Testing the Sensitivity of Machine Learning Classifiers to Attribute Noise in Training Data.

Willem Theodorus Schooltink
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
w.t.schooltink@student.utwente.nl

ABSTRACT

As datasets in the real world are often filled with some degree of noise in the data, emerging from several possible factors such as human error, a lot of research has been done on data cleaning algorithms. A notably less studied aspect of the data quality problem is research on the degree that noise in data affects classifier accuracy. This paper provides insights through an experimental approach to determine the impact different levels of noise in training data has on the accuracy of a resulting classifier, for Support Vector Classifiers and Random Forest Classifiers. The experiments show a high tolerance for noise in sensor data across both classifiers. With these results, one might be able to tune data cleaning algorithms or make an informed decision on what machine learning technique to choose based on a known data dirtiness.

Keywords

Machine learning, Classifier, Accuracy, Attribute Noise, Data Quality Impact, Support Vector Machine, Random Forest.

1. INTRODUCTION

In the modern world, where engineers are ever so more aiming to solve complex tasks[1, 4, 13, 15], such as digital artificially intelligent personal assistants, machine learning has become increasingly prevalent and complex; The development of artificially intelligent personal assistants, as well as image recognition algorithms, and big-data analysis, to name a few fields, is of great focus to many institutions and companies[7]. Such classifiers are ideally trained with perfect data, however, in the real world, data often contains noise. A lot of research has been performed on algorithms that clean the noise from datasets, with varying levels of accuracy and complexity. There is, however, a lack of research on the the actual impact of noise in data on the accuracy of classifiers.

This research will provide insight in the degree of sensitivity of certain machine learning classifiers to varying noise in training data. The results allows noise reduction algorithms to be tuned such that the resulting dataset is at a certain level of dirtiness which is deemed acceptable to the performance of machine learning classifiers. The research will specifically focus on the accuracy of machine learning classifiers for multi-dimensional sensor data.

Another insight that is aimed to be provided by this research is some added knowledge in the field of data imputation. The research will not only compare the results of different classifiers against one another, but different types of noise will be tested, which can be extrapolated to a comparison on what noise provides the best results if used to fill missing values.

33rd Twente Student Conference on IT July. 3rd, 2020, Enschede, The Netherlands.

2. BACKGROUND

2.1 Machine Learning Techniques

This research will look at the impact of different levels of noise in training data over multiple machine learning techniques. The techniques that will be tested are Random Forests and Support Vector Machines (SVMs).

2.1.1 Random Forests

Random forests are an extension on the decision tree machine learning technique. A decision tree is, as the name suggests, a tree graph in which each node the following path is determined by analysing the (a part of) the value(s) put into the decision tree. The tree will end in terminal leafs which contain a classifying label. The leaf that is reached by following the decision tree will be the prediction made by the decision tree. Such a decision tree can be trained by providing labeled training data, resulting in a, ideally, accurate classifier.

Random forests ensemble a multitude of decision trees, all of which are trained with different folds of the training data that is given to the Random Forest. As each decision tree in the forests has received different training data, the trees are not necessarily the same, and can therefore predict different answers to the same input. A Random Forest Classifier lets each of its trees vote on a classification and, based on these votes, determines the classification that is most likely to be correct. This way of classifying has been shown to provide more accurate classification[3].

2.1.2 Support Vector Machines

Support Vector Machine Classifiers are a machine learning algorithms that take a input data and classifies it in a classification. A Support Vector Machine that is used for classification of data is also referred to as Support Vector Classifier (SVC). This classification is achieved by taking all values in an input and map each of them to a dimension, such that the input is a point in a multi-dimensional space. A SVC trains by receiving labeled training data, which it inputs in the multi-dimensional space, after which vectors are generated that separate the space into areas, each of which will be associated with a classification. After a SVC

is trained inputted data will then be put in the space, and be classified by the area the value falls within.

Figure 1 shows a somewhat simplistic representation of a SVC with 2 dimensional data, and 2 classes. Note that the dashed line splits the space, such that each created area represents a given class. As data gains more dimensions the dimensions of the SVC space increase as well.

