A Genetic Programming Approach to Feature Selection and Construction for Ransomware, Phishing and Spam Detection

Harith. Al-Sahaf
School of Engineering and Computer Science
Victoria University of Wellington
P.O. Box 600, Wellington 6140, New Zealand
harith.al-sahaf@ecs.vuw.ac.nz

Ian Welch
School of Engineering and Computer Science
Victoria Unoversity of Wellington
P.O. Box 600, Wellington 6140, New Zealand
ian.welch@ecs.vuw.ac.nz

ABSTRACT

Feature selection and construction can potentially help reduce dimensionality and build more effective features that aim to improve performance. This paper utilises Genetic Programming (GP) to automatically select and construct high-level features for three cybersecurity-related tasks namely Ransomware detection, Spam detection, and Phishing website detection. The effectiveness of the features constructed by the proposed method has been assessed using three commonly-used machine learning algorithms on three datasets and compared against the performance of these machine learning algorithms applied to the original set of features and those features selected and constructed by another GP-based. The experimental results show that the proposed method has significantly improved the performance compared to the other methods.

CCS CONCEPTS

• **Computing methodologies** → **Genetic programming**; *Feature selection*; • **Information systems** → *Wrappers (data mining)*;

KEYWORDS

Genetic programming, feature construction, classification

ACM Reference Format:
Harith. Al-Sahaf and Ian Welch. 2019. A Genetic Programming Approach to Feature Selection and Construction for Ransomware, Phishing and Spam Detection. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322083

1 INTRODUCTION

Rapid development in technology has made data acquisition easier, faster and more accurate. However, such comprehensive collection often produces a very large amount of data in terms of the number of instances and features. Dealing with large volumes of data is very challenging as many machine learning algorithms perform poorly when there are thousands of features. Furthermore, data is often composed of noisy, redundant and irrelevant features, which can significantly reduce performance. Dimensionality reduction aims at reducing the number of features, which can be achieved by performing feature selection and feature construction.

Feature selection and construction methods can broadly be categorised into three groups: *filter*, *wrapper*, and *embedded* [5]. Each of those approaches has its own advantages and disadvantages. Utilising both filter and wrapper approaches has been shown to combine their advantages and achieve good results [5].

Genetic Programming (GP) is an evolutionary algorithm that mimics Darwin evolution theory by natural selection and survival of the fittest. The flexibility and tree-based representation of GP have attracted many researchers to apply this technique to tackle various problems, e.g., feature selection [1], and feature construction [5].

A two-stage approach is adopted in [1] for biomarker discovery using mass spectrometry. The top-ranked features by the entropy and ANOVA test methods are selected in the first stage and GP is then used in the second stage to further reduce the number of features. Similarly, the terminal nodes of an individual evolved by GP are used to improve the classification of breast masses in [3].

Both Neshatian *et al.* [4] and Tran *et al.* [5] proposed class-dependent GP-based feature construction methods to improve the classification performance, where each feature (or set of features) aims to improve classifying the instances of one class.

In [2], multi-tree GP has been utilised to construct multiple features and combine them with some hidden features to improve the performance of the decision trees classification algorithm.

Unlike the aforementioned studies, the proposed method in this paper utilises the conventional single-tree GP with a modified individual representation to automatically select and construct features simultaneously. Moreover, human intervention to determine the number of features to be constructed is not needed and the features can be constructed from any level of the evolved program tree.

2 THE PROPOSED METHOD

The proposed method, *GP Feature selection and constructor* (GPFsc), iterates over the list of instances and the corresponding feature vector based on the selected and constructed features is generated for each instance. The transformed training set is then used to measure the average *balanced* accuracy of a wrapped classifier by adopting the 3-fold cross-validation approach. Hence, the measured accuracy is used as an indicator of the program goodness to tackle the problem at hand, which is defined as,

$$fitness = 1 - \left(\frac{1}{C} \sum_{i=1}^{C} \frac{correct\,(\mathrm{S}_i)}{\mathrm{S}_i} \right), \tag{1}$$

where C is the number of classes, S_i and $correct\,(\mathrm{S}_i)$ is the total number and correctly classified instances of the i^{th} class, respectively.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322083

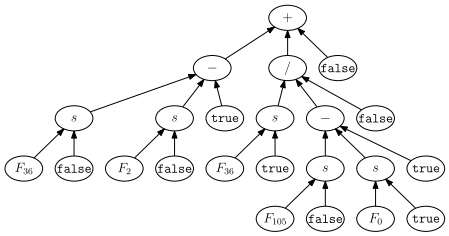

Figure 1: An individual evolved by GPFsc.

Table 1: A summary of the benchmark datasets.

Dataset	Task	Classes	Instances	Features
Spam	Spam detection	2	4601	58
Phishing	Phishing website detection	2	11055	31
Ransomware	Ransomware detection	2	1524	30968

Table 2: The GP parameter settings for GPFsc and MFGP.

Parameter	Value	Parameter	Value
Generations	50	Maximum Tree Depth	10
Population Size	1024	Reproduction	The best individual
Crossover Rate	80%	Selection Type	Tournament
Mutation Rate	20%	Tournament size	7
Minimum Tree Depth	2	Initial Population	Ramped Half-and-half

As depicted in Figure 1, the terminal set comprises two node types that are *flag* and F_i, where i is the index of a feature. The *flag* node type takes either `true` or `false` value. When a *flag* node is `true`, the result of its parent node will be used as a feature. The F_i node type, is a non-negative integer value ranges between 0 and M, where M is the number of features in the original feature vector.

