

# Prediction of SLAM ATE Using an Ensemble Learning Regression Model and 1-D Global Pooling of Data Characterization

Islam Ali  
Department of Computing Science  
University of Alberta  
Edmonton, Canada  
iaali@ualberta.ca

Bingqing(Selina) Wan  
Department of Engineering Science  
University of Toronto  
Toronto, Canada  
b.wan@mail.utoronto.ca

Hong Zhang  
Department of Computing Science  
University of Alberta  
Edmonton, Canada  
hzhang@ualberta.ca

**Abstract**—Robustness and resilience in simultaneous localization and mapping (SLAM) are critical requirements for modern autonomous robotic systems. One of the essential steps to achieving robustness and resilience is the ability of SLAM to have an integrity measure for its estimates, thus having internal fault tolerance mechanisms to deal with performance degradation. In this work, we introduce a novel method for predicting SLAM localization error based on the characterization of raw sensor inputs. The proposed method relies on using a random forest regression model trained on 1-D global pooled features generated from characterized raw sensor data. The model is validated by using it to predict the performance of ORB-SLAM3 on three different datasets running in four different operating modes, resulting in an average prediction accuracy of up to 93.1 % and 80.45 % for ATE and APE, respectively. Then, the paper studies the quality of prediction with limited training data and proves that we can maintain proper ATE and APE prediction quality when training on only 20 % and 40% of the data, respectively. Finally, the paper discusses the impact of out-of-distribution predictions on prediction accuracy.

**Index Terms**—SLAM performance, random forest regression, SLAM robustness, SLAM resilience

## I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is a fundamental building block that gives modern robotic systems the ability to estimate its location while building a map of the navigated environment [1]. Over the last few decades, SLAM research has evolved significantly in terms of architecture, accuracy, requirements, and challenges [2]. One of the major challenges faced by SLAM is the robustness and resilience of the system when deployed in the real world [3]. Robustness of SLAM is the ability of the system to provide acceptable performance when operating under predefined conditions. Resilience is the ability of a system to converge to an acceptable performance when operating outside of the predefined conditions, which implicitly highlights the importance of having internal error prediction and tolerance mechanisms in SLAM to allow for this convergence to happen effectively [4]. For that reason, researchers have directed their attention towards the introduction of integrity indicators of either some blocks in the SLAM pipeline [5], or the final SLAM outcome [6].

*Absolute Trajectory Error (ATE)* [7] is considered the de-facto metric for measuring the accuracy of localization in SLAM and is used by most state-of-the-art solutions such as ORB-SLAM3 [8], VINS-Mono [9], among many others. *ATE*

is defined to be the root mean square (RMS) of the *Absolute Pose Error (APE)*, which is the instantaneous error between corresponding poses in the traversed trajectory. The relation between *ATE* and *APE* is given by:

$$APE = \|\hat{X}_i - X_i\| \quad (1)$$

$$ATE = \sqrt{\left(\frac{1}{N} \sum_{n=0}^N (APE)^2\right)} \quad (2)$$

Where  $\hat{X}_i$  and  $X_i$  are the estimated and ground truth pose at keyframe  $i$  respectively.

Therefore, on-line prediction of SLAM ATE is an integral part of the quest to reach robust and resilient SLAM as it provides SLAM systems with internal indicators of the integrity of their estimates, which can be used to correct estimation errors, govern switching between localization alternatives, and improve robotics safety when deployed.

In this paper, we propose a novel methodology for predicting the absolute trajectory error (ATE) of a SLAM algorithm using 1-D global pooling of input data characteristics and an ensemble learning-based regression model. This methodology is motivated by the high correlation observed and reported in our previous work [4] between the SLAM performance of multiple algorithms on one side, and the characterization metrics measured on different SLAM datasets on the other side. Since *ATE* is considered a coarse performance metric, the method was also evaluated for suitability to predict *APE* as well.

The rest of the paper is organized as follows. Section II presents a brief review of related work. Then, Section III provides a background overview. Next, Section IV describes our proposed method in detail. After that, results are presented and discussed in Section V. Finally, Section VI presents our conclusions from this study.

## II. RELATED WORK

Predicting and estimating system performance is crucial for the safe use of robots and autonomous systems and has been extensively studied in closely related disciplines. For example, navigation systems like INS/GPS use statistical models to estimate errors in sensor measurements and improve localization through Kalman filters [10] [11]. Additionally,

