A Support Vector Machine Water Wave Optimization Algorithm Based Prediction Model for Metamorphic Malware Detection

Mohd Mursleen, Ankur Singh Bist, Jaydeep Kishore

Abstract*: In this paper, we proposed a novel method based on coupling of SVM (Support Vector Machine) and WWO (Water Wave Optimization) for detection of metamorphic malware. The working of SVM model depends upon the proper selection of SVM parameters. Malware signatures have been taken from G2, MWOR, MPCGEN and NGVCK (Next Generation Virus Creation Kit).Benign signatures have been taken from Gygwin, GCC, TASM, MingW and Clang .ClustalW and T-Coffee are used for signature alignment during primary pairwise alignment and secondary multiple alignment in order to avoid the problem of variable length of code. In this study WWO has been employed for determining the parameters of SVM. The performance of SVM-WWO method has been compared with LAD Tree, Naive Bayes, SVM and ANN(Artificial Neural Network). Furthermore, The results obtained show that the newly proposed approach provides significant accuracy. Satisfactory experimental results show the efficiency of proposed method for metamorphic malware detection. Further, it has been recommended that this method can be used to facilitate commercial antivirus engines.

Index Terms*: metamorphic malware detection, support vector machine (SVM), water wave optimization (WWO).

I. INTRODUCTION

Malicious software, in any manner, reflects a threat for every user. Antivirus designers face the problem of detecting an ever-growing malwares with large number of polymorphic and metamorphic variants [1, 2, 3]. Malware is a program which is designed with malicious intent. Malware can be categorized into virus, worm, Trojan horse, backdoor, spyware, adware, botnet etc as shown in Figure1. Computer virus is a program that has the property of self replication and the main target to hide its identity. Due to various advancements in the field of antivirus design, malware designers have also modified complexity of virus code. Computer viruses can be classified into boot sector virus, encrypted virus, oligomorphic virus etc. Our main purpose is to detect metamorphic malware. Metamorphism is a process where the computer program mutates in such a manner that structural appearance gets modifies but functionality remains same. To prevent detection malware designers use the concept of metamorphism. The methods used to design metamorphic malware are subroutine permutation, dead code insertion, register swap, equivalent code substitution, random jump insertion and code transposition [4].

Signature detection has been widely used by antivirus engine for malware detection. The signature is a sequence of string expected to present in a particular malware. Malware designers adopted various methods in order to evade signature based techniques. In the evolution path of malware, metamorphic malware emerged after polymorphic viruses.

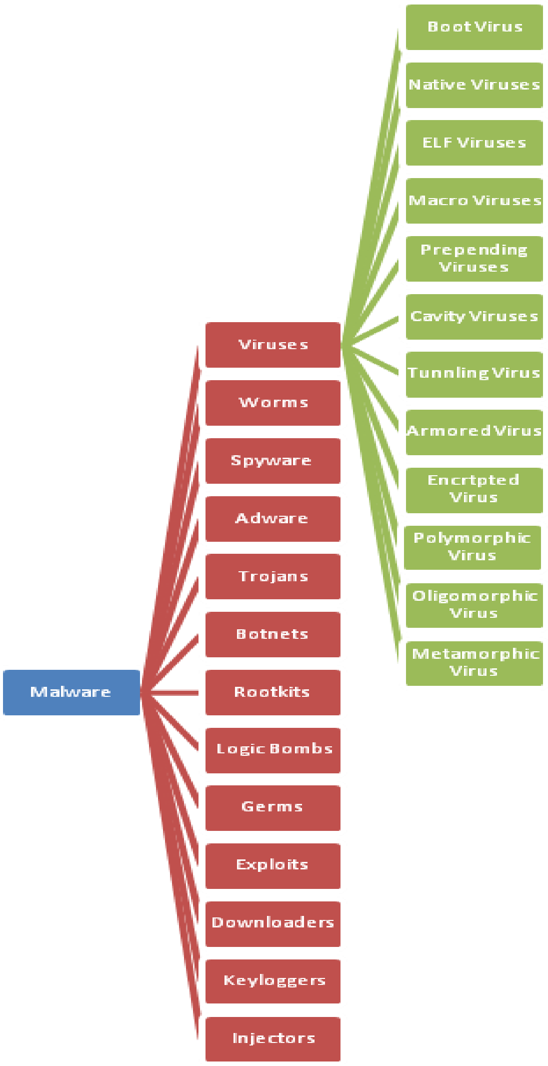


Figure1. Block diagram of malware classification

Revised Manuscript Received on 30 January 2019.
***** Correspondence Author
Mohd Mursleen*, Nanhi Pari Seemant Engineering Institute Pithoragarh, India.
Ankur Singh Bist, KIET Group of Institutions, Ghaziabad,India
Jaydeep Kishore, Nanhi Pari Seemant Engineering Institute Pithoragarh, India

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

Many techniques have been designed in order to detect metamorphic malware. It can be categorized as static analysis, dynamic analysis and hybrid analysis. Static technique involves the detection of malware without executing the tested file. Static technique is widely used and safe also but sometimes malware sample evade from static techniques then dynamic analysis is used. Dynamic analysis involves the behaviour analysis of tested file after its execution. In some specific cases like logic bombs dynamic detection does not work efficiently. Hybrid technique takes the advantage from previous two methods and has shown good results for the classification of malwares. Existing methods have several challenges. They are not very efficient for zero day malwares [5]. Secondly they require significant time and require timely definition updates. Machine learning is widely used for classification tasks, past literature shows the use of machine learning techniques for malware detection also. Data to be classified may be large number of features and efficient model is the demand of time so role of optimization algorithms cannot be denied. Our proposed method is inspired from these two facts.

