ASPS Conference Proceedings 1: 1283-1291 (2022)

Proceedings of

12th Structural Engineering Convention-An International Event (SEC 2022)

Available at https://asps-journals.com/index.php/acp

Naïve Bayes to Machine Learning Approach for Structural Dynamic Complications

P.V. Ramana[1,*]

[1] Professor, National Centre for Disaster Mitigation & Management, MNIT Jaipur, India

Paper ID - 090540

Abstract

Learning is one of the most powerful concepts in artificial intelligence research. It allows a system to learn from its environment and automatically modify its behavior to suit its needs. On par with human champions, the world's best computer backgammon player is a computer program that learns by playing against itself. The learning algorithm and available data set limit computer learning. Several techniques are available to perform machine learning. Decision trees, the naïve Bayes approach, and the more general Bayes net approach are a few choices. The naïve Bayes approach is an instance of the more general Bayes nets. This paper examines and analyzes the naïve Bayes and decision tree approaches to learning. Various techniques to avoid over-fitting, such as ensemble construction and cross-validation, are also implemented and analyzed. A novel hybrid between the naïve Bayes approach and the decision tree method is presented. The hybrid system produces a spectrum of options that could be used for learning by merely changing parameter values. At one end lies the naïve Bayes approach, while at the other lies the decision tree technique. The proposed hybrid scheme solves the problem of poor naïve Bayes performance in a domain with dependent attributes and the memory consumption problem of the decision tree. One can analyze this idea and show encouraging experimental data that backs the need for such a solution.

Keywords: Artificial Intelligence, Naïve Bayes, Earthquake & Structural Dynamics, Machine Learning.

1. Introduction

Learning, one of the most prominent fields of Artificial Intelligence research, has been extensively worked on. The learning component of a system allows it to learn from the environment automatically and modify its behavior. This is different from the conventional notion of a deterministic program. Learning techniques have been employed in a wide range of domains. Current spam filtering systems [1-4] and the world's best computer backgammon player [6-10] use learning. Traditional learning methods include the naïve Bayes model, a Bayes net structure, and decision trees. The naïve Bayes approach uses a model that can be constructed very fast and is suited for learning when the attributes to be learned are independent of each other. Performance is poor if the details exhibit dependencies. A more complicated Bayes net structure could be discovered in such a case if the independence property does not hold amongst attributes. This gives rise to more accurate predictions, but the construction takes an enormous amount of time. As for a decision tree, while construction is a simple and easy-to-understand process, it is time- and memory-consuming. Memory consumption problems are tackled with the use of pruning techniques. These include boosting, cross-validation, and ensemble construction [21-24]. These techniques tend to solve the problem of over-fitting. This would arise when a learning algorithm is too concerned and takes a narrow approach to learning based on the data set used for training. This would lead to false predictions when the algorithm can predict new data [25-28].

In this paper, one can implement and examine the naïve Bayes and decision tree approaches. One can choose decision trees and implement these techniques to study the effects of boosting, cross-validation, and ensemble construction. A novel hybrid approach to learning using decision trees and naïve Bayes is proposed. As shown in Section 6, the hybrid system displays a spectrum of learning solutions. At one end of the spectrum lies the naïve Bayes method, while at the other lies the decision tree approach [29-33]. In other parts of the range, the two techniques are combined by learning a decision tree using pruning methods and including a Naïve Bayes model at the leaf nodes, consisting of the remaining attributes that have not been branched upon. This method takes advantage of the speed of the naïve Bayes approach. It uses the decision tree to break as many dependencies as possible to suit the data to the naïve Bayes assumption of attribute independence [34-38]. To avoid over-fitting, pruning on a decision tree is done due to memory concerns. If the amount of data to be learned and the number of attributes is large, then pruning is required to keep the tree manageable. This would result in the loss of data. In such a case, using a naïve Bayes model at the leaf nodes would allow for better predictions by recovering some of the lost data [39-42]. It would be fast and not memory-consuming, thus leading

*Corresponding author. Tel: +919549654189; E-mail address: pvramana.ce@mnit.ac.in

Published online: December 19, 2022

doi:10.38208/acp.v1.652

to an overall good predictor. Section 2 examines related work, while Section 3 contains the details of decision trees and their construction process. Section 4 discusses the naïve Bayes approach and Section 5 discusses techniques used for improving the performance of the naïve Bayes and the decision tree techniques. Section 6 proposes our novel hybrid technique and explains the intuition and the need for such an approach. In section 7, one can evaluate the naïve Bayes and Decision Tree classifier and the effect of various improvements. Experiments also show that the Hybrid inherits desirable properties from both classifiers. One can conclude and discuss future work in Section 8.