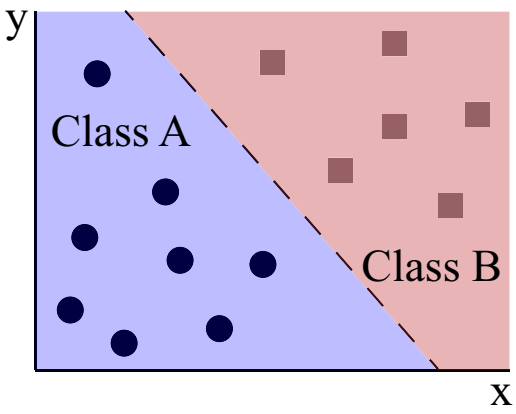

Figure 1. Support Vector Classifier example

2.2 Accuracy calculation

As this research will focus on the impact of noise in training data on the accuracy of classifiers the accuracy of a the classifiers will be determined with the standard metric of accuracy:

$$\frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Where TP are true positives: Values are classified correctly as class A. FP are false positives: Values are wrongly classified as class A. TN are true negatives: Values are correctly classified as class B. And FN are false negatives: Values are wrongly classified as class B.

2.3 Data Noise

Commonly, in data science, there are 2 different kinds of noise defined for machine learning data: Attribute noise and Class noise[16].

1. **Attribute noise** in the values of data attributes, this can include missing fields, or fields with incorrect values.
2. **Class noise** is noise in the labels of data, so any mislabeled data would fall into class noise.

furthermore, previous research has defined several forms of missingness of data, the proposed categories define patterns in the noise. There are 3 categories proposed for the missingness of data[14]:

1. **Missing Completely At Random (MCAR):** The missing values are completely at random, there is no correlation between the aspects of an entry and the likelihood of a value missing.
2. **Missing At Random (MAR):** Also sometimes referred to as missing conditionally at random, this means that there is a correlation on the likelihood that a value is missing and some other aspect of the data. An example of this could be that in a survey between people that there is a greater chance that a certain value is missing if that person is over the age of 70.
3. **Not Missing At Random (NMAR):** This one is often a bit more difficult to grasp: The likelihood of value missing varies, but because of unknown factors[12]. An example of this could be unnoticed wear in sensors such that it produces more noise over time.

This research will focus on the impact of attribute noise on machine learning accuracy, following the MCAR method of introducing noise.

3. RESEARCH QUESTIONS

The aim of this research is to answer the following research questions:

- To what degree do different levels of attribute noise in training data impact the accuracy of a Support Vector Classifier using multi-dimensional sensor data?
- To what degree do different levels of attribute noise in training data impact the accuracy of a Random Forest Classifier using multi-dimensional sensor data?
- How does Gaussian noise compare to mean value imputation on the accuracy of classifiers, when introduced on training data.

4. RELATED WORK

4.1 Data Quality

Data quality is often referred to as dirtiness of data, where high dirtiness is synonymous to poor quality data. This paper already defined attribute- and class noise, the causes and more specific sub-classifications of noise have been discussed in previous research papers[9, 11]. Attribute noise, in the form of missing data, can arise from a multitude of factors, these include possible errors while processing or saving data, or fields intentionally left empty in surveys and questionnaires. Furthermore, as data science is a growing field of interest, datasets can be used for purposes not initially intended, this might also introduce noise in the data as it is being transformed into a desired format.

4.2 Impact of Data Quality on the Accuracy of Machine Learning

Existing research on the impact of data noise has shown how data noise can be detrimental to machine learning classifiers trained on that data[16]. This previous research provide insights into how to handle noise in data, suggesting methods to reduce noise in data, through either correction or deletion of noise. It has been opted that the best method of noise reduction in data is prevention of noise in the first place[9], however, if such prevention is not possible best practices for data handling are suggested.

Research on the impact of noise on machine learning has resulted in several unique approaches to the problem. One of these approaches is to define a manner to determine, through a mathematical approach, to what degree noise in training data limits the accuracy of trained classifiers, non-specific to the machine learning technique used[5].

Another approach taken up is to, through experiments, determine the sensitivity of certain machine learning techniques to data noise. Such research can and have provided insights into how sensitive machine learning techniques are to data noise, compared to one another[10]. This research

aims to provide new insights in the latter approach, by experimenting on - and comparing the results of yet unexplored combinations of machine learning techniques.