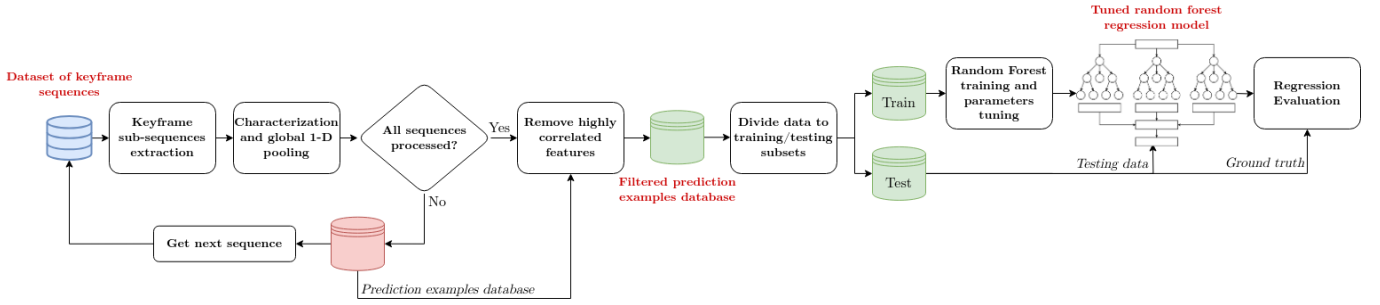


Fig. 1: A block diagram of the proposed SLAM errors prediction methodology

integrity measures for robot localization have been proposed to minimize deployment risks in real-world scenarios [12]–[13]. Despite being essential for robust and resilient perception and navigation, there is limited research on the integrity of localization outcomes of SLAM due to unclear design requirements, as reported in [14].

The existing literature on integrity measures for SLAM has primarily focused on two areas. The first area is predicting the overall performance of the entire SLAM pipeline. This approach aims to evaluate the integrity of the entire SLAM pipeline as a whole and provide a single measure for SLAM output integrity. For instance, the traveled path of a robot is modeled using Voronoi Graphs to train a model for predicting SLAM performance [6]. Training data are acquired using the method outlined in [15], which generates training examples by simulating a SLAM algorithm several times on selected environments. The same training examples generation methodology is used in [16] where a number of univariate and multivariate linear regression models are trained to predict the normalized relative translational error and the absolute trajectory errors. The two approaches proposed are useful in determining the overall performance of SLAM but would not allow for on-line prediction of localization integrity. That is due to their reliance on an overall descriptor of the whole path the robot will traverse rather than incremental raw sensor data (e.g., images). On the other hand, other methods are proposed to identify an upper bound of the localization uncertainty in SLAM providing a guarantee of the system performance when the spatial distribution of features is known [17].

The second area is investigating the integrity of specific components within the SLAM pipeline. This approach aims to evaluate the integrity of individual SLAM components and how the quantification of this integrity measure can be utilized to properly correct potential anomalies in localization estimates. For example, in [5] a learning-based integrity measure for visual loop closure is proposed to decrease false positives and ultimately improve the overall performance of SLAM localization accuracy through the reduction of loop closure false positives.

Our approach is unique in both its design and goal. We use a sequence of key-frame measurements such as images and/or inertial measurements as inputs, which are characterized to generate a corresponding characterization matrix

of the sequence traversed. Then, we apply a 1-D global pooling function on the rows of the characterization matrix, which results in a 1-D vector descriptor of the sequence. After that, the descriptor is sent to a prediction model to predict the expected ATE at the end of the input sequence. Consequently, this approach allows for on-line monitoring of the system performance and provides an integrity measure of localization estimates at any time. This is essential for ensuring the robustness and resilience of the SLAM system, particularly in challenging environments or under uncertain conditions.

### III. BACKGROUND

To provide a foundation for this study, this section explores two concepts: 1-D global pooling and random forest regression. We discuss how these techniques are used and examine their applicability to the proposed work.

#### A. 1-D Global Pooling

This technique was introduced in [18] as a solution to the problem of overfitting in neural networks. The technique does this by reducing the spatial dimension of a feature map to a single value using a global pooling function (e.g., average, min, max, etc.) across all features. Essentially, this reduction replaces a detailed feature map with an abstract, descriptive characteristic of it. Learning those characteristics instead of the examples themselves was proven to enhance the generalization of the learned model [19]. In this work, we examined different statistical 1-D global pooling functions and their impact on prediction quality, which resulted in choosing 1-D global average pooling (GAP) due to its superior performance compared to others.

#### B. Random Forest Regression

Random forest is an ensemble learning technique that relies on the concept of bagging [20] where several decision tree prediction models are trained on independent random sub-samples of the input features in the bootstrapping phases. Bootstrapping is a statistical technique that involves random sub-sampling of the training data pool while allowing replacement [21] to generate bootstraps. For each decision tree in the random forest, a bootstrap is selected for training on a random subset of available descriptor features. The outcomes of all decision trees are then combined either by averaging (regression) or by

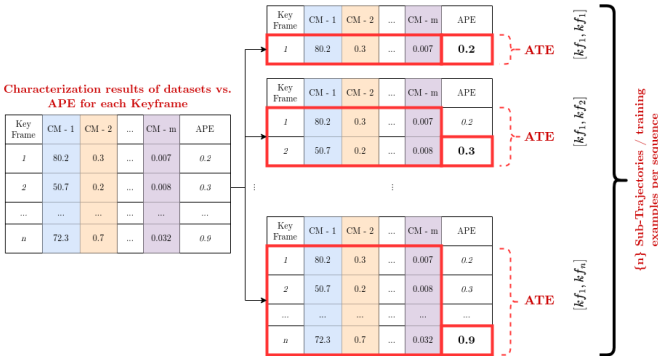


Fig. 2: Extraction of training examples based on sub-trajectories and corresponding SLAM errors (APE and ATE)

majority voting (classification) [22]. Due to the independence and low correlation among decision trees, the prediction error is not accumulated or propagated, thus resulting in a lower prediction error.