2. Related Work

Machine learning was a result of the quest to mimic human intelligence. It has received tremendous amounts of attention in artificial intelligence research. [1] examines various approaches to machine learning and classifies the methods under three broad categories: (i) data mining, (ii) neural network, and (iii) reinforcement learning techniques. Learning is applied to various domains, such as spam filtering [2,3] and games [10]. [2] Spam Assassin is a mail filter that uses text analysis, Bayesian filtering, DNS blocklists, and collaborative filtering databases. A part of our experimental database was from [2]. This paper concentrates on decision trees [4-9] and the naïve Bayes approach [11-17]. Decision trees are a highly easy-to-understand method of classifying training data and using the classification to predict. [4,9] contain approaches to choosing attributes to split upon in decision trees. [4] presents a family of measures called C-SEP (Class Separation). The method utilizes the cosine of the angle between vectors associated with the nodes in the tree. [9] discusses an approach that splits on an attribute with maximum information gain. Decision trees consume memory at rapid rates. Pruning [5-9] is required to keep memory utilization at modest levels. [6] is a backtracking approach that grows the tree and prunes it as needed. Though the final tree may be small, the process dictated in [6] is memory-consuming during tree growth. [7] discusses pruning from the viewpoint of increasing simplicity. A complex, yet the actual tree is pruned to increase its simplicity, enhancing understanding. Pruning should not tradeoff the accuracy beyond tolerable limits, however. [8] takes the exciting approach of searching for pruned trees in a search space. [9] contains a system that uses a statistical significance test and chi-square value cut-offs to prune trees. [5] is a general comparison of numerous methods. The Naive Bayes Classifier has been well-studied, even in the spam domain. [11] advocates the use of specific domain knowledge to get better recall and precision. In the context of spam classification, the authors manually identify about 20 non-phrasal domain-specific features such as overemphasized punctuation, time of receiving, number of attachments, subject lines, etc. One can use similar rules to preprocess emails and parameterize them along 613 attributes. Along with [13], [11] does a detailed evaluation of Naïve Bayes-based spam classifiers. [12] proposes heuristics like Complement NB (to deal with skewed training data) and introducing weights (to deal with dependence) to improve the accuracy of Naïve Bayes text classifiers; however, many of their heuristics assume multinomial attributes, which one can already get rid of during our pre-processing stage. Finally, one can implement in our Naïve Bayesian classifier many simple hacks suggested in [43-47] by authors out of practical experience designing spam classifiers. These include setting priors, weighting to deal with skewed data, etc.

3. The Decision Tree Approach

Decision trees have been used for various purposes, such as pattern matching and machine learning. The reader is encouraged to examine the references provided for an in-depth discussion of decision trees. A brief description is provided here. A decision tree essentially classifies the training data into sets. These sets are formed by branching on the attribute values that the examples in the training data**.** A naïve decision tree would sequentially branch on all attributes. Techniques to handle problems with the naïve decision tree are discussed in Section 5. A perfect decision tree would be structured so that all the training examples at a node in the tree are the same classification. But this rarely happens since there would always be a few outliers due to noise. The prediction at the node is the majority of the type of the various examples, with biasing incorporated if required. When the formation of the tree is completed, prediction can take place. Given a piece of data, one can traverse a path from the root to the tree's leaf by taking edges corresponding to the value that the data depicts for the attribute at the node. The prediction of the leaf node is then the prediction that is returned. As the number of attributes increases, it is impossible to use the naïve technique. This is due to the explosion of nodes if one can branch on every attribute. For example, if one can have 20 binary details, then the number of leaf nodes alone would be 2^{20}. The chi-square value [48, 49] is used as a means of testing for statistical significance. The attribute chosen for branching is usually selected to maximize the information gained.

4. The Naïve Bayes Approach

Naïve Bayes classification is used widely for its simplicity, efficiency, and excellent performance in various applications, including text classification and spam detection. Naïve Bayes estimates the probability that an instance x belongs to class y as

$$P(y|x)=\frac{P(y)}{P(x)}P(x|y) \quad (1)$$

$$=\frac{P(y)}{P(x)}\prod_i P(x_i|y) \quad (2)$$

and predicts the class with the highest P value $(y|x)$. Step (1) is the Bayes theorem, and (2) is the Naïve Bayes assumption. The latter assumption is made because it is generally tough to learn P(x|y) for all x, and in contrast, it is much easier to understand $P(x_i|y)$. For binomial attributes x_i, this can be done by simply counting the number of occurrences in the training set S of x_i in each class and the number of instances in each category. Even in this basic form, the naïve Bayes classifier performs reasonably well. Its biggest weakness, though, is the assumption that makes it so simple – that the attributes are independent. Many heuristics have been proposed in the literature [Section 2] to improve the classifier's performance in domains where the assumption leads to bad performance. Its other drawback is that it does not work well when the training data set is skewed, for

instance, when one can get no training instance for a particular class. For such cases, as suggested in [15], one can assign $P(x_i|y) = \varepsilon > 0$. Finally, in the spam-detection domain, there are only two classes (*i.e.,* spam and ham), and a false positive is much more expensive (say λ-times) than a false negative. One can predict a test example as spam when $P(y|x)/P(\hat{y}|x) > \lambda$. [13] suggests how to choose λ depending on the particular configuration in which the spam classifier is deployed; one can use 9 for spam classification. Note that λ=1 corresponds to selecting the class with the highest ("higher," since there are only two classes) conditional probability.

5. Improvements

Over-fitting [9] is often a problem for learners. Given a set of training data, the learner would tend to take a narrow approach and learns such that its predictions for the given training set is accurate, while it fares poorly on unseen data. This is usually due to either outliers or incomplete representation of the various classes in the data that misleads the construction of the learner. Such a problem can be solved through a variety of approaches. The following subsections discuss a few: (i) cross-validation, (ii) ensemble, and (iii) pruning. Pruning is only relevant in the context of decision trees. [9] discusses all these approaches in greater detail.

5.1 Cross-Validation

Cross-validation is constructing a set of learners from a given training set and choosing the best of the generated group. A parameter, k, to indicate the number of learners to be constructed is required. The given training data set is divided into k different parts. To build the learner, one of the k parts is used as the test data, while the rest of k-1 is used for the training data. Each tree has a specific test data set. In this manner, k learners are constructed, and the best of them is chosen. The intuition behind cross-validation is that it would look for the most generic decision learner that performs well. In this manner, it is ensured that the learner does not learn unwanted patterns. It is to be noted that this is not a fool-proof technique since a particular learner may perform very well on a given test set, but the test set may not be representative of the entire domain.