The goal of this article is twofold: first we used different alignment techniques in order to pre-process the malware signature extracted from data files taken for experiment. Different substitution matrices are used for making in depth analysis. Secondly, we proposed the unique combination of SVM-WWO and compared the same with popular techniques. The experiment involved malware samples from G2, MWOR, MPCGEN and NGVCK and benign files taken from Gygwin, GCC, TASM, MingW and Clang.

The contributions of this study includes: 1. Development of an efficient SVM-WWO technique for detecting metamorphic viruses. 2. Performance evaluation of different learning models. 3. Investigating the role of different alignment techniques and substitution matrices with different learning model. 4. Investigating the impact of proposed learning model for metamorphic malware detection on dataset generated from various kits. The organization of paper is as follows: Section 2 gives a complete description on the techniques used by various researchers to detect metamorphic viruses. Section 3 discusses on the proposed technique used to detect metamorphic malware. Section 4 discusses on the performance analysis of the proposed algorithm with comparative analysis. Conclusion and future work are given in the final section. The section below discusses on the type of metamorphic malware detection techniques.

II. Survey

Various techniques have been proposed to detect metamorphic malware. Reference [6] experimentally proved that OPCODE (Operational Code) can be utilized to detect malware samples. Analysis was carried out on different malware samples. After getting disassembled, opcode frequency of malware samples was compared with benign files. It was found that frequency distribution of opcode in malware sample is quite different from benign files. Results approved the same patterns as classifying feature for malware classification problem. The techniques in reference [7] use geometric detection but biased to false positive. In past few years various heuristic techniques came into existence for detecting computer viruses [8, 9]. Various authors [10, 11] proposed metamorphic malware detection based on HMM (Hidden Markov Models). Srinivasan [12] proposed technique based on new scoring named as SSCT (Simple Substitution and Column Transposition). The authors [13] were able to classify metamorphic malware based on function call score graph. Reference[14] propose metamorphic malware detection technique based on SVD (Singular Value Decomposition) approach. Reference [15] utilized the concepts of clustering techniques with EM (Expectation Maximization) for metamorphic malware detection and found satisfactory results. Reference [16] proposed a technique for metamorphic java script malware and compared the impact of different static techniques. Reference [17] introduced function call graph scoring technique for identification of metamorphic malware. Reference [18] proposed opcode based similarity score calculation motivated by simple substitution distance. Structural entropy [19] has been used for collecting data variation. Another important steps of this research included file segmentation and sequence comparison. Authors obtained good results for critical cases. Reference [20] introduced a method motivated from face recognition. The concept of eigen vector scoring achieved interesting results for highly metamorphic malware. Reference [21] proposed a graph technique and obtained comparable results with HMM. Authors in [22] created malware data sample by built in buffer overflow and finally showed that proposed technique perform well for such types of attacks. Reference [23] proposed a technique based on syntax and semantic analysis of files to be classified. Reference [24] proposed a technique named VILO based on NN (Nearest Neighbour) and TFIDF (Term Frequency and Inverse Document Frequency) with nperm used for similarity analysis. Experimental results demonstrated the strength of proposed technique used for malware detection. Reference [25] introduced a framework named as MARD for the analysis of metamorphic malware. Results found to be improved when compared with ACFG (Automated Control Flow Graph) and SWOD-CFWeight (Sliding Window of Difference and Control Flow Weight). Reference [26] introduced API call graph based on disassembled pattern taken from program but proposed technique was not suitable for real time detection. Reference [27] used KNN (K-Nearest Neighbour, Naive Bayes, J48 Decision Tree, SVM, MLP (Multilayer Perceptron) for malware detection. Reference [28] used machine learning approach with hybrid feature selection for detecting suspicious files. Reference [29] proposed a metamorphic malware detection technique based on similarity analysis.

Weighted opcode graph was used to trace out similarity score generation. Reference [30] proposed a technique based on chi-squared test to identify metamorphic malware. Firstly authors used different compilers for experimental study. Same methodology further utilized to trace instructions produced by metamorphic malware generators.

The authors [31] found dynamic birthmarks in malware sample and then utilized concept of HMM and PHMM (Profile Hidden Markov Models) for metamorphic malware detection.

