The Application of Machine Learning Algorithms in Classification of Malicious Websites

Tabassom Sedighi
Reza Montasari
Amin Hosseinian-Far

Centre for Environmental and Agricultural Informatics, School of Water, Energy and Environment, Cranfield University, Cranfield.
Hillary Rodham Clinton School of Law,
Swansea University, Swansea.
Department of Business Systems and Operations,
University of Northampton, Northampton.

{t.sedighi@cranfield.ac.uk}
http://www.cranfield.ac.uk
{Reza.Montasari@Swansea.ac.uk}
http://www.swansea.ac.uk
{amin.hosseinianfar@northampton.ac.uk}
https://www.northampton.ac.uk

Abstract

This chapter compares three different machine learning techniques, i.e. Gaussian process classification, decision tree classification and support vector classification, based on their ability to learn and detect the attributes of a malicious website. The data used have all been sourced from HTTP headers, WHOIS lookups and DNS records. As a result, this does not require parsing of the website content. The data are first subjected to multiple steps of pre-processing including: data formatting, missing value replacement, scaling and principal component analysis.

Keyword: *Gaussian Process; Decision Tree; Support Vector; Classification; Malicious Applications; WHOIS; HTTP Headers; DNS Records; Correlation Matrix*

1. Introduction

In today's society, our inescapable reliance on technology makes it almost impossible to ignore the ever-present dangers of malicious online resources and the threat that they pose to financial, personal and business security. The computer security company Kaspersky states in their 2016 statistics report that 31.9% of their customers computers were "subjected to at least one Malware-class web attack over the year" and that "261,774,932 unique URLs were recognized as malicious by web antivirus components" (Garnaeva et al., 2016). These statistics show how important it is to be cautious when using the web. In this chapter, the competency of machine learning algorithms are compared and evaluated with a view to determining how effective these are to detect malicious websites.

These comparison and evaluation are based only on the data that can be obtained from HTTP headers, WHOIS data and DNS records. The advantage of using only this data is that it can all be obtained without the need to parse any code located on the client or the server which could have potentially harmful effects. To achieve the stated objectives, first, the dataset will be heavily pre-processed into an appropriate format so that it can efficiently be utilised. Next, the dataset will be subject to sampling, scaling and dimensionality reduction before three machine learning algorithms are applied with the aim of successfully identifying whether the data are describing a malicious or benign website.

2. The Dataset

The dataset selected for this study is "Malicious and Benign Websites", provided by Urcuqui (2018) on Kaggle. The dataset contains information about 1781 unique websites. Out of these websites, 1565 are benign and 216 are malicious. For each website, the dataset contains 21 attributes of metadata that describe information about the application and network layers of the website, all of which are freely available for public access. The first attribute is named URL, which have all of its values replaced with unique identifying values in order to protect the anonymity of the data. This attribute is therefore relatively futile to a machine learning algorithm as every value is unique and unrelated and will therefore not be used. The last attribute is the 'Type' of the website, i.e. malicious or benign, and is a binary value of 0 or 1 with 0 being benign. Therefore, there are 19 potentially useful attributes in the dataset which can be refined later using dimensionality reduction techniques.

3. Data Preparation

This section provides an outline of how the raw dataset was modified and prepared so that it would be ready to feed into machine learning algorithms. This includes processes such as missing value handling, dimensionality reduction and data normalisation.

3.1 Data Analysis

Of the 19 attributes in the dataset, excluding the URL and Type, 13 contain numerical data, 4 are categorical and 2 contain date-time values. Tables 1, 2 and 3 provide information concerning each of these attributes.

Table 1. Information about numerical attributes

Name	Min	Max	Mean
URL_LENGTH	16	249	56
NUMBER_SPECIAL_CHARACTERS	5	43	11
CONTENT_LENGTH	0	649263	11726
TCP_CONVERSATION_EXCHANGE	0	1194	16
DIST_REMOTE_TCP_PORT	0	708	5
REMOTE_IPS	0	17	3
APP_BYTES	0	2362906	2982
SOURCE_APP_PACKETS	0	1198	18
REMOTE_APP_PACKETS	0	1284	18
SOURCE_APP_BYTES	0	2060012	15892
REMOTE_APP_BYTES	0	2362906	3155
APP_PACKETS	0	1198	18
DNS_QUERY_TIMES	0	20	2.26

Table 2. Information about Categorical Attributes

Name	Unique Count	None Count
CHARSET	9	7
SERVER	240	175
WHOIS_COUNTRY	49	306
WHOIS_STATEPRO	182	362

Table 3. Information about Timestamp Attributes

Name	Unique Count
WHOIS_REGDATE	DD/MM/YYYYHH:MM
WHOIS_UPDATED_DATE	DD/MM/YYYYHH:MM

3.2 Data Formatting and Conversion

3.2.1 Numerical Attributes

The numerical data attributes were naturally in the correct format to be used by a machine learning algorithm with the only issue being any values that were set to N/A. These were all replaced with the value -1 as there were no other negative values in the dataset and this allowed for a clear distinction between true values and missing values.

