Towards Data Analytics Based PIM Detection in Wireless Networks

Eren Deniz*§, Gokcan Cantali†, Ozcan Ozay*, Onur Yildirim*, Gürkan Gür‡, and Fatih Alagoz†

* Yupana Inc., Walnut Creek, USA, name.surname@yupanatech.com
† Dept. of Computer Eng., Bogazici University, İstanbul, Turkey, name.surname@boun.edu.tr
‡ Zurich University of Applied Sciences (ZHAW), Switzerland, gurkan.gur@zhaw.ch
§ Dept. of Computer Eng., Ege University, İzmir, Turkey, eren.deniz@mail.ege.edu.tr

***Abstract*—Passive Intermodulation (PIM) is a physical layer Radio Access Network (RAN) problem observed in both 4G and 5G networks. It is caused by internal physical processes such as inferior cabling or rusting, and external factors such as metallic obstacles in the radio propagation path. PIM degrades the user experience and radio resource efficiency while leading to an operation overhead for detecting and mitigating it on the operator side. Nevertheless, current solutions for PIM typically rely on costly hardware and site visit-based investigation by technicians. This work proposes a Machine Learning (ML) based PIM detection scheme for identifying PIM problems in RAN sites. Our approach relies on network KPI data already collected in the infrastructure for various purposes, including network monitoring, performance control, and maintenance. We investigate the performance of our proposed technique using empirical data collected from actual network cells.**

I. Introduction

Wireless and mobile networks have become the backbone of our civilization with the emergence of anytime-anywhere connected services and advances in end-user devices. This phenomenon was also starkly evident during the COVID-19 pandemic when ubiquitous broadband access and applications like remote working were critical for our daily lives [1]. Accordingly, the cellular network infrastructure has been expanding with more base stations and radio access nodes to facilitate sufficient connectivity and capacity. This trend is accompanied by more complex radio technologies and diverse radio environments for network devices with the deployment of 5G networks. As a result, new challenges regarding RAN management, maintenance, and user-experienced signal quality have arisen. A fundamental technical problem in that regard is the Passive Inter-Modulation (PIM) problem, which has been in effect since 4G networks, adversely impacting network coverage and signal quality in current wireless systems [2].

PIM occurs due to the nonlinearity of RF antennas or passive network elements such as connectors and cables, leading to an inferior signal quality caused by impairment of Quality of Service (QoS) and spectral efficiency [3]. In such scenario, the nonlinear function of the passive device can be characterized by the power series $y = \Sigma_{k=1}^{\infty} a_k x^k$ where x and y represent instantaneous input and output signals, respectively. The coefficient a_k depends on the nonlinear characteristics of the device, with k being the order of the power series. For instance, when the input signal is composed of two frequencies f_1 and f_2, there will be intermodulation signals in the output signal in addition to harmonic ones due to the combination of these two frequencies as shown in Fig. 1. These intermodulation signals around the fundamental frequency lead to the PIM issue. If this problem is caused by the physical conditions of network equipment degrading over time, it is called *Internal PIM*. Occasionally, the root cause of the PIM is an external factor unrelated to the base station itself. This so-called *External PIM* can occur when metallic objects such as a fence, billboard, or roof material are around the signal path. Unlike Internal PIM, typically emerging and showing its symptoms gradually over time, External PIM causes an abrupt degradation in signal and connection quality.

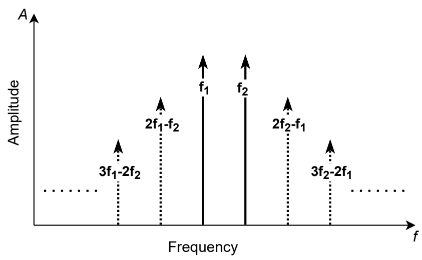


Fig. 1: Intermodulation products for two signals at f1 and f2 in PIM.