III. DESCRIPTION OF THE PROPOSED APPROACH

If WWO is meta-heuristic technique inspired from natural computing of water waves [32]. The phenomena's related to water waves like propagation, refraction, and breaking can be formulated into mathematical model in order to retrieve solutions in high-dimensional space of solutions.WWO is used for tuning SVM parameters. The basic theme of WWO is based on shallow water wave models. Crucial steps of WWO include propagation, refraction and breaking. Following equation depicts the workflow of WWO.

$$x'(k) = x(k) + rand(-1,1).\delta\, L(k) \qquad (1)$$

Where x is the original wave and adjustment of k is originates new wave x' . L(k) is the length of k^{th} dimension (1≤k≤n).

$$\delta = \delta\alpha^{-(f(x)-fmin+c)/(fmax-fmin+c)} \qquad (2)$$

Where δ shows updated wavelength, α denotes wavelength reduction coefficient, *fmin* and *fmax* denotes are minimum and maximum values of fitness.

$$x'(k) = N((x^*(k)+x(k))/2_, (x^*(k)+x(k))/2) \qquad (3)$$

Where μ is mean, σ denotes standard deviation, N(μ, σ) is Gaussian random number and x^* is the best solution.

$$\delta'_=\delta(f(x)/f(x')) \qquad (4)$$

Where δ' is the wavelength after refraction.

$$x'(k) = x(k) + N(0,1).\beta\, L(k) \qquad (5)$$

Where $x'(k)$ is the solitary wave at k dimension. If no value of $x'(k)$ is better than x^* then x^* is fittest otherwise get replaced by fittest one.

The steps of optimization are thus explained as follows.

Step0: Wavelength reduction coefficient (α)=1.02, Breaking coefficient (β)=0.002 and Maximum wave height (h_{max})=4 are used as initial values and maximum iteration is taken as stopping condition.

Step1: Initially random values are taken for population P_0 of z waves where z denotes solutions.

```
Step2: Until termination of base condition do
Step3:   for $x \in P_0$ do
Step4:     Update x to $x'$ depending upon equation (1)
Step5:       if f($x'$)>f(x) then
Step6:         if f($x'$)>f($x^*$) then
Step7:           Break $x'$ depending upon equation (5)
Step8:         Change $x^*$ with $x'$
Step9:       Exchange x with $x'$
Step10:    else
             Decrease x.h -1
Step11:      if x.h=0 then
Step12:        Refract x to $x'$ depending upon equation (3) & (4)
Step13:  Wavelength updation based on equation (2)
Step14: return $x^*$
```

In machine learning support vector machines are very popular. SVM have been used in various applications, including prediction of river water pollution[42], prediction of electrical and thermal performance in PV/T system[43], crack detection[44] and liver fibrosis diagnosis[45]. The main concern is to study the effectiveness of svm for malware detection. Lot of articles have been written for malware classification using svm and its variants[46, 47,48,49,50]. Let $\{(x_1, y_1),(x_2\ \ y_2$),………......., (x_p, y_p) } denotes datasets for training, where x_iC R_n and y_iC R_n (i=1....l). To handle the regression following equation is solved.

$$\text{minimize } \frac{1}{2}||w||^2 + C^*\Sigma_{i=0}^{p}(£ + £'')$$

where y_i-<w, x_i>-b<=$£_i$ +$£$,   <w, x_i>+b-y_i<=$£_i$ +$£^*$ and $£$, $£_i$* >=0 must be satisfied.

Generic equation can be formulated with the help of following equation.

$$f(x)=\Sigma_{i=1}^{l}(\alpha_i-\alpha_{i^*})\ (x_i, x)+b$$

This method can be modified for non-linear regression and can be rewritten as with the help of kernel function K (x_i, x_j).

$$f(x)=\Sigma_{i=1}^{l}(\alpha_i-\alpha_{i^*})\ K(x_i, x_j)+b$$

RBF (radial basis functions) is used in our experiment and WWO is used to tune γ, ε, C.

Neural network is a mathematical computational model inspired from working style of brain. Brain in general learns from experience, similarly the underlying mathematical formulations of neural networks contains the capability of getting trained based of certain input sets. Learning techniques can be classified as supervised, unsupervised and reinforced learning. After the training phase, trained neural networks attain the capability to predict. Neural networks are widely used for classification tasks. In our experiment we use MatlabR2013a for designing NN predictive model for the identification of metamorphic malware.

Bayes' theorem is the basic inspiration for naive bayes classifier. In our experiment we use MatlabR2013a for designing probabilistic model based on bayes' theorem i.e. naive bayes classifier.

LAD algorithm is a technique that utilizes LAD (least absolute deviation) step to design regression trees. It can manage discrete variables with the help of recursive partitioning. We used weka tool for applying lad tree algorithm. Thus various machine learning algorithms are used with our newly proposed approach i.e. svm-wwo. In the field of computer virology dataset is very crucial part for research. At the same time standard, updated and real time complex datasets are not easily available. Next section describes the datasets used for our experiment.