5. METHODOLOGY

5.1 Selecting a Dataset

There exist a multitude of online platforms where publicly available datasets for machine learning are published[1][2][3]. For this research a dataset containing gyroscopic and accelerometer data of people running or walking. This dataset can be found on the online platform Kaggle[4].

The specific dataset that will be used for this research has entries with the structure as described in table 1.

Value Description	Data type
Activity	Boolean; 0 (Walking) or 1 (Running)
On wrist	Boolean; 0 (True) or 1 (False)
Acceleration X	Floating point number
Acceleration Y	Floating point number
Acceleration Z	Floating point number
Gyro X	Floating point number
Gyro Y	Floating point number
Gyro Z	Floating point number

Table 1. Data structure

As table 1 shows, the data contains gyro and accelerometer readings. The gyroscope displays, to a certain degree of accuracy, the angle of the phone in a 3-dimensional space. And the accelerometer shows the acceleration of the device, also in 3 dimensions relative to the device. Furthermore the data contains 2 Boolean values: Activity, which tells whether the person was running or walking at the time. And 'On wrist', which tells whether the measuring device was on the wrist of the person at the time, this could for example be a smartwatch.

5.2 Generating Noise

This noise is only generated over the training data set, the test set has not been altered in order to measure the impact of noisy training data on classifying non-noisy data. For the purpose of this research the following levels of data noise have been tested: 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100%, each percentage meaning that the given ratio of values are being replaced with noise.

5.2.1 Introducing Noise

In order to introduce noise in a dataset, one must determine which of the values will be replaced with noise. As stated before, different levels of noise will be tested, described in percentages. Then a number of values in the data set is selected such that the amount selected is equal to the the percentage of noise times the amount of total values[10]. Note the difference of entries and values here: An entry contains multiple values, so a row in a dataset is called an entry, and a specific value of an entry is called a value.

The pseudo-code will clarify how such an selection, as described above is made.

[1]https://www.kaggle.com/
[2]http://deeplearning.net/datasets/
[3]https://www.datasetlist.com/
[4]https://www.kaggle.com/vmalyi/run-or-walk/data

Algorithm 1 Select Values

```
1: procedure SelectValues
2:     columns ← Indices of alterable columns.
3:     dataset ← The dataset to introduce noise on.
4:     noiseCount ← length(columns) ∗ length(dataset).
5:     sample ← randomSample(noiseCount, dataset).
6:     i ← 0.
7:     loop:
8:     if i < length(sample) then
9:         entry ← sample[i]/len(columns).
10:        value ← columns[sample[i]%len(columns)]
11:        dataset[entry][value] ← generateNoise()
12:        i ← i + 1.
13:        goto loop.
```

5.2.2 Gaussian Noise

Considering the nature of the signals that are to be processed, one can introduce noise to the data in the following way: For all values that are sensor readings a Gaussian distribution with the minimum and maximum of the values of the column is created. Then for all of the selected values a random number will be generated using the Gaussian distribution of that column. This process is repeated over all columns. This type of noise can represent sudden spikes that accelerometer and gyroscopic sensors might (incorrectly) detect, due to faults in the sensor themselves, or vibrations of the sensors within a device which could occur if the sensors are note fitted exactly right. Another possible source of noise in sensor readings, is electronic interference from either within the circuit or external sources.

Algorithm 2 Gaussian Noise

```
1: procedure GenerateGaussianNoise
2:     mean ← The mean of the column of the value.
3:     stDev ← The standard deviation of the column.
4:     max ← The maximum value of the column.
5:     min ← The minimum value of the column.
6:     noise ← randGauss(mean, stDev).
7:     if noise > max then
8:         noise ← max.
9:     if noise < min then
10:        noise ← min.
11:    return noise.
```

5.2.3 Missing values

A second type of noise that will be tested separately is noise through missing data. Missing data can occur in many ways, for example in sensor systems if the sensor encountered an error while reading the state of the sensor or through failure to save them. This noise will be generated through deleting values completely at random, using the selection system as described for the Gaussian noise before. In order to make the dataset then usable for the classifiers, all missing fields will be filled with the mean value of that column[9], this will be referred to as *Mean Noise*. This replacement has to be done, as classifiers often do not work well with empty values. The replacement of missing values in a dataset is studied within the field of data imputation.

