Optimization machine learning models for selecting transmit antennas in 5G/6G systems

Abdellah Ouldammar, Abdelmounaim Moulay Lakhdar, Ahmed Bouida, Khaled Merit
Laboratory of Information Processing and Telecommunication (LTIT), University Tahri Mohammed Bechar, Bechar, Algeria

Article Info

Article history:

Received Jul 2, 2024
Revised Sep 15, 2024
Accepted Sep 29, 2024

Keywords:

5G/6G
GridSearchCV
KNN
Supervised learning models
SVM
TAS

ABSTRACT

Transmit antenna selection (TAS) plays a crucial role in improving the performance and spectral efficiency of 5G/6G systems. This study proposes to use the GridSearchCV method for hyperparameter optimization in two supervised learning models, support vector machine (SVM) and K-nearest neighbors (KNN), to optimally select antenna peers based on channel gain. These models were applied to Alamouti's space-time block coding to improve performance, resulting in increased signal-to-noise ratio (SNR) and reduced bit error rate (BER). The results show that optimizing the hyperparameters led to a significant improvement in the performance of the SVM and KNN models. The SVM and KNN models were evaluated using a variety of metrics, with the SVM demonstrating superior predictive performance in terms of accuracy, average macro recall, average macro precision, average macro F1 score, and cross-validation score. Even before optimization, the SVM outperforms the KNN in terms of performance metrics. After optimization, this gap widens further, demonstrating the robustness of SVM for classification tasks. Although KNN is faster to train.

Corresponding Author:

Abdellah Ouldammar
Laboratory of Information Processing and Telecommunication (LTIT)
University Tahri Mohammed Bechar
Bechar, Algeria
Email: ouldammar.abdellah@univ-bechar.dz

1. INTRODUCTION

Multipath propagation is an inherent feature of the wireless channel, leading to signal fading. One way of improving the capacity of wireless channels is to use multiple antennas at the transmitter and receiver [1]-[13]. However, in many real-world systems, the receiver must be small by design (cell phones, PDAs, and smartphones). in such physically small receivers, it's not practical to have several antennas. It is therfore impractical to deploy multiple antennas at the receiver. In contrast, the base station can easily accommodate multiple antennas, making it possible to use multiple transmit antennas. In such systems, transmit antenna selection (TAS) algorithms can help reduce the number of active transmit antenna, which can lead to impoved bit error rate (BER) performance and reduced hardware complexity [3]-[16]. For multi-antenna systems, TAS plays an important role in the trade-off between transmission rate and power cosumption, as a greater number of active antenna results in higher channel gain, but also increases circuit power [15].

The antenna selection approch (TAS) can also be combined with other techniques, such as orthogonal space-time block coding (OSTBC), to further improve system performance [17]. Li *et al.* [18], the authors discussed strategies for improving the performance of multiple-antenna communication systems in the context of low-rate channel -state feedback. More specifically, it focuses on the different transmitter strategies, such as TAS, Alamouti space-time coding and adaptive power allocation in multiple-input singleoutput (MISO) communication systems. This approach to antenna selection in MISO systems involves an exhaustive search for all possible combinations of available antennas. This can become complex due to the large number of calculations involved. Fast antenna selection algoritms are therefore essential to reduce this complexity. Therefore, it is crucial to develop efficient algorithms for antennas selection in MIMO systems in order to reduce the complexity associated with exhaustive searches for all possible combinations among the available antennas and optimize real-time performance.

There are optimization approaches, such as a greedy step-wise optimization approach, which can provide near-optimal solution [2]-[13]. So, antenna selection in wireless communications is an optimization-based decision, where selection is based on predefined criteria or algorithms. However, Machine-Learning techniques offer a data-driven approach to antenna selection, where selection is based on learning from available data and making predictions. Joung [19] compared machine learning-based antenna selection whith conventional optimization-based methods in terms of performances, complexity and feedback, where he applies multi-class classification, which is a main task in machine learning, in a multiple-input multiple-output (MIMO) system with TAS. Yang *et al.* [14], the authors propose a new approach called pattern recognition aided transmission antenna selection MIMO (PR-TAS), which uses pattern recognition algorithms such as K-nearest neighbors (KNN) and support vector machine (SVM), to select antennas in a vertical bell laboratories layered space time (V-BLAST) system, which divides serial data into several sub-streams and transmits them via different transmission antennas respectively, the authors have optimized the features extraction of the TAS algorithm based on the pattern recognition demonstrated in reference [19], to run the antenna selection algorithm and effectivelly reduce redundant calculations in the antenna set search process. In paper [20], the authors proposed a learning-based antenna selection and power allocation (L-ASPA) algorithm that overcomes the high computational complexity of the joint antenna selection and power allocation (JASPA) algorithm. In the study presented in article [21], the authors proposed a machine learning -based approach, in particular the K-NN algorithm, for transmitter antenna selection to improve system performance in real-time environments. Machine Learning-based antenna selection and power allocation designs were investigated in paper [22], to fully exploit spatial diversity while minimizing the power consumption of avtive radio frequency (RF) modules. The authors propose an assisted antenna selection scheme (multi-label convolutional neural network) MLCNN for MIMO Internet of things communication systems, which shows promising results in improving prediction accuracy. The conclusion of the study in article [23], is that the proposed Machine Learning-Enabled Joint Antenna Selection and Precoding Designs algorithm (L-ASPD) significantly outperforms baseline schemes, achieves better effictive sum rate compared to traditional methods like joint antenna selection and precoding design (JASPD) under limited processing time, and reduces computation complexity by 95% while retaining over 95% of optimal performance. This indicates that using machine learning for antenna selection and precoding design can lead to improved system efficiency in wireless communication systems.