3.2.2 Categorical Attributes

The first decision that was made about the categorical attributes was that the WHOIS_STATEPRO attribute would not be used as it has a very large number of missing values. As a result, it would be unlikely to be useful to the machine learning algorithms. The server attribute also has a high number of unique values and 'None' values. However, the server attribute would almost certainly be an effective indicator if there were enough data entries and adequate pre-processing were performed on it. Therefore, it was decided to keep this attribute.

The categorical attributes required much more pre-processing before they could be ready to be used by a ML algorithm. First, the datasets were analysed, and any values that represented the same category were combined into one category. For instance, given that, in the WHOIS_COUNTRY column, there were values 'us' and 'US', these were converted so that all values referring to the United States would be 'US'. This was applied to all categorical data. For each categorical attribute, the full set of unique values were then indexed, and each occurrence of each attribute was replaced with its corresponding index value. All missing values and none values were then set to -1 for the same reason as with the continuous data.

3.2.3 Timestamp Attributes

The third and final datatype in the dataset is timestamp data; these required a few steps of processing before they could be used. First, any data values that were not in a timestamp format were either converted manually to the correct format or were converted to 'NaT' meaning Not a Time, if the value did not represent date and time information. Then, the datetime values were all converted into integers that represented the time in seconds so that they were in pure integer form and finally, all NaT values were converted to be equal to a value that would simulate the -1 that has been used for the other 2 data formats, the function for this is displayed below.

$$NaT = Date_{min} - Date_{range,}$$

Where,

- NaT = Not a Time values
- $Date_{min}$ = Lowest time value in the attribute
- $Date_{range}$ = The range of date values

3.3 Random Under Sampling

The dataset being used in this project has a heavy majority of one class over the other. As a result, there are many more benign websites than there are malicious websites. This imbalance can cause overfitting of ML algorithms if it is not dealt with effectively. Since the imbalance is so large, the most appropriate way to address this was to use random under-sampling. Random under-sampling involves reducing the size of that dataset by removing entries from the larger class until the classes contain the same number of instances. This leaves the dataset much smaller than it was originally, but, in some cases, it can significantly improve the accuracy of the ML algorithms, usually on the initially smaller class. Whilst there was the option to employ some forms of oversampling, this would have left the dataset with many copies of every instance in the minority class and could potentially cause bad overfitting to this class.

3.4 Scaling

Once the data have been processed, it could then be utilised for ML; however, the significant variation in data ranges and values can cause the ML algorithms to apply imbalanced importance to the attributes. This issue can be addressed by converting the dataset, so that all attributes have similar statistical attributes such as range, standard deviation or mean. For this study, min-max scaling was selected for a range of -1 to 1. This was due to the fact that the missing data had all been set to equal -1 and that this form of scaling would retain the distinction that was desired when this decision was made. The followings provide the functions for min-max scaling.

$$X_{std} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$X_{scaled} = (X_{std} * (max - \min)) + min$$

- X = Value to be scaled
- X_{min} = Minimum of attribute values

One of the main disadvantages of min-max scaling as opposed to standardisation is that the resulting standard deviations are smaller, denoting that outliers are less easily detected. This has not created an issue for the dataset used in this study since the data are all exact and no errors were made during the collection and production of the data.

3.5 Dimensionality Reduction

Having a large quantity of data is extremely important to ensure the accuracy and quality of ML algorithms. However, this amount of data requires substantial processing power to be able to perform the required calculations. In order to alleviate this issue, there exists several techniques that can be performed to reduce the amount of data without losing the usefulness that it provides. For this study, principle component analysis (PCA) technique was selected. Before the dimensionality reduction was performed, it was essential to visualise the relationships between the attributes in the dataset. For this purpose, a correlation matrix was generated and plotted as shown in Figure 1.