A. Dataset -

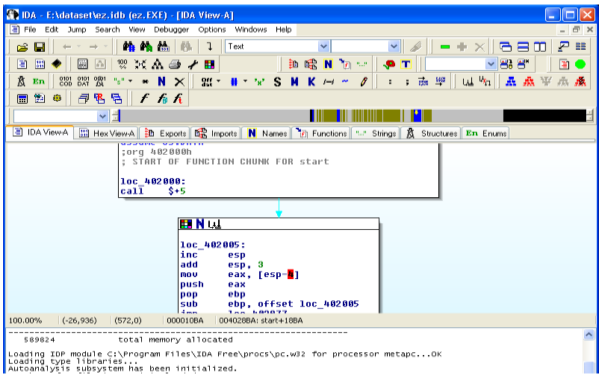
For measuring the effectiveness of proposed approach following metamorphic malware and benign files are used. Well established malware generation kits and benign files are taken for analysis in our experiment.

1. MWOR[33]
2. NGVCK[34]
3. MPCGEN[35]
4. G2 virus generation kit [36]
5. Clang[37]
6. Cygwin[38]
7. GCC[39]
8. MingW[40]
9. TASM[41]

Lot of research articles have been written using same dataset [7, 11, 43]. Datasets used for analysis contains clang, cygwin, gcc, mingw and tasm as normal files. Malware data sample contains mwor, ngvck, mpcgen, and g2. These challenging datasets are taken for testing the effectiveness of svm-wwo algorithm. From these malware and benign data samples dataset1, dataset2, dataset3 and dataset4 are prepared. Dataset1 contains malicious samples from mwor and benign files are taken in equal proportion from clang, cygwin, gcc, mingw and tasm. Similarly dataset2, dataset3 and dataset4 contains ngvck, mpcgen and g2 as malware samples and normal files are taken in equal proportion from clang, cygwin, gcc, mingw and tasm. The purpose of this hybrid combination is to analyse the variation of accuracy under different data environment.

IV. EXPERIMENTAL RESULTS

This section provides details of our experimental results. Initially we disassemble data files taken for analysis. The purpose of this step is to extract opcode sequences. Past literature shows that opcode sequence can be effectively used as classifying feature. We have direct assembly codes generated from various kits but in order to analyze real time malware detection scenario, firstly assembly files are converted into executables. After this step disassembling process is performed. Actually exact and accurate disassembling is another aspect that requires further exploration due to various issues. Our main concern is to work with executables in order to attain practical accuracy values. Figure2 shows the disassembled code produced with the help of IDA.



IDA - E:\dataset\ez.idb (ez.EXE) - [IDA View-A]

File Edit Jump Search View Debugger Options Windows Help
Text
IDA View-A Hex View-A Exports Imports Names Functions Strings Structures Enums

```
;org 402000h
; START OF FUNCTION CHUNK FOR start

loc_402000:
call    $+5
```

```
loc_402005:
inc     esp
add     esp, 3
mov     eax, [esp-4]
push    eax
pop     ebp
sub     ebp, offset loc_402005
```

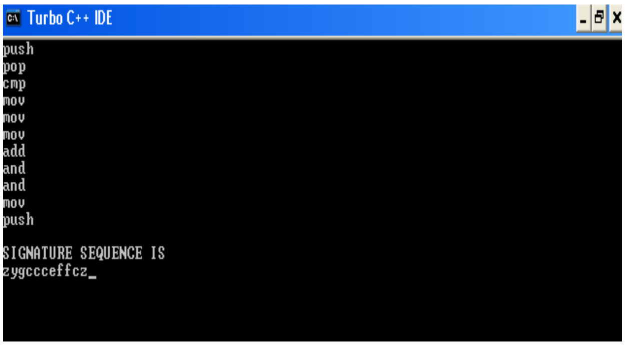
100.00% (-26,936) (572,0) 000010BA 004028BA: start+18BA

```
-----------------------------------------------------------------
   589824              total memory allocated

Loading IDP module C:\Program Files\IDA Free\procs\pc.w32 for processor metapc...OK
Loading type libraries...
Autoanalysis subsystem has been initialized.
```

Figure 2: Diassembled code

The extracted opcode sequence was mapped with english alphabets. Each opcode is replaced with unique letter. It gives rise to unique signature corresponding to each file to be tested. Figure 3 shows the signature corresponding to a file. Similar unique signatures are generated for all samples belongs to dataset1, dataset2, dataset3 and dataset4. Variable length of signatures that comes out from various files is another point of concern. In order to solve the same problem next step i.e. alignment is performed. Turbo C++ is used to implement algorithm for scanning files and to prepare signatures. It has been shown in past literature that there are more than 130 opcodes in any specific processor. If a matrix is designed on the basis of all existing opcodes then comparison time will be too large. In order to avoid this situation only a subset of opcodes is taken for analysis. There are two important factors associated with antimalware software i.e. accuracy and fast scanning. These two factors are diametrically opposite in nature. It makes the challenge difficult for antimalware designers. We are following the heuristics of past literature in terms of opcode subset selection.