Despite the large amount of work on this topic, no previous study has investigated in depth how to find optimal values for the hyperparameters of a machine learning model. In almost all machine learning projects, different models are used and the one with the best performance on the dataset is selected. However, there is room for improvement because we cannot be sure that this particular model is the best one to solve the broblem in question [14], [19], [24].

This study investigated the effects of hyperparameter optimization on machine learning models for TAS in 5G/6G systems. While earlier studies have explored the impact of machine learning techniques on antenna selection and system performance, they have not explicitly addressed its influence on finding optimal hyperparameter values to maximize model performance and efficiency, particularly using systematic approaches like GridSearchCV.

As mentionede earlier, the performance of a model depends on the values for the hyperparameters. It's worth noting that there is no way to know the best values for the hyperparameters in advance, so ideally, we need to try all possible values to find the optimal ones. Doing this manually could take a lot of time and resources, so we use GridSearchCV to automate the tuning of the hyperparameters.

2. DATA-DRIVEN PREDICTION METHOD IN TRANSMIT ANTENNA SELECTION

Data-driven prediction (DDP) method in TAS involves using machine learning algorithms to predict the best transmit antennas based on the channe conditions [25]. This approach can be used in mutiple-input mutiple-output (MIMO) systems to improve the system capacity and reiability. The DDP method can be used in combination with other antenna selection methods, such as the (KNN) algorithm or the (SVM) algorithm [14], [19], [21]. These methods aim to reduce the computational compexity associated with traditional TAS algorithms by avoiding exhaustive searches of all possible antenna subsets. Data driven prediction (DDP) methods in TAS can achieve lower complexity and higher efficiency, making them suitable for real-time applications. In our work, we applied multi-class classification and defined two classifiers, KNN and SVM, performing nested cross-validation using GridSearchCV to find the best hyperparameters for each classifier.

2.1. K-nearest neighbor (KNN)

KNN is a simple instance-based learning agorithm that classifies a data point according to the majority class among its K nearest neighbors in feature space [26]-[29]. KNN can be used to map the CSI to the optimal antenna subset by considering the most similar past instances [14], [19], [21]. This method is particularly useful in dynamic envirenments where wireless channe conditions change freqquently.

2.2. Support vector machine (SVM)

SVM is a supervised learning model that can be used for classification and regression tasks. It works by finding the hyperplane that best separates the data into different classes [26], [28], [30]-[33]. It is possible to train an SVM to classify the optimal set of antennas based on channel state information (CSI) [14], [19]. This classification helps to select antennas that maximize system performance metrics such as signal-to-noise-ratio (SNR) BER.

2.3. GridSearchCV method

GridSearchCV is a technique for systematically searching for the hyperparameters of a machine learning model by trying all possible combinations of specific values for those hyperparameters. It uses cross-validation to evaluate the performance of each combination. GridSearchCV is used to optimize the hyperparameters of a model to find the best possible configuration that maximizes model performance on a given dataset. It is implemented in the scikit-learn library in Python. After testing all combinations, GridSearchCV selects the model with the best hyperparameters and retrains it on the dataset.

3. PROPOSED METHOD

This section presents the proposed methodology for predicting the best antenna pairs for data transmission using the Alamouti code in the MISO 2X1 system. A full description of the methodology is provided in this section.

3.1. Data generation and preprocessing

We have generated a synthetic data set representing the channel coefficient of a MISO system with eight transmit antennas and ten thousand samples. The channel coefficients are generated using the Rayleigh fading model. The channel gain is then calculated by squaring the channel coefficients. We then generate labels for the antenna pairs, create all possible combinations of antenna pairs, and check which pair has the highest channel gain for each sample. These Boolean labels are then converted to class labels. Using the channel gain as input to the machine learning model, the antenna pairs are used as aabels. The data is diveded into training and test sets: 80% of the data is used for training and 20% for testing. The features are then normalized. The goal off this operation is to give the same scale to all input data.

3.2. Machine learning models

We used two established machine learning models to predict the best antenna pairs for the Alamouti spâce-time block code system. These models are KNN and SVM. We chose these two models because of their different approaches and different prediction track records.

3.3. Tuning hyperparameters

For each classifier, nested cross-validation is performed to find the best hyperpârameters. The GridSearchCV scikit-learn is used for this purpos. The different hyperparameters for the two models used are shown in Table 1.

Table 1. Hyperparameters for both models

KNN		SVM	
Parameter	Values/Options	Parameter	Values/Options
n-neighbors	[3, 5, 7, 9, 11]	C	[0.1, 1, 10, 100]
Weights	[‘uniform’, ‘distance’]	Gamma	[0.001, 0.01, 0.1, 1]
Metric	[‘euclidean’, ‘manhattan’, ‘minkowski’]	Kernel	[‘linear’, ‘rbf’, ‘poly’, ‘sigmoid’]

3.3.1. Parameters grid for KNN

a) n_neighbors: values [3, 5, 7, 9, 11] are chosen to test different numbers of neighbors. The number of neighbors (n_neighbors) is a crucial parameter for the KNN model, as it determines how many neighbors are taken into account when making a prediction. Too small a number risks making the model sensitive to noise, while too large a number can dilute the influence of close neighbors. By testing several odd values, we avoid ties in the majority vote, which is particularly useful for classification problems.

b) Weights: the [‘uniform’, ‘distance’] options can be used to test two different approaches for weighting neighbors.