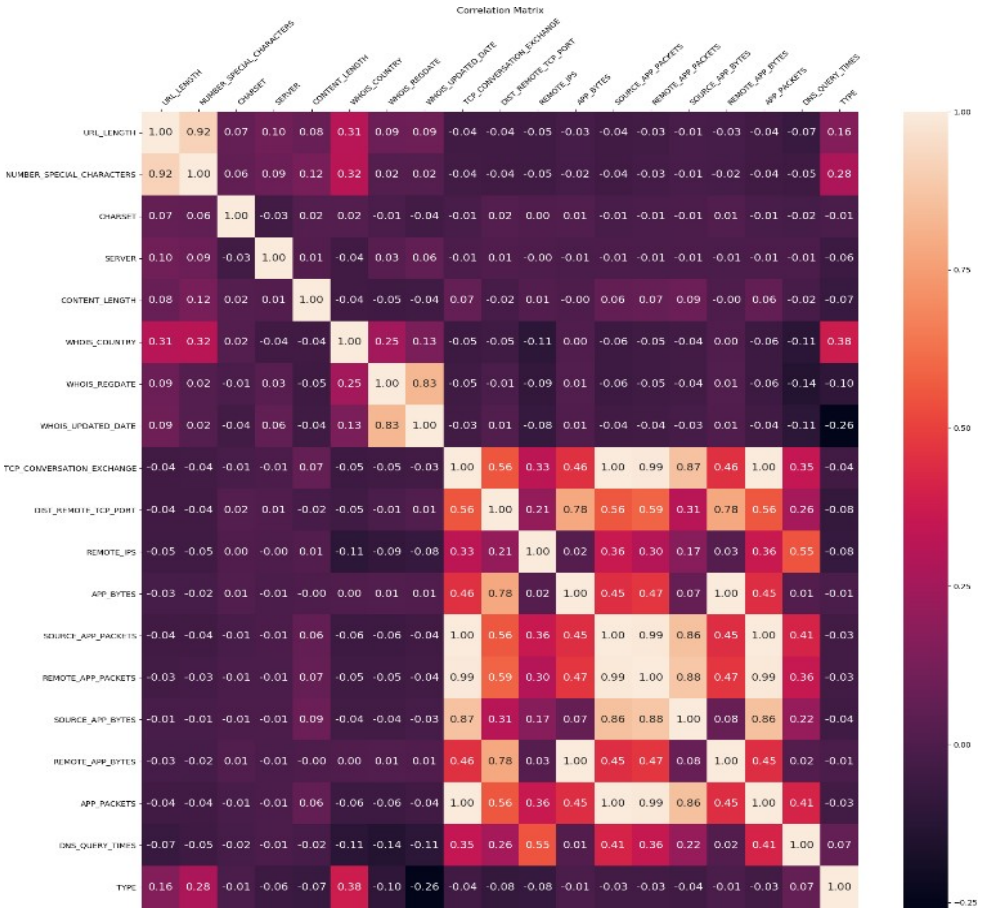

Figure 1. Correlation Matrix prior to PCA.

As represented by Figure 1, a large percentage of the attributes in the bottom left of the plot are very closely correlated to one another. However, they show very little relationship to the classification. On the contrary, *URL_LENGTH, NUMBER_SPECIAL_CHARACTERS, WHOIS_COUNTRY and WHOIS_UPDATED_DATE* are all strongly correlated with the classification. Nevertheless, they have low correlation with each other, indicating that these features were most likely to be useful to the ML algorithms.

3.5.1 Principal Component Analysis

Having plotted the correlation matrix revealed that some forms of dimensionality reduction could have a large impact on the performance of the dataset. PCA is a dimensionality reduction technique that generates a decreased number of features whilst retaining the highest percentage of the underlying information as possible. In order to achieve this, it combines them by projecting all the original data into lower dimensional space in a manner that makes them no longer realistically interpretable by a human. PCA was then performed in a loop for every number of output attributes up to the original count. Next, the classification algorithms (discussed in the next section) were used to analyse the effectiveness of the PCA. The results of this analysis are depicted in Figures 2, 3 and 4.