Several methods have been proposed in the past to detect and cancel PIM in base stations. However, such methods usually depend on manual work and incur further operational costs [4]. For instance, one of the most widely used methods for PIM detection is to generate an artificial load on the network cell to draw out the problematic branch or component. However, this is an undesirable approach due to deterioration in the network coverage during tests. Instead of such methods, digital and remote solutions for PIM detection have emerged recently [5]. Several methods have been studied for computer-aided PIM detection, utilizing various data-driven approaches, including machine learning (ML), statistics, and rule-based techniques. Although machine learning approaches for PIM detection usually use supervised learning models such as neural networks [6], some studies introduce novel ideas [7] as potential solutions. These methods usually work well

DOI: 10.1109/CAMAD59638.2023.10478405

under the assumption that a site is either healthy or not for the entire monitoring process. For scenarios where the PIM issue slowly develops on a site, anomaly detection techniques using time-series data [8] along with other classifiers such as random forests [9] seem to provide better results. In that vein, using data analytics approaches for anomaly detection [10] is promising for detecting PIM issues since PIM can be considered an anomaly in the network site.

Data analytics is already a key element for network management and maintenance in wireless systems. In that regard, data collection and processing in near real-time for anomalies, optimization, modeling, and performance monitoring essentially lend themselves to PIM problems. However, the surge in mobile devices such as IoT and more heterogeneous systems has made it increasingly difficult to identify and troubleshoot network issues promptly. The evolution of wireless technology and the development of next-generation networks, such as Beyond 5G or 6G, is expected to exacerbate this problem due to more complex radio environments, more stringent QoS requirements, and massive connectivity. Therefore, more advanced PIM solutions relying on data analytics and remote monitoring are an essential research topic. As such, novel anomaly detection techniques for PIM are crucial for ensuring mobile networks' efficient and secure operation.

To address this PIM detection challenge in wireless networks, we provide the following contributions in our work:

- We propose a novel data analytics technique, namely PIM Detection via One-class SVM (The P-DOS), for detecting cellular base stations with PIM problems. To this end, we frame the PIM detection as an inherent anomaly detection problem and exploit machine learning to enable this function. P-DOS processes temporal data of network Key Performance Indicators (KPIs) fetched from base stations.
- We investigate and identify the dominant KPIs relevant for the PIM detection objective based on our dataset.
- We provide an empirical performance characterization of P-DOS by evaluating its performance on a dataset entailing hourly KPI data of 1348 sites over two weeks. This examination is accompanied by a comparison of P-DOS performance to a baseline method using Random Forest models.

The rest of our paper is structured as follows. In Section II, we introduce our PIM detection model P-DOS. Next, we describe the Random Forest based approach, which we use for benchmarking our proposal in Section III. In Section IV, we present the data analysis of the collected KPI dataset followed by model details in Section V. We provide and discuss our experimental results in Section VI. Finally, Section VII concludes the paper and outlines future work.

II. PIM Detection via One-Class SVM (P-DOS)

A promising approach, which could potentially be an accurate PIM detection approach, is using one-class SVM models [11] due to the unbalanced data set nature. In this work, we propose the *PIM Detection via One-Class SVM (P-DOS)* method, which relies on one-class SVM, as the name suggests.