– Uniform: all neighbors have the same weight.
– Distance: closer neighbors are given a higher weight, which can improve accuracy by giving more weight to the most relevant neighbors.
Test these two options shows whether distance weighting improves model performance compared with uniform weighting.

c) Metric: metrics [‘euclidean’, ‘manhattan’, ‘minkowski’] are distances commonly used to measure similarity between points.

– Euclidean: Euclidean distance, the distance “as the crow flies”.
– Manhattan: Manhattan distance, the sum of absolute distances along the axes.
– Minkowski: A generalization of the previous two, with the parameter p=3.

Testing different metrics helps to determine which is most appropriate for the specific data, as KNN performance can vary depending on the metric used.

3.3.2. Parameters grid for SVM

a) C: values [0.1, 1, 10, 100] are chosen to test different levels of regularization.

– C low (0.1): high regularization, which may avoid overlearning, but may lead to underlearning.
– C high (100): low regularization, which may allow a better fit to the training data, but risks overlearning.

Testing a range of values helps to find the right balance between bias and variance.

b) Gamma: values [0.001, 0.01, 0.1, 1] are chosen to test the influence of the decision function.

– High gamma (1): each data point has a greater influence, which can lead to overlearning.
– Low Gamma (0.001): each data point has less influence, which can lead to underlearning.

Testing different gamma values helps to find the right level of model complexity.

c) Kernel: kernel types [‘linear’, ‘rbf’, ‘poly’, sigmoid’] allow different data transformations.

– Linear: linear kernel, useful for linearly separable data.
– rbf: gaussian kernel (Radial Basic Function), useful for non-linearly separable data.
– Poly: polynomial kernel, useful for capturing more complex relationships.
– Sigmoid: sigmoid kernel, less commonly used but can be useful in certain cases.

Testing different kernels allows you to see which one best captures the relationships in the data.

3.3.3. Cross validation

We use cross-validation to evaluate the performance of each hyperparameter combination. We use 5-fold cross-validation to divide the data into five subsets and evaluate the generalizability of the results of a statistical analysis to an independent data set. This allows us to avoid overlearning and provide a more reliable estimate of model performance. Using these hyperparameter grids, GridSearchCV can explore a variety of configurations for each model, automating the process of finding the best hyperparameters. This makes it possible to optimize KNN and SVM models to improve prediction accuracy, without having to manually test each combination.

3.4. Performance evaluation

To evaluate a machine learning model in a multi-class classification task, several performance metrics can be used, such as accuracy, precision, recall, F1 score, confusion matrix, ROC-AUC (Receiver Operating Characteristic-Area Under Curve), and micro and macro averages. Although we have 28 classes, it can indeed be difficult to interpret the results due to the complexity. In this case, it can be useful to visualize the model performance using the following metrics:

a) Confusion matrix: a matrix that visualizes model performance by showing true positives, false positives, true negatives, and false negatives for each class. It helps to understand where the model is going wrong.

b) Accuracy (Overall Precision): the percentage of correct predictions out of all predictions made. It’s a simple measure, but can be misleading if the classes are unbalanced.

c) Macro-Average: calculates the metric independently for each class and averages it. This treats all class equally, regardless of their frequency.
d) Precision: the proportion of true positive predictions out of all positive predictions.
e) Recall: the proportion of true positive predictions out of all the true positive instances.
f) F1 score: the harmonic mean of precision and recall, providing a balance between the two.

We also evaluated the performance of the adopted models by calculating the 5-fold cross-validation score to provide an estimate of model performance on novel data, as well as measuring training and prediction times for the SVM and KNN classifiers.

4. RESULTS AND DISCUSSION

4.1. Results

In this section, we present the results of simulations performed in the Python environment, i.e., confusion matrices for a test set. We also present tables showing the results of the calculations for the different metrics mentioned above. These tables allow us to compare the confusion matrices obtained by the SVM and KNN models before and after hyperparameter optimization, for model evaluation purposes.

4.1.1. Model performances

After adjusting the hyperparameters, we initialized and trained the SVM and KNN classifiers on the training data, and then performed the predictions on the test data. The confusion matrices in Figures 1-4 show that there are 28 classes (0-27) into which the models classify the data. They also show the number of correct and incorrect predictions made by the SVM and KNN models. The values on the diagonal represent correct predictions, while the other values represent errors. Models optimized in terms of hyperparameters show overall accuracy, with fewer false positives (FP) and false negatives (FN) than non-optimized models. And in both cases, SVM has a better overall accuracy, with fewer false positives (FP) and false negatives (FN) than KNN, the latter having more FP and FN, indicating a slightly inferior performance in terms of correct classification. And, Tables 2 and 3 show the evaluation of the SVM and KNN models. After optimization, the performance of the SVM improved in all metrics (accuracy, precision, recall, F1 score, average cross-validation score). For example, accuracy increased from 0.9190 to 0.9295. KNN also improved, from 0.7820 to 0.8200 for accuracy, but this improvement was less significant than that of SVM.