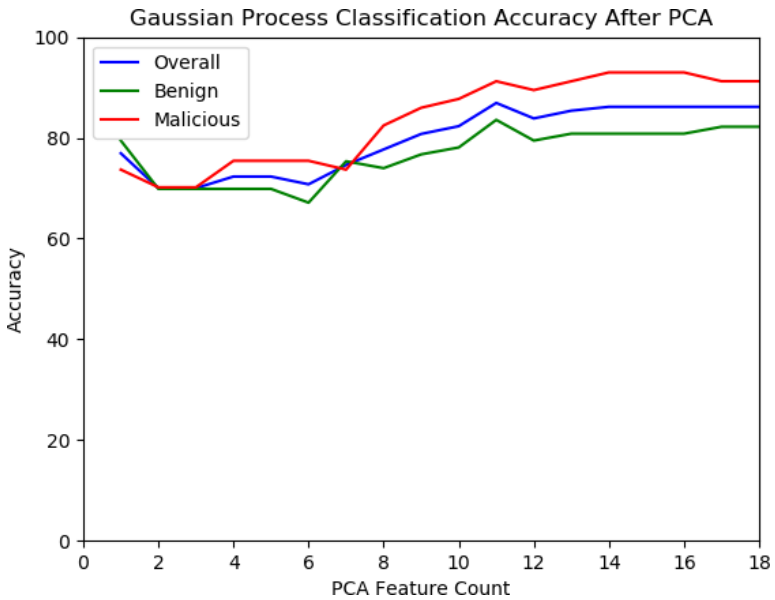

Figure 2. PCA Analysis using Gaussian Process Classification.

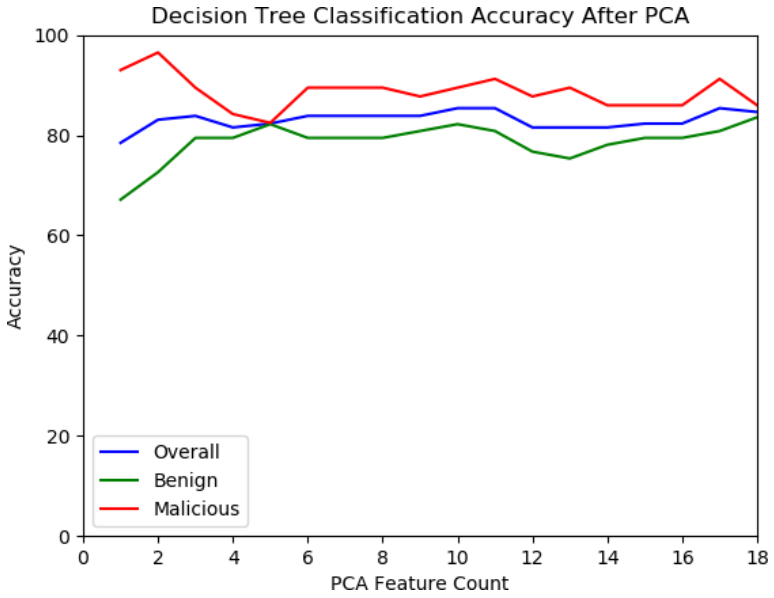

Figure 3. PCA Analysis using Decision Tree Classification.

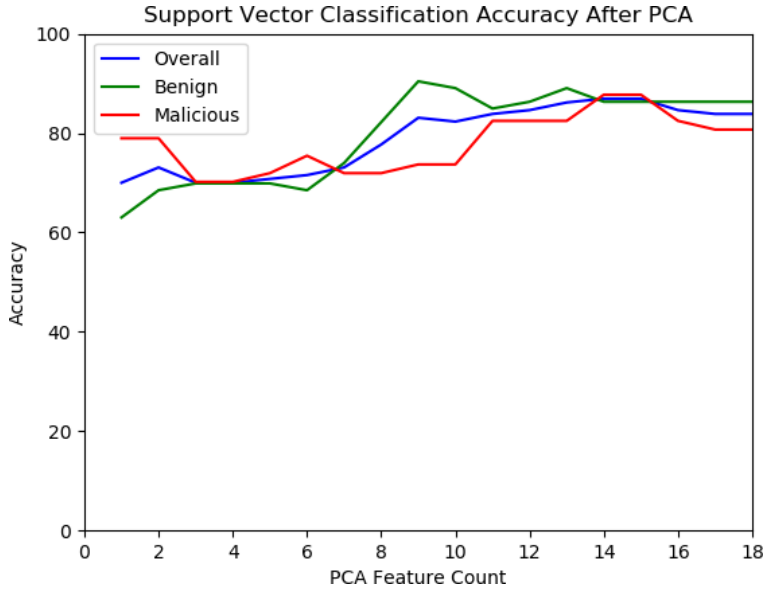

Figure 4. PCA Analysis using Support Vector Classification.

Figures 2, 3 and 4 clearly demonstrate the relevance and usefulness of the PCA to be applied to this dataset as the results for all 3 classifiers do not show any significant improvement above 11 features, i.e. less than 2/3 of the initial amount of data. In view of this, it was decided to use the feature count of 11

through PCA as this appeared to be the point of plateau/ convergence of the Gaussian Process Classification and the Support Vector Classification. Had the Decision Tree Classification was the sole focus of this study, a much lower feature count could have been selected since the reduction of features appears to have much less effect on this classifier. Once the PCA had been completed with a feature count of 11, a new correlation matrix as displayed in Figure 5 was produced to show the relationships between the new features.