Random forests provide a way to balance accuracy and generalization and were proven to be superior to competing methods. For instance, they were proven to outperform neural networks [23] on tabular structured data and handle overfitting well when compared to boosting algorithms [24].

#### IV. METHODOLOGY

This section presents the proposed methodology to predict error in SLAM, along with an overview of the design choices made in this work. Figure 1 illustrates the proposed pipeline and shows how the different system components interact with each other. The figure provides a visual representation of the flow of data and processes in the proposed approach.

Given a dataset  $\mathcal{D}$  of  $N$  sequences, defined as:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \dots, N\} \quad (3)$$

where  $\mathbf{x}_i$  is a characterization matrix of size  $(m \times n)$  corresponds to  $m$  characterization metrics applied on an input sequence of  $n$  measurements (e.g. images/inertial measurements) and  $y_i$  is a scalar corresponding to the ATE of the trajectory.

Each characterization matrix  $\mathbf{x}_i$  is transformed to a 1-D vector  $\mathbf{z}_i$  of size  $(m \times 1)$  by applying a 1-D global pooling function  $f(\cdot)$  as such:

$$\mathbf{z}_i = f(\mathbf{x}_i) \quad (4)$$

Consequently, the transformed dataset  $\mathcal{D}'$  is defined as:

$$\mathcal{D}' = \{(\mathbf{z}_i, y_i), i = 1, \dots, N\} \quad (5)$$

We seek to learn a model to predict SLAM errors  $\hat{y}_i$  given an unseen 1-D vector  $\mathbf{z}_i$  from the transformed dataset  $\mathcal{D}'$ .

##### A. Data example generation

To generate examples for training the error prediction model, we run a SLAM algorithm on all sequences available in several datasets. For each of the run sequences, a number of sub-sequences are calculated that corresponds to the keyframes

TABLE I: Tuned Hyperparameters in the random forest and their corresponding ranges

Hyperparameter	Range
Number of tree estimators	[10, 1000]
Minimum sample required for a split	{2,5,10}
Minimum samples required at a leaf	{1,2,4}
Maximum number of features used for a split	{None, sqrt, log2}
Maximum depth a tree can grow up to	[10,100]
Bootstrapping for tree building	{True, False}

selected by the SLAM algorithm. Thus, we utilize the concept of sub-trajectories [7] in order to expand the number of training examples.

Given an input sequence of size  $\mathcal{K}$  keyframes, we can extract  $\mathcal{K}$  examples, where each example is a sub sequence of keyframes  $(kf)$  in the inclusive range of  $[(kf)_1, (kf)_k]$  where  $k = \{1, 2, \dots, \mathcal{K}\}$ . The corresponding error of each sub trajectory is calculated and is associated with each trajectory to construct a training example. Figure 2 illustrate the process in detail and shows how the training examples extraction and error association take place. Sub-trajectories used for model training and testing are generated sequentially from available data sequences in a dataset running in a specific operation mode. The split of the available data for training and testing is done without randomization, meaning that training and testing are conducted on sub-sequences generated from the same dataset but from different sequences.

##### B. Sequence characterization and 1-D global pooling

Each generated sub-sequence is considered to be an independent sequence of images/sensor readings. We apply the characterization framework introduced in [4] which contains an array of characterization metrics (e.g. measuring brightness, contrast ... etc.) that generate a characterization vector for each image/sensor reading in the sequence. As seen in Figure 3, characterization generates a 2D matrix of size  $(m \times n)$ , where each row represents a characterization metric outcome, and each column represents an input sub-sequence. Due to the variability in sequence sizes, the generated 2D matrices are not of the same dimension. Thus, to reduce the dimensionality and provide unified feature vectors for training, we apply a 1-D global pooling function on 2-D matrices to generate 1-D vectors of unified size of  $(m \times 1)$ . This is achieved by reducing each row in the characterization matrix into a single scalar value using the pooling function. In this work, we utilize one of 12 different pooling functions that include statistical pooling functions (e.g. mean, min, max ... etc.) and diversity pooling functions (e.g. entropy, simpson diversity index and its variants). In order to provide the prediction model with more descriptive features, we also concatenate all 1-D global pooled features into a single feature vector, study its impact on the prediction quality, and compare its performance to using a single 1-D global pooling function.

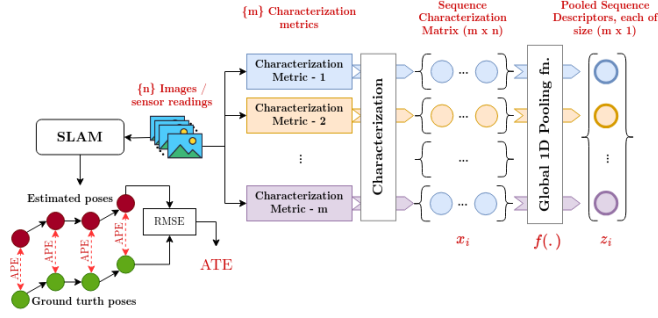


Fig. 3: Generation of feature vectors using input sub-sequence characterization and 1-D global pooling

### C. Removal of highly correlated features

The existence of collinearity between independent input features is a potential problem in regression and can lead to numerically unstable results [25]. To detect highly correlated features, we calculate Pearson correlation coefficient (PMCC) [26] between each feature and all other available features. After that, highly correlated features are grouped where the PMCC between any two features in a group is greater than a threshold of 95%. Then, only one feature is selected from each group which is then used for the training of the regression model in order to ensure prediction stability of the trained regression model.