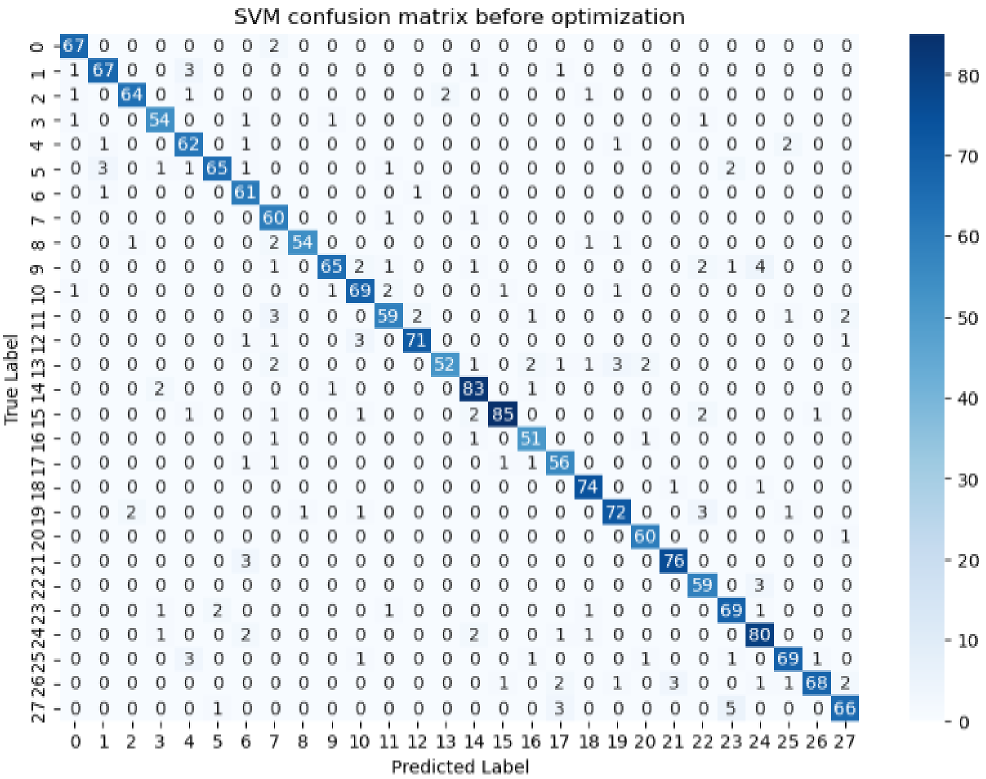

Figure 1. Confusion matrix (SVM) for a set of tests using the default hyperparameters

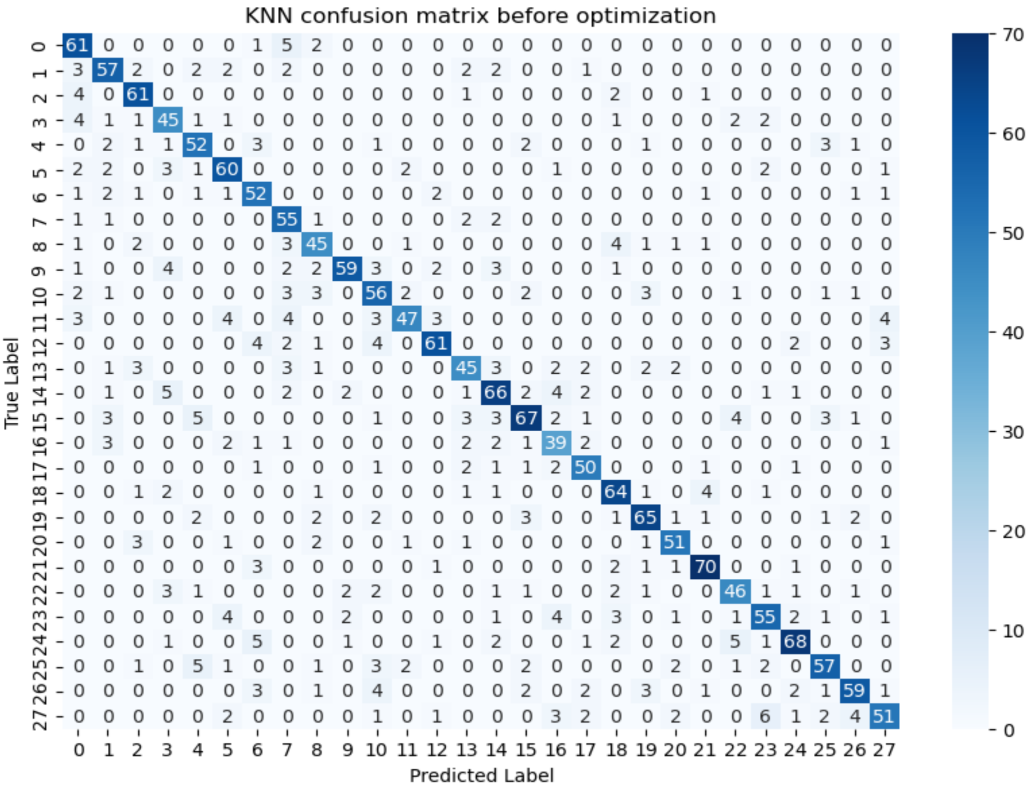

Figure 2. Confusion matrix (KNN) for a set of tests using the default hyperparameters