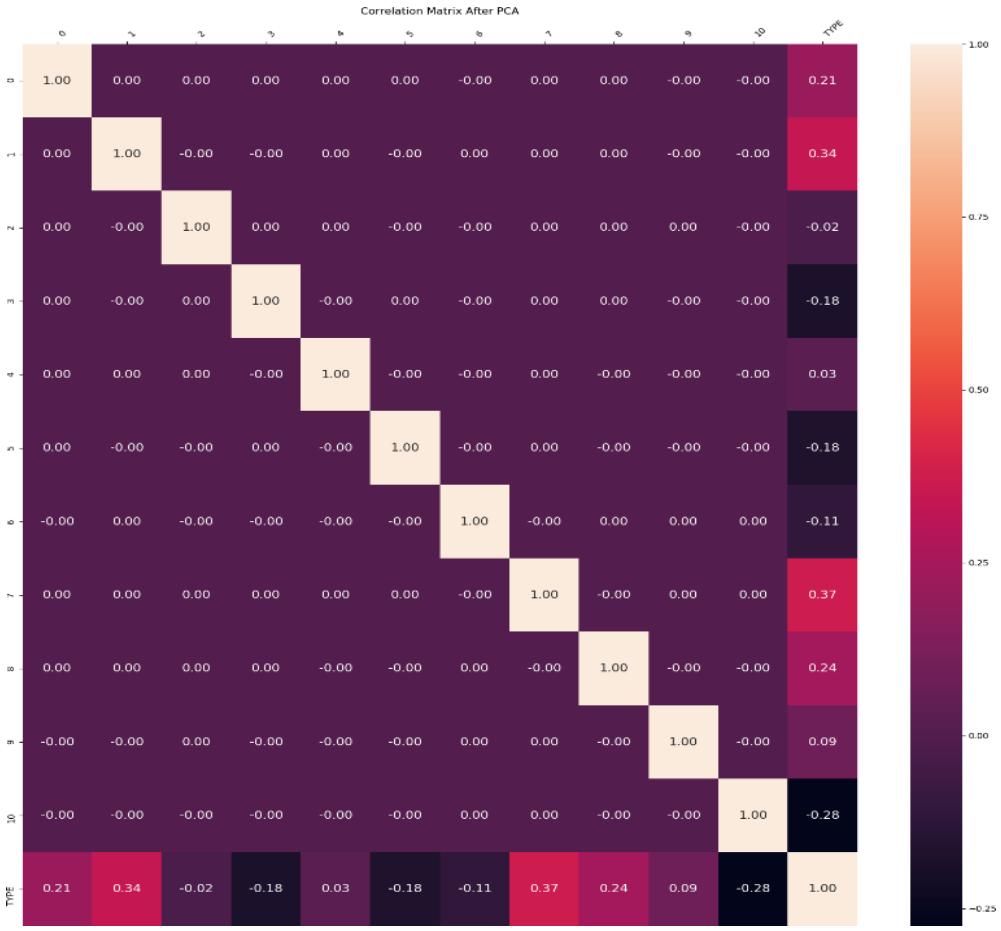

Figure 5. The new Correlation Matrix after the completion of the PCA.

As the new correlation matrix reveals, the PCA has produced a set of features that have very little correlation with each other. However, these features are almost all correlated to the classification, illustrating that it has successfully achieved its intended purpose.

4. Classification Methods

As previously stated, this study involved applying 3 ML classification algorithms to the dataset with the aim of achieving significant prediction accuracy of the classification. The remainder of this section provides an in-depth discussion of these classification methods.

4.1 Gaussian Process Classification (Binary)

The first classification algorithm implemented in this study was Gaussian Process Classification. This ML algorithm uses a regression model to fit the data and then calculates a probability for each class using this. It then determines the most effective probability to use as a splitting point between the prediction classes to find its output predictions. In this case, the regression model is Laplace Approximation. This algorithm is different when dealing with multiple output classes (Williams and Rasmussen, 2006; Daneshkhah et al., 2017; Batsch et al., 2019).

4.2 Decision Tree Classification

Decision Tree Classification is a ML algorithm which progressively splits the dataset by incrementally adding rules that provide the largest increase in prediction accuracy. This process terminates when the accuracy is no longer increasing (Grąbczewski, 2014).

4.3 Support Vector Machine Classification

The Support Vector Classifier attempts optimally to separate classes by constructing hyperplanes that split them. These hyperplanes use linear boundaries if possible but can become much more complex when dealing with non-linearly separable output classes (Friedman et al., 2001).

