overfits. Overcomplex trees may be created with low predictive ability. However, the decision tree's ability to detect patterns and interactions in the training data may be compromised if the max depth is too low. Consequently, the testing error also increases. Therefore, achieving an optimum depth value between the two extremes of being either too high or too low is a challenge.

(c) Random Forest

RF is a classification and prediction technique that employs many decision trees and two tuning parameters, ntree and mtry. The ntree and mtry parameters considerably affect the performance of the RF classifier, as detailed in Section 3.2. As summarised in [44], ntree is the number of generated randomised trees. Predicting with a large number of trees improves the stability and accuracy of the results and the ntree value of 500 is usually used when the model is deployed for real-world practice. In comparison, mtry is the size of the randomly selected feature set, where the small value of mtry results in diversity in the model. The range of the values of mtry used in this study is from 1 to 10, and the values of ntree which further evaluate the characteristics of the RF model towards

this tuning parameter are 100, 200, 500 and 1000. The graph in Figure 8 shows that a constant 100% cross-validation accuracy was obtained when the model was fed with all mtry and ntree values. We anticipate that the model can make 100% correct predictions regardless of the predictors utilised because of the small size and simplicity of the training dataset. However, the accuracy decreases when a large, sophisticated training dataset is used because of the increase in the number of predictors used, as demonstrated in [44]. Therefore, further investigation into the number of predictors is required to determine the model's best performance.

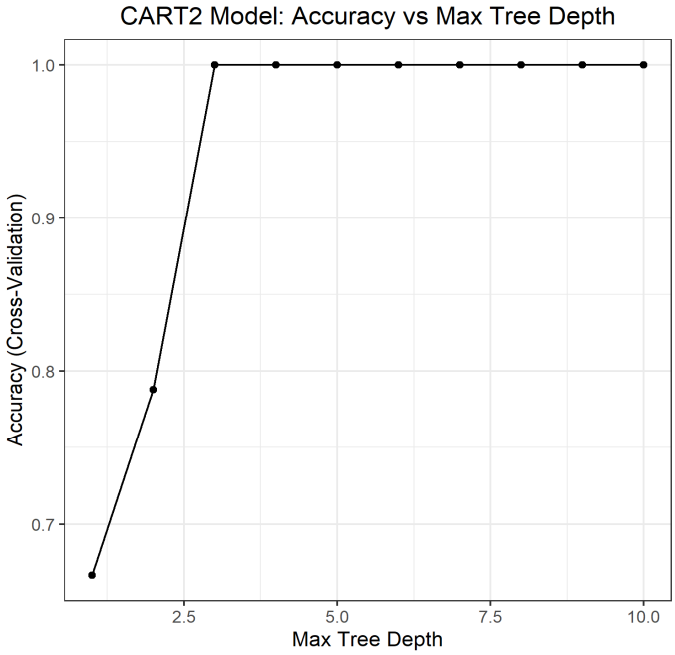

Figure 7. CART model accuracy versus max tree depth.

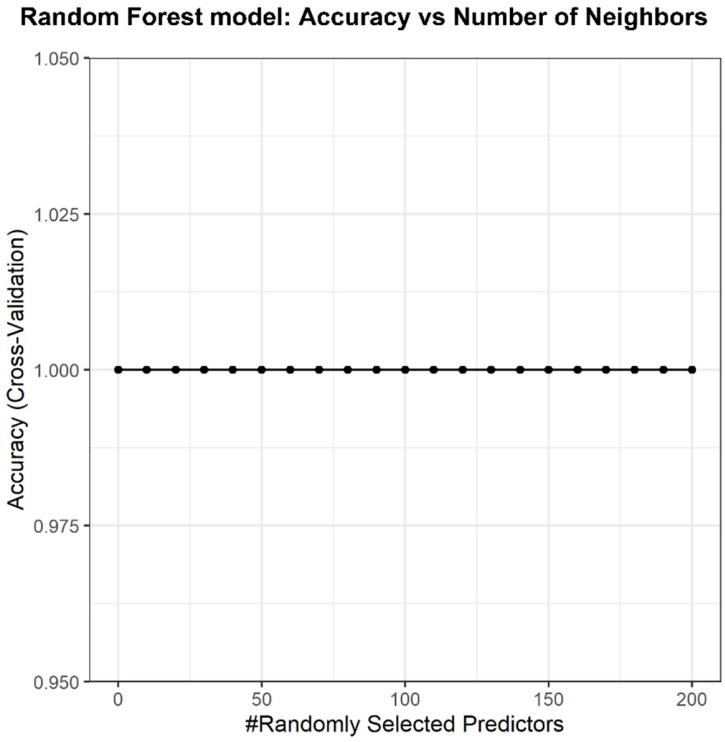

Figure 8. RF model accuracy versus number of randomly selected variables.