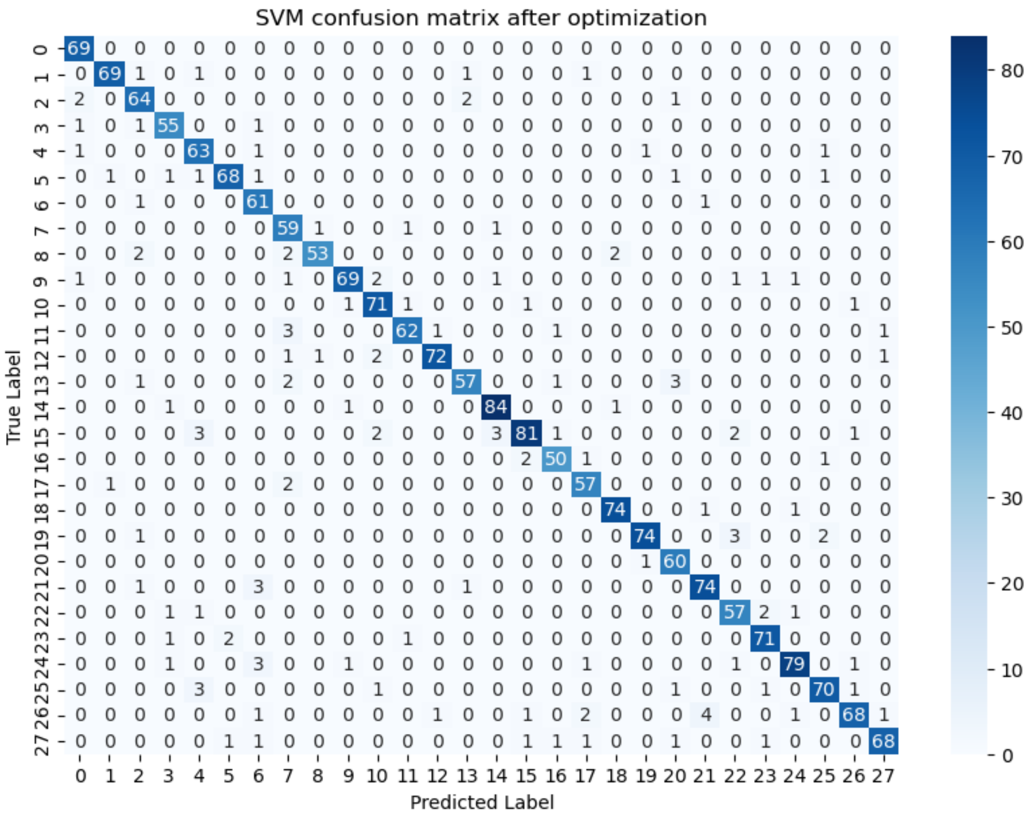

Figure 3. Confusion matrix (SVM) for a set of tests using GridSearchCV

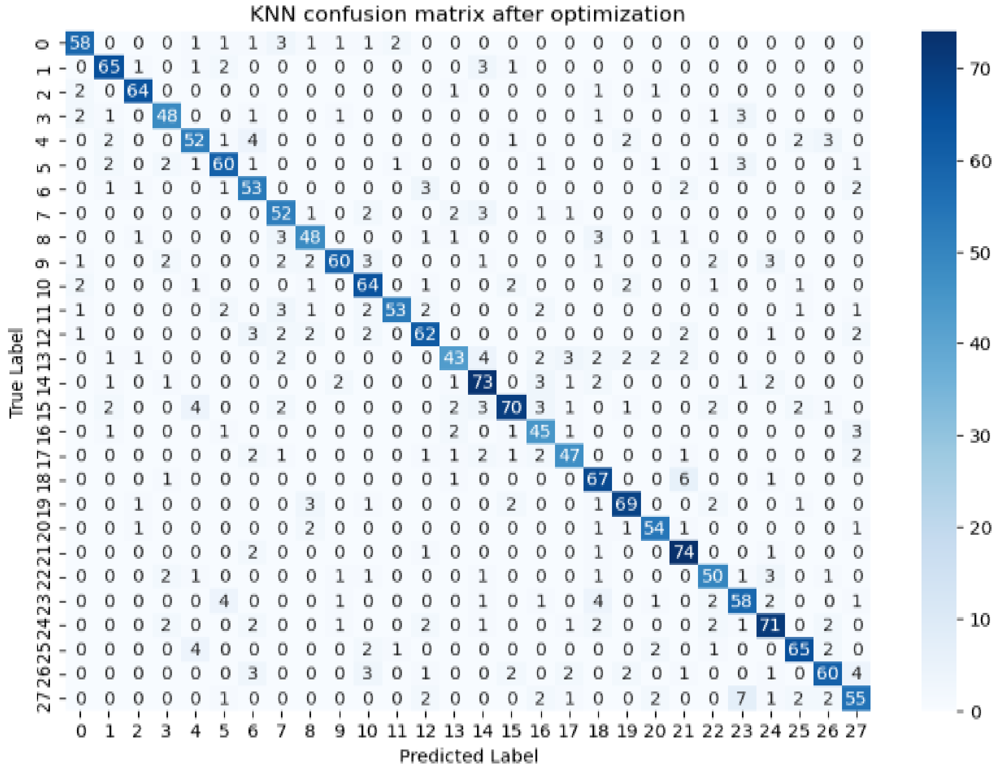

Figure 4. Confusion matrix (KNN) for a set of tests using GridSearchCV

Table 2. Results of evaluation metrics obtained for SVM and KNN models before optimization