5.2 Ensembles

Another method to limit over-fitting is the ensemble construction method. This involves the construction of multiple learners, with each one contributing to every prediction. This paper used the AdaBoost [19] algorithm for ensemble construction. This technique trains a learner based on the input training data. It then evaluates the performance of the learner on the same training data. The examples that were incorrectly predicted are then given a higher weight. Learning occurs again, and the learner construction process concentrates more on the accuracy of models with high importance. This process carries on. Each learner has a weight associated with it, and when a prediction is required, each learner's output is weighed appropriately, and a final projection is formed. The ensemble method takes advantage of the probability that a majority of learners being wrong is lower than that of a single learner being bad. This property, of course, requires the errors to be independent of each learner, which is not valid. Theoretically, as the number of learners grows, the number of erroneous predictions decreases.

5.3 Pruning

Pruning serves two purposes while constructing Decision Trees: (i) it ensures that the final tree has not learned unwanted patterns, and (ii) it restricts the number of resources consumed. Pruning can be done during or after the learning process (post-pruning). There are various approaches to pruning [5-9]. This paper uses the one suggested in [9]. It uses a statistical significance test based on chi-value cutoffs to determine whether branching on an attribute would provide benefits beyond a threshold. Excessive pruning would result in the loss of data. Such pruning would be required to fit a vast decision tree into memory.

6. Hybrid Approach

The Hybrid classifier tries to capture the desirable properties of the Naive Bayes and the Decision Tree-based classifiers. A Naive Bayes Classifier is trivial to implement, requires little memory, and is quick to learn from a training set. Its biggest drawback, however, is a large number of incorrect classifications it does when the attributes are dependent. If somehow the dependent qualities could be identified, and for each such pair, only one used both in the learning and the inferring stages, one can cut down on the misclassifications. Decision Trees classifiers, unless pruned, even for a small number of attributes, require prohibitively large memory and running times. Pruning the tree using moderate-to-large χ-values gets around this problem but introduces another. Quite often, many training examples get cluttered into the same leaf node because no matter which attributes one chooses, the information gained is never sufficient to split the node. Such leaves often have training instances of many different classes and end up misclassifying test examples of all types except the majority class. What is needed is a more ``intelligent" inference mechanism at such nodes that does a better job at classifying than just assigning the course with the maximum number of instances [49 – 50].

A side-effect of using the information-gain heuristic, which one can exploit, is that if there are two attributes dependent on each other, one can hope that one of them would be chosen to split a node of the decision tree. As a result, for attributes left in the leaf nodes, it would be more acceptable to make the independence assumption (that when all points were considered together), as other attributes that some of the characteristics in the leaves were dependent on would have been selected higher up in the tree to split a node. The Hybrid Classifier combines the Decision Tree Classifiers' propensity to separate dependent attributes and the effective classification by the Naïve Bayes Classifier on independent features. The idea is simple. In the learning phase, the Hybrid Classifier grows a tree like the Decision Tree. The only difference is at the leaves, where a naive Bayes learner is now implemented. This Naive Bayes learner learns only from the training examples that arrive at that particular leaf, using only those attributes that the Decision Tree has not used along the path from the root to the leaf. During the inference phase, just as in Decision Trees, the attribute values

of the test example determine the way it takes down the tree and hence the particular leaf node it reaches. The decision at the leaf node is taken by the Naive Bayes classifier based on the attributes of the test, which has still not been considered. This classifier is parameterized by the χ-value used for cut-off. If it is very high, no attribute exists that will give an information gain sufficient to split even the root node, making the root node the Naive-Bayesian leaf node. On the other hand, if the cut-off is zero, one can get the same tree constructed by Decision Trees. Moreover, getting a zero information gain upon choosing any attribute means that the number of all instances in the leaf node belong to the same class, in which case, the Naive Bayesian leaf node will return the type that the leaf would have produced in a Decision Tree.

7. Evaluation

This section evaluates the Naïve Bayes Classifier and the Decision Tree classifier. One can then discuss the effect of using modifications such as cross-validation and ensembles. Finally, one can consider our Hybrid Classifier. One can use the following two datasets in our experiments. Spam Assassin[2] is an email corpus containing 18110 emails, of which around 10000 are spam and the rest ham. Spamassassin has two datasets, a *SMALL* one and a *LARGE* one. The *SMALL* dataset consists of 2970 examples for training and 330 samples for testing. The *LARGE* dataset contains 16299 examples for training and 1811 examples for testing.

7.1 Evaluating Naïve Bayes Classifier

In its basic form, the Naïve Bayes classifier got 60 misclassifications (32 false positives and 28 false negatives) on the *LARGE* dataset and 23 misclassifications (all false negatives) on the *SMALL* database. The cost of misclassifying is not symmetric in classifying spam – false positives are more expensive than false negatives. To tune the classifier's performance, one can study the effect of varying λ on the number of false positives and negatives. As expected, as λ grows, the number of false positives decreases while the number of false negatives increases. Unfortunately, their sum (the total number of misclassifications) also increases – growing to almost 200 (11%) by the time the number of false positives comes down to 0.

7.2 Evaluating Decision Trees

The chi value [18] was set at 0.5 for these experiments. The decision tree had 13 erroneous predictions for the *SMALL* dataset, out of which all 13 were false negatives. There were 72 errors for the *LARGE* dataset, with 41 false negatives and the other 31 false positives. Figure 2 shows the curve representing the change in the total number of errors with χ when the SMALL dataset was used. The graph takes shape.

Of an exponential curve, except a plateau. One can be observed that as χ increases, the number of nodes pruned increases exponentially. Hence, the amount of information loss, which is proportional to node loss, is also exponential, resulting in many errors. The plateau in the graph is due to a region of χ values with very few nodes between them. This is a property of the dataset.