The function set comprises five operators that are s, $+$, $-$, $\times$, and protected / (returns '0' if the dominator is zero). The s node (stands for *select*) has two terminal children F_i and *flag*, and returns the value of the F_i node to its parent if the second child (*flag*) is `true`. Each of the other four operators has the corresponding arithmetic meaning and has three children. The first two children can be any non-terminal nodes whereas the third child is a *flag* node.

3 EXPERIMENT DESIGN AND RESULTS

Three datasets vary in type, e.g., Spam[1], Phishing website[2] and Ransomware[3] detection, number of classes, number of features, and number of instances per class, as summarised in Table 1, are used in this study to assess the performance of the proposed method. 10-fold cross-validation is adopted here and the original ratio of instances for each class is maintained. Furthermore, the decision trees (J48), naïve bayes (NB) and random forest (RF) methods are used to measure the impact of the GPFsc selected and constructed features compared to the use of all the features (original) and that selected and constructed by the Multi Features (MFGP) method [1].

The evolutionary parameters of both GPFsc and MFGP are kept consistent in this study, and are summarised in Table 2.

The experiments for GPFsc and MFGP have been independently repeated 30 times, each of which is 10-fold cross validation, using different seed values and the maximum, average and standard deviation performance over those 30 runs are reported and presented in Table 3. The single-sample t-test is used with a 95% confidence interval and the symbols '↑', '↓', '+' and '−' indicate that GPFsc is,

[1] Available at: https://archive.ics.uci.edu/ml/datasets/spambase
[2] Available at: https://archive.ics.uci.edu/ml/datasets/phishing+websites
[3] Available at: http://rissgroup.org/ransomware-dataset/

Table 3: The experimental results on the three dataset.

		Spam	Phishing	Ransomware
J48	Original	93.0	95.9	96.8
	MFGP ($\bar{x} \pm s$)	86.1 ± 2.5	91.4 ± 2.2	68.4 ± 3.5
	GPFsc ($\bar{x} \pm s$)	94.5 ± 0.5↑ +	96.7 ± 0.3↑ +	94.1 ± 1.7↓ +
	GPFsc (Max)	95.4	97.3	**97.4**
NB	Original	79.3	93.0	85.2
	MFGP ($\bar{x} \pm s$)	79.7 ± 2.0	86.9 ± 2.2	67.7 ± 2.7
	GPFsc ($\bar{x} \pm s$)	88.1 ± 1.5↑ +	91.5 ± 1.0↓ +	88.3 ± 2.2↑ +
	GPFsc (Max)	90.2	94.0	92.1
RF	Original	94.7	97.3	94.8
	MFGP ($\bar{x} \pm s$)	85.8 ± 2.7	91.4 ± 2.2	68.4 ± 3.5
	GPFsc ($\bar{x} \pm s$)	95.1 ± 0.4↑ +	97.8 ± 0.2↑ +	94.6 ± 1.3 +
	GPFsc (Max)	**95.7**	**98.1**	97.3

respectively, significantly better and significantly worse compared to the use of the original features and significantly better and significantly worse compared to the use of MFGP features. The highest accuracy on each dataset is indicated using a **bold** font.

The results on the Spam dataset show that GPFsc features has significantly improved the performance of the three classifiers, where RF has scored the best performance with 95.7% accuracy.

Apart from NB with the original features, GPFsc features have significantly improved the performance of the other classifiers on the Phishing dataset with a 98.1% best accuracy scored by RF.

The results on the most challenging dataset (Ransomware) show that GPFsc features have significantly degraded the performance of J48 and slightly (not significant) that of RF with all features. However, GPFsc features have significantly improved the performance in all the other cases and scored the highest accuracy (97.4%).

4 CONCLUSIONS

This paper proposed a new wrapper feature selection and construction method utilising GP to automatically identify the nodes of an evolved individual that best contribute toward generating the feature vector. Using three datasets, the performance of the proposed method has been assessed and the impact of its automatically selected and constructed features on the performance of three classification algorithms have been investigated. The results show that the proposed method's features have significantly improved the performance of those classification algorithms compared to the original and other GP-based selected and constructed features.

In the future, we would like to study the effectiveness of the proposed method to tackle image classification problems.

REFERENCES

[1] Soha Ahmed, Mengjie Zhang, and Lifeng Peng. 2013. Feature Selection and Classification of High Dimensional Mass Spectrometry Data: A Genetic Programming Approach. In *Proceedings of the European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer, 43–55.

[2] Krzysztof Krawiec. 2002. Genetic Programming-based Construction of Features for Machine Learning and Knowledge Discovery Tasks. *Genetic Programming and Evolvable Machines* 3, 4 (2002), 329–343.

[3] R. J. Nandi, A. K. Nandi, R. Rangayyan, and D. Scutt. 2006. Genetic Programming and Feature Selection for Classification of Breast Masses in Mammograms. In *Proceedings of the 2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 3021–3024.

[4] Kourosh Neshatian, Mengjie Zhang, and Mark Johnston. 2007. Feature Construction and Dimension Reduction Using Genetic Programming. In *Proceedings of the Australasian Joint Conference on Artificial Intelligence*. Springer, 160–170.

[5] Binh Tran, Mengjie Zhang, and Bing Xue. 2017. Class Dependent Multiple Feature Construction Using Genetic Programming for High-Dimensional Data. In *Proceedings of the Australasian Joint Conference on Artificial Intelligence*. Springer, 182–194.