### D. Random forest regression model

A random forest regression model is trained and tuned on 70 % of the data examples available for each test case. After that, the model is tested on the remaining unseen 30 % in order to determine its performance. We utilize the random forest regression implementation provided in scikit learn library [27] due to its efficiency and ease-of-use. It also exposes a number of hyperparameters that we can tune for optimal performance of the model.

Tuning the random forest hyperparameters is essential to achieve the best prediction performance. For that, we perform a randomized grid search with cross validation on the multi-dimensional space of hyperparameters provided in Table I. This method is proven efficient in selecting the best hyperparameters while maintaining reasonable complexity and execution time [28].

### E. Performance Evaluation

To quantitatively evaluate the regression quality of our method, four different metrics are utilized, which are defined as follows.

- 1) Coefficient of determination ( $R^2$ )

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=0}^n (y_i - \bar{y})^2} \quad (6)$$

- 2) Mean absolute percentage error (MAPE)

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (7)$$

TABLE II: The number of sub-trajectories available from each dataset at different operation mode.

Dataset	# Seq	# sub-trajectories available			
		M	S	M-I	S-I
KITTI	22	11799	23201	-	-
EuroC-MAV	11	3348	1956	3043	1484
TUM-VI	28	2049	1161	4230	1924

\* M, S, M-I, and S-I refer to monocular, stereo, monocular-inertial, and stereo-inertial respectively.

- 3) Mean absolute error (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (8)$$

- 4) Root mean squared errors (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (9)$$

Where  $y$  is the ground truth,  $\hat{y}$  is the predicted value, and  $n$  is the number of testing samples.

Those metrics differ in terms of their allowable range, and their indication of the quality of performance. Together, they give a clear indication of the performance and suppress any corner case or anomalies any metric can suffer from.

## V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we describe and discuss our experimental setup and associated experimental results. As mentioned in Section IV, we run ORB-SLAM3 [8] on three different datasets and in four different modes of operations, resulting in 10 test cases as illustrated in Table II. For each test case, we train and tune the random forest, and evaluate the model performance. Additionally, we study the impact of reducing the amount of training data on the ATE prediction quality to show how our proposed prediction model can still perform relatively well when limited data is available for training.

After that, the same experiments are repeated to predict SLAM APE instead of ATE to evaluate the suitability of the proposed method for the prediction of instantaneous errors of SLAM.

The experimental results show that the proposed method is able to predict SLAM ATE with a mean accuracy of 93.1 %. On the other hand, the same methodology was able to predict APE with a mean accuracy of 80.45 %, which is a direct indication of the efficacy of our method, the validity of using the characterization metrics as data descriptors, and the proper choice of the 1-D global pooling function for the SLAM ATE/APE prediction task.

### A. Training data generation

To generate examples for training the prediction model, we ran ORB-SLAM3 [8] on all sequences available in three different datasets, which are: KITTI [29], EuroC-MAV [30], and TUM-VI [31]. We apply our proposed data example generation process, which resulted in a great increase in

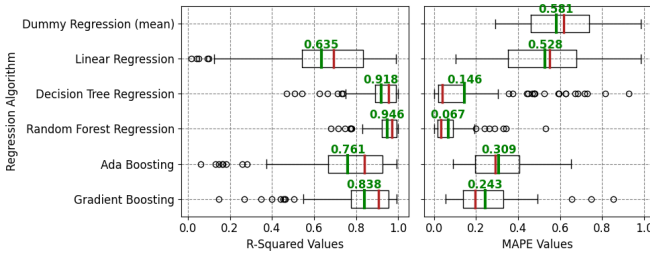


Fig. 4: Quantitative comparison of different regression Models for ATE prediction

TABLE III: Comparison between using the APE and ATE at 40 % and 20 % of the trajectory respectively as a predictor for the whole trajectory APE and ATE vs. our proposed random forest method when trained on similar available data

Mode - Dataset	Baseline (APE)		Random forest (APE)		Baseline (ATE)		Random forest (ATE)	
	$R^2$	MAPE	$R^2$	MAPE	$R^2$	MAPE	$R^2$	MAPE
M-EuroC	-12.9044	1.3333	0.9606	0.1061	0.1557	0.3683	0.9973	0.0106
M-KITTI	0.8171	0.5775	0.9976	0.0280	0.7621	0.3710	0.9992	0.0079
M-TUMVI	-1.2853	0.7225	0.6807	0.3281	0.9036	0.2503	0.9663	0.1246
MI-EuroC	0.1706	0.5437	0.8006	0.1686	-0.0637	0.2909	0.9888	0.0199
MI-TUMVI	-1.3701	0.8984	0.7945	0.2999	0.8571	0.2539	0.9835	0.1221
S-EuroC	0.3305	0.8435	0.9896	0.1562	0.9899	0.2699	0.9993	0.0125
S-KITTI	0.5943	0.4515	0.9995	0.0147	0.9559	0.1627	0.9999	0.0013
S-TUMVI	0.0979	0.5595	0.6494	0.3324	0.6202	0.1904	0.8458	0.0359
SI-EuroC	0.2107	1.1129	0.4613	0.2364	0.6864	0.1954	0.9900	0.0189
SI-TUMVI	-5.6590	1.3394	0.6079	0.3760	0.9381	0.1681	0.9723	0.1979
Mean	-1.8998	0.8382	0.7942	0.2046	0.6805	0.2521	0.9743	0.0552

