OPTIMIZATION OF SUPPORT VECTOR MACHINE FOR CLASSIFICATION OF SPYWARE USING SYMBIOTIC ORGANISM SEARCH FOR FEATURES SELECTION

BY

GANA, Noah Ndakotsu
MTech/SICT/2017/6903

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF TECHNOLOGY IN CYBER SECURITY SCIENCE

June, 2021

ABSTRACT

Malware's key target is to compromise system security pillars, the confidentiality, integrity and availability. Spyware is a form of malware program that collect entity's information including personal confidential information, activity logs on computing system, financial transaction, password and geolocation precision through monitoring target without prior knowledge of victim. The integration of computing devices into daily existence, as well as the exponential development experienced in application development including the expansion of interconnected computing devices serve as goldmine to malicious entities for target and exploit using spyware. In previous literature, Support Vector Machine (SVM) was employed for the classification of spyware, but has suffered setbacks of low performance as a result of untuned parameters as well as the use of irrelevant dataset features for training and classification. The optimization of SVM for classification of spyware using Symbiotic Organisms Search (SOS) algorithm for feature selection was therefore deployed to enhance performance. The results obtained from the study indicate that the technique performed optimally in spyware classification recording the following; 97.40% and 2.3% respectively for accuracy and false positive rate consecutively. Therefore, revealed that the optimization of SVM with SOS for classification enhances performance and reduces the rate of false alarm which is an improvement on existing literatures. This points the fact that tuned parameters of the model can be implemented for proper classification of spyware attacks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

API	Application Programming Interface
ARFF	Attribute Relation File Format
BA	Bees Algorithm
CIA	Confidentiality, Integrity and Availability
CV	Cross Validation
DE	Differential Evaluation
EULA	End User Lenience Agreement
GS	Grid Search
ICT	Information Communication Technology
MV	Mutual Vector
PSO	Particle Swarm Optimization
RAT	Remote Administration Trojan
RBF	Radial Basis Function
SMOTE	Synthetic Minority Oversampling Technique
SOS	Symbiotic Organism Search
SVM	Support Vector Machine

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

Compromise of system Confidentiality, Integrity and Availability (CIA) is normally the key target of malware according to (Javaheri *et al.*, 2018). A malware can be referred to as malicious software that execute series of codes in systems that have been compromised by its nefarious activities, exploiting the security defense mechanism in place. Virus, worm, trojan, rootkit and ransom-ware according to Nawfal and Wesam (2016), are examples of malware for which this research focuses mainly on spyware.

Spyware is a program that collects entity's information ranging from personal information, activities performed by entity, financial transaction, password, geo-location precision through monitoring without entity's prior knowledge Kumar *et al.* (2019). Spyware was first recorded in 1995 by Microsoft's business model which denotes as an espionage software, also spyware is an espionage ransomware code that exfiltrates sensitive information

Spyware software are akin to malicious program as it entices users into application execution, while being stealthy by circumventing removal activities as it uses subliminal channel. Although some spyware program is embedded in End User License Agreement (EULA) as backdoors to obtain user's consent, some spyware is installed without the user's consent. Spyware also have the capability of transmitting harvested information to third party after stealthily monitoring user behavior, web surfing habits, confidential details and user profile.

The exponential engagement of the world population in Information Communication and Technology (ICT) usage as at the 2018 was 51.2% which is 3.9 billion population, there have been a steady upward trend in the incorporation and usage of ICT, likewise cybercrime projection is estimated at USD 2 billion by the end of 2019, as a result of skyrocketing rate of ICT usage (Bustamante *et al*., 2018). This becomes a challenge to ICT users based on the fact that malicious activities need to be exterminated to avoid breach of CIA, spyware which is one of the stealthy unwitting trending malicious activity needs to be checkmated.

Support Vector Machine (SVM) a supervised machine learning model and noted for its ability to map feature vectors from nonlinear space to a higher multidimensional dimensional space, thereby making use of linear classifier obtained from the new space, suppose H represents the generated new feature space and $\emptyset$ represents mapping function so that $\emptyset : R^d \rightarrow H$, the feature vector $\bar{x} \in R^d$, the mapping of feature vector is denoted by $\emptyset(\bar{x})$, while the y label stands same, thus, $(\bar{x}_i, y)$ which is the training sample becomes $(\emptyset(\bar{x}_i), y)$, furthermore, H defines the hyperplane in the transformed space, which segregates the training sample $(\emptyset(\bar{x}_i), y_i), \ldots, (\emptyset(\bar{x}_n), y_n)$. This leads to obtaining a hyperplane in the space H which permits the mapping of feature vector $\emptyset(\bar{x}_i)$ to be segregated on one side of the hyperplane for label $y_i = -1$, and $\emptyset(\bar{x}_i)$ to the other side of the hyperplane for $y_i = 1$, (Arya and Bedi, 2018; Kulkarni and Harman, 2011)

However, SVM perform better once the parameters such as the kernel function are optimized and optimal feature for classification are also well defined (Alwan and Ku-Mahamud, 2017; Huang and Wang, 2006; Lin and Zhang, 2013).

Symbiotic Organism Search (SOS) is a metaheuristic algorithm that is based on organism symbiotic association in an ecosystem widely employed in various fields for optimization of problems ranging from scheduling of task, construction project and engineering structure design optimization. SOS was first introduced by Cheng and Prayogo (2014) and refer to as a symbiotic based relation of organisms in ecosystem for numerical optimization and engineering design problem, an initial population known as the ecosystem is defined at the initialization stage, which further generates a random organism population, for each related problem, an organism stands as a candidate solution which indicates the adaptiveness degree of an organism, each organism is bound to pass through the three major phases of symbiotic interaction namely; mutualism, commensalism and parasitism respectively in an iterative process Liao and Kuo (2017), symbiotic organism search algorithm is effective solution to solve complex numeric computations regardless of it few control parameters compared to some other optimization algorithms (Çelik and Öztürk, 2018).

Grid search algorithm is an exhaustive search algorithm, which completely explores a search space, while the variable to be optimized is represented by each dimension of a grid coordinate, the grid search works with a define range of value, known as the maximum and minimum value, that aids in establishing an optimal variable (Liu *et al.*, 2006).

Grid search algorithms concept is based on setting the parameter values such as the SVM kernel function of C, γ and step sizing, in order to determine a grid search points. Thus, for each parameters (C, γ) in the grid Support Vector Machine model is trained, in which the sample data is evaluated using the optimal selected model of training results (Yuanyuan *et al.*, 2017). Therefore, this research intends to apply SOS for selection of optimal spyware features that will be trained for classification.