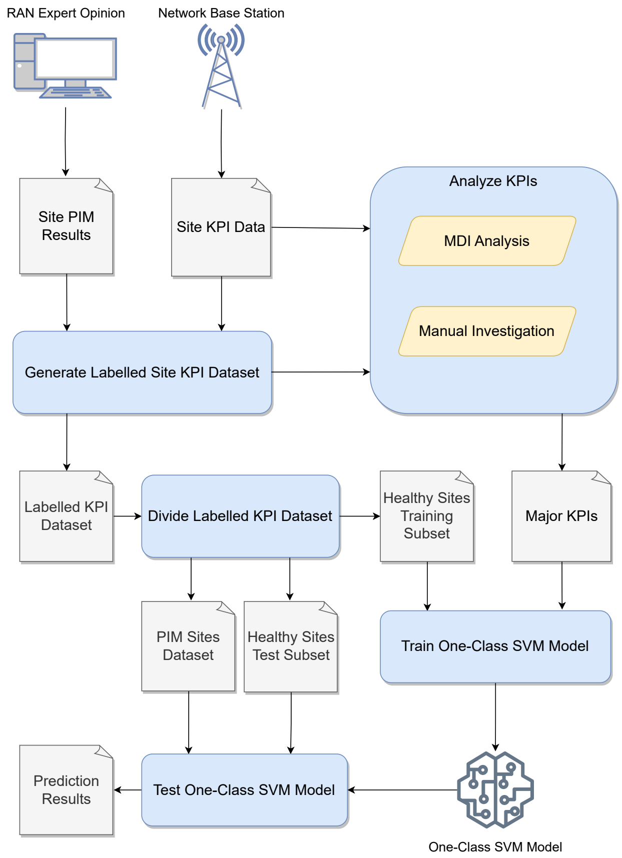


Fig. 2: High-level block diagram of the P-DOS approach

A. Rationale

Among various classification methods, a different approach is presented by the concept of one-class classification (OCC), which is a specific type of binary classification [12]. In OCC, the aim is to differentiate the instances of a particular class and eliminate any data that does not belong to that class. Thus, the learning process only utilizes the samples of this one class to learn its characteristics so that any foreign data can be filtered out. Due to its nature, OCC is a good candidate, hence commonly used, for anomaly detection and outlier detection problems where collected data samples tend to contain a large number of samples of the expected case and rare samples of unusual cases. In other words, OCC is a well-established approach for classification tasks with imbalanced datasets.

One-Class SVM is a widely used implementation of OCC, utilizing SVMs as the underlying model. Unlike regular SVM, a one-class SVM is an unsupervised machine learning model trained on data belonging to a single class and mainly used for anomaly/outlier detection problems. Considering a healthy cellular network, we can assume that the existence of PIM is an anomaly that should not occur in the usual operation of a base station. This rationale allows us to treat the PIM issue as an outlier detection problem that can be solved with a one-class SVM model.

B. Architecture

A high-level view of the P-DOS approach is shown in Fig. 2. As one can observe from the diagram, we use a collected dataset of healthy (no-PIM) sites that do not exhibit PIM effects and some unhealthy (PIM) sites that suffer from PIM problems. A subset of the healthy sites is used for training our one-class SVM model to allow the model to learn what kind of KPI values a non-PIM network site is supposed to manifest. The remaining healthy sites and all unhealthy sites are used in the testing step to evaluate the accuracy of the developed detection model.

III. Baseline Approach: Random Forest

For evaluating and benchmarking the P-DOS proposal, we compare it to a conventional binary classification-based approach. For the baseline method, we train a random forest classifier. Random forest [13] is a popular ensemble algorithm for classification tasks. The basic idea behind the random forest algorithm is to ensemble several decision trees. By aggregating multiple decision trees, the robustness and accuracy of the model increase. The predictions made by each tree in the random forest are combined to produce the final prediction.

There are many hyper-parameters available for the random forest model. In this work, we investigate two of them, namely $n\_estimators$ and max_depth. The parameter $n\_estimators$ represents the number of trees in the forest. Increasing the number of trees in the forest generally leads to a decrease in the variance of the model and an increase in its overall performance. However, a larger number of trees also increases the computational cost of training and prediction. The parameter max_depth represents the maximum depth of each decision tree in the forest. Increasing the value of max_depth will improve the model's performance; however, increasing it excessively causes the overfitting problem that limits the generalization of the model to new and unseen data. In summary, there is an implicit trade-off: a good balance between $n\_estimators$ and max_depth is vital for achieving good performance with random forest models.

IV. Data Analysis

In our collected dataset, we have 1288 healthy and 60 unhealthy sites, with two weeks of hourly KPI data per site. Among the healthy 1288 sites, 1200 sites are used for training, and 88 healthy sites along with 60 unhealthy sites are used for testing the model, as shown in Table I. Two weeks of KPI data are available for each site in the dataset. For identifying the salient KPIs, we followed a two-pronged methodology. First, during this study, we manually investigated different KPIs to understand which KPIs differ the most between healthy and unhealthy sites. For each KPI we have in our dataset, we plotted the values of that KPI for a healthy site and an unhealthy site, both selected randomly, in the same graph for manually detecting the differences. We repeated this combinatorial process for multiple sites and for each KPI to increase our confidence in making a decision on KPI relevance for the PIM phenomenon.