Turbo C++ IDE

```
push
pop
cmp
mov
mov
mov
add
and
and
mov
push

SIGNATURE SEQUENCE IS
zygccceffcz_
```

Figure 3: Extracted opcodes and corresponding signature

Sequence alignment technique is broadly used for managing various sequences like amino acids, RNA or DNA to compute similarity index. It may be due to following course of action between the sequences.

- Functional
- Structural
- Evolutionary

Aligned sequences of DNA, amino acid or nucleotide residues are characteristically represented as rows within a matrix.

Gaps are introduced between the residues so that identical or similar letters are aligned in consecutive columns. In the first phase of alignment experiment Clustalw is used with identity or unitary substitution matrix (CLI) for initial pairwise and secondary multiple alignment. In second phase of alignment experiment clustalw is used with Gonnet substitution matrix (CLG). In third phase of alignment experiment clustalw was used with BLOSUM substitution matrix (CLB). In the fourth phase of experiment T-coffee was used with BLOSUM as substitution matrix (TCB). Table1, Table2, Table3 and Table4 depicts accuracy values for dataset1, dataset2, daraset3 and dataset4 respectively using svm-wwo, svm, nn, naive bayes and lad tree.

Table1, Table2, Table3 and Table4 shows the accuracy values generated from svm-wwo, svm, nn, naive bayes and lad tree under various alignment schemes for dataset1, dataset2, dataset3 and dataset4. Table 1 shows that svm-wwo attains highest accuracy under CLG alignment for dataset1. The performance of svm is best under CLG alignment for dataset1. NN attains highest accuracy under CLG alignment for dataset1 and lad tree attains highest accuracy under CLG alignment for datset1. Table 2 shows that svm-wwo attains highest accuracy under CLG alignment for dataset2. The performance of svm is best under CLG alignment for dataset2. NN attains highest accuracy under CLG alignment for dataset2 and lad tree attains highest accuracy under CLG alignment for datset2. Table 3 shows that svm-wwo attains 99.0% accracy under CLG and CLB alignment for dataset3. The performance of svm is best under CLB alignment for dataset3. NN attains highest accuracy under CLG and CLB alignment for dataset3 and lad tree attains highest accuracy under CLB alignment for datset3. Table 4 shows that svm-wwo attains highest accuracy under CLG alignment for dataset4. The performance of svm is best under CLG alignment for dataset4. NN attains highest accuracy under CLB alignment for dataset4 and lad tree attains highest accuracy under CLG alignment for datset4. Observation on various datasets using five predictive models reflects the effectiveness of machine learning techniques for metamorphic malware detection. In all experiments CLG and CLB alignment combination with svm-wwo, svm, nn, naive bayes and lad tree outperforms as compared to CLI and TCB. The comparison of CLG and CLB for various test cases shows that CLG is performing well in most of the cases.

Table 1. Accuracy values (round-off) for dataset 1 using svm-wwo, svm, nn, naive bayes and lad tree

Dataset1	Svm-wwo	Svm	NN	Naive Bayes	Lad Tree
CLI	73.0	71.0	63.0	74.0	70.0
CLG	97.0	96.0	96.0	93.0	92.0
CLB	79.0	73.0	78.0	72.0	81.0
TCB	86.0	84.0	85.0	90.0	86.0
Average	83.75	81.0	80.5	82.25	82.25

Table 2. Accuracy values (round-off) for dataset 2 using svm-wwo, svm, nn, naive bayes and lad tree

Dataset2	Svm-wwo	Svm	NN	Naive Bayes	Lad Tree
CLI	72.0	70.0	61.0	72.0	69.0
CLG	97.0	96.0	92.0	89.0	91.0
CLB	80.0	72.0	76.0	71.0	80.0
TCB	87.0	79.0	81.0	88.0	88.0
Average	84.0	79.25	77.5	80.0	82.0

Table 3. Accuracy values (round-off) for dataset 3 using svm-wwo, svm, nn, naive bayes and lad tree

Dataset3	Svm-wwo	Svm	NN	Naive Bayes	Lad Tree
CLI	88.0	86.0	79.0	73.0	80.0
CLG	99.0	98.0	98.0	99.0	93.0
CLB	99.0	99.0	98.0	90.0	90.0
TCB	89.0	86.0	90.0	90.0	89.0
Average	93.75	92.25	91.25	88.0	88.0

Table 4. Accuracy values (round-off) for dataset 4 using svm-wwo, svm, nn, naive bayes and lad tree

Dataset4	Svm-wwo	Svm	NN	Naive Bayes	Lad Tree
CLI	87.0	85.0	80.0	74.0	79.0
CLG	99.0	98.0	96.0	97.0	95.0
CLB	98.0	96.0	99.0	91.0	91.0
TCB	90.0	87.0	89.0	92.0	90.0
Average	93.5	91.5	91.0	88.5	88.75

Table 5. Training and validation proportion used in experiment (for all data samples)

Prediction techniques	% of data used for training	% of data used for training	No. of training sets	No. of testing sets
SVM-WWO	78.0	22.0	54.0	16.0
SVM	78.0	22.0	54.0	16.0
NN	78.0	22.0	54.0	16.0
Naive Bayes	78.0	22.0	54.0	16.0
LAD Tree	78.0	22.0	54.0	16.0

Table5 shows the training and validation proportion used in experiment for dataset1, dataset2, dataset3 and dataset4. Same proportion of training and validation is used for all datasets in order to maintain the uniformity. The main purpose is to estimate accuracy values on same scale. Artificial neural network have been used for analysis in this work using Matlab2013a. Table 6 demonstrates the parameter used for ANN. These parameters are selected based on past literature of malware detection using neural network.