5. Results

Having carried out all the experiments, the results were tabulated and plotted. This section first provides the overall results in Tables 3 and Table 4, where

- True Negative represents values that were correctly predicted as benign,
- False Positive refers to values that were falsely predicted as malicious,
- False Negative outlines values that were falsely predicted as benign, and
- True Positive define values that were correctly predicted as malicious.

Table 4. Data showing the overall accuracy

Overall Results	Accuracy	Benign Accuracy	Malicious Accuracy
Gaussian	86.92%	83.56%	91.23%
Gaussian no PCA	87.69%	86.30%	89.47%
Decision Tree	84.62%	79.45%	91.23%
Decision Tree no PCA	89.23%	84.93%	94.74%
Support Vector	83.85%	84.93%	82.46%
Support Vector no PCA	82.31%	89.04%	73.68%

Table 5. Data representing overall results counts

	True Negative	False Positive	False Negative	True Positive
Gaussian	61	12	5	52
Gaussian no PCA	63	10	6	51
Decision Tree	58	15	5	52
Decision Tree no PCA	62	11	3	54
Support Vector	62	11	10	47
Support Vector no PCA	65	8	15	47

The remainder of this section offers the results concerning the individual classification methods.

5.1 Gaussian Process Classification Results

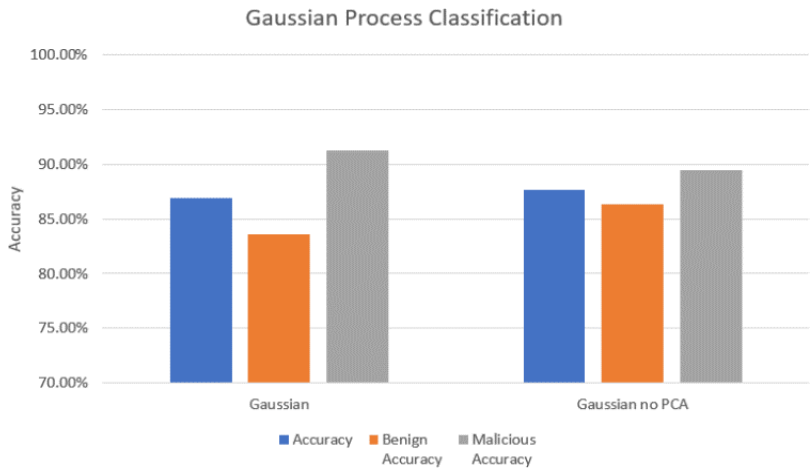

Figure 6. Gaussian Process Classification Results.

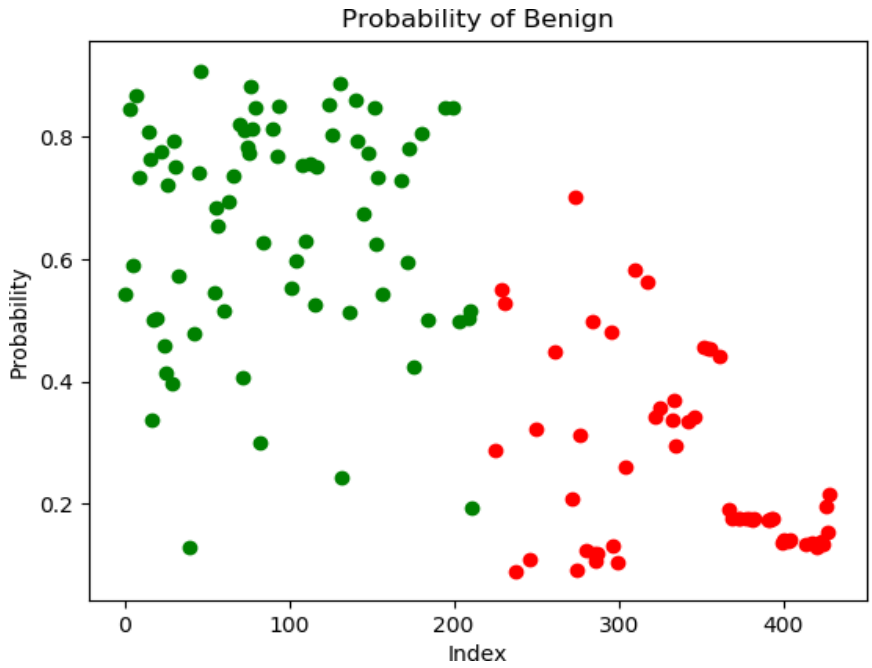

Figure 7. Gaussian Process Benign Probability.