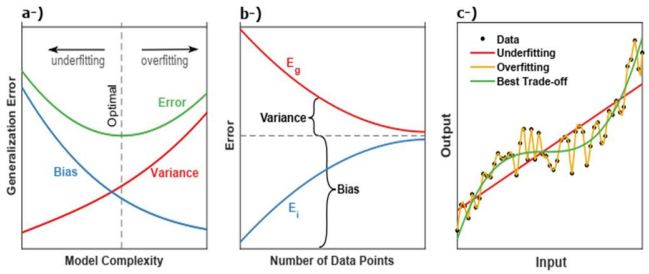

Fig. 1: This graph shows the variation of total misclassifications and false positives with λ. False negatives are the difference between the two curves.

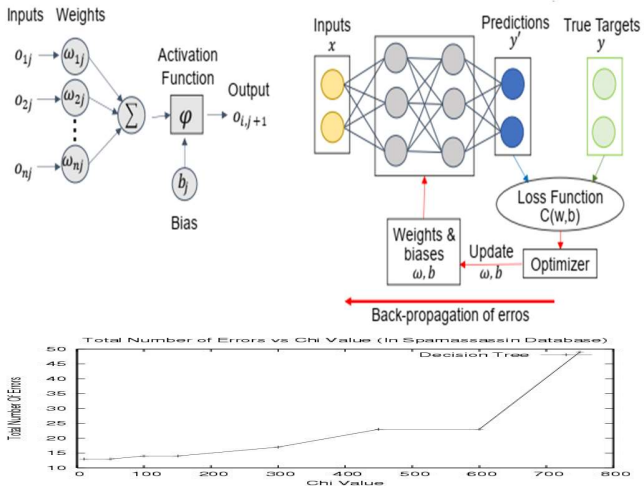

Fig. 2: This graph shows how the total number of errors varies with χ & ANN scheme.

7.3 Effect of Improvements

This section evaluates the performance of ensembles and cross-validation. A naïve learner suffers from the problem of learning on a complete training set. This narrow approach would imply that the learner's performance on unseen data might be poor. To avoid such over-fitting, one can use techniques like ensembles and cross-validation. One can evaluate these approaches for decision trees. Figure 3 shows the effect of using costumes. The *SMALL* dataset was used. Chi-square values were set at 0.5 for all experiments in this section. The general
The trend to note is that of the reduced number of total errors. When the ensemble number is less than or equal to 4, the ensemble has insufficient rees to make a difference. This is due to the large number of trees required to classify a hard-to-learn example correctly. This is why the standard decision tree and the ensemble have identical performances till ensemble number 4. After eight trees in the choir, the total number of errors remains constant. This indicates either one of two possible scenarios. One possibility is that the remaining misclassified examples are complicated and require many trees. This is not likely, since even with 16 trees, no difference is observed. The second possibility can be due to outliers in the data set and other inconsistencies in the dataset, making fault-free prediction difficult.

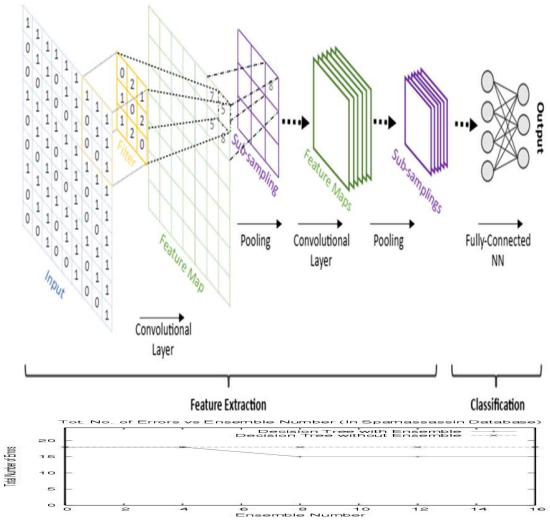

Fig. 3: This graph depicts how the number of errors varies as the ensemble number increases.

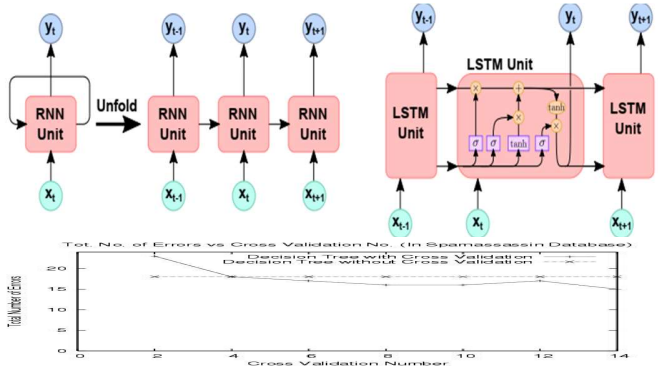

Fig. 4: This graph depicts how errors vary as the cross-validation number increases.

Figure 4 depicts how the total number of errors decreases as the number of trees in the cross-validation increases. The general trend is as expected, with the total number of mistakes reducing with the number of trees in the cross-validation algorithm. There are, however, two irregularities worth noting. Firstly, for Cross-validation number equal to 2, the number of total errors is higher than for a single simple tree. This is a property of the data set. When the data set is split randomly, it could happen that split sets are not representative of the domain. It should also be noted that the number of examples a cross-validation tree trains upon is fewer than the average tree. This is because one can set aside some training data for testing during the cross-validation process. The second irregularity occurs when the cross-validation number is equal to 12. This performs worse than with a cross-validation number equal to 10. This, again, is attributed to the random splitting of the data set. This random splitting in cross-validation can lead to a slight increase in the total number of errors. But from our experimentation, one can be observed that such misclassification increases are small in magnitude and are removed when the cross-validation number is increased further.