Table 6. ANN parameters (for all data samples)

Parameters	Dataset1, Dataset2, Dataset3 & Dataset4
Learning rule	Levenberg-Marquardt
epochs	412
Activation function	Sigmoid
Neurons in hidden layer	15

SVM technique has been taken for analyzing different malware dataset. The parameters C, γ, ε of SVM is taken 2.2, 0.4, 1.1 respectively for four different datasets taken for study. Matlab R2013 is used to develop SVM code.

WWO is used to calculate the optimal value of parameters γ, ε, C for SVM-WWO algorithm. We have taken four different datasets in order to strengthen our observations. SVM parameters are calculated for all different datasets. Table 7 gives hyper parameter values obtained from experiment

Table 7. SVM parameters calculation by WWO.

Datasets taken for analysis	C	ε	γ
Dataset1	1.92	0.310	0.710
Dataset2	1.36	0.151	0.631
Dataset3	1.87	0.392	0.780
Dataset4	1.48	0.291	0.744

Accuracy is calculated in expressions of the ratio of data samples that are classified acceptably or not. We have the following definitions.

TP (True Positive)—The number of malware samples correctly classified as malicious code.

TN (TrueNegative)—The number of benign samples correctly classified as normal file sample.

FP (False Positive)—The number of normal file samples incorrectly classified as malware.

FN(False Negative)—The number of malicious code samples incorrectly classified as benign.

Then accuracy is calculated as

Accuracy = (TP+TN)/ (TP+FN+ TN+FP) (6)

Figure4 shows the relationship between average accuracy for four different datasets obtained from svm-wwo, svm, nn, naive bayes and lad tree. Dataset1, dataset2, dataset3 and dataset4 are taken with popular classification techniques to justify the strength of proposed technique. Lad tree performs best for dataset1. The performance of lad tree is comparable for dataset3 and dataset4 and obtains minimum accuracy score for dataset2. Naive bayes performs best for dataset1. The performance of naive bayes is comparable for dataset3 and dataset4 and obtains minimum accuracy score for dataset2.

Neural network performs best for dataset2. The performance of neural network is comparable for dataset3 and dataset4 and obtains minimum accuracy score for dataset1. Support vector machine performs best for dataset2. The performance of support vector machine is comparable for dataset3 and dataset4 and obtains minimum accuracy score for dataset1. Support vector machine water wave optimization algorithm performs best for dataset2. The performance of neural network is comparable for dataset3 and dataset4 and obtains minimum accuracy score for dataset1. Finally experimental results show that svm-wwo performs better than other conventional machine learning techniques used for classification of metamorphic malware.

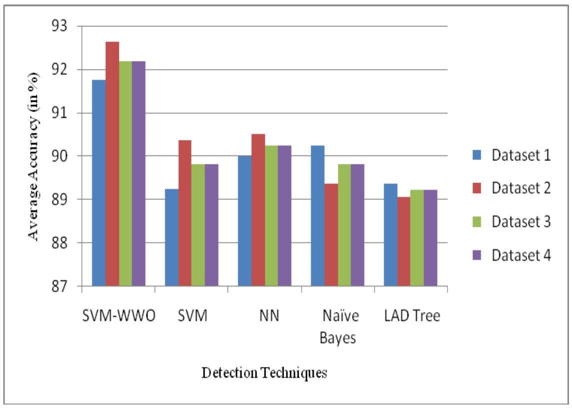


Figure 4: Accuracy analysis proposed methods on dataset1, dataset2, dataset3 and dataset4

V. CONCLUSION

In this paper, we have proposed a novel approach based on the combination of WWO and SVM for the identification of metamorphic malware. WWO is used to tune the hyper parameters of svm. The unique mixture of WWO and SVM has reflected better accuracy in predicting metamorphic malware for dataset1, dataset2, dataset3 and dataset4 as compared to SVM, NN, Naive Bayes and LAD Tree. Further it is recommended to explore the scope and utility of SVM-WWO for other complex malware datasets. The experimental results show that the proposed method is robust. For future developments, we are planning to increase the classification accuracy by using hybrid/ensemble learning methods. It would be interesting to explore the utility of SVM-WWO for different category of malwares like botnet, Trojan horses, spyware etc. Not only problems related to malware can be solved with the help of proposed approach but variety of problems related to classification can be solved with proposed approach. As we have seen in experiments tuning of parameters with the help of optimization algorithm plays vital role in terms of accuracy. In this paper water wave optimization algorithm is used for tuning svm parameters but the use of other optimization algorithms for malware classification or for other classification algorithms will give new direction to concerned field. Role of effective techniques in the field of optimization theory and intelligent computing have produced ground breaking results in terms of accuracy and computation time. Same approach will be helpful in the domain of antivirus capability enhancement.