(d) SVM—Support vector machine accuracy versus sigma

Multiple values for cost and sigma were examined to determine the ideal values for the SVM model. In particular, the values chosen were sigma = 2^(−25, −20, −15, −10, −5, 0) and cost C = 3^(−7, −6, −5, −4, −3, −2, −1, 1, 2, 3, 4, 5, 6, 7). The resulting cross-validation accuracy values are shown in Figure 9. We observe that the cost value acquires saturation accuracy at approximately 65% for all cases. However, we can break the saturation point and increase the prediction accuracy by an additional 10% to 11% by further increasing the sigma and cost parameter. In our case, a low-cost value results in low accuracy, as a high tolerance level of misclassification causes poor performance as the cost parameter controls the model's flexibility and ability to generalise. Therefore, with a low cost, samples inside the margins are penalized less than with a higher cost. The changes in the sigma parameter in our SVM models also show that this parameter controls the level of nonlinearity introduced in the model. Figure 9 shows that high accuracy is achieved when the sigma value is minimal because a low sigma value results in a highly linear decision boundary. In contrast, the decision boundary tends to be linear as we increase the sigma value, resulting in high inaccuracies in prediction. This result emphasises the importance of carefully determining the sigma and cost values as it may results in high computational time and prediction accuracy.

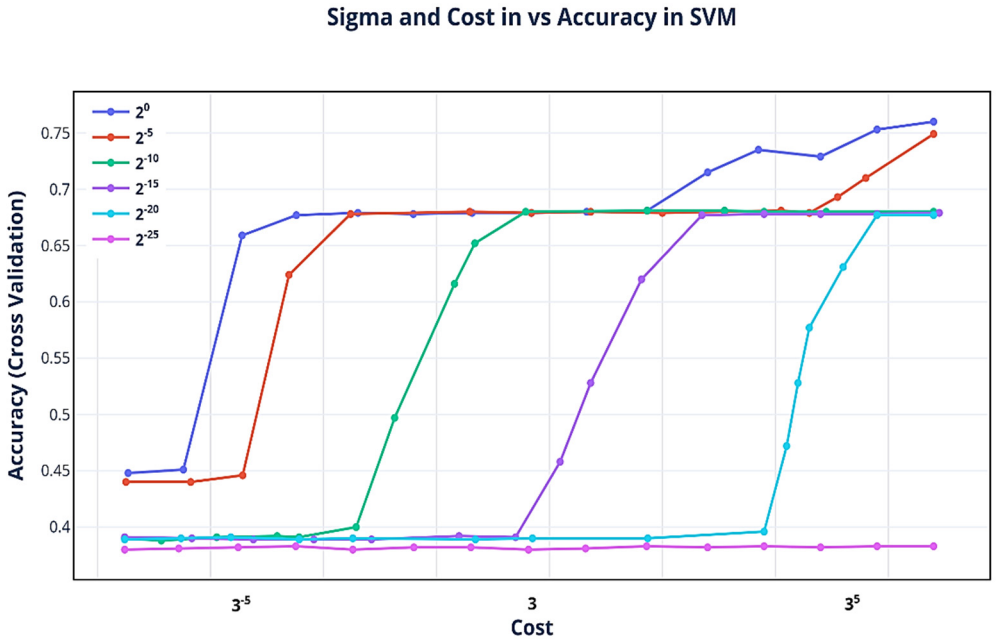

Figure 9. SVM model accuracy for various sigma and cost values.

Our results indicate that RF consistently generates the most accurate outcomes, followed by KNN, CART and SVM. This result is supported by [45–48], which indicated that RF generally has the best predictive performance. Although SVM contributes the least to cross-validation accuracy, its difference from the other ML models is not considerable; the average accuracy of these models is approximately 67%. Although SVM can perform well in unbalanced datasets, research in [49] demonstrated that the sample size and imbalanced data of the training samples have a larger effect on the classification accuracy of KNN and RF than on SVM, particularly when small and balanced training datasets are used. Additionally, our research highlights the importance of traditional parametric approaches in ML modelling when the datasets show a considerable degree of variation, which may cause the overfitting of the ML model. Thus, the interpretability, speed and generalisation capabilities of a simple algorithm make it a feasible alternative, particularly when the expected accuracy differences compared to other models are small [50].