TABLE I: Experimental Dataset

Label\Phase	Training	Test
Healthy (no-PIM)	1200	88
Unhealthy (PIM)	-	60

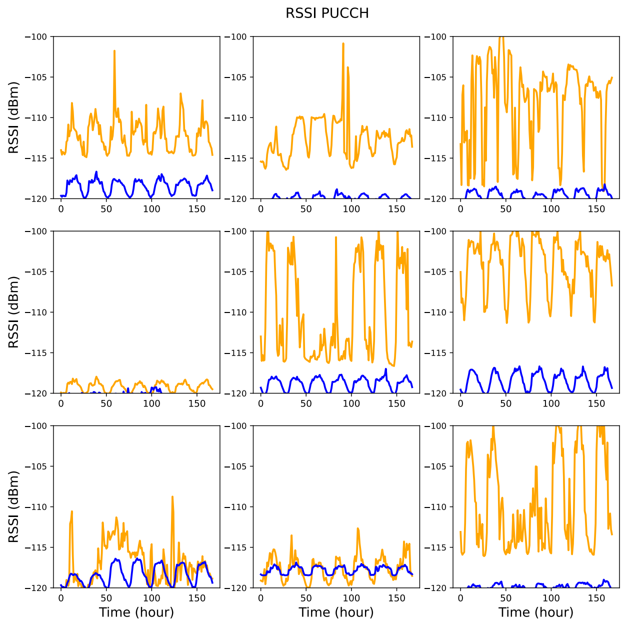


Fig. 3: Each plot displays the `RSSI_UL_INTERFERENCE_PUCCH` values for a pair of unhealthy (orange) and healthy (blue) sites.

After checking plots for multiple sites for each KPI, the most noticeable differences in values were observed in `RSSI_UL_INTERFERENCE_PUCCH` and `RSSI_UL_INTERFERENCE_PUSCH` KPIs (PUCCH and PUSCH, in short). They entail RSSI (Received Signal Strength Indicator) values for interference in the Uplink Control Channel (UCCH) and Uplink Shared Channel (USCH) in LTE networks. We illustrate their plots for nine example sites in Fig. 3 and Fig. 4, respectively. In addition to the manual investigation of KPIs via plots, the Mean Decrease in Impurity (MDI) method was also utilized as a feature selection technique [14] for verification of relevant KPIs. MDI allows us to detect the features that a classifier benefits the most when making predictions. The values obtained per KPI from the MDI analysis are provided in Fig. 5. One can observe that **Feature 1** and **Feature 2** have the biggest MDI value among other KPIs, which correspond to `PUCCH` and `PUSCH`, respectively. Since MDI results are consistent with our manual investigation, we proceed with these two KPIs and train one-class SVM models.

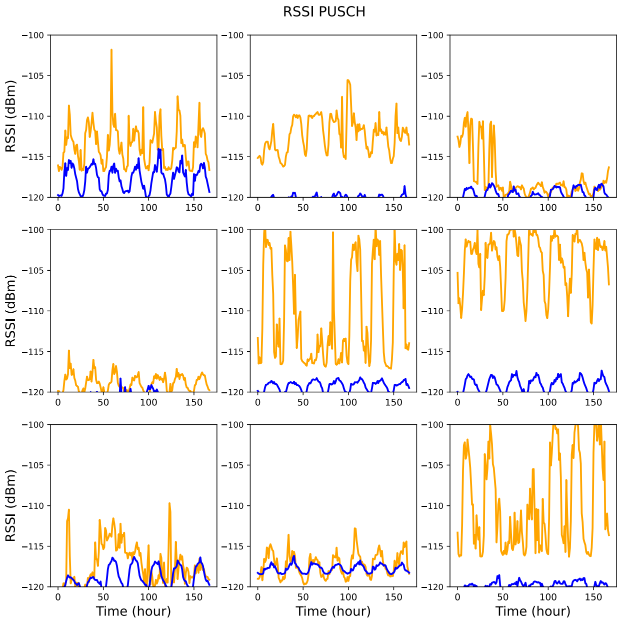


Fig. 4: Each plot displays the `RSSI_UL_INTERFERENCE_PUSCH` values for a pair of unhealthy (orange) and healthy (blue) sites.