REFERENCES

1. U. Bayer, A. Moser, C. Kruegel and E. Kirda, " Dynamic Analysis of Malicious Codes," *Journal of Computer Virology and Hacking Techniques*, vol. 2, no.1, pp. 67-77, 2006.
2. F. Cohen, "Computer Viruses: Theory and Experiments." *Computers & Security*," vol. 6, no. 1, pp. 22-35, 1985.
3. K. Fu, and J. Blum, "Controlling for Cybersecurity Risks of Medical Device Software," Communications of the ACM," vol. 56, no. 10, pp.35-37, 2013.
4. Z. Zhou, Q. Zhu and M. Zhou," On The Time Complexity of Computer Viruses," *IEEE Transaction of Information Theory*," vol. 51, no. 8, pp. 2962-2966, 2005.

5. L. Bilge and T. Dumitras, " Before We Knew it: An Empirical Study of Zero-Day Attacks in The Real World,"In: Proceedings of The ACM Conference on Computer and Communication Security," pp. 833-844, 2012.
6. B. Bilar," Opcodes are Predictor for Malware," *IJESDF*, vol. 1, no. 2, pp. 156-168, 2007.
7. P. Szor," The Art of Computer Virus Research and Defense,"*Pearson Education*, 2005.
8. A. S. Bist, "Detection of Metamorphic Viruses: A Survey,"In *Advances in Computing, Communications and Informatics (ICACCI, International Conference on*, pp. 1559-1565, 2014.
9. V. P. Nair, H. Jain, Y. K. Golecha, M. S.Gaur, & V. Laxmi, " MEDUSA: MEtamorphic Malware Dynamic Analysis Using Signature from API," In *Proceedings of the 3rd International Conference on Security of Information and Networks*, pp. 263-269.
10. C. Annachhatre, T. H. Austin, and M. Stamp, "Hidden Markov models for malware classification," *Journal of Computer Virology and Hacking Techniques*, vol. 11, no. 2, pp.59-73, 2015.
11. W. Wong, and M. Stamp, "Hunting for metamorphic engines," *Journal in Computer Virology*, vol. 2, no. 3 , pp. 211-229, 2006.
12. S. Srinivasan, "SSCT Score for Malware Detection," *SJSU Master Thesis*, pp. 10-40, 2015.
13. D. Rajeswaran, "Function Call Graph Score for Malware Detection." *SJSU Master Thesis*, pp. 11-47, 2015.
14. R. K. Jidigam, T. H. Austin & M. Stamp, "Singular value decomposition and metamorphic detection," *Journal of Computer Virology and Hacking Techniques*, vol. *11*, no. 4, pp. 203-216, 2015.
15. U. Narra, F. D. Troia, V. A. Corrado, T.H. Austin, and M. Stamp, "Clustering versus SVM for malware detection," *Journal of Computer Virology and Hacking Techniques*, pp.1-12. 2015.
16. M. Mangesh, T. H. Austin, and M. Stamp. "Hunting for Metamorphic JavaScript malware," *Journal of Computer Virology and Hacking Techniques* , vol. 11, no. 2, pp. 89-102, 2015.
17. D. Sayali, Y. Park, and M. Stamp. "Eigenvalue analysis for metamorphic detection," *Journal of computer virology and hacking techniques*, vol. 10, no. 1 , pp. 53-65, 2014.
18. S. Gayathri, R. M. Low, and M. Stamp, "Simple substitution distance and metamorphic detection," *Journal of Computer Virology and Hacking Techniques*, vol. 9, no. 3 , pp. 159-170, 2013.
19. B. Donabelle, R. M. Low, and M. Stamp. "Structural entropy and metamorphic malware," *Journal of computer virology and hacking techniques*, vol. 9, no. 4 , pp. 179-192, 2013.
20. D. Sayali, Y. Park, and M. Stamp. "Eigenvalue analysis for metamorphic detection," *Journal of computer virology and hacking techniques* 10, no. 1, pp. 53-65, 2014.
21. R. Neha, R. M. Low, and M. Stamp. "Opcode graph similarity and metamorphic detection," *Journal in Computer Virology* , vol. 8, no. 1-2 , pp. 37-52, 2012.
22. S. Ronak. "METAMORPHIC VIRUSES BUFFER OVER." *PhD diss.*, San Jose State University, 2010.
23. D. P., Mila, R. Giacobazzi, A. Lakhotia, and I. Mastroeni, "Abstract symbolic automata: mixed syntactic/semantic similarity analysis of executables," In *ACM SIGPLAN Notices*, vol. 50, no. 1, pp. 329-341, 2015.
24. A. Lakhotia, A. Walenstein, C. Miles, and A. Singh, "VILO: a rapid learning nearest-neighbor classifier for malware triage," *Journal of Computer Virology and Hacking Techniques* , vol. 9, no. 3, pp.109-123, 2013.
25. A. Shahid, R. Nigel Horspool, and I. Traore, "MAIL: Malware Analysis Intermediate Language: a step towards automating and optimizing malware detection," In *Proceedings of the 6th International Conference on Security of Information and Networks*, pp. 233-240, 2013.