5. Conclusions

This article provides a comprehensive observation of a scenario in which different hyperparameters of multiple ML model methods for 5G communication system subcarrier spacing prediction are manipulated. Resource optimisation is crucial because 5G is expected to be an important enabler in the information and communication technology industry by supporting diverse incoming services with various requirements. Our results show that the optimal hyperparameter setting for ML models directly impacts the model's performance. This observation emphasises the importance of hyperparameter tuning to understand how the ML models respond to the data used. Although multiple automated optimisation solutions exist, their advantages and drawbacks vary when they are applied to various levels of difficulty, most notably the challenges associated with the 5G technology. Given the modest size and simplicity of the dataset used, the RF model showed the greatest overall performance in terms of accuracy and consistency across all parameters employed. The relationship between the size of neighbours and the classification error rate is not necessarily linear for models such as KNN because small datasets may not always benefit from a large k value. Moreover, our observation of the CART model indicates that trees with a minimal number of nodes are preferable. They are straightforward to print and present to subject matter experts. They also have a low chance of overfitting the dataset whilst achieving 100% correctness. The deployment of ML as a subcarrier spacing prediction for 5G systems will make the wireless systems highly flexible and capable of accommodating varying traffic loads and data requirements in the future.

Author Contributions: F.S.S. (Writing—original draft preparation, algorithm, methodology and result); N.A.M.R. (Investigation, project administration and supervision); K.H.M.A. (Investigation and writing—review and editing); N.A.A. (Investigation and resources); N.M.A. (Investigation, project administration and supervision). All authors have read and agreed to the published version of the manuscript.

Funding: This study was financially supported by the Fundamental Research Grant Scheme (FRGS) grant (FRGS/1/2020/TK0/UNITEN/02/7).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the Ministry of Higher Education Malaysia (FRGS/1/2020/TK0/UNITEN/02/7) and UNITEN iRMC BOLD Publication Fund (J510050002) for funding the project.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study, in the collection, analyses or interpretation of data, in the writing of the manuscript or in the decision to publish the results.

References

1. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [CrossRef]
2. How Artificial Intelligence Improves 5G Wireless | DeepSig. Available online: https://www.deepsig.ai/how-artificial-intelligence-improves-5g-wireless-capabilities (accessed on 21 June 2022).
3. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [CrossRef]
4. Bergstra, J.; Kégl, B.; Bengio, Y.; Bardenet, R.; Bengio, Y. Algorithms for Hyper-Parameter Optimization Unsupervised Learning of Speech Representations View project Algorithms for Hyper-Parameter Optimization. Available online: https://www.researchgate.net/publication/216816964 (accessed on 12 April 2022).
5. Samidi, F.S.; Radzi, N.A.M.; Ahmad, W.S.H.M.W.; Abdullah, F.; Jamaludin, M.Z.; Ismail, A. 5G New Radio: Dynamic Time Division Duplex Radio Resource Management Approaches. *IEEE Access* **2021**, *9*, 113850–113865. [CrossRef]
6. Morocho-Cayamcela, M.E.; Lee, H.; Lim, W. Machine Learning for 5G/B5G Mobile and Wireless Communications: Potential, Limitations, and Future Directions. *IEEE Access* **2019**, *7*, 137184–137206. [CrossRef]
7. Ullah, H.; Nair, N.G.; Moore, A.; Nugent, C.; Muschamp, P.; Cuevas, M. 5G Communication: An Overview of Vehicle-to-Everything, Drones, and Healthcare Use-Cases. *IEEE Access* **2019**, *7*, 37251–37268. [CrossRef]