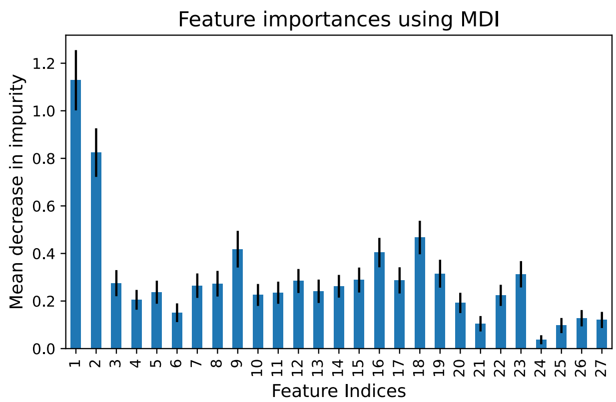


Fig. 5: MDI analysis on network site KPIs.

V. Configuration of ML Models

In this section, we provide details regarding the training and hyper-parameter selection of our models.

A. Random Forest

For the random forest model, two hyper-parameters, $n\_estimators$ and max_depth, were investigated. Several different values have been tried for both hyper-parameters to obtain the optimum result. Table III shows all the parameter values we have investigated for training the model. In our case, the best result has been observed with the parameter values $n\_estimators$ = 10, max_depth = 2.

TABLE II: One-class SVM Parameters

Name	Range	Step
ν	0.01 - 0.20	0.01
γ	0.01 - 0.10	0.01
kernel	"Linear" | "RBF" | "Poly" | "Sigmoid"	-

TABLE III: Random Forest Parameters

Name	Range	Step
n_estimators	10 - 100	10
max_depth	2 - 20	2

B. The P-DOS Scheme

While training our one-class SVM models, we use a variety of values for each of the model parameters. One-class SVM algorithm has three main parameters: nu (ν), gamma (γ), and kernel. These parameters are fine-tuned, as given in Table II, to provide a more accurate prediction for the trained model. Based on the results of our training and validation phases, the optimal values for the parameters are obtained as $\nu = 0.01$, $\gamma = 0.05$, and kernel = "rbf". We train two one-class SVM models based on PUCCH and PUSCH signals using these parameter values.

VI. Experiment Results

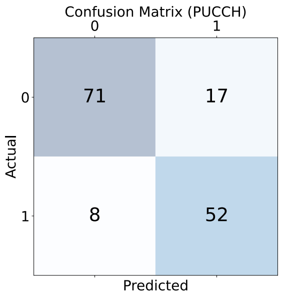
In this section, we present detailed performance metrics for the P-DOS models. Also, as a baseline method to compare the success of our models, we train two Random Forest (RF) models based on *PUCCH* and *PUSCH* values. We apply our test dataset to each of the four models for a comprehensive study.

A. Performance Results

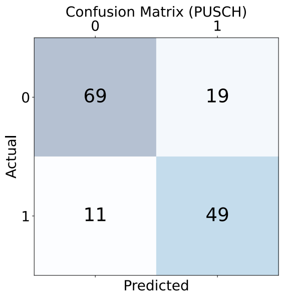
For evaluating the performance of the trained models, we use the following performance metrics: Precision, Recall, Accuracy, and F1 score. Results of our experiments for both the P-DOS and RF models are given in Table IV. We deduce from the experiment results that the performance of one of the P-DOS models is quite similar to the results obtained from the other model. Both models achieve at least a score of 70% for each performance metric, which can be considered a success. We also observe that the model for PUCCH performs slightly better than the model for PUSCH. This minor performance difference is consistent with our previous study since both the MDI analysis and manual investigation show that PUCCH is more relevant than PUSCH for the PIM problem. We also compare the success of our models with the trained RF models. Based on the results, one can observe that although each P-DOS model has a lower precision score than their RF counterparts, each P-DOS model achieves a vastly greater recall score than both RF models. Moreover, when we consider F1 scores, the P-DOS models significantly outperform RF

TABLE IV: Performance Results

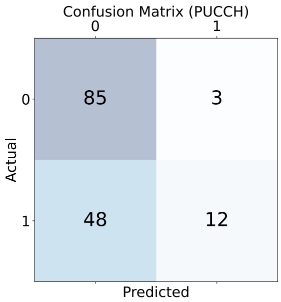
	OC-SVM PUCCH	OC-SVM PUSCH	RF PUCCH	RF PUSCH
Precision	0.75	0.72	0.80	1.0
Recall	0.87	0.82	0.20	0.11
Accuracy	0.83	0.80	0.65	0.64
F1 Score	0.81	0.77	0.32	0.21