1.2 Statement of Problem

The upsurge of the integration of computing devices and expansion of interconnected devices into daily existence serves as a goldmine to malicious entities which are the basic target of spyware. Support Vector Machine (SVM) machine learning classification algorithms have been widely used in classification of spyware, however, SVM suffers a setback as a result of the kernel parameter definition, which causes a high False Positive Rate (FPR) and low accuracy rate in classification of spyware such as experienced in researches conducted by Javaheri *et al*. (2018); Xu *et al*. (2015) resulting in 93% and 67.4% accuracy respectively with a false positive rate of 7%, undisclosed high FPR respectively, while an imbalanced dataset serve as a drawback in performance evaluation as experienced in the research by Kumar *et al.* (2019), which achieved an accuracy of 86.93% and FPR of 3.3%.

1.3 Aim and Objectives of the Study

The aim of this research is to design an optimized Support Vector Machine (SVM) classification algorithm for spyware classification to achieve better performance in accuracy with low false positive rate. The objectives are to:

i. design a Symbiotic Organism Search (SOS) algorithm for spyware feature selection to obtain optimal features,
ii. classify spyware attacks using Support Vector Machine (SVM) optimal parameter with optimal spyware features, and
iii. evaluate the performance with the technique using standard parameters as obtained in some relevant literatures.

1.4 Significance of the Study

While lots of machine learning classification techniques were employed in the detection and classification of spyware, high false positive rate and low accuracy performance are the associated drawback, this research will enhance performance of spyware classification.

It extends the present body of knowledge about SVM optimization for classification of spyware, likewise enriches and enhances the research field as spyware is given little attention in terms of research, not taking into cognizance the ranging effect of information espionage.

In order to have a robust classification algorithm for spyware, the optimization of existing technologies will aid in forestalling the stealth techniques used by spyware program in breaching the confidentiality, integrity and availability of computing entities through presenting an optimized machine learning model for the classification of spyware.

1.5 Scope of the Study

The research will be limited to feature selection-based SOS meta-heuristic algorithm and optimize SVM kernel parameters for classification of spyware using existing spyware dataset from well-known dataset repository and evaluating the performance of proposed design.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Spyware Preamble

Spyware had its first appearance on October 16, 1995 in a Usenet post in which hardware which can be used for espionage was termed spyware, depicted in Figure 2.1 is a flowchart of spyware working process, capturing and recording of the following from compromised device;

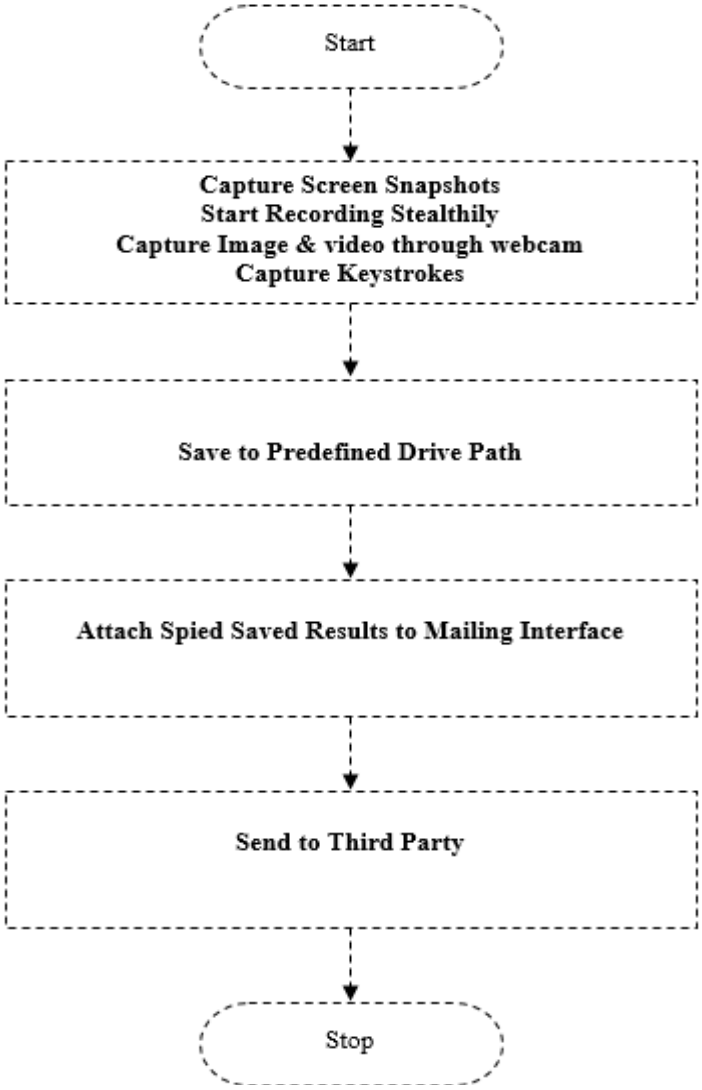

Figure 2.1 Flow Chart of Spyware Working Process (Patel, 2015)

screenshots, audio and image, keystroke and saving spy results to a predefined drive path, attaching to mailing interface and channeling same to third party all in a stealthy and subliminal manner, while in 2000 the founder of Zone Labs, Gregor Freund, used spyware as a term in a press release for their firewall product. Spyware is termed an application which stealthily and unwittingly installs on devices in order to monitor user activity, track and convey same to third party without user knowledge.

2.2 Related Literature

Kumar *et al*. (2019) in an experiment to classify spyware affected files through the implementation of data mining technique. More than eight thousand malware samples with hundreds of benign samples were used in spyware classification, which was based on Application Programming Interface (API) call dataset, J48 Decision Tree classification algorithm was used, an accuracy, true positive rate and false positive rate of 86.93%, 86.9% and 3.3% respectively was achieved, however, there exist an imbalance dataset sample based on malware to benign ratio, while focus was majorly based on API system call.

Xu *et al*. (2015) proposed a SpyAware framework that encompasses of a profiler, a feature extraction and a classifier, which were to aid in automatic profiling of app execution as it relates to binder calls and system, calls, obtaining feature vectors from execution traces and predicting and training of spyware execution in terms of feature vectors; support vector machine (SVM) and Naïve Bayes classifiers was incorporated at the classification stage. Furthermore, performance level of 67.4% and 64.2% accuracy was achieved in detecting spyware execution respectively. However, the research focuses majorly on smartphone privacy leakage issues, and also based on a define version of Android OS platform, accuracy rate is low and a high undisclosed FPR is said to be achieved.