7.4 Evaluating the Hybrid Classifier

In this subsection, one can evaluate the performance of the Hybrid Classifier. One can turn off cross-validation and ensemble to best illustrate the effect of going in for the hybrid. Figure 5 shows how the number of incorrect predictions by the Hybrid classifier varies with χ for the two datasets. It also plots a corresponding curve for decision trees and a horizontal line representing the Naïve Bayes numbers. On the large dataset, one can be observed that Naïve Bayes performs better than Decision Trees, irrespective of the χ-value one can choose. However, the curve for the Hybrid classifier serves exactly as the Naïve Bayes classifier for $\chi \sim 9000$. On the other hand, on the small dataset, Decision Trees perform better than the Naïve Bayes classifier, potentially due to more outstanding dependencies in the attributes. Here, the Hybrid classifier behaves precisely like the Decision Trees for $\chi \sim 0$. Thus, the Hybrid classifier comes across as a very flexible and powerful classifier, which gives us the best of both worlds – Naïve Bayes and decision trees – of course, depending on the choice of χ. One can envision the Hybrid classifier running *cross-validation* over different values of χ and choosing the best value for a given dataset. Cross-validation would prove especially useful in datasets where neither Naïve Bayes nor Decision Trees perform well. Instead, here is some intermediate value of χ for which the Hybrid classifier outperforms them. Unfortunately, though, none of the two datasets showed this trend. There are other advantages of using the Hybrid over these different classifiers. Decision trees experience an exponential blow-up in their memory requirement as the χ-value decreases and the tree depth increases. And so does the Hybrid classifier. Figure 6 shows this blow-up for both datasets by plotting the number of nodes in the tree.

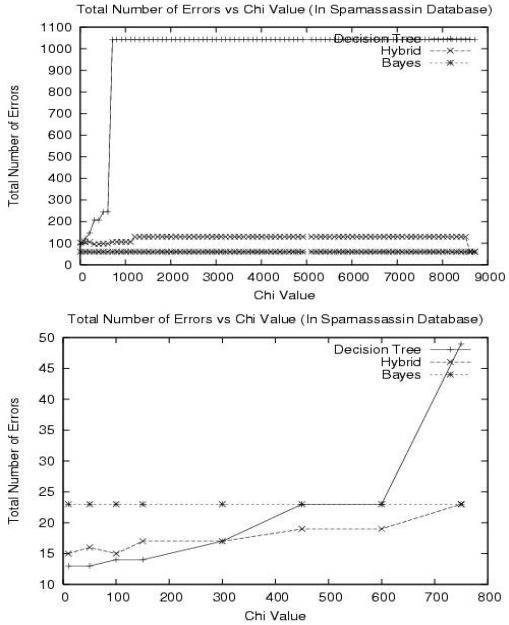

Fig. 5: This graph depicts how errors vary for the three classifiers for different χ values on the large (above) and small (below) spam datasets.

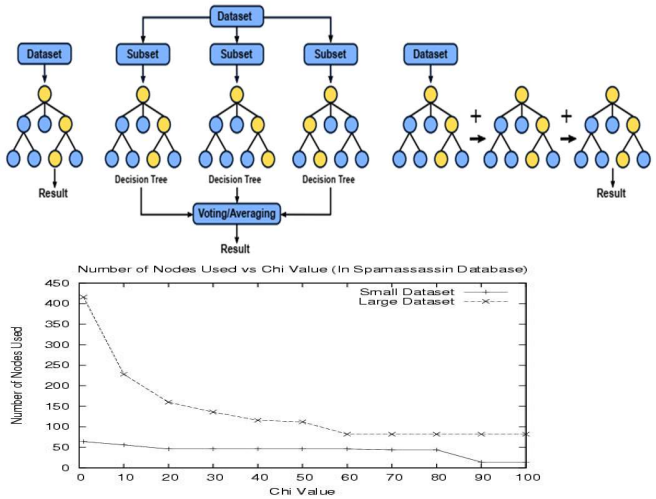

Fig. 6: This graph plots the memory requirement for different χ values on the two datasets.

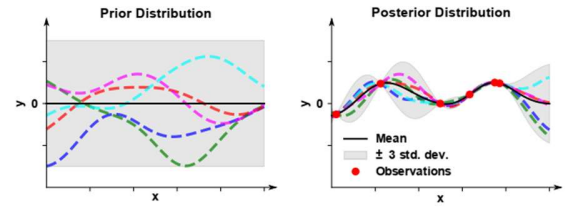

Fig. 7: In Gaussian Process Regression, the prior distribution (left) is defined by kernel functions, and the posterior distribution (right) is updated with information from the observation points using Bayesian inference.

Classifiers perform nearly as one can will as decision trees, even for higher χ-values. Hence, given a decision tree with some χ value v1, one can obtain a Hybrid classifier, with χ value v2, such that v1 < v2, but the Hybrid still performs as well as the decision tree. Since v1 < v2, the hybrid would have fewer nodes than the corresponding decision tree. Thus if memory is a constraint, one can get nearly the same performance as decision trees without running into the memory problems that the latter does.

Figure 7 illustrates updating the prior with input data and evaluating the posterior distribution. Usually, a maximum likelihood estimator approach is applied to optimize the kernel hyperparameters. In other words, the search process looks for the marginal likelihood of data and evaluates the best fit of the output data to the input data. A detailed explanation, including the formulation for GP with non-zero mean and inference considering a Gaussian white noise in the observation's outputs, can be found in the works.

Even raw data can be input in DL methods due to its ability to learn highly complex and nonlinear patterns. On the other hand, DL usually demands more data. The different procedures of feature selection and extraction in both Classical ML and Deep Learning are illustrated in Figure 8.