the number of available examples for training, testing, and validation. The method is applied on four different modes of ORB-SLAM3 [8] which are monocular, monocular-inertial, stereo, and stereo-inertial, resulting in 10 different test cases. Table II shows the number of training examples generated for each of the test cases.

### B. Selection of regression algorithm

In order to validate our selection of the regression algorithm, we examined a number of famous regression models with their default hyperparameter values provided in [27]. These algorithms are: dummy regression that takes the average of input features, linear regression, decision tree, random forest, Ada boosting, and gradient boosting. The evaluation is done using  $R^2$  and  $MAPE$  metrics to allow comparison of different test cases as they provide an absolute measure of performance regardless of the value and range of the predicted variable. As shown in Figure 4, random forests outperform other regression algorithms resulting in the highest  $R^2$  value and the lowest  $MAPE$  value as well. Additionally, we can clearly observe the overfitting problem of boosting algorithms [24] when we examine Ada boosting and gradient boosting performance compared to random forests.

### C. Performance comparison to baseline

Due to the nature of SLAM errors and how they evolve over time, we aspire to compare our model when trained on only 20 % of the data to in case of ATE and 40 % of the data in case of APE to corresponding observed errors after traversing the same percentage of the a given data example. This experiments

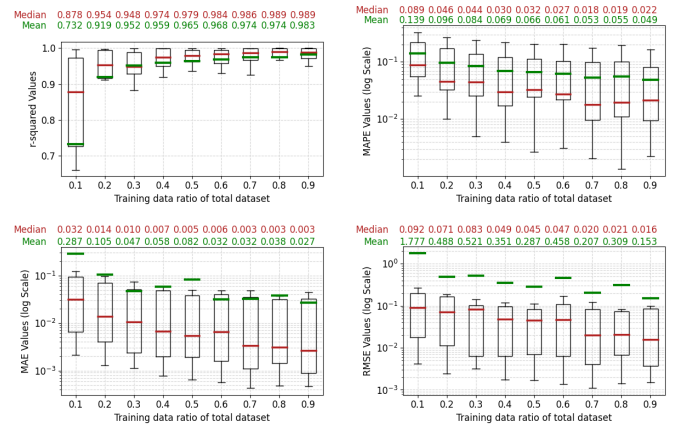


Fig. 5: Effect of reducing training data size on ATE prediction quality using 1-D GAP

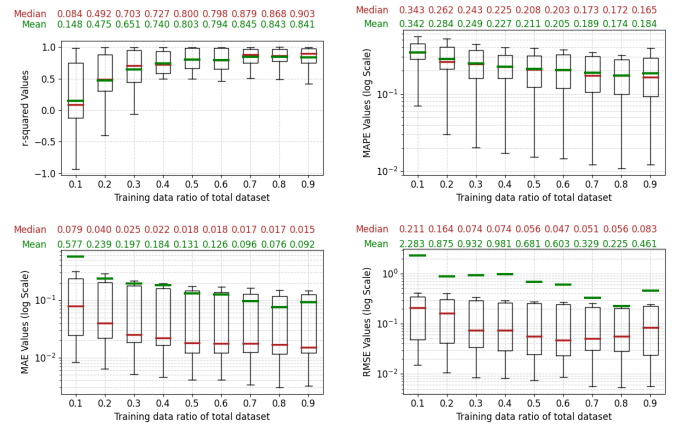


Fig. 6: Effect of reducing training data size on APE prediction quality using 1-D GAP

highlights the need for training a prediction model to predict SLAM errors since the observed errors during the course of the trajectory are not correlated to that at the end of the trajectory. For that, we compute  $R^2$  and  $MAPE$  between the SLAM errors at 20 % for ATE and 40% for APE of the trajectory and ATE/APE at the end of the trajectory. Then, we compare the outcomes with that of our prediction model. As shown in Table III, we can observe that the prediction model outperforms the baseline (ATE at 20 % and APE at 40%) and provides more accurate outcomes.

### D. Impact of limited training data on error prediction

The supervised learning formulation of the problem requires ground truth data to produce training examples. The availability of such data may be challenging and limited, thus, we examine our method against limited training data in order to provide evidence of its adaptability to such challenging situations.