In order, to counteract the challenges faced by existing anti-spyware tools such as detecting of spyware that have capacity to modify self against detection, Wu *et al*.s (2007) proposed a Stateful Threat Aware Removal System (STARS) that has the potent to track critical activities of running process, monitoring spyware removal task effectiveness over a period, and establishing a trade-off between system dependability and system performance as it relates to severity threat of spyware system. To this end, it was established that STARS has the capacity to detect and remove self-healing spyware, a challenge faced by existing

commercial anti-spyware tools based on an experiment performed in this study. However, there is a shortfall in overall performance in removing self-healing spyware likewise the proposed method lacks the capacity to detect hidden registry entries.

Wazid *et al*. (2013) opine a prevention mechanism against key logger spyware attacks, the proposed methodology include the following phase, key logger spyware attack, honeypot based detection and prevention of key logger spyware, generation of spyware attack in order to help track attack behavior, detection of keylogger spyware, monitoring of malicious system activities and permanently disabling the key logger spyware by a prevention server is achieve respectively by the aforementioned phases. It was stated that proposed mechanism if employ can tackle key logger spyware attack, however, focus was designated on key logger spyware attack alone.

Cloud theory model was used to develop an interest model to enable spyware detection, the research, Wang and Chen (2009) presents a novel spyware detection technique that employs an abstract characterization of popular classes of spyware programs through the use of data mining approach as a result of its capacity to discover program of interest in large amount of behaviors, thereby leading to overcoming the drawback associated with unknown signature based detection as the proposed model can detect unknown spyware as well as variant of known spyware. Theoretically, it was further reveal that the define model was able to detect spyware programs optimally, though, this research was theoretically based and not implemented in real system scenario.

Extraction and selection of optimal features to detect spyware was proposed by Sheta *et al*. (2016) in the research that leads to optimal features selection is based on the frequency and appearance of the feature in the dataset as opine in this study. Accuracy performance

metrics was employed in this study as well as the following classification algorithms; ZeroR, Naïve Bayes, C4.5 decision Tree (J48), Support Vector Machine (SVM), JRip and Random Forest attaining an accuracy of 91.50%, 99.49%, 99.86%, 99.80%, 99.24% and 99.86% respectively with n-gram equal to 5 out of 100 selected features, J48 classifier outperforming all competing approaches, however, more performance metrics will enhance result interpretation and gauging.

In order to attain an optimal and accurate detection of Adware, data mining algorithms such as Naïve Bayes, Support Vector Machine algorithm SMO, IBk, J48, and JRip were employed in the proposed approach for accurate detection of Adware using Opcode sequence extraction to identify unseen and novel instances of adware along n-gram size, detection rate, false alarm rate, and accuracy were used as performance evaluation metric including area under receiver operation characteristics curve (AUC). ZeroR serve as the baseline classifier, IBk achieve AUC, FNR, FAR of 0.949, 0.022 and 0.115 respectively with the value of n-gram equal to 4 and a 70% split. IBk was said to have outperform other classification algorithms used, however, focus was based on adware, with a little collection of dataset used in experimental evaluation (Shahzad, 2011).

The research performed by Shazhad *et al*. (2010) basically based on evaluation of Windows platform executable achieve through the implementation of machine learning algorithms such as ZeroR, Naïve Bayes, Support Vector Machine (SMO), J48, Random Forest, and JRip classification algorithms, also, a 10 fold cross validation was employ to classify unseen binaries. Accuracy and Area Under Receiver Operation Characteristics (ROC) curve was used as metrics in terms of performance evaluation. J48 classification algorithms achieve 90.5% accuracy using n as 6, denoting the highest accuracy compared to other

classification algorithms used in this study, while ZeroR, Naïve Bayes, SVM, Random Forest and JRip achieved 86.92%, 89.80%, 89.65%, 89.48%, 89.45% accuracy respectively, Random Forest algorithm give an AUC score of 0.83 using n as 6, ZeroR, Naïve Bayes, SVM, J48, Random Forest and JRip achieved AUC of 0.50, 0.62, 0.71, 0.65 and 0.66 respectively. Furthermore, common feature-based extraction and frequency feature extraction was employed in order to obtain Reduced Feature Set (RTS) which was further used in generation of attribute relation file format (arff) files, nevertheless, attention dwell mostly on Windows executable and dataset used in experiment is of small size.

Tripathy *et al*. (2016) in their research opined a framework with the capability of detection and classification of spyware. The following classification algorithms: Decision Tree, ZeroR, JRip, J48 and Naïve Bayes was applied in classifying existing spyware, Decision Tree scored the best accuracy of 97.7854 with Kappa Statistic and ROC area scoring 0.723 and 0.9356 respectively, the opined framework serve as a robust rule-based algorithm to enhance the proposed framework.

Data mining based detector optimized by Breadth-First Search algorithms was employed to achieve an accuracy of 90.5% and 0.731 FPR in the research by Bahraminikoo *et al*. (2012) in order to detect spyware, feature set generate form Common Feature-based Extraction (CFBE) feature selection technique with n-gram value of 4 achieving an accuracy of 89.49%, 88.21% and 88.02% respectively for Random Forest, Naïve Bayes, and Support Vector Machine, the following FPR was also recorded against each of aforementioned classifier respectively 0.731, 0.665 and 0.665 was used as a comparative factor. However, the research experiment was majorly based on executable files in evaluating the performance of employed method that was developed and experience a high FPR and low

accuracy. Javaheri *et al*., (2018) proposed a kernel level system routine interception in detecting and eliminating spyware and ransom ware, Linear Regression, JRip and J48 decision tree classifiers was employed in the research in order to achieve the spyware and ransom ware detection as well as elimination, experiment performed based on the designed methodology give an accuracy of 93% with a FPR of 7%, however, the resulting performance evaluation indicates a low accuracy couple with high FPR.