Figure 9 shows the steps to build a surrogate model: apply DOE to define the supporting point's location; sample results with the high-fidelity model; preprocess the data; train the surrogate model; predict new outputs using the surrogate model. Adaptive Learning is optional and may be applied to select new

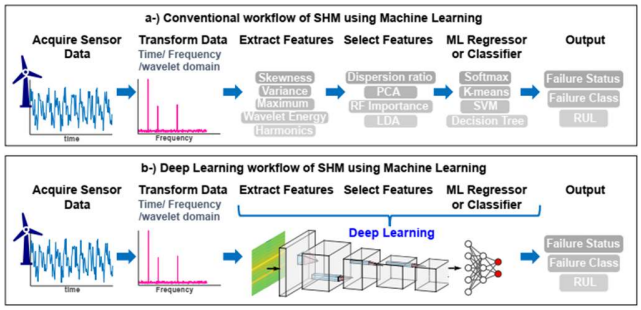

Fig. 8: Structural Health Monitoring workflow: in the classic approach, feature extraction and selection are handcrafted and followed by an ML method (a); if Deep-Learning is used, feature extraction and selection are performed automatically by the ML method (b)

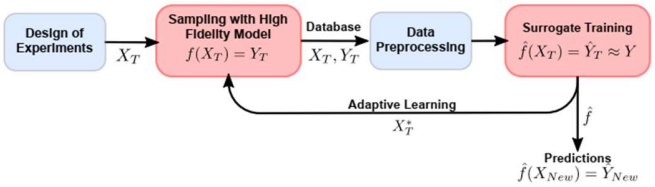

Fig. 9: Steps to build a surrogate model

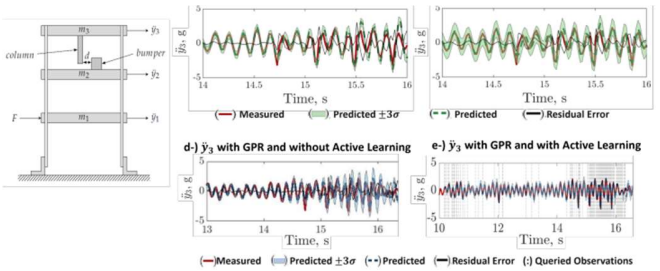

Fig. 10: Three-storey structure to construct an operational DT (a) and predicted acceleration response of the third floor ¨ y3 when column and bumper are in contact (nonlinear response) for different stages of the DT implementation (b-e).

supporting points to update the surrogate model in regions of interest or uncertainty. The representations are illustrative and do not show the arrangement precisely nor include all configurations.

The basic steps to build a surrogate model are schematized in Figure 11. The first step is to generate the support points. As the approximate function is

Based on their information, it is essential to generate an informative set of support points according to the Design of Experiments (DOE) theory.

First, measured data was used to calibrate the physical model parameters. Then the outputs of this model are used as input of a GPR, which ameliorates the output prediction to care for uncertainties and nonmodeled physics using online adaptive sampling. The methodology is demonstrated in the model of the three-storey structure, as shown in Figure 10. Although the physical model used was linear, the DT could predict the nonlinear behavior resulting from the contact between the column and bumper, which occurs at specific excitations.

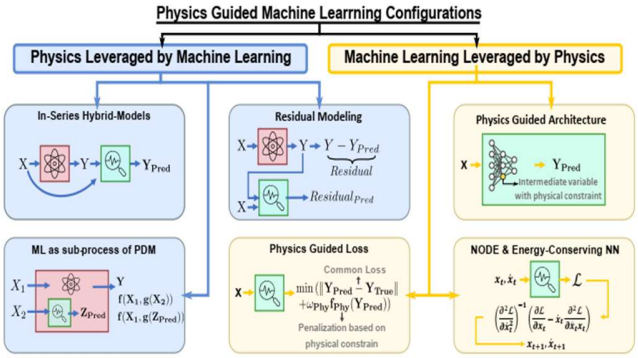

Fig. 11: Configurations of Physics-Guided Machine Learning merging Data-Driven Models (DDM) and Physics-Driven Models (PDM).

The state-of-art of PGML is described here according to the configurations in which the physical knowledge is merged with the ML algorithm, as illustrated in Figure 11. Two categories can be defined: Physics leveraged by Machine Learning, in which ML models are used to improve the results from the simplified physical models, and Machine Learning leveraged by Physics, in which physical laws and constraints are intrinsically embedded into the ML, guiding it to have physically consistent results.

8. Conclusion and Future Work

The capabilities of ML in extracting and recognizing fault patterns from monitored signals in the time and
frequency domain makes SHM the most developed and explored of these areas. Recurrently used data transformation and preprocessing methods are presented with examples of their significance in prediction accuracy. The prominence of Deep-learning methods in SHM is noteworthy since they automatically extract relevant features and reveal more complex patterns. SHM allows early failure detection and enables Remaining Useful Life predictions. Consequently, SHM avoids catastrophic failures and might implement a preventive maintenance schedule to optimize uptime, maximizing the use of component lifetime. In addition, ML methods locate and classify damage and plan actions to minimize risks. Therefore, ML for SHM results in operational safety and economic savings, with more straightforward implementations and better efficiency than model-based methods. The correspondence between ML and Active Control of Vibration and Noise is natural since both have data-based optimization as a pillar. Thus, active control theory exploits ML tools by various means. System identification and reduced-order modeling techniques use ML to model the system under control. ML methods also support the study of the optimal location of sensors and actuators. Various control system approaches and configurations use ML methods
to optimize controller parameters and policies, offline or online. Usually, ML applications in Active Control
are justified and perform better than traditional methods in nonlinear and complex scenarios. Although the memory and processing costs of ML algorithms are prohibitive in many control applications, the computer science advances occurring in parallel should make them affordable in more scenarios.