Table 6. Gaussian Process Confusion Matrix (PCA)

Gaussian		**Predicted**	
		Benign	**Malicious**
Actual	**Benign**	61	12
	Malicious	5	52

Table 7. Gaussian Process Confusion Matrix (No PCA)

Gaussian No PCA		**Predicted**	
		Benign	**Malicious**
Actual	**Benign**	63	10
	Malicious	6	51

5.2 Decision Tree Classification Results

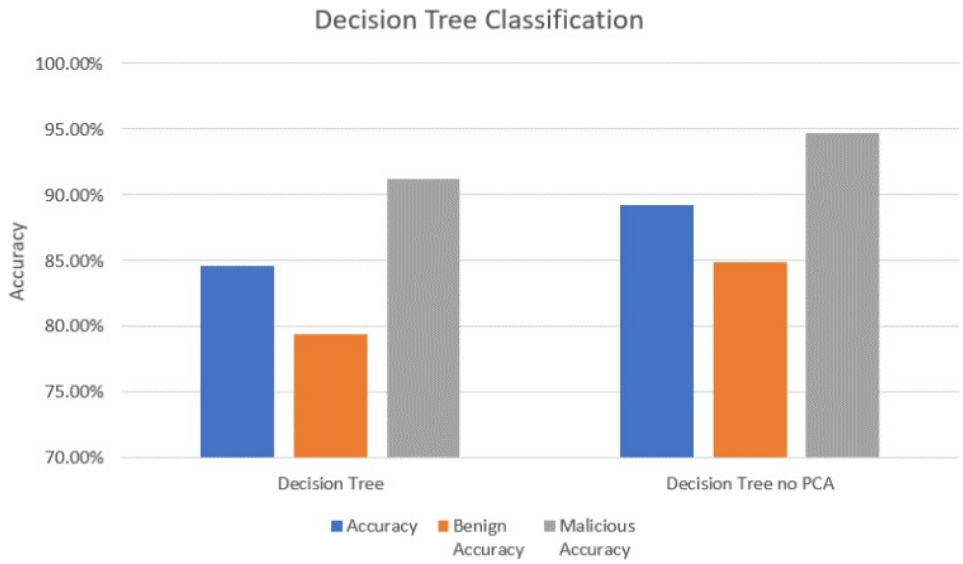

Figure 8. Decision Tree Classification Results.

Table 8. Decision Tree Confusion Matrix (PCA)

Decision Tree		**Predicted**	
		Benign	**Malicious**
Actual	**Benign**	58	15
	Malicious	5	52

Table 8. Decision Tree Confusion Matrix (No PCA)

Decision Tree No PCA		**Predicted**	
		Benign	**Malicious**
Actual	**Benign**	62	11
	Malicious	3	54

5.2 Support Vector Classification Results

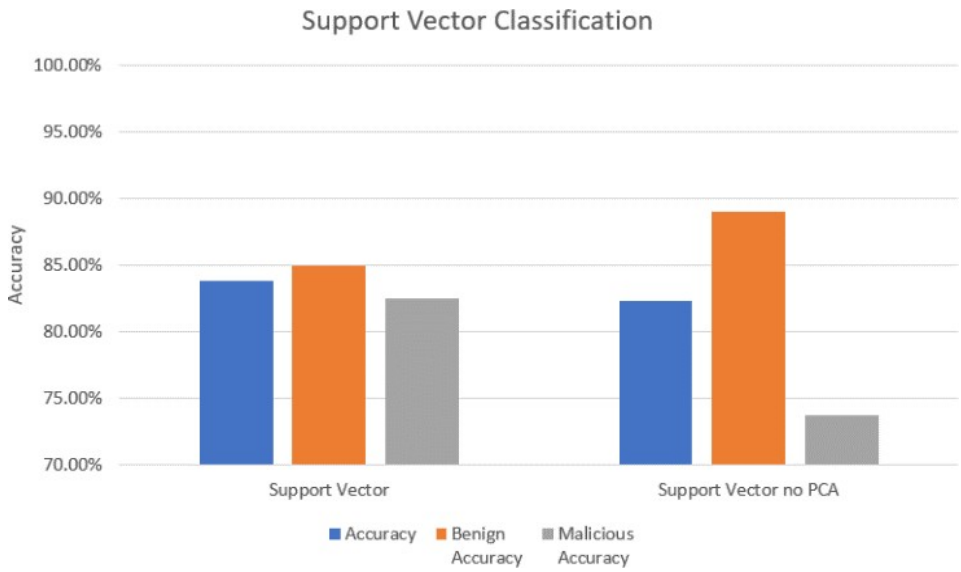

Figure 9. Support Vector Classification Results.