Wang. (2006) opined a surveillance spyware detection system that encompasses both static and dynamic analysis, in order to classify spyware SVM classification algorithm optimal features generated from information gain ranking was trained, an accuracy of 97.91% for known spyware and 96.4% for the unknown spyware was achieved and a false positive rate of 0.68% and TPR of 95.33% based on the static and dynamic analysis performed. However, the research based the experiment basically on executable and a resulting high FPR. Boldt *et al*. (2008) proposed a hypothesis of the possibility of classifying software that have spyware functionality embedded based on the software End User License Agreement (EULA). The experiment performed by the study using data information obtained from 100 software application through the means of anti-spyware application in order to aid in determining software applications that have spyware features embedded therein based on the application EULA, 15 different classification algorithms was employed and multi-nominal Naïve Bayes, SVM and Voter Perception algorithms was assert to have outperformed other classification algorithms while achieving optimal AUC and accuracy rate and low false positive rate, the obtained result indicates that the proposed hypothesis about EULAs can serve as an evaluating mechanism for classification of soft

wares with embedded spyware functionality. However, the result indicates a very high FPR and Low accuracy.

Sai *et al*. (2019) developed a novel malware detection technique known as Memory management with API Call mining (MACA-I) to detect malware that transit in memory management API, monitored and tracked based on dynamic analysis, the evaluation of the developed technique was based on accuracy and sensitivity performance metrics, using the following machine learning algorithms; Logistics Regression, Support Vector Machine (SVM), and Decision Tree, which achieved the following accuracy 78.78%, 77.27% and 89.89% respectively, while the sensitivity of 91.17%, 85.28% and 97.05% was attained respectively, however, the research majored on API calls only.

2.3.1 Concept of Spyware Form

The following are some classified part of spyware which in some cases represents forms of spyware; Adware, Key logger, and Trojan horse as buttress below and tends to exploit device resource while serving as security threat (Stafford *et al*., 2004).

2.3.2 Adware

This are applications characterized by it functionality such as tracking user browser activities and relaying adverts to the compromised device based on tracked browsing activities, also adware have the capability of altering browser functionality through installation of browser helper objects as well as modifying browser default setting and redirecting searches (Stafford *et al*., 2004).

2.3.3 Keystroke logger

This are program or hardware developed to record keystroke. It has the ability to violate device privacy and used to infiltrate target network, most at time keyloggers are installed as part of Trojan Horse attacks (Stafford *et al*., 2004).

2.3.4 Remote administration trojans (RATs)

RATs is regard as unknown or unexpected delivery of embedded malicious content in an authentic application such as free-wares, games, cracked versions of software in order to get a foothold of exploited device install self and make a remote communication with third party relaying compromised information and receiving command in a covert manner (Stafford *et al*., 2004).

2.4 Symbiotic Organism Search Concept

The nature inspired algorithms symbiotic organism search (SOS) metaheuristic algorithm is a direct simulation of the relationship that exist between organisms in the ecosystem for survival.

The adoption of algorithm features; mutualism, commensalism as well as parasitism feature of interaction is to aid in emerging suitor solutions.

2.4.1 Mutualism

Entails the cohabitant of two organisms of which both benefit from each other without losing on both side of the symbiotic.

2.4.2 Commensalism

This entails the symbiotic relationship between pair of an organism one benefit, while the other organism neither gain or loses from the interaction.

2.4.3 Parasitism

In this form of relationship one of the organisms stands to gain, while the other is harmed, the sequential iteration as a result of progressive population of organisms offer potential solution through the use of SOS algorithm (Pal *et al*., 2020). The population of an ecosystem which can be tagged as an initial ecosystem of organism (candidate solutions) with the number of organisms in the ecosystem (ecosize) is generated and represented as:

$$e = \{X_1, X_2, X_3, \dots, X_{ecosize}\}$$

The organism position i is defined as $X_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{id}\ ]$ for optimization of d – dimensional problem.

Mutualism, commensalism and parasitism phases are used to update the positions of each organism found in the ecosystem respectively.

The phases of SOS Algorithms

a) Mutualism Phase

To enhance the survival in the mutual relationship of both X_i and X_j, organism is X_i randomly chooses an organism X_j $(i \neq j)$. The emergence of a new organism is achieved as indicated in equation (2.1) and (2.2).

The mutual interaction vector between X_i and X_j represents mutual vector (MV) as reflected in equation (2.3).

X_{best} indicates organism with the best fitness value.

β_1 and β_2 indicates the benefit factors among organism X_i and X_j.

β_1 and β_2 establishes either 1 or 2 stochastically.

While light and heavy benefit denotes the values 1 and 2 respectively.

The new organism fitness values $f(x_i^{new})$ and $f(x_j^{new})$ are evaluated, then X_i and X_j are updated to x_i^{new} and x_j^{new}, respectively, if the fitness of the new organisms is excellent as represented in equation (2.4) and (2.5), respectively.

$$X_i, new = X_i + R_1(0,1 - \beta) * (X_{best} + MV * \beta_1) \quad (2.1)$$

$$X_j, new = X_j + R_2(0,1 - \beta) * (X_{best} + MV * \beta_2) \quad (2.2)$$

$$MV = \frac{1}{2}(x_i + x_j) \quad (2.3)$$

$$X = \begin{cases} X_i^{new} & if\ f(X_i^{new}) > f(X_i) \\ X_i & if\ f(X_i^{new}) \leq f(X_i) \end{cases} \quad (2.4)$$

$$X = \begin{cases} X_j^{new} & if\ f(X_j^{new}) > f(X_j) \\ X_j & if\ f(X_j^{new}) \leq f(X_j) \end{cases} \quad (2.5)$$

where $R_1\ (0,1)$ and $R_{12}(0,1)$ are vectors of random numbers on the range 0-1; $f(.)$ is the fitness function.

b) Commensalism Phase

A random relationship exists between organism X_i and X_j $(i \neq j)$ for the enhancement survival of X_i. The emergence of new organism is achieved using equation (2.6); X_{best} is regarded as the fittest organism.

X_i is updated to X_i^{new}, if $f(X_i^{new})$ is better than that of $f\ (X_i)$ in respect to relation indicated in equation (2.7).

$$X_i^{new} = X_i + R(-1,1) * \left(X_{best} + X_j\right) \tag{2.6}$$

$$X_i = \begin{cases} X_i^{new} & if\ f(X_i^{new}) > f(X_i) \\ X_i & if\ f(X_i^{new}) \leq f(X_i) \end{cases} \tag{2.7}$$

where $R_1\ (0,1)$ is a vector of random numbers between -1 and 1.

c) Parasitism Phase

A uniform generated random number is applied on mutating organism X_i to achieve an artificial parasite know as parasite vector.