Not surprisingly, reducing the data available for training will reduce the quality of ATE and APE predictions. However,



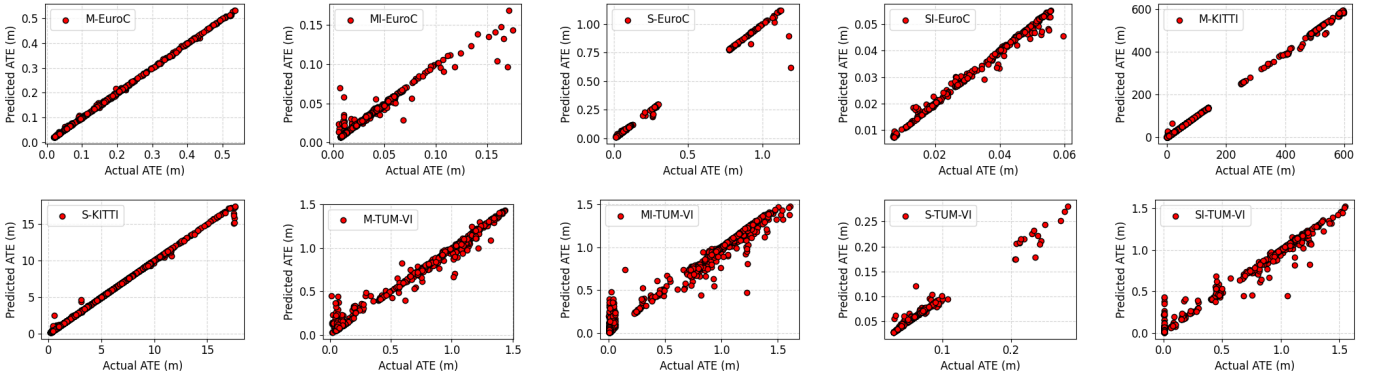


Fig. 7: Actual vs. predicted ATE for all testcases using the 1-D GAP and random forests after training on 70 % of the data

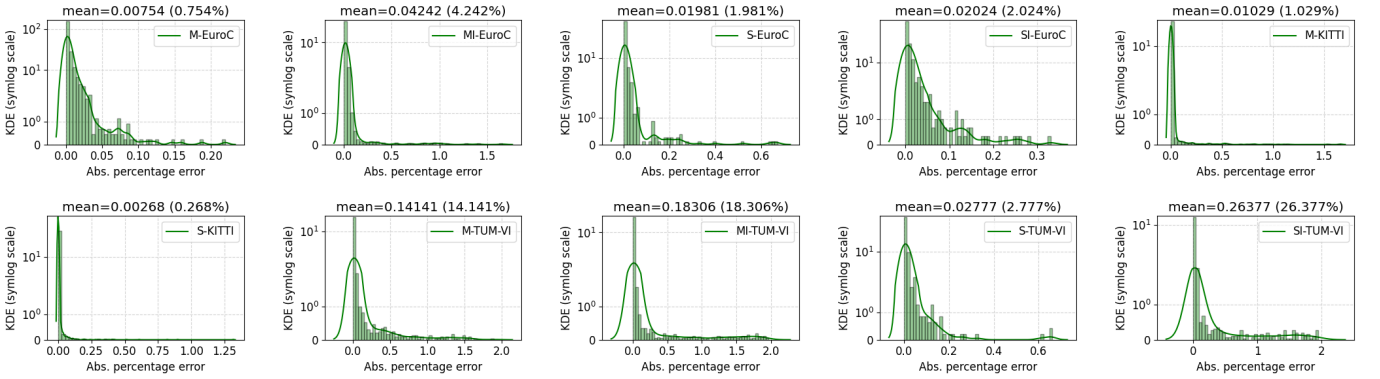


Fig. 8: Absolute error percentage of all testcases using 1-D GAP and random forests after training on 70 % of the data

the question is how much reduction should one expect in case of having limited amount of data available for training. In addition to that, we seek to prove that the proposed method can be utilized to reduce evaluation efforts of SLAM by training on a small portion of the dataset and the prediction of the rest of the dataset.

As shown in Figure 5, we can observe the normal behaviour of increased prediction quality when more training data is utilized. When we look at the  $R^2$  and  $MAPE$  metrics, we can observe that we are able to properly predict ATE while training on only 20% of each test case. In that case, the reduction in  $R^2$  is limited to only 6.51% on average. On the other hand,  $MAPE$  also dropped by only 4.7%.

On the other hand, in Figure 6 we can observe same increase in predication quality with the usage of more training data. A closer look at  $R^2$  and  $MAPE$  metrics show the ability to properly predict APE with only 40% of the data produces a reduction of 10.1% and 4% in  $R^2$  and  $MAPE$  respectively.

#### E. ATE prediction accuracy

The comparison between the actual ATE and the predicted ATE using 1-D GAP and random forests is presented in Figure 7 for the 10 testcases we examined. Moreover, we present the kernel distribution estimate (KDE) of the absolute error

percentage of all testing example in each of the 10 testcases in Figure 8. One can observe that we are able to predict the ATE value within an average error of 7% of the actual ATE with peak performance of an average error that is less than 5% of the actual ATE value in 7 out of 10 testcases examined.

#### F. APE prediction accuracy

Additionally, we compare the actual APE and the predicted APE using the same setup in Figure 9. A closer look to the KDE of the absolute error percentage provided in Figure 10 shows that we are able to predict the APE value within an average error of 19% with peak performance of average error that is less than 14% of the actual APE in 7 out of 10 testcases examined.