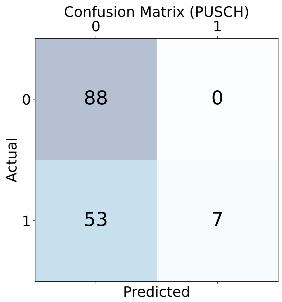
(a) The P-DOS Confusion Matrix (PUCCH-based model)



(b) The P-DOS Confusion Matrix (PUSCH-based model)



(c) RF Confusion Matrix (PUCCH-based model)



(d) RF Confusion Matrix (PUSCH-based model)

Fig. 6: Confusion matrix for the P-DOS and RF with different KPIs.

models. Regarding accuracy, the P-DOS models still perform better than RF models; however, the difference between their scores is less significant compared to recall or F1 scores. Overall, the better-performing RF model only achieves a 32% F1 score and 65% accuracy, while the better-performing P-DOS model achieves an 81% F1 score and 83% accuracy.

Along with our performance metrics, we also plot the confusion matrix for the PUCCH-based and PUSCH-based models, as shown in Fig. 6a and Fig. 6b, respectively. In both matrices, *0* represents the healthy case, and *1* represents the PIM case. One can observe from these matrices that the PUCCH-based and PUSCH-based models successfully detect PIM issues for 52 and 49 sites, respectively, out of all the 60 unhealthy sites. Moreover, out of all the 88 healthy sites, PUCCH-based and PUSCH-based models successfully mark 71 and 69 of them as healthy sites, respectively. Overall, the PUCCH-based model has 17 false positives and eight false negatives, while the PUSCH-based model has 19 false positives and 11 false negatives.

TABLE V: F1 scores with changing hyperparameters using PUSCH.

	ν=0.01	ν=0.05	ν=0.1	ν=0.2
γ=0.01	0.54	0.54	0.60	0.69
γ=0.02	0.67	0.67	0.67	0.69
γ=0.05	0.77	0.77	0.77	0.77
γ=0.1	0.76	0.76	0.76	0.76

TABLE VI: F1 scores with changing hyperparameters using PUCCH.

	ν=0.01	ν=0.05	ν=0.1	ν=0.2
γ=0.01	0.58	0.58	0.60	0.72
γ=0.02	0.71	0.71	0.71	0.74
γ=0.05	0.81	0.81	0.81	0.81
γ=0.1	0.80	0.80	0.80	0.80

In addition to one-class SVM models, we also provide the confusion matrices for the results of RF models, based on PUCCH (in Fig. 6c) and PUSCH (in Fig. 6d). Results show that PUCCH-based and PUSCH-based RF models correctly identify 85 and 88, respectively, sites out of all 88 healthy sites. However, the PUCCH-based model only detects 12, and the PUSCH-based model only detects 7 unhealthy sites out of all 60 sites with PIM problems. Therefore, the PUCCH-based RF model contains three false positives and 48 false negatives, while the PUSCH-based RF model contains 0 false positives and 53 false negatives. Although these models have a very high precision score, especially the PUSCH-based model, the recall score is much lower than an acceptable value, significantly decreasing the accuracy and F1 score. Overall, with such a high number of false negatives, these RF models are not well suited for a successful PIM detection task.

B. Effects of Hyper-Parameters

During our study, we find the optimal values for the hyper-parameters of the P-DOS, γ and ν, as 0.05 and 0.01, respectively. To provide an insight into how the hyper-parameter values affect PIM detection results, the obtained F1 scores using different γ and ν values are given in Table V and Table VI. The reason for choosing the F1 score for this analysis, instead of other metrics such as accuracy, precision, or recall, is that the PIM detection problem usually relies on an unbalanced training set, with a much higher number of healthy sites compared to sites that are actually suffering from a PIM problem. Thus, choosing the F1 score as the main metric allows us to make a healthier interpretation.

According to the data in Table V, selecting low values for both ν and γ results in a visible decrease in F1 score on the PUSCH-based model. Increasing ν from 0.01 to 0.2 seems to improve the results when $\gamma = 0.01$. Furthermore, increasing γ

from 0.01 to 0.1 seem to improve the result, although $\gamma = 0.05$ produces a slightly better score than $\gamma = 0.10$. Also, note that the effect of ν on the F1 score seems to diminish when γ has large values. For instance, when $\gamma = 0.01$, increasing ν from 0.01 to 0.2 allows F1 score to increase by 15%. On the other hand, when $\gamma = 0.02$, changing ν from 0.01 to 0.2 increases F1 score by only 2%. In fact, when $\gamma = 0.05$ or $\gamma = 0.1$, changing ν does not have any effect on the F1 score at all. From the analysis, we observe that γ has a more direct effect on the results than ν, which shows its effect when we provide low values for γ.