The parasite vector (PV) is evaluated against a randomly chosen organism X_j, and the parasite vector replaces X_j if the parasite vector is filter. Equation (2.8) reflects the relation.

$$X_j = \begin{cases} PV\ if\ f(PV) > f\left(X_j\right) \\ X_j\ if\ f(PV) \leq f\left(X_j\right) \end{cases} \tag{2.8}$$

where PV is the parasite vector (Pal *et al*., 2020).

2.5 Applications of Symbiotic Organism Search (SOS)

Symbiotic Organism Search have been applied in various fields to optimize performance, fields ranging from cloud data center for energy efficient scheduling of virtual machine as postulated by Abdulhamid *et al*. (2019) cloud computing environment for task scheduling by Abdullahi *et al*. (2015) power system for load frequency control of multi area power system Guha *et al*. (2018) nonlinear channel equalization using wavelet neural network trained by SOS and the prediction of sea wave height as well as in automatic data clustering respectively, Akbarifard and Radmanesh (2018); Nanda and Jonwal (2017); Zhou *et al*. (2019) all of which the implementation of symbiotic organism search performed excellently in solving all the various complex problem as associated with each aforementioned fields.

SOS algorithm was applied by Jaffel and Farah (2018) in the research study 'a Symbiotic Organisms Search algorithm for feature selection in satellite image classification' in order to select optimal image features for classification by artificial neural network which will further aid in excellent classification and thereby reducing error rate, the performance of the opined method was said to have outperformed other state of the art feature selection techniques as reflected from the obtained experimental result. SOS performance compared against other met-heuristic algorithms such as Bees Algorithm (BA), Particle Swarm Optimization (PSO), Differential Evolution (DE) as well as Genetic algorithm in different numerical computational test analysis and challenges in engineering indicates that SOS had an excellent optimal performance with regards efficiency and effectiveness (Cheng and Prayogo, 2014).

2.6 Optimization of Support Vector Machine Parameter with Grid Search Algorithms

Liu *et al*. (2006) In order to determine the optimal parameters of SVM, Grid Search (GS) algorithm was implemented in defining SVM hyper-parameters based on the exhaustive searching capability of GS, Syarif and Wills, (2016); Yuanyuan *et al.* (2017) state that SVM performance is strongly based on parameter tuning, grid search algorithm was applied in tuning the parameter of SVM in order to improve performance.

2.7 Support Vector Mechanism

Vapnik and his research fellow in the 1990's came up with Support Vector Machine (SVM) theory, a concept that was envisage from neural network or referred to as a mathematical adjunct of neural network (Olson and Delen, 2008). SVM is also known to be an optimal hyper-plane that is based on mathematical computational scheme. SVM is a supervised learning algorithms Kremic and Subasi (2016), SVM can be employed in classification or regression field problem according to (Olson and Delen, 2008).

And is widely used to optimize problems in diverse aspect of power, machine learning, health, and time series forecasting with and excellent performance record in solving complex numerical problems. Support Vector Machine (SVM) known to be a type of classification algorithm that has been adopted in various field of classification, with the capability of implicitly performing a transformation of non-linear feature space (Boswell, 2002), due to ability of SVM in segregating data samples through the employment of best hyperplane, SVM has witness a tremendous application in various fields ranging from multidimensional data classification of microarrays, windspeed prediction, voltage stability

to contingency ranking (Boswell, 2002), however if there exist a great significant over or underrepresentation of a class over another, this leads to class imbalance which has a significant performance limitation on machine learning classification (Sukhanov *et al.*, 2018).

The classification of linear and non-linear data can be achieved using SVM, in order to classify a training data using SVM, the original data is transformed into a multidimensional space while a hyper-plane is constructed in a higher dimension as reflected in Figure 2.1. A hyper-plane is built in higher dimensions in order to achieve classification through SVM, the hyper-plane is also known as decision plane as depicted in Figure 2.1, a particular class of training data is distinguished from another class type using a decision plane.

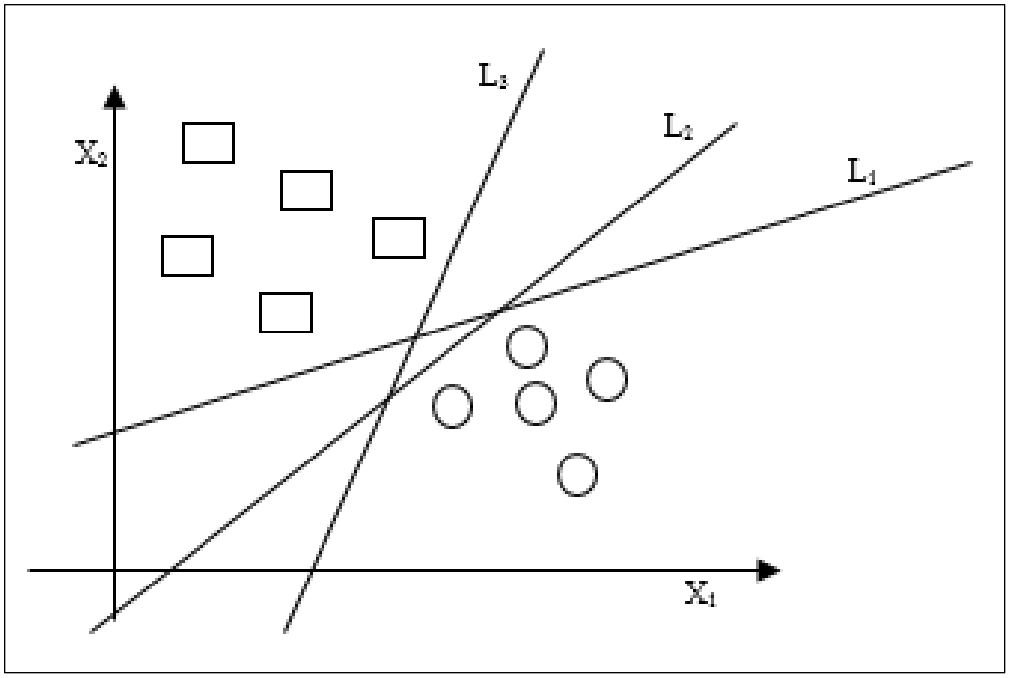

Figure 2.2 Many linear classifiers (hyper planes) separating the data (Boswell, 2002)

An a[illegible] training data by constructing a hyper-plane, SVM locates the vector points, known also as support vectors, which establishes the decision boundary and defines a maximum marginal

separation between the classes. SVM separates the classes with highest marginal distance in the decision plane.