5.3 Implementation of Machine Learning Classifiers

This research tests SVMs and Random Forests. In order to mitigate the chance on human error in implementing these techniques, preexisting implementations have been used. These implementations specifically came from the

sklearn package for Python[5].

For the research, the default implementation of the SVC and the Random Forest Classifier from the sklearn package have been used.

5.4 Experimental Testing

In order to get results to draw conclusions from, several experiments have been performed. For each level of noise both the SVC and the Random Forest Classifier were trained on a dataset with the given amount of noise. After the training, the accuracy of both classifiers was tested on a test dataset which contains no introduced noise.

5.4.1 Result Validation

In order to validate the results of the experiments, K-fold cross validation is used. This technique splits a dataset into different so-called folds. Each fold has subsections of the original dataset shuffled in order. A part of such a fold is then used as test data, and the remaining larger set is used as training data.

Furthermore, the generation of both the noise in the classifiers and the datasets require some randomness. This randomness is achieved true seeded randoms. These randoms are provided with an initial seed, such that they will produce the same randomness every time the experiment is rerun. In order to test whether the selected seed generated outlying results, several different seeds were tested on a subset of the experiments.

6. RESULTS

The experiments as described previously have been been performed, and the results of them are available in the appendix A.1. The results show the average of the classifier accuracy over 10 different folds of the dataset, as well as the standard deviation of the results. The results of both types of noise have been plotted in order to visualize the data:

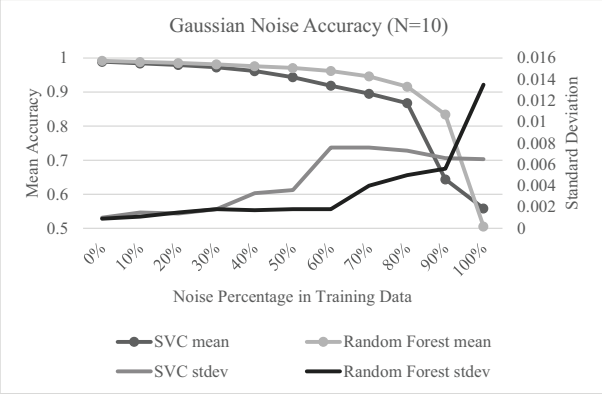

Figure 2. Gaussian Noise Accuracy

Figures 2. and 3. show the decline of accuracy of the machine learning classifiers as the level of noise in the training data increases. Figure 2. shows that the Support Vector Classifier has a greater sensitivity to high levels of of Gaussian noise up until the 100% noise level. However, at 100% noise all of the original sensor values are replaced with noise, so the results at this level do not say much. Furthermore, note that while the Gaussian noise shows a more gradual drop in accuracy as compared to the more sudden plummet of accuracy of the mean noise.

[5]https://scikit-learn.org/

Figures 2. and 3. also display the standard deviation of the accuracy over the percentages, the graph show a rise in standard deviation as the percentage of noise increases. However, standard deviation is in all cases very low: almost exclusively under 1%. This low deviation show that the results are consistent over all folds.

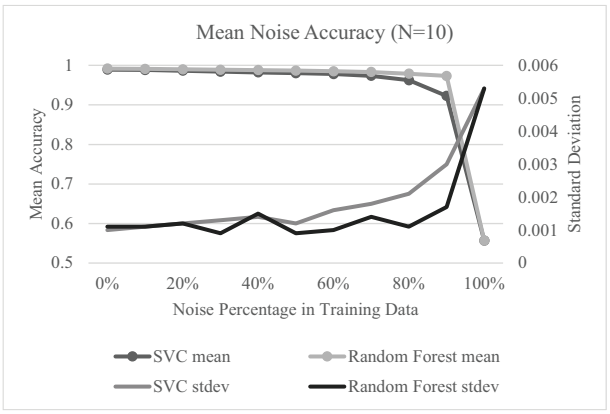

Figure 3. Mean Noise Accuracy

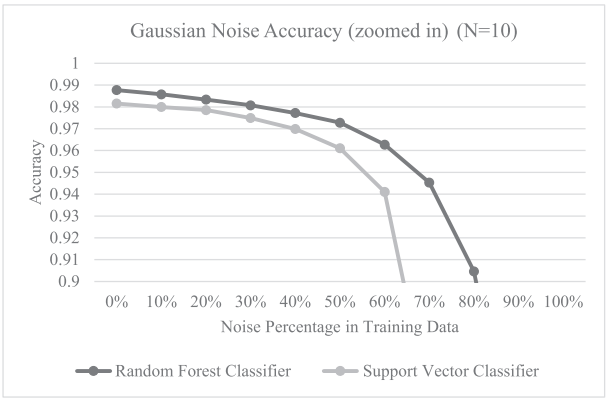

Figure 4. Gaussian Noise Accuracy (zoomed in)