Although the method proposed is able to adapt to both ATE and APE, the performance when predicting ATE is much better in terms of accuracy. That's due to fact that ATE is a smoothed signal compared to APE which observes more changes over time.

#### G. Out of distribution (OOD) prediction

Supervised machine learning operates under the closed-world assumption, such that both training and testing assume independent and identical distributions (*i.i.d.*) [32]. When input data shifts from this distribution, we witness what is called covariate shift [33]. This causes the predictor to provide

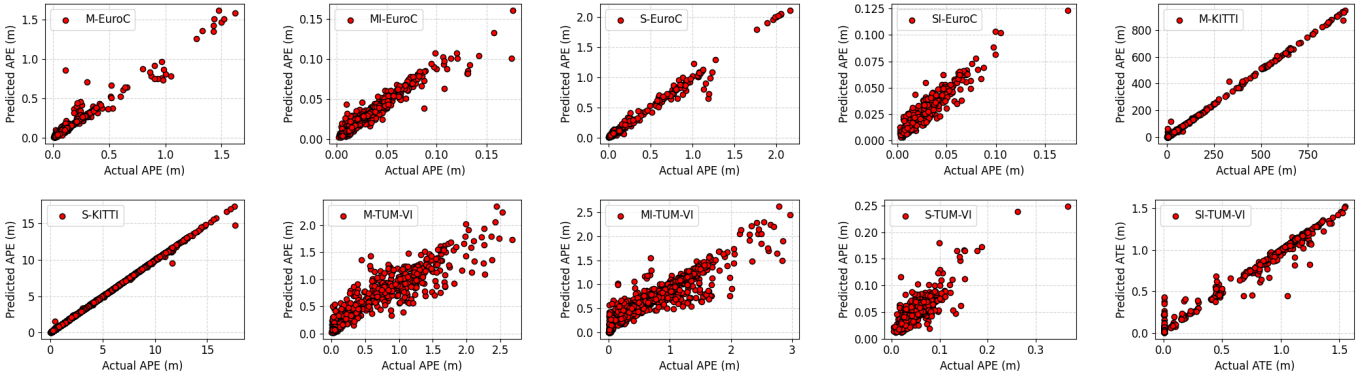


Fig. 9: Actual vs. predicted APE for all testcases using the 1-D GAP and random forests after training on 70 % of the data

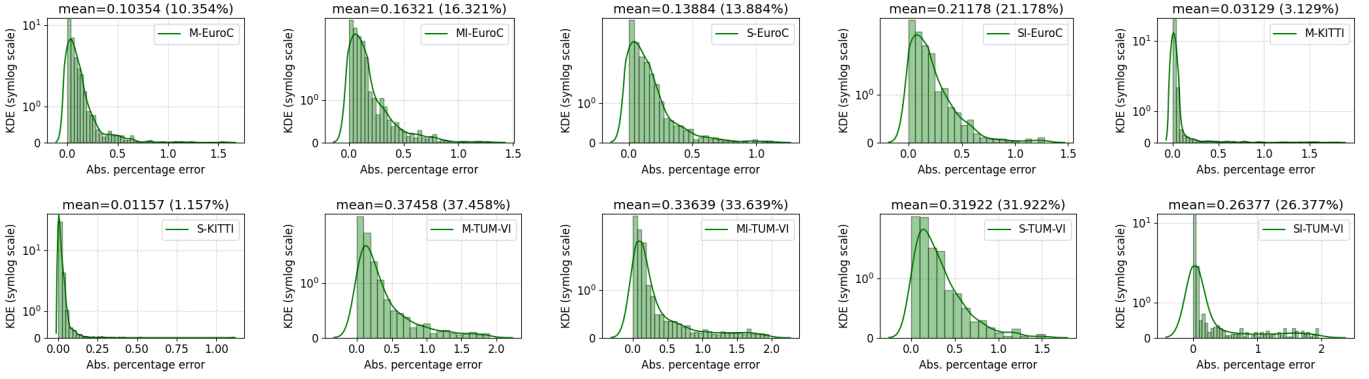


Fig. 10: Absolute error percentage of all testcases using 1-D GAP and random forests after training on 70 % of the data

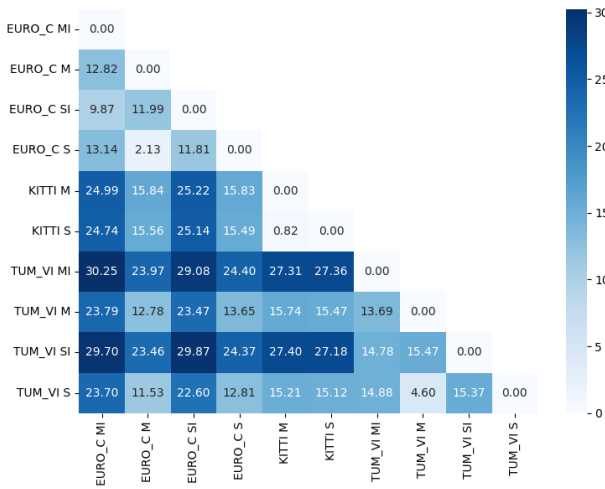


Fig. 11: Sinkhorn distance between different testcases

non-accurate estimates as the testing data follows a different distributions compared to the training one.