The maximum marginal hyper-plane is defined by the middle line, a boundary line would be selected which separates the two classes at a maximal distance to the closest data point, given a two-class situation, a decision plane or hyper-plane for classification is stated by Vapnik's theory.

SVM possess an excellent and standard statistical base, implemented in various field of knowledge ranging from malware detection, imaging optimization and big data classification (Demidova, 2016; Maigida *et al*., 2019; Takeuchi *et al*., 2018).

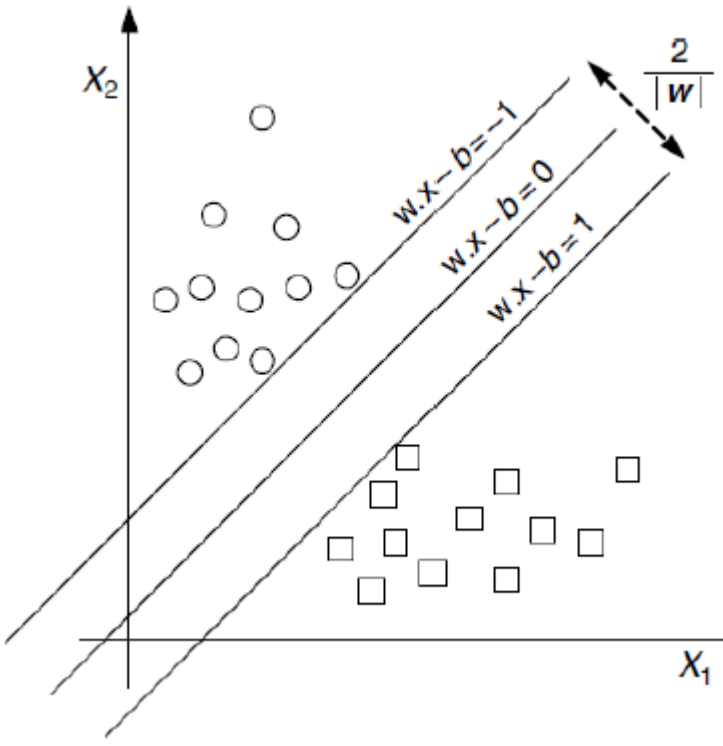

Figure 2.3 Maximum separation hyper-plane (Boswell, 2002)

Given input x_i is $m \times n$ dimension with a corresponding class $y_i \in \{-1, 1\}$.

$$\omega^T \cdot (x_i) + b = 0 \qquad (2.9)$$

The hyper-plane is built such that it fulfills the following inequality function for both the classes.

$$\omega^T \cdot (x_i) + b \geq +1 \qquad y_i = 1 \qquad (2.10)$$

$$x^T \cdot (x_i) + b < +1 \qquad y_i = -1 \qquad (2.11)$$

But using equation (1) and (2) equation (4) is generated;

$$y_i(\omega^T \cdot (x_i) + b) \geq +1 \; i = 1 \ldots n \qquad (2.12)$$

where ω^T is the normal to the line and also known as the weight vector, while b is the bias.

SVM also have a kernel function that which is a standard mathematical method employed for a nonlinear mapping for a higher dimensional data. The kernel functions which are linear kernel, polynomial kernel, radial basis kernel and sigmoid functions aids in providing solution to higher dimensional classification of original input data, basically evaluating the value of dot product mapped data point into feature space. The figure 2.4 represents the pseudocode for SVM.

Algorithm 1 Training an SVM

Require: $\boldsymbol{X}$ and $\boldsymbol{y}$ loaded with training labeled data, $\boldsymbol{\alpha} \Leftarrow 0$ or $\boldsymbol{\alpha} \Leftarrow$ partially trained SVM

1: $C \Leftarrow$ some value (10 for example)
2: **repeat**
3: **for all** $\{x_i, y_i\}, \{x_j, y_j\}$ **do**
4: Optimize α_i and α_j
5: **end for**
6: **until** no changes in $\boldsymbol{\alpha}$ or other resource constraint criteria met

Ensure: Retain only the support vectors ($\boldsymbol{\alpha_i} > 0$)

Figure 2.4 Support Vector Machine pseudocode (Maigida *et al.*, 2019)

2.7.1 Linear kernel function

Simple kernel function known as Linear Kernel Function (Arya and Bedi, 2018)

$$k(x, x') = x \,.\, x^T \qquad (2.13)$$

x^T represents the input transpose of a metrics x .

2.7.2 Polynomial kernel function

It is a non-linear kernel function; it is directional that means output depends on the direction of input in a low dimensional space. This is due to the dot product in the kernel (Arya and Bedi, 2018).

$$k(x, x_i) = (1 + x \,.\, x_i^T)^P \qquad (2.14)$$

'P' defines the kernel function degree "poly" polynomial function.

2.7.3 Radial basis kernel function

The radial basis kernel function select the solution for smooth kernel (Arya and Bedi, 2018).

$$k(x, x_i) = e^{-\gamma x - x_i^2} \gamma > 0 \qquad (2.15)$$

γ represent parameter that determine the kernel 'RBF' spread.

2.7.4 Sigmoid function

The Sigmoid Function is also known as hyperbolic tangent function (Arya and Bedi, 2018).

$$k(x, y) = tanh(ax^T y + c), a, c > 0 \qquad (2.16)$$

'a' and 'c' in the above equation stand for kernel parameters.

2.8 Imbalance dataset

Imbalance class can be noted as a scenario where the difference in training sample that exist in a classification task greatly varies, in order to improve the data imbalance challenge, thereby obtaining relatively balanced class, oversampling techniques known as Synthetic Minority Oversampling Technique (SMOTE) is applied on imbalanced class (Lv *et al*., 2018). SMOTE implements an algorithm of nearest neighbor of k homogenous samples, where k is normally an odd number greater than 1, afterward, linear interpolation is implored through randomly choosing one of the k nearest neighbor data. The technique interpolates between two sample points, creating a new sample class data, adds the class data to a few class sets and balances the distributed imbalance of the dataset. Sukhanov *et al*., (2018) noted that imbalanced class are part of the challenges exhibited in terms of classification using machine learning models, as it regards to performance degradation. SMOTE algorithm was deployed in the research by Zhang *et al.* (2019) in order to balance the training target for better performance as achieved in the clutter suppression experiment.