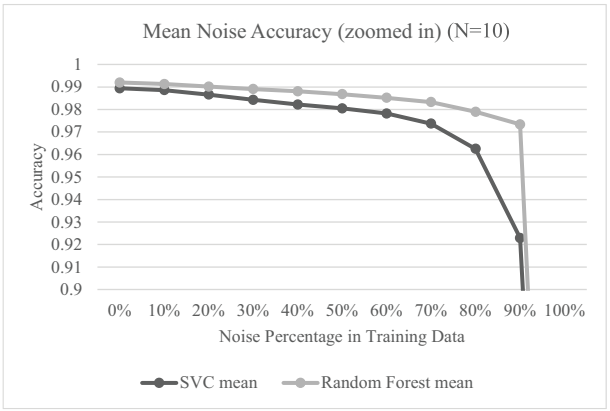

Figure 5. Mean Noise Accuracy (zoomed in)

In order to compare the classifiers more closely figures 4. and 5. show zoomed in views of figure 2. and 3. respectively. The zoomed in graphs show one interesting finding: Overall, the accuracy of the Random Forest Classifier is slightly higher than that of the Support Vector classifier. While the difference in accuracy is small, it seems to be consistent. This holds for both the experiments with the

Gaussian noise as well as the experiments where values were replaced by the means of the column.

7. CONCLUSION

The results show that both the Support Vector Classifier and the Random Forest Classifier have a high tolerance for noise in the specific dataset. In the experiment with Gaussian noise the Support Vector classifier shows a significantly higher sensitivity to noise levels over 60% than the Random Forest Classifier.

The results also show that the Support Vector Classifier performs slightly worse overall in both tests, the difference between the classifiers is minimal. However, the standard deviation of the accuracy between the folds show that this performance is consistent.

Finally, the experiments show that the type of noise greatly affects the accuracy of trained classifiers. In the experiments with Gaussian noise the graphs show a more gradual decline in accuracy, whereas the experiments where values are replaced with the mean of the column show accuracy of over 90% up until the entire training set is replaced with noise, at which point values will be indistinguishable.

8. DISCUSSION

The fact that the Support Vector Classifier showed an overall lower accuracy over all the tests, and a higher sensitivity to data noise is an interesting finding that seems to align with previous research. Previous research on the impact of Class Noise on Random Forest Classifiers, for example, have show that Random Forest Classifiers do perform well in those circumstances[6]. And without the introduction of noise, a study on the comparison of the accuracy a multitude of classifiers, including Random Forests and Support Vector Machines, has show that in those circumstances the Random Forest Classifier also outperforms the Support Vector Classifier[8].

Another interesting result is the comparison of the different types of noise, as the results show a gradual decline in accuracy when introducing Gaussian noise, as compared to the mean noise. One possible reason for this could be that the classifiers learn to ignore the noise in mean replacement easier, as it is consistently the same value for each entry in a column. Possibly, the classifiers learn to then emphasise the values that deviate from this mean. This is, however, speculation and further research will have to done in order to confirm or refute this.

One must, however, note the limitations of the conclusions of this paper: The research has been preformed on a dataset of mostly sensor readings. The findings of the research may not necessarily apply to different types of datasets.

9. FUTURE WORK

This research on the sensitivity of machine learning classifiers on noise still allows for many directions to be taken. This paper discusses a select subset of types of noise and machine learning classifiers, this can be greatly expanded on.

One of the proposed future topics is research on different, other types of trained classifiers. For example, one might expect that certain machine learning techniques show a higher sensitivity to noise in training data, resulting partly from the assumptions each model makes about aspects of data; Bayesian Networks, for example, rely on the assumption of conditional independence of the data[2], although real world data often does have weak conditional dependencies in the data. Research on this topic could lead to novel and valuable new insights.