They implemented and analyzed decision trees and the naïve Bayes approach to machine learning. One can discuss problems with such learning techniques. Optimizations, such as cross-validation, ensembles, and pruning, were discussed and implemented to eliminate over-fitting issues. A novel hybrid approach to machine learning was also presented. The hybrid system produces a spectrum of solutions that can be used to learn. At one end of the spectrum lies the naïve Bayes approach, while at the other end lies the decision tree approach. All points elsewhere in the range constitute a decision tree with a naïve Bayes model at the leaf nodes. This hybrid approach provides excellent learners for domains with many dependent attributes. Many points would make the domain unsuited to decision trees, while naïve Bayes will not work well due to the lack of independence among features. Future work can examine if a naïve Bayes approach is required at each leaf node or if a naïve Bayes at just a subset of the leaf nodes would be sufficient. Another interesting question is whether the naïve Bayes model at a leaf node should contain all the remaining attributes or if many irrelevant details can be eliminated. Numerous such questions arise about the interaction of the hybrid technique with various parameters. The more questions one can answer, the greater the understanding obtained. Incorporating more theoretical and expert knowledge into ML models, as studied in Physics Guided Machine Learning models, is another trend subject of research, one it leads to more interpretable models, less need for training data, and more physically consistent predictions. The drawbacks and difficulties identified in the application of ML in SHM, Active Control and Product design also show the open discussions and room for progress.

Disclosures

Free Access to this article is sponsored by SARL ALPHA CRISTO INDUSTRIAL.

References

1. Ganesh, E., Ramana, P. V., & Shrimali, M. K. (2022). Solved structural dynamic mathematical models via a novel technique approach. Materials Today: Proceedings.
2. P.V. Ramana (2021), Modified Adomian Decomposition Method for Uni-Directional Fracture Problems, Sadhana, unique addition, 1-9.
3. B K Raghu Prasad, P.V. Ramana (2012)," Modified Adomian decomposition method for fracture of laminated uni-directional composites," Sadhana Vol. 37, Part-1, pp.33-57.
4. Ayush Meena, P.V. Ramana (2021), Mathematical Model for Recycled PolyEthylene Terephthalate Material Mechanical Strengths, Materials Today: Proceedings, 38, Part 5.
5. Arigela Surendranath, P.V. Ramana (2021), Mathematical Approach on Recycled Material Strength Performance Via Statistical Mode, Materials Today: Proceedings, 38, Part 5.
6. Ayush Meena, P.V. Ramana (2021), High-Rise Structural Stalling and Drift Effect Owe to Lateral Loading, Materials Today: Proceedings, 38, Part 5.
7. P.V. Ramana (2021), Statistical concert of solid effect on fiber concrete, Materials Today: Proceedings, 38, Part 5.
8. Ganesh, E., Ramana, P. V., & Shrimali, M. K. (2022). Unsolved

structural dynamic mathematical models via Novel technique approach. Materials Today: Proceedings.