Chawla *et al.* (2002) proposed SMOTE algorithm backed by the idea of establishing a balanced dataset through interpolating into few samples, given a minor sample x, through searching for k value of recent samples of a few classes, the under-sampling magnification is N.

In the samples of k nearest neighbours, N samples are randomly selected and are denoted as $y_1, y_2, \dots, y_n$. In a few sample, randomly linear interpolation is performed between $x\ and\ y_j (j = 1,2, \dots N)$ to create a new minority sample P_j.

$$P_j = x + rand(0,1) * (y_i - x), \qquad j = (1,2, \dots, N) - 1 \tag{2.17}$$

In equation (2.17), rand (0,1) denotes a random number with an interval (0,1), while few of these newly synthesised samples are placed in new dataset for training to produce new datasets, x_i = example of a few classes x_{i1} x_{i2} x_{in} , table 2.1, depict the pseudocode for SMOTE algorithm (Lv *et al*., 2018).

Table 2.1 **Algorithm SMOTE(T,N,k**) (Lv *et al*., 2018).

Input: Number of minority class samples T; Amount of SMOTE N%; Number of nearest neighbors k
Output:(N/100)*T synthetic minority class samples

```
(* If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTE. *)
if N < 100
then Randomize the T minority class samples
T = (N/100) * T
N = 100
end if
N = (int)(N/100) (*The amount of SMOTE is assumed to be in integral multiples of 100. *)
K = Number of nearest neighbors
Numattrs = Number of attributes
Sample[][]: array for original minority class samples
Newindex: keeps a count of number of synthetic samples generated, initialized to 0
Synthetic[][]: array for synthetic samples
(* Compute k nearest neighbors for each minority class sample only. *)
for 1-i to T
Compute k nearest neighbors for i, and     save the indices in the nnarray
Populate(N, i, nnarray)
 endfor
Populate(N, i, nnarray) (* Function to generate the synthetic samples *)
while N != 0
Choose a random number between 1 and k, call it nn. This step choose one of
the k nearest neighbors of i.
for attr-1 to numattrs
Compute: dif =
      Sample[nnarray[nn]][attr] − Sample[i][attr]
Compute: gap = random number between 0 and 1
Synthetic[newindex][attr] =
       Sample[i][attr] + gap * dif
endfor
newindex++
N = N − 1
endwhile
 return (* End of Populate. *)
End of Pseudo-Code.
```

2.9 Findings from Literature

A summary of related literature is presented in this section as shown in table 2.2 (appendix), presented in this section also is the limitations that this research proposed to bridge in order to contribute to knowledge.

The related literature reviewed revealed that spyware classification using machine learning algorithm have been experimented, however, performance recorded indicates a relatively high false positive rate and insufficient accuracy rate, researches in the field of spyware is given less attention while the spread of spyware threat continue to penetrate computing platform at same pace as the growth of computing platform exponentially.

Hence, there is need for an enhanced machine learning algorithm that will prove better in terms of performance evaluation metrics, this research bridges the limitation experienced by designing optimize SVM classification algorithm for spyware classification to achieve a better performance in respect to accuracy, true positive rate, false positive rate and recall.

CHAPTER THREE

3.0 RESEACH METHODOLOGY

3.1 Research Procedure

The research aims at optimizing the classification of spyware using Support Vector Machine (SVM), through the selection of optimal spyware features and parameters for SVM by Symbiotic Organism Search (SOS) algorithm and Grid Search (GS) algorithm respectively. The research methodology employed in this research is represented in Figure 3.1.

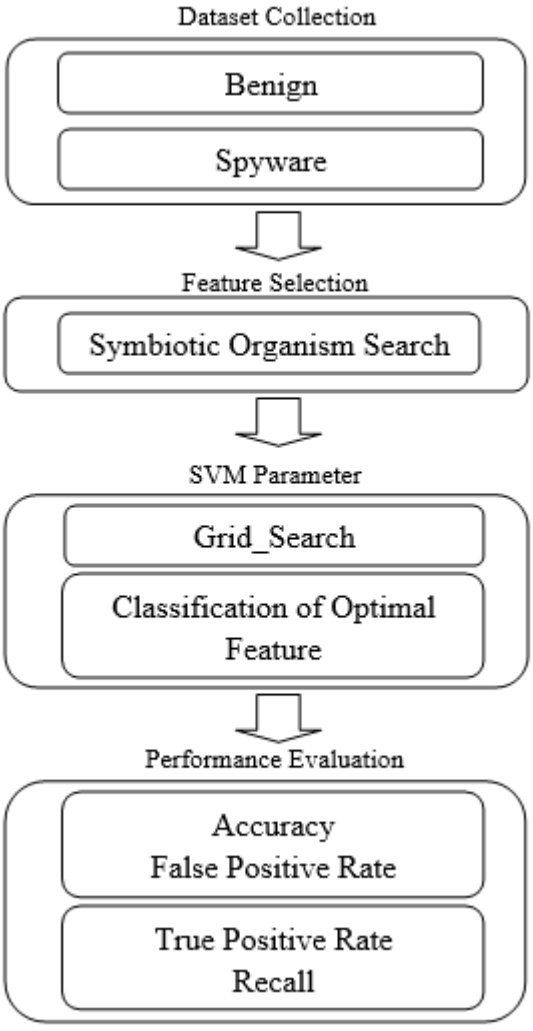

Figure 3.1 Research Process Block Diagram

3.2 Spyware Classification Datasets

The research makes use of the available dataset of Advance Persistent Threat (APT) dataset accessible for research and development found in Microsoft subsidiary repository known as Github, the dataset is comprised of 189 attributes and 291 instances, the following form part of the attributes; techniques, tactics and procedure (TTP), spyware, backdoor, Trojan and rootkit.

3.3 Data Preprocessing

Data preprocessing is a key component in machine learning classification algorithm in other to optimize learning process, imbalance dataset can result to overfitting, affecting the performance of the model, as a result of the nature of the dataset deployed in this research. Synthetic Minority Oversampling Technique (SMOTE) was used in order to achieve excellent classification performance, SMOTE aids to overcome the challenge of biasness associated with imbalanced dataset.

3.4.1 Feature selection

Dataset of high dimensionality are difficult to manage manually, in other to read meaning of large set of data, data mining come in place, however, redundancy exist in collection of huge datasets as a result of technology imperfection which may alter the performance of machine learning classification algorithm evaluation. Feature selection is a process that aids in reducing redundancy in order to optimize the quality of a target in terms of classification through selection of feature subset from the main dataset features while maintaining the actual feature accuracy representation.