Table 9. Support Vector Confusion Matrix (PCA)

Support Vector		**Predicted**	
		Benign	**Malicious**
Actual	**Benign**	62	11
	Malicious	10	47

Table 8. Decision Tree Confusion Matrix (No PCA)

Support Vector No PCA		**Predicted**	
		Benign	**Malicious**
Actual	**Benign**	65	8
	Malicious	15	47

5. Discussion and Conclusion

The analysis of the results obtained reveal that all three of the ML techniques used in this study have achieved their intended purpose to predict the nature of a website from the provided data. All of the overall accuracies are between 80% and 90% with similar values for each of the classes alone indicating that there is no overfitting of one class. The application of the principle component analysis showed an overall minor reduction of accuracy. However, for the improvement of efficiency, this trade-off is almost certainly worthwhile. The nature of the application incurs a higher cost to the misclassification of malicious websites than it causes to the benign. As a result, the ML technique that would best apply to this context would be one that achieves a higher accuracy on malicious websites than it applies to the benign ones. Furthermore, the results reveal that the Gaussian Process Classifier or the Decision Tree Classifier would fit that role whereas the Support Vector Classifier would not be appropriate for the stated role.

The fact that the Gaussian Process Classifier is technically a regression model with a classification layer on top means that the probability of each prediction can be determined as displayed as in Figure 6. The results represented by Figure 6 reveals the probability that each website is benign with the truly benign websites being in green and the truly malicious applications being in red. Furthermore, the results demonstrate that whilst the classifier incorrectly predicted around 13% of websites, only a few had a probability that was significantly bad. Most of the incorrect predictions were very close to the cut-off value of approximately 0.4, denoting that, with some fine tuning, this model could potentially be brought much closer to 100% accuracy. Considering the correlation matrix in Figure 5, it could be deduced that forward selection might have been a better choice of dimensionality reduction due to the high number of attributes with very limited usefulness, or potentially even a combination of forward selection and PCA. Therefore, this could be investigated as a future work.

In summary, ML algorithms offer many opportunities to detect malicious websites without the need for high risk website content parsing. Instead, as the study has shown, this can be achieved by using data from HTTP headers, WHOIS lookups and DNS records. Of the three classifiers used in this study, the Gaussian Process Classifier is the most appropriate option for the application. This is due to the fact that it is a good balance between effectively managing dimensionally reduced data and achieving a high accuracy on specifically the malicious websites. Another point of consideration for future work could be a further investigation into the possibility of forward feature selection alongside additional other similar ML methods.

References

Dal Pozzolo, A., Caelen, O., Johnson, R. A. and Bontempi, G. (2015). Calibrating probability with under-sampling for unbalanced classification. *IEEE Symposium Series on Computational Intelligence,* pp. 159-166. IEEE.

Pressable. (2019). What are DNS Records? Types and How to Use Them. Available at: https://pressable.com/2019/10/11/what-are-dns-records-types-explained-2/.

Garnaeva, M., Sinitsyn, F., Namestnikov, Y., Makrushin, D. and Liskin, A. (2016). Overall Statistics for 2016: Kaspersky Security Bulletin. *Blue Book*.

Grąbczewski, K. (2014). *Meta-learning in decision tree induction* (Vol. 1). Cham: Springer International Publishing.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York: Springer series in statistics.

Urcuqui, C. (2018). Malicious and Benign Websites: Classify by Application and Network Features (Dataset). *Kaggle*. Available at: https://www.kaggle.com/xwolf12/malicious-and-benign-websites.

Kaggle. (2018). Malicious and Benign Websites Learning. Available at: https://www.kaggle.com/dmrickert3/malicious-and-benign-websites-learning.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning* (Vol. 2, No. 3). Cambridge, MA: MIT press.

Daneshkhah, A., Hosseinian-Far, A., & Chatrabgoun, O. (2017). Sustainable maintenance strategy under uncertainty in the lifetime distribution of deteriorating assets. In Strategic Engineering for Cloud Computing and Big Data Analytics (pp. 29-50). Springer, Cham.

Batsch, F., Daneshkhah, A., Cheah, M., Kanarachos, S., & Baxendale, A. (2019). Performance boundary identification for the evaluation of automated vehicles using Gaussian process classification. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 419-424). IEEE.

InterServer. (2017). WHOIS Lookup Explained. Available at: https://www.interserver.net/tips/kb/whois-lookup-explained/.