26. F. Parvez, V. Laxmi, M. S. Gaur, and P. Vinod, "Mining control flow graph as API call-grams to detect portable executable malware," In *Proceedings of the Fifth International Conference on Security of Information and Networks*, pp. 130-137, 2012.
27. F. Ivan, A. Erwin, and A. S. Nugroho. "Analysis of machine learning techniques used in behavior-based malware detection," In *Advances in Computing, Control and Telecommunication Technologies (ACT), Second International Conference on*, pp. 201-203, 2010.
28. P. V. Shijo, and A. Salim. "Integrated static and dynamic analysis for malware detection, " *Procedia Computer Science* , vol. 46. pp. 804-811, 2015.
29. R. Neha, R. M. Low, and M. Stamp. "Opcode graph similarity and metamorphic detection," *Journal in Computer Virology, vol.* 8, no. 1, pp.37-52, 2012.
30. T. H. Annie, and M. Stamp. "Chi-squared distance and metamorphic virus detection," *Journal of Computer Virology and Hacking Techniques* , vol. 9, no. 1 ,pp. 1-14, 2013.
31. R. Hardikkumar, and M. Stamp "Hunting for Pirated Software Using Metamorphic Analysis." *Information Security Journal: A Global Perspective, vol.* 23, no. 3 ,pp. 68-85, 204.
32. Y. J. Zheng, "Water wave optimization: a new nature-inspired metaheuristic. *Computers & Operations Research", 55*, 1-11,2015.
33. S. M. Sridhara, & M. Stamp, (2013). "Metamorphic worm that carries its own morphing engine". *Journal of Computer Virology and Hacking Techniques*, *9*(2), 49-58.
34. NGVCK. VX Heavens, Retrieved from: *http://vxheaven.org/vx.php?id=tn02*
35. MPCGEN. VX Heavens, Retrieved from: *http://vxheaven.org/vx.php?id=tn02*
36. G2. VX Heavens. Retrieved from: http://download.adamas.ai/dlbase/Stuff/VX%20Heavens%20Library/static/vdat/creatrs1.htm
37. Clang. Retrieved from http://clang.llvm.org/.
38. Cygwin. Retrieved from: http://www.cygwin.com/
39. GCC. Retrieved from http://gcc.gnu.org/.
40. MinGW. Taken from: http://www.mingw.org/.
41. TASM. Retrieved from:
42. trimtab.ca/2010/tech/tasm-5-intel-8086-turbo-assemblerdownload
43. O. Kisi , & K. S. Parmar, "Application of least square support vector machine and multivariate adaptive regression spline models in long term prediction of river water pollution". *Journal of Hydrology*, *534*, 104-112.
44. J. C. Mojumder, J. C. Ong, W. T. Chong, & S. Shamshirband, "Application of support vector machine for prediction of electrical and thermal performance in PV/T system". *Energy and Buildings*, *111*, 267-277. 2016.
45. W. D. Fisher, T. K. Camp & V. V. Krzhizhanovskaya, "Crack detection in earth dam and levee passive seismic data using support vector machines". Procedia Computer Science, 80, 577-586.2016.
46. S. Ch, S. K. Sohani, D. Kumar, A. Malik, B. R. Chahar, A. K.Nema, ... & R. C. Dhiman, "A support vector machine-firefly algorithm based forecasting model to determine malaria transmission". *Neurocomputing*, *129*, 279-288.2014.
47. S. Huda, J. Abawajy, M. Alazab, M. Abdollalihian, R. Islam, & J. Yearwood, "Hybrids of support vector machine wrapper and filter based framework for malware detection". *Future Generation Computer Systems*, *55*, 376-390.2016.
48. J. Sahs, & L. Khan,"A machine learning approach to android malware detection". In *Intelligence and security informatics conference (eisic), 2012 european* (pp. 141-147). IEEE. 2012.
49. A. D. Schmidt, R. Bye, H. G. Schmidt, J. Clausen , O. Kiraz, K. A. Yuksel, ... & S. Albayrak, "Static analysis of executables for collaborative malware detection on android". In *Communications, ICC'09. IEEE International Conference on* (pp. 1-5). IEEE.2009.
50. Q. K. A. Mirza, I. Awan, & M. Younas, "CloudIntell: An intelligent malware detection system" *Future Generation Computer Systems*.2017.
51. S. Peddabachigari, A. Abraham, C. Grosan, & J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems" *Journal of network and computer applications*, *30*(1), 114-132.2007.
52. R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3)*. pp. 876—880.Available:http://www.halcyon.com/pub/journals/21ps03-vidmar