	Evaluation				
Metrics	Accuracy (%)	Macro average precision (%)	Macro average recall (%)	Macro average F1 score (%)	Average cross-validation score (%)
SVM	91.90	92.13	91.90	91.90	91.07
KNN	78.20	78.67	78.20	78.21	78.41

Table 3. Results of evaluation metrics obtained for SVM and KNN models after optimization

	Evaluation				
Metrics	Accuracy (%)	Macro average precision (%)	Macro average recall (%)	Macro average F1 score (%)	Average cross-validation score (%)
SVM	92.95	93.12	92.95	92.96	92.96
KNN	82.00	82.28	82.00	81.98	82.80

4.1.2. Computational complexity

Table 4 shows how the SVM model optimization has reduced the training and prediction times: the training time is reduced from 0.3756 to 0.2408 seconds. And the prediction time decreased from 0.6111 to 0.2981 seconds. This indicates that the optimized hyperparameters have made the model more efficient. KNN maintained a similar training time, but the prediction time was slightly reduced from 0.0819 to 0.0629 seconds. This shows a slight improvement in prediction speed.

We found that optimizing hyperparameters using the GridSearchCV method significantly enhances the classification performance of machine learning models in selecting transmit antennas in 5G/6G systems. The proposed method in this study tended to have an inordinately higher proportion of correct classifications as shown by improved metrics such as accuracy, precision, recall, and F1 score across multiple classes.

Table 4. Results of the computational complexity of classifiers

	Before optimization		After optimization	
	Training time (Seconds)	Prediction time (Seconds)	Training time (Seconds)	Prediction time (Seconds)
SVM	0.3756	0.6111	0.2408	0.2981
KNN	0.0050	0.0819	0.0050	0.0629

4.2. Discussion

Optimizing hyperparameters for SVM can include selecting the type of support kernel, the C parameter, and the gamma parameter. These adjustments improve the model's ability to optimally separate classes. For KNN, optimization may involve the number of neighbors (K) and the distance weighting method. Optimal choice of K can improve model accuracy by reducing noise. After optimizing the SVM model's hyperparameters, a reduction in computational time can be achieved through more efficient parameter choices that reduce the number of computations required or simplify the model. For example, a simpler kernel or a weel-tuned C-value can speed up the convergence of the algorithm. KNN is generally faster to train because it relies on simple distance calculations. However, optimization can further reduce prediction time by choosing an optimal K that minimizes the number of computations required.

Even befor optimization, SVM far outperforms KNN in terms of metric performance. After optimization, the gap widens even further, demonstrating the robustness of SVM for classification tasks. Although KNN is faster to train, the optimized SVM offers a better trade-off between computation time and prediction quality. In addition, the performance of the SVM and KNN models was consistent on different data, as shown by the cross-validation results in Tables 2 and 3, making the a more reliable choice for the antenna selection task.

Thus, hyperparameter optimization had a significant impact on the performance and complexity of the SVM and KNN models. In particular, SVM showed a significant improvement in all metrics after optimization, while reducing the computational time. KNN also benefied from optimization, but to a lesser extent than SVM. These results demonstrate the importance of hyperparameter optimization in the development and improvement of machine learning models.

Our study suggests that higher model complexity is not associated with poor performance in antenna selection for 5G/6G systems. The proposed method may benefit from the integration of advanced ensemble techniques without adversely impacting the processing time and efficiency compared to traditional approaches noted in previous research. This study explored a comprehensive application of GridSearchCV for hyperparameter optimization in machine learning models like SVM and KNN. However, further and in-depth studies may be needed to confirm its robustness across different datasets and real-world scenarios, especially regarding the variability in wireless channel conditions and hardware constraints.

5. CONCLUSION

In this work, we proposed a method to predict the best antenna pairs for data transmission using the Alamouti code in the 2X1 system to further improve the overall performance of the MISO system. We used the GridSearchCV method to optimize the hyperparameters of the SVM and KNN machine learning models to select the antenna pairs. The results show that optimizing the hyperparameters led to a significant improvement in the performance of the SVM and KNN models. The significant improvement in the SVM indicates that the choice of hyperparameters allowed the algorithm to converge faster and make predictions more efficiently. When making a trade-off between performance and computation time, it is important to consider the size of the dataset and the resources available. SVM may be preferred for small datasets where perforance is critical, while KNN may be useful for large datasets with limited resources. In all cases, GridSearchCV will only test the combinations of hyperparameters specified in the grid. If the grid is too restricted, potentially better combinations may be missed. Our study demonstrates that optimized machine learning models, such as SVM and KNN, are more robust than traditional methods in selecting transmit antennas for 5G/6G systems. Future studies could explore the integration of more complex models or ensemble techniques, or the use of techniques such as Bayesian search or RandomizedSearchCV with feasible ways of producing further performance gains in real-time wireless communication systems.

Recent observations suggest that the optimization of machine learning models, specifically through hyperparameter tuning using GridSearchCV, significantly enhances the performance of TAS in 5G/6G systems. Our findings provide conclusive evidence that this phenomenon is associated with improved classification accuracy and reduction in computational complexity, not due to elevated numbers of exhaustive search methods.

ACKNOWLEDGMENTS

This work is partly supported by the Algerian Ministry of Higher Education and Research (PRFU) project N° A25N01UN080120220001, titled "Smart Techniques for 5G and 6G Wireless System".