A similar analysis can also be performed for the PUCCH-based model, using the data in Table VI. Again, ν affects the F1 score more when γ has lower values. When $\gamma = 0.01$, increasing ν from 0.01 to 0.2 improves the F1 score by 14%. Similarly, a 3% change is observed for the same adjustment for ν when $\gamma = 0.02$. As seen in the analysis of the PUSCH-based model, changing ν when $\gamma = 0.05$ or $\gamma = 0.1$ does not have any effect on the F1 score for the PUCCH-based model either. Increasing γ from 0.01 to 0.10, on the other hand, seems to improve the results significantly, except when $\gamma = 0.05$, which produces the optimal result.

VII. Conclusion

In this work, we have proposed a novel PIM detection technique based on an anomaly detection approach, denoted as the P-DOS. We compared its performance to a baseline method that adopts an alternative ML model. For this purpose, the proposed approach was tested on a KPI data set collected over two weeks from operational wireless network cells. We have also identified the optimum configuration of the P-DOS (i.e., hyperparameters) by investigating different parameter values. The P-DOS has outperformed the baseline method and provided favorable detection results.

In future research, we would like to test our proposal with different datasets from different network locations and frequency bands for a more comprehensive performance evaluation.

Acknowledgment

The authors would like to thank H. Ferit Eniser from MPI-SWS, Germany, for his valuable comments and insights.

References

[1] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, and G. Smaragdakis, "A year in lockdown: How the waves of COVID-19 impact internet traffic," *Commun. ACM*, vol. 64, p. 101–108, jun 2021.

[2] G. Cantali, E. Deniz, O. Ozay, O. Yildirim, G. Gür, and F. Alagoz, "PIM detection in wireless networks as an anomaly detection problem," in *2023 International Balkan Conference on Communications and Networking (BalkanCom)*, pp. 1–6, 2023.

[3] Z. Cai, L. Liu, F. de Paulis, and Y. Qi, "Passive intermodulation measurement: Challenges and solutions," *Engineering*, vol. 14, pp. 181–191, 2022.

[4] H. M. Karaca, "Passive inter-modulation sources and cancellation methods," *The European Journal of Research and Development*, vol. 2, p. 75–91, Jun. 2022.

[5] M. Z. Waheed, D. Korpi, L. Anttila, A. Kiayani, M. Kosunen, K. Stadius, P. P. Campo, M. Turunen, M. Allén, J. Ryynänen, and M. Valkama, "Passive intermodulation in simultaneous transmit–receive systems: Modeling and digital cancellation methods," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 9, pp. 3633–3652, 2020.

[6] B. Liang, X. Bu, M. Li, P. Guo, and C. Liu, "A novel RTRLNN model for passive intermodulation cancellation in satellite communications," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 18–23, 2018.

[7] F. B. Mismar, "Intermodulation interference detection in 6G networks: A machine learning approach."

[8] S. Banerjee, R. R. Martin, and A. Pardo, "Frequency-aware time series forecasting, anomaly detection, classification and granger causality," in *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pp. 217–221, 2022.

[9] H. Ranjani, S. P. Puthennurakel, A. Brisebois, S. Banerjee, and V. Umaashankar, "Time series based approach for detecting passive intermodulation occurrences in cellular network," in *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pp. 630–638, 2022.

[10] G. Muruti, F. A. Rahim, and Z.-A. bin Ibrahim, "A survey on anomalies detection techniques and measurement methods," in *2018 IEEE Conference on Application, Information and Network Security (AINS)*, pp. 81–86, 2018.

[11] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computing*, vol. 13, no. 7, pp. 1443–1471, 2001.

[12] M. M. Moya and D. R. Hush, "Network constraints and multi-objective optimization for one-class classification," *Neural Networks*, vol. 9, no. 3, pp. 463–474, 1996.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Advances in Neural Information Processing Systems* (C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.