Furthermore, there are many different types of noise that can be tested. One could opt to research specific datasets and look at noise which would be relevant to that certain type of data, or look at what types of techniques are studied and used in the field of data imputation. Additionally, one could look more into the different types of noise in the sense of on which values noise is introduced (MAR, MCAR, NMAR). This could be more extensively studied: Is there a difference in sensitivity to these three different methods of noise introduction?

And a final path to take, as an extension to this research would be to test different types of data. This research was focused on training classifiers with sensor data. Possible other types of data could be surveys, questionnaires, or event-logs for example.

10. REFERENCES

[1] B. Baruque, E. Corchado, A. Mata, and J. M. Corchado. A forecasting solution to the oil spill problem based on a hybrid intelligent system. *Information Sciences*, 180(10):2029–2043, 2010.

[2] I. Ben-Gal. *Bayesian Networks*. American Cancer Society, 2008.

[3] L. Breiman. Random forests. *Machine Learning*, Jan 1985.

[4] S.-B. Cho. Pattern recognition with neural networks combined by genetic algorithm. *Fuzzy Sets and Systems*, 103(2):339–347, 1999.

[5] C. Cortes, L. D. Jackel, and W.-P. Chiang. Limits on learning machine accuracy imposed by data quality. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, KDD'95, page 57–62. AAAI Press, 1995.

[6] A. Folleco, T. M. Khoshgoftaar, J. Van Hulse, and L. Bullard. Software quality modeling: The impact of class noise on the random forest classifier. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3853–3859, 2008.

[7] Grand View Research. Artificial intelligence market size, share: Ai industry trends report 2025, Dec 2019.

[8] M. Liu, M. Wang, J. Wang, and D. Li. Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and chinese vinegar, Dec 2012.

[9] P. Lodder. To impute or not impute: That's the question. *Advising on research methods: Selected topics 2013*, January 2014.

[10] D. F. Nettleton, A. Orriols-Puig, and A. Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306, 2010.

[11] P. Oliveira, F. Rodrigues, and P. Rangel Henriques. A formal definition of data quality problems. 01 2005.

[12] A. Pedersen, E. Mikkelsen, D. Cronin-Fenton, N. Kristensen, T. M. Pham, L. Pedersen, and I. Petersen. Missing data and multiple imputation in clinical epidemiological research. *Clinical Epidemiology*, Volume 9:157–166, 03 2017.

[13] J. Perols, K. Chari, and M. Agrawal. Information

market-based decision fusion. *Management Science*, 55(5):827–842, 2009.

[14] D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[15] G. Yang, Y. Lin, and P. Bhattacharya. A driver fatigue recognition model based on information fusion and dynamic bayesian network. *Information Sciences*, 180(10):1942–1954, 2010.

[16] X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22:177–210, November 2004.

APPENDIX

A. TABLES

A.1 Classifier Accuracy

Gaussian Noise		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Random Forest	μ	0.9919	0.9887	0.9856	0.9815	0.9761	0.9710	0.9622	0.9463	0.9161	0.8343	0.5055
	σ	0.0009	0.0011	0.0015	0.0018	0.0017	0.0018	0.0018	0.0040	0.0050	0.0056	0.0135
SVC	μ	0.9894	0.9845	0.9797	0.9728	0.9618	0.9439	0.9189	0.8956	0.8677	0.6440	0.5587
	σ	0.0010	0.0015	0.0014	0.0018	0.0033	0.0036	0.0076	0.0076	0.0073	0.0066	0.0065
Mean Noise		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Random Forest	μ	0.9920	0.9913	0.9902	0.9891	0.9881	0.9868	0.9852	0.9833	0.9790	0.9734	0.5566
	σ	0.0011	0.0011	0.0012	0.0009	0.0015	0.0009	0.0010	0.0014	0.0011	0.0017	0.0053
SVC	μ	0.9894	0.9886	0.9866	0.9843	0.9822	0.9805	0.9782	0.9737	0.9625	0.9229	0.5566
	σ	0.0010	0.0011	0.0012	0.0013	0.0014	0.0012	0.0016	0.0018	0.0021	0.0030	0.0053