REFERENCES

[1] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, no. 3, pp. 311–335, 1998, doi: 10.1023/A:1008889222784.

[2] Yang-Seok Choi, A. F. Molisch, M. Z. Win, and J. H. Winters, "Fast algorithms for antenna selection in MIMO systems," in *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484)*, Orlando, FL, USA: IEEE, 2003, pp. 1733-1737, vol. 3, doi: 10.1109/VETECF.2003.1285322.

[3] S. Sanayei and A. Nosratinia, "Antenna selection in MIMO systems," *IEEE Communications Magazine*, vol. 42, no. 10, pp. 68–73, Oct. 2004, doi: 10.1109/MCOM.2004.1341263.

[4] G. Tsoulos and G. V. Tsoulos, Eds., *MIMO System Technology for Wireless Communications*. in Electrical Engineering and applied Signal Processing Series. Boca Raton [Fl.]: Taylor & Francis, 2006.

[5] M. Lee and S. K. Oh, "A modified iterative water-filling algorithm with per-iteration power normalization in multiuser MMSE-precoded MIMO systems," in *2008 10th International Conference on Advanced Communication Technology*, Gangwon-Do, South Korea: IEEE, Feb. 2008, pp. 372–376, doi: 10.1109/ICACT.2008.4493781.

[6] A. Sibille, C. Oestges, and A. Zanella, *MIMO: from Theory to Implementation*. Burlington, MA: Academic Press/Elsevier, 2011.

[7] J. R. Hampton, *Introduction to MIMO Communications*, 1st ed. Cambridge University Press, 2013. doi: 10.1017/CBO9781107337527.

[8] J. V. Cuan-Cortes, C. Vargas-Rosales, and D. Munoz-Rodriguez, "MIMO channel capacity using antenna selection and water pouring," *Journal of Wireless Communications and Networking*, vol. 2014, no. 1, p. 228, Dec. 2014, doi: 10.1186/1687-1499-2014-228.

[9] B.-J. Lee, S.-L. Ju, N. Kim, and K.-S. Kim, "Enhanced transmit-antenna selection schemes for multiuser massive MIMO systems," *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–6, 2017, doi: 10.1155/2017/3463950.

[10] W. A. Shehab and Z. Al-qudah, "Singular value decomposition: principles and applications in multiple input multiple output communication system," *IJCNC*, vol. 9, no. 1, pp. 13–21, Jan. 2017, doi: 10.5121/ijcnc.2017.9102.

[11] V. S. Hendre, M. Murugan, and S. Kamthe, "Performance analysis of transmit antenna selection with MRC in MIMO for image transmission in multipath fading channels using simulink," *IJECE*, vol. 5, no. 1, p. 119, Feb. 2015, doi: 10.11591/ijece.v5i1.pp119-128.

[12] V. Hendre and P. Chinchamalatpure, "Implementation of TAS in MIMO using software defined radio," in *ICCCE 2019*, vol. 570, 2020, pp. 317–323, doi: 10.1007/978-981-13-8715-9_38.

[13] S. Kim, "Efficient transmit antenna subset selection for multiuser space–time line code systems," *Sensors*, vol. 21, no. 8, p. 2690, Apr. 2021, doi: 10.3390/s21082690.

[14] P. Yang, J. Zhu, Y. Xiao, and Z. Chen, "Antenna selection for MIMO system based on pattern recognition," *Digital Communications and Networks*, vol. 5, no. 1, pp. 34–39, Feb. 2019, doi: 10.1016/j.dcan.2018.10.001.

[15] Z. Wang and L. Vandendorpe, "Antenna selection for energy efficient MISO systems," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2758–2761, Dec. 2017, doi: 10.1109/LCOMM.2017.2753782.

[16] G. Jin, C. Zhao, Z. Fan, and J. Jin, "Antenna selection in TDD massive MIMO systems," *Mobile Networks and Applications*, vol. 26, no. 5, pp. 1831–1837, Oct. 2021, doi: 10.1007/s11036-019-01297-5.

[17] L. V. Nguyen, B. C. Nguyen, X. N. Tran, and L. T. Dung, "Transmit antenna selection for full-duplex spatial modulation multiple-input multiple-output system," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4777–4785, Dec. 2020, doi: 10.1109/JSYST.2019.2960599.

[18] L. Li, S. A. Vorobyov, and A. B. Gershman, "Transmit antenna selection based strategies in MISO communication systems with low-rate channel state feedback," *IEEE ransactions on Wireless Communications*, vol. 8, no. 4, pp. 1660–1666, Apr. 2009, doi: 10.1109/TWC.2009.080384.

[19] J. Joung, "Machine learning-based antenna selection in wireless communications," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2241–2244, Nov. 2016, doi: 10.1109/LCOMM.2016.2594776.

[20] T. X. Vu, L. Lei, S. Chatzinotas, and B. Ottersten, "Machine learning based antenna selection and power allocation in multi-user MISO Systems," in *2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, Avignon, France: IEEE, Jun. 2019, pp. 1–6. doi: 10.23919/WiOPT47501.2019.9144088.

[21] S. Gecgel, C. Goztepe, and G. K. Kurt, "Antenna selection on spatial modulation: a machine learning approach," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, Sivas, Turkey: IEEE, Apr. 2019, pp. 1–4, doi: 10.1109/SIU.2019.8806300.