A dataset shift was observed when trying to test the model on data from a testcase that is different from the training one, which resulted in a huge reduction in prediction quality of the model. In order to prove the dataset shift, sinkhorn distance

[34] was measured between different testcases and is provided in Figure 11. Sinkhorn distance is used to measure the shift in distributions between training and testing data when both are coming from different testcases.

## VI. CONCLUSIONS

In this paper, the problem of performance prediction in SLAM is addressed as a fundamental requirement for robustness and resilience in SLAM. The study starts by giving a brief review of the literature related to this topic, and provides a basis for the proposed algorithm. After that, we introduce our methodology for predicting SLAM errors using an ensemble learning technique and 1-D global average pooling of input data characterization results. Our methodology is first compared to a multitude of regression models to validate our selection of random forests as our regression model. Then, the methodology is tested on 10 different test cases to quantify its adaptability to different datasets. The experimental results showed superiority in using random forest compared to our selected baseline and provided evidence for the ability to predict ORB-SLAM3 errors using characterized and pooled features with accuracy that can reach 93.1 % and 80.45 % for ATE and APE respectively. Additionally, the paper studied the impact of reducing the amount of training data on error prediction quality, and it is shown that we are able

to use only 20% for ATE and 40% for APE to maintain proper prediction quality. This is critical due to the limited availability of ground truth data in practical settings. Finally, we study the suitability of the method proposed to predict out-of-distribution data and provide evidence on the dataset shift observed between different testcases examined. The study illustrated the possibility to predict SLAM both coarse and fine SLAM error metrics, that can equip SLAM algorithms with ability to self-asses its estimates and enhance its robustness and resilience capabilities.

## REFERENCES

- [1] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "The slam problem: a survey," *Artificial Intelligence Research and Development*, pp. 363–371, 2008.
- [2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [3] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, "Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems," *arXiv preprint arXiv:2109.12343*, 2021.
- [4] I. Ali and H. Zhang, "Are we ready for robust and resilient slam? a framework for quantitative characterization of slam datasets," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2810–2816.
- [5] H. Carson, J. J. Ford, and M. Milford, "Predicting to improve: Integrity measures for assessing visual localization performance," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9627–9634, 2022.
- [6] M. Luperto, V. Castelli, and F. Amigoni, "Predicting performance of slam algorithms," *arXiv preprint arXiv:2109.02329*, 2021.
- [7] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7244–7251.
- [8] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [9] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [10] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, "Achieving integrity in an ins/gps navigation loop for autonomous land vehicle applications," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 4. IEEE, 1998, pp. 3437–3442.
- [11] A. Nouredin, T. B. Karamat, M. D. Eberts, and A. El-Shafie, "Performance enhancement of mems-based ins/gps integration for low-cost navigation applications," *IEEE Transactions on vehicular technology*, vol. 58, no. 3, pp. 1077–1096, 2008.
- [12] G. D. Arana, O. A. Hafez, M. Joerger, and M. Spenko, "Recursive integrity monitoring for mobile robot localization safety," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 305–311.
- [13] O. A. Hafez, G. D. Arana, M. Joerger, and M. Spenko, "Quantifying robot localization safety: A new integrity monitoring method for fixed-lag smoothing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3182–3189, 2020.
- [14] Y. D. Yasuda, L. E. G. Martins, and F. A. Cappabianco, "Autonomous visual navigation for mobile robots: A systematic literature review," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–34, 2020.
- [15] F. Amigoni, V. Castelli, and M. Luperto, "Improving repeatability of experiments by automatic evaluation of slam algorithms," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7237–7243.
- [16] E. Piazza, P. U. Lima, and M. Matteucci, "Performance models in robotics with a use case on slam," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4646–4653, 2022.
- [17] A. I. Mourikis and S. I. Roumeliotis, "Predicting the performance of cooperative simultaneous localization and mapping (c-slam)," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1273–1286, 2006.
- [18] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.
- [19] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning (adaptive computation and machine learning series)," 2016.
- [20] C. D. Sutton, "Classification and regression trees, bagging, and boosting," *Handbook of statistics*, vol. 24, pp. 303–329, 2005.
- [21] B. Efron, *Bootstrap methods: another look at the jackknife*. Springer, 1992.
- [22] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [23] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.
- [24] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999.
- [25] J. I. Daoud, "Multicollinearity and regression analysis," in *Journal of Physics: Conference Series*, vol. 949, no. 1. IOP Publishing, 2017, p. 012009.
- [26] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.
- [27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
- [28] P. Probst, M. N. Wright, and A.-L. Boulesteix, "Hyperparameters and tuning strategies for random forest," *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, vol. 9, no. 3, p. e1301, 2019.
- [29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.
- [30] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
- [31] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The tum vi benchmark for evaluating visual-inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1680–1687.
- [32] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, "Towards out-of-distribution generalization: A survey," *arXiv preprint arXiv:2108.13624*, 2021.
- [33] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. MIT Press, 2008.
- [34] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, 2013.