9. Ganesh, E., Shrimali, M. K., & Ramana, P. V. (2021). Estimation of Seismic Fragility in Structural Systems. i-Manager's Journal on Structural Engineering, 10(2), 1.
10. Arigela Surendranath, P.V. Ramana (2021), Interpretation of bi-material interface through the mechanical and microstructural possessions Materials Today: Proceedings,17, Part 3.
11. Ganesh, E., Ramana, P. V., & Shrimali, M. K. (2022), "Solved structural dynamic mathematical models via a novel technique approach," Materials Today: Proceedings, Volume:62P6 / 3133-3138 / 2022
12. B K Raghu Prasad, P.V. Ramana (2012)," Modified Adomian decomposition method for fracture of laminated uni-directional composites," Sadhana Vol. 37, Part-1, pp.33-57
13. Ayush Meena, P.V. Ramana (2021), "Mathematical Model for Recycled PolyEthylene Terephthalate Material Mechanical Strengths," Materials Today: Proceedings, 38, Part 5
14. Arigela Surendranath, P.V. Ramana (2021), "Mathematical Approach on Recycled Material Strength Performance Via Statistical Mode," Materials Today: Proceedings, 38, Part 5
15. Ayush Meena, P.V. Ramana (2021), "High-Rise Structural Stalling and Drift Effect Owe to Lateral Loading," Materials Today: Proceedings, 38, Part 5
16. P.V. Ramana (2021), "Statistical concert of solid effect on fiber concrete," Materials Today: Proceedings, 38, Part 5
17. Ganesh, E., Shrimali, M. K., & Ramana, P. V. (2021), "Estimation of Seismic Fragility in Structural Systems," i-Manager's Journal on Structural Engineering, 10(2), 1
18. Arigela Surendranath, P.V. Ramana (2021), "Interpretation of bi-material interface through the mechanical and microstructural possessions," Materials Today: Proceedings,17, Part 3
19. Ayush Meena, P.V. Ramana (2021), "Assessment of endurance and microstructural properties effect on polypropylene concrete," Materials Today: Proceedings, 38, Part 5
20. P.V. Ramana (2020), "Functioning of bi-material interface intended for polypropylene fiber concrete," Materials Today: Proceedings,14, Part 2
21. Ganesh E, Ramana P.V, Shrimali M.K (2022), "Unsolved structural dynamic mathematical models via Novel technique approach," Materials Today: Proceedings Volume:62(2) / 684-691 / 2022
22. E. Ganesh, P.V. Ramana, M.K. Shrimali (2022), "Inelastic materials and mathematical variables for obstacle bridge problem evaluation," Materials Today: Proceedings Volume:75 / 45-53 / 2022
23. Ganesh E, Ramana P.V, Shrimali M.K (2022), "3D Wave Problems Evaluation and Forecasting Through an Innovative Technique," Materials Today: Proceedings Volume:68(2) / 634-639 / 2022
24. Ramana P.V. (2022), "Temperature effect and microstructural enactment on recycled fiber concrete," Materials Today: Proceedings Volume:65 / 56-64 / 2022
25. Meena A, Ramana P.V (2022), "Highly nonlinear mathematical regression for polypropylene material strength prognostication," Materials Today: Proceedings Volume:62(2) / 696-702 / 2022
26. Ayush Meena, P.V. Ramana (2022), "Mathematical model valuation for recycled material mechanical strengths," Materials Today: Proceedings Volume:60(1) / 753-759 / 2022
27. Ayush Meena, P.V. Ramana (2021), " Mechanical strength over reinforced concrete without infill structure," Materials Today: Proceedings (Scopus) Volume:46 / 8783-8789 / 2021
28. P.V. Ramana (2021), "GGBS: fly-Ash Evaluation and Mechanical properties within high strength concrete," Materials Today: Proceedings (Scopus) Volume:56 / 2920-2931 / 2021
29. Ramana P.V (2021), "Assessment of Mechanical Properties and Workability for Polyethylene Terephthalate Fiber Reinforced Concrete," Materials Today: Proceedings (Scopus) Volume:38p5 / 2960-2970 / 2021
30. P.V. Ramana (2021), "Salvaged malleable concrete and its perfunctory, micro-structural and robustness competencies," Materials Today: Proceedings (Scopus) Volume:53 / 2970-2981 / 2021
31. Surendranath A, Ramana P.V (2021), "Recycled materials execution through digital image processing," Materials Today: Proceedings (Scopus) Volume:46 / 8795-8801 / 2021
32. Kunal Bisht, Ramana P.V (2020), "Gainful utilization of waste glass for the production of sulphuric acid resistance concrete," Construction and Building Materials Volume:235 / 117-124 / 2020
33. Ramana P.V (2018), "Sustainable production of concrete containing discarded beverage glass as fine aggregate," Construction and Building Materials Volume:177 / 116-124 / 2018
34. P.V. Ramana (2018), "Influence of iron content on the structural and magnetic properties of Ni-Zn ferrite nanoparticles synthesized by PEG assisted sol-gel method," Materials & Design Volume:465 / 747-755 / 2018
35. Ramana P.V (2017), "Evaluation of mechanical and durability properties of crumb rubber concrete," Construction and building material Volume:155 / 811-817 / 2017
36. P.V. Ramana (2014), "Modified Adomian Decomposition Method for Van Derpol Equations," International Journal of Non-Linear Mechanics Volume:65 / 121 / 2014
37. Ramana P.V (2013), "A Novel Procedure for Solving Multi-Support Obstacle Problems'', International Journal on Mechanical Engineering and Robotics (IJMER) Volume :1 / 108-119 / 2013
38. P.V. Ramana (2013), "An FSM Analysis for Box Girder Bridges," International Journal of Advanced Electrical and Electronics Engineering (IJAEEE) Volume:2 / 9Decomposition
39. Ramana P.V and Raghu Prasad B.K (2013), "Modified Adomian Decomposition Method for Non-linear Dynamic Problems," International Journal of Nonlinear Mechanics Volume:01 / 101-109 / 2013
40. P.V. Ramana and B K Raghu Prasad (2012), "Modified Adomian Decomposition Method for Uni-Directional Fracture Problems," Academy of Engineering Sciences Volume:37 / 33-57 / 2012
41. Ramana P.V and Raghu Prasad B.K (2011), "Modified Adomian Decomposition Method for Multi-Directional Fracture Problems," International Journal of SEWC Volume:27 / 15-22 / 2011
42. P. V. Ramana and B K Raghu Prasad (2011), "Modified Adomian Decomposition Method for Helmholtz Problems," IJEST Volume:7 / 1-17 / 2011

43. Ramana P.V. and Raghu Prasad B.K (2011), "Modified Adomian Decomposition Method for Elliptical Problems," Applied Mathematics and Computation Volume:109 / 8-29 / 2011

44. Ayush Meena, P.V. Ramana (2022), "Mechanical strength evaluation for morsel rubber material via mathematical models," Materials Today: Proceedings Volume:62(2) / 24-32 / 2022

45. Naveen Tiwari, Ganesh, E., & Ramana, P. V. (2022), "Seismic Characterization of Symmetrical Structure Damage and Ductility Obligation." Materials Today: Proceedings

46. Rohit Naharla, Ganesh, E., & Ramana, P. V. (2022), "Seismic Effects on Reinforced Concrete Multi-Storey Structure NDTHA Assessment," Materials Today: Proceedings

47. Ayush Meena, P.V. Ramana (2021), Assessment of endurance and microstructural properties effect on polypropylene concrete, Materials Today: Proceedings, 38, Part 5.

48. P.V. Ramana (2020), Functioning of bi-material interface intended for polypropylene fiber concrete, Materials Today: Proceedings,14, Part 2.

49. Anamika Agnihotri, P. V. Ramana "Strength and Durability Analysis of GGBS & Recycled Materials," International Conference on Advances in Civil and Structural Engineering (ICACSE-2020), May 28-30, 2020.

50. Ganesh, E., Ramana, P. V., & Shrimali, M. K. (2022). Inelastic materials and mathematical variables for obstacle bridge problem evaluation. Materials Today: Proceedings.