[22] W. An, P. Zhang, J. Xu, H. Luo, L. Huang, and S. Zhong, "A novel machine learning aided antenna selection scheme for MIMO internet of things," *Sensors*, vol. 20, no. 8, p. 2250, Apr. 2020, doi: 10.3390/s20082250.

[23] T. X. Vu *et al.*, "Machine learning-enabled joint antenna selection and precoding design: from offline complexity to online performance," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3710–3722, Jun. 2021, doi: 10.1109/TWC.2021.3052973.

[24] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, 1-52, 2018, doi: 10.48550/ARXIV.1603.06560.

[25] B. Clerckx and C. Oestges, "Space-time coding with partial transmit channel knowledge," in *Mimo Wireless Networks*, Elsevier, 2013, pp. 335–384, doi: 10.1016/B978-0-12-385055-3.00010-9.

[26] Md. S. I. Sagar *et al.*, "Application of machine learning in electromagnetics: mini-review," *Electronics*, vol. 10, no. 22, p. 2752, Nov. 2021, doi: 10.3390/electronics10222752.

[27] AKSOY, Selim, "Non-bayesian classifiers Part I: k-nearest neighbor classifier and distance functions," B.m.: Department of Computer Engineering Bilkent University, 2009. [Online]. Available: https://www.cs.bilkent.edu.tr/~saksoy/courses/cs551-Spring2005/slides/cs551_nonbayesian1.pdf

[28] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, Apr. 2017, doi: 10.1109/MWC.2016.1500356WC.

[29] QI, Hairong, "COSC 522 – machine learning lecture 6 – nonparametric learning – kNearest Neighbor (kNN)," B.m.: Electrical Engineering and Computer Science University of Tennessee, Knoxville, 2020. [Online]. Available: https://web.eecs.utk.edu/~hqi/cosc522/

[30] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998, doi: 10.1023/A:1009715923555.

[31] AKSOY, Selim, “Non-Bayesian Classifiers Part II: linear discriminants and support vector machines,” B.m.: Department of Computer Engineering Bilkent University., 2009. [Online]. Available: https://www.cs.bilkent.edu.tr/~saksoy/courses/cs551/slides/cs551_nonbayesian2.pdf
[32] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, “Machine learning for wireless communications in the internet of things: a comprehensive survey,” *Ad Hoc Networks*, vol. 93, p. 101913, Oct. 2019, doi: 10.1016/j.adhoc.2019.101913.
[33] QI, Hairong, “COSC 522 – machine learning lecture 13: support vector machine,” B.m.: Electrical Engineering and Computer Science University of Tennessee, Knoxville., 2020. [Online]. Available: https://web.eecs.utk.edu/~hqi/cosc522/

BIOGRAPHIES OF AUTHORS

Abdellah Ouldammar was born in Tipaza (Algeria), and received a Bachelor’s degree in 2016, a License degree in Telecommunication, and a Master’s in Telecommunications Systems from Bechar University in 2021. Actually, He is a member of the Laboratory of Information Processing and Telecommunication (LTIT) at the University of Bechar, where he is preparing a Ph.D. thesis in Telecommunication Systems. His current research interests include the application of machine learning in 5G systems. He can be contacted at email: ouldammar.abdellah@univ-bechar.dz.

Abdelmounaim Moulay Lakhdar got the Engineering Degree in Telecommunication in 2000 at the Institute of Telecommunications in Oran. Magister was the second degree in Signal and telecom at Djillali LIABES university of Sidi Bel Abbes in 2003. From 2004 up present, he worked in the Bechar University as lecturer. Since May 2009, he graduated Ph.D. Es Sciences at Sidi Bel Abbes. He does his research activity at the Bechar University and Communications, Architecture and Media Laboratory (CAMR) (Djillali LIABES University) and he is being LTIT Laboratory director at Béchar university since 2015. His research interests are image transmission, image processing, and digital transmission performance. Correspondence address: Bechar University, Department of Electrical engineering, Bechar, Algeria. He can be contacted at email: moulay.abdelmouaim@univ-bechar.dz.

Ahmed Bouida was born in Bechar (Algeria), received a Bachelor's in 1989, an Engineer's degree in Electrical Engineering from the University Djillali LIABES of Sidi Bel Abbes (Algeria) in 1994, the Magister degree in Electronic from the University Djillali LIABES of Sidi Bel Abbes (Algeria) in 2008 (Algeria), and the Ph.D. degree in Embedded Electronic Systems from the University of Bechar. He is an assistant professor at the University of Bechar and a Laboratory of Information Processing and Telecommunication (LTIT) member. His current research interests include image processing, biometric image compression, wavelet transform, image quality assessment, image segmentation, and deep learning application. He can be contacted at email: Bouida.ahmed@univ-bechar.dz.

Khaled Merit is an Assistant Professor in Telecommunications at Tahri Mohammed University of Bechar (UTMB), Bechar, Algeria. He received a Magister degree in Telecommunications and an Engineering degree in Telecommunications with majors from National Institute of Telecommunication & ICT Oran (Algeria) in 2007 and 2011, respectively. Currently pursuing a Ph.D. degree at Tahri Mohammed University of Bechar, Bechar, Algeria. Mr. MERIT’s research interests include Pattern Recognition, Computer Vision, and Machine Learning. Correspondence address: Information Processing and Telecommunication Laboratory (LTIT), Tahri Mohammed University, Bechar 08000, Algeria. He can be contacted at email: merit.khaled@univ-bechar.dz.