8. Meng, H.; Shafik, W.; Matinkhah, S.M.; Ahmad, Z. A 5G Beam Selection Machine Learning Algorithm for Unmanned Aerial Vehicle Applications. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 1–16. [CrossRef]
9. Khan, S.; Khattak, H.A.; Almogren, A.; Shah, M.A.; Din, I.U.; Alkhalifa, I.; Guizani, M. 5G Vehicular Network Resource Management for Improving Radio Access Through Machine Learning. *IEEE Access* **2020**, *8*, 6792–6800. [CrossRef]
10. Thakkar, A.; Lohiya, R. A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges. Arch. Comput. *Methods Eng.* **2020**, *28*, 3211–3243. [CrossRef]
11. Huang, C.-W.; Chiang, C.-T.; Li, Q. A study of deep learning networks on mobile traffic forecasting. In Proceedings of the IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–6. [CrossRef]
12. Ma, B.; Guo, W.; Zhang, J. A Survey of Online Data-Driven Proactive 5G Network Optimisation Using Machine Learning. *IEEE Access* **2020**, *8*, 35606–35637. [CrossRef]
13. Asadi, A.; Muller, S.; Sim, G.H.; Klein, A.; Hollick, M. FML: Fast Machine Learning for 5G mmWave Vehicular Communications. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 15–19 April 2018; pp. 1961–1969. [CrossRef]
14. Sim, G.H.; Klos, S.; Asadi, A.; Klein, A.; Hollick, M. An Online Context-Aware Machine Learning Algorithm for 5G mmWave Vehicular Communications. *IEEE/ACM Trans. Netw.* **2018**, *26*, 2487–2500. [CrossRef]
15. Baz, A. Bayesian Machine Learning Algorithm for Flow Prediction in SDN Switches. In Proceedings of the 1st International Conference on Computer Applications and Information Security, ICCAIS, Riyadh, Saudi Arabia, 4–6 April 2018. [CrossRef]
16. Qin, M.; Yang, Q.; Cheng, N.; Zhou, H.; Rao, R.R.; Shen, X. Machine Learning Aided Context-Aware Self-Healing Management for Ultra Dense Networks with QoS Provisions. *IEEE Trans. Veh. Technol.* **2018**, *67*, 12339–12351. [CrossRef]
17. Farooq, H.; Forgeat, J.; Bothe, S.; Bouton, M.; Shirazipour, M.; Karlsson, P. Coordinated Hyper-Parameter Search for Edge Machine Learning in Beyond-5G Networks. In Proceedings of the IEEE International Conference on Communications Workshops, ICC Workshops, Montreal, QC, Canada, 14–23 June 2021. [CrossRef]
18. Isabona, J.; Imoize, A.L.; Kim, Y. Machine Learning-Based Boosted Regression Ensemble Combined with Hyperparameter Tuning for Optimal Adaptive Learning. *Sensors* **2022**, *22*, 3776. [CrossRef] [PubMed]
19. Fang, H.; Wang, X.; Tomasin, S. Machine Learning for Intelligent Authentication in 5G and Beyond Wireless Networks. *IEEE Wirel. Commun.* **2019**, *26*, 55–61. [CrossRef]
20. Osman, H.; Ghafari, M.; Nierstrasz, O. Hyperparameter optimization to improve bug prediction accuracy. In Proceedings of the IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE), Klagenfurt, Austria, 21 February 2017; pp. 33–38. [CrossRef]
21. Akanbi, O.A.; Amiri, I.S.; Fazeldehkordi, E. Implementation and result. In *A Machine-Learning Approach to Phishing Detection and Defense*; Elsevier: Amsterdam, The Netherlands, 2015; pp. 55–73. [CrossRef]
22. Geler, Z.; Kurbalija, V.; Radovanović, M.; Ivanović, M. Comparison of different weighting schemes for the kNN classifier on time-series data. *Knowl. Inf. Syst.* **2015**, *48*, 331–378. [CrossRef]
23. Geler, Z.; Kurbalija, V.; Ivanović, M.; Radovanović, M. Weighted kNN and constrained elastic distances for time-series classification. *Expert Syst. Appl.* **2020**, *162*, 113829. [CrossRef]
24. Ray, S. A Quick Review of Machine Learning Algorithms. In Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 35–39. [CrossRef]
25. Starzacher, A.; Rinner, B. Evaluating KNN, LDA and QDA classification for embedded online feature fusion. In Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Sydney, Australia, 15–18 December 2008; pp. 85–90. [CrossRef]
26. Bailey, R.R.; Srinath, M. Orthogonal moment features for use with parametric and non-parametric classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 389–399. [CrossRef]
27. Sen, P.C.; Hajra, M.; Ghosh, M. Supervised Classification Algorithms in Machine Learning: A Survey and Review. In *Advances in Intelligent Systems and Computing*; Springer Science and Business Media: Berlin, Germany, 2020; Volume 937, pp. 99–111. [CrossRef]
28. A Debiased MDI Feature Importance Measure for Random Forests. Available online: https://proceedings.neurips.cc/paper/2019/hash/702cafa3bb4c9c86e4a3b6834b45aedd-Abstract.html (accessed on 2 March 2022).
29. Stevens, F.R.; Gaughan, A.E.; Linard, C.; Tatem, A.J. Disaggregating Census Data for Population Mapping Using Random Forests with Remotely-Sensed and Ancillary Data. *PLoS ONE* **2015**, *10*, e0107042. [CrossRef]
30. Fox, E.W.; Hill, R.A.; Leibowitz, S.G.; Olsen, A.R.; Thornbrugh, D.J.; Weber, M.H. Assessing the accuracy and stability of variable selection methods for random forest modeling in ecology. *Environ. Monit. Assess.* **2017**, *189*, 316. [CrossRef]
31. Wang, Z.; Wang, Y.; Zeng, R.; Srinivasan, R.S.; Ahrentzen, S. Random Forest based hourly building energy prediction. *Energy Build.* **2018**, *171*, 11–25. [CrossRef]
32. Burges, C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 121–167. [CrossRef]
33. Güven, I.; Şimşir, F. Demand forecasting with color parameter in retail apparel industry using artificial neural networks (ANN) and support vector machines (SVM) methods. *Comput. Ind. Eng.* **2020**, *147*, 106678. [CrossRef]

34. Jordaan, E.; Smits, G. Estimation of the regularization parameter for support vector regression. In Proceedings of the International Joint Conference on Neural Networks. IJCNN'02, Honolulu, HI, USA, 12–17 May 2022; pp. 2192–2197. [CrossRef]
35. SVM Hyperparameters Explained with Visualizations | by Soner Yıldırım | Towards Data Science. Available online: https://towardsdatascience.com/svm-hyperparameters-explained-with-visualizations-143e48cb701b (accessed on 2 March 2022).
36. C and Gamma in SVM. A | by A Man Kumar | Medium. Available online: https://medium.com/@myselfaman12345/c-and-gamma-in-svm-e6cee48626be (accessed on 2 March 2022).
37. Pes, B. Learning from High-Dimensional Biomedical Datasets: The Issue of Class Imbalance. *IEEE Access* **2020**, *8*, 13527–13540. [CrossRef]
38. Muthukrishnan, S.; Pallekonda, A.K.; Saravanan, R.; Meenakshi, B. Fault Detection in the Wind Farm Turbine Using Machine Learning Based on SVM Algorithm. *J. Phys. Conf. Ser.* **2021**, *1964*, 052015. [CrossRef]
39. Saez, J.A.; Galar, M.; Krawczyk, B. Addressing the Overlapping Data Problem in Classification Using the One-vs-One Decomposition Strategy. *IEEE Access* **2019**, *7*, 83396–83411. [CrossRef]
40. Zhang, Z. Introduction to machine learning: K-nearest neighbors. *Ann. Transl. Med.* **2016**, *4*, 218. [CrossRef]
41. Grubinger, T.; Zeileis, A.; Pfeiffer, K.-P. evtree: Evolutionary Learning of Globally Optimal Classification and Regression Trees inR. *J. Stat. Softw.* **2014**, *61*, 1–29. [CrossRef]
42. Rao, A.R.; Lohse, G.L. Towards a texture naming system: Identifying relevant dimensions of texture. *Vis. Res.* **1996**, *36*, 1649–1669. [CrossRef]
43. Khoshgoftaar, T.M.; Allen, E.B. Controlling Overfitting in Classification-Tree Models of Software Quality. *Empir. Softw. Eng.* **2001**, *6*, 59–79. [CrossRef]
44. Ao, Y.; Li, H.; Zhu, L.; Ali, S.; Yang, Z. Identifying channel sand-body from multiple seismic attributes with an improved random forest algorithm. *J. Pet. Sci. Eng.* **2018**, *173*, 781–792. [CrossRef]
45. Talukdar, S.; Singha, P.; Mahato, S.; Shahfahad; Pal, S.; Liou, Y.-A.; Rahman, A. Land-Use Land-Cover Classification by Machine Learning Classifiers for Satellite Observations—A Review. *Remote Sens.* **2020**, *12*, 1135. [CrossRef]
46. Iranitalab, A.; Khattak, A. Comparison of four statistical and machine learning methods for crash severity prediction. *Accid. Anal. Prev.* **2017**, *108*, 27–36. [CrossRef] [PubMed]
47. Rao, R.S.; Pais, A.R. Detection of phishing websites using an efficient feature-based machine learning framework. *Neural Comput. Appl.* **2018**, *31*, 3851–3873. [CrossRef]
48. Deist, T.M.; Dankers, F.J.W.M.; Valdes, G.; Wijsman, R.; Hsu, I.; Oberije, C.; Lustberg, T.; van Soest, J.; Hoebers, F.; Jochems, A.; et al. Machine learning algorithms for outcome prediction in (chemo)radiotherapy: An empirical comparison of classifiers. *Med. Phys.* **2018**, *45*, 3449–3459. [CrossRef]
49. Thanh Noi, P.; Kappas, M. Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery. *Sensors* **2018**, *18*, 18. [CrossRef]
50. Schratz, P.; Muenchow, J.; Iturritxa, E.; Richter, J.; Brenning, A. Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. *Ecol. Model.* **2019**, *406*, 109–120. [CrossRef]