SOS was employed in the selection of optimal spyware features that will optimize the classification of spyware using SVM classifier. Figure 3.2 depicts SOS algorithm pseudo code.

```
Initialize Ecosystem: Define the Number of Organisms
Define Termination Criteria
Initialize the Ecosystem Randomly
Calculate the fitness and find the initial best solution
while (termination criteria not met)
{
        for i = 1: ecosize

        Mutualism Phase
        Select one organism randomly, Xj where Xj ≠ Xi
        Determine Mutual Vector = (Xi + Xj)/2
        Determine Benefit Factors BF1 & BF2 = 1 or 2
        Modify Xi and Xj according to equations (2.1) & (2.2)
        Calculate fitness values of modified organisms

        if(modified organisms are fitter than previous)
          accept modified organism to replace the previous
        else
          reject modified organisms and keep the previous

        Commensalism Phase
        Select one organism randomly, Xj where Xj ≠ Xi
        Modify organism Xi according to equation (2.6)

        Parasitism Phase
        Select one organism randomly, Xj where Xj ≠ Xi
        Create a Parasite Vector from Organism Xi
        Calculate fitness values of new organisms

        if(Parasite Vector fitter than Xj)
          Replace organism Xj with Parasite Vector
        else
          Keep organism Xj and delete Parasite Vector

        End

}
```

Figure 3.2 Pseudocode for Symbiotic Organism Search Algorithm

The achievement of optimal spyware features will be attained through the procedure establishment as depicted in Figure 3.2; initialization of the ecosize, definition of termination criteria, initialization of the ecosystem randomly, calculating the fitness and establishing the initial optimal solution, selection of candidate feature subset, evaluation of generated candidate feature subset through mutualism, commensalism and parasitism respectively with the defined equations (2.1) through equation (2.8) in chapter two respectively and output of optimal relevant value if the termination criteria is achieved.

3.4.2 Optimization of SVM parameter using grid search algorithm

Grid Search algorithm was employed for the performance of SVM classifier in order to classify optimal features of spyware dataset by first predefining SVM parameters and training the optimal spyware features while tracking the performance evaluation of SVM classifier based on spyware testing dataset. Grid search algorithm was used to define SVM model optimal parameter to achieve a reduced training error through regulating the penalty (C) and kernel function (γ) respectively. This is to aid in optimizing SVM parameters, a retraining of SVM classifier with the testing spyware dataset was evaluated with performance metrics Figure 3.3 depicts the optimization flowchart of GridSearch-SVM classifier.

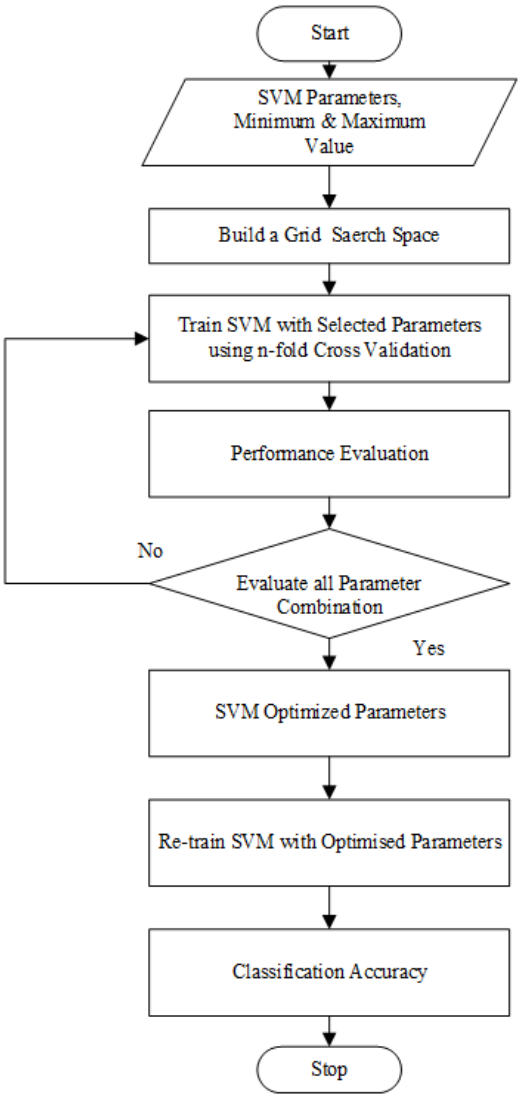

Figure 3.3 GridSearch SVM Classifier

3.5 Proposed Optimization of SVM using Gridsearch Algorithm for Spyware Classification

The proposed Gridsearch a search spyware detection using Support Vector Machine classifier will encompass the represented flowchart phases represented in Figure 3.4. The supply of training dataset which is comprise of both spyware and benign data into the initialization phase of symbiotic organism search algorithms to determine the optimal

global solution for features selection, initialization of SVM parameters, training of SVM model with initialized optimal features through regulated means by SOS function to obtain optimized parameters. The obtained optimal selected features will be trained using Support Vector Machine classifier alongside testing spyware dataset for classification of spyware dataset. The evaluation performance parameter will be employed through building a grid search to determine the efficiency of spyware classification using Support Vector Machine classifier. This research employed a 5-fold and 10-fold cross validation in the experiment.

The grid search which majorly is an exhaustive search-based algorithm on defined subset of the hyper-parameter space is employed in this research. The hyper-parameters are specified using minimal value (lower bound), maximal value (upper bound) and number of steps. The performance of every combination is evaluated using some performance metrics.

Grid search optimizes the SVM parameters (C, and Gamma) using a cross validation (CV) method as a performance metric. The goal is to identify good hyper-parameter combination so that, the classifier can predict unknown data accurately. To define C and Gamma using k-fold CV, first the spyware dataset is splited into k subsets, a subset is used as a testing data and then evaluated using

the remaining k-1 training subsets. Then, the CV error is calculated using this split error for the SVM classifier using different values of C and γ parameters. Various combination of hyper-parameters value is entered and the one with the best cross-validation accuracy (or the lowest CV error) is selected and used to train an SVM on the whole dataset.

RBF kernel been a default SVM kernel function was adopted having the parameters C and γ. The optimization of SVM parameter using grid search is depicted in Figure 3.4, while

figure 3.5 represents the algorithm in this experiment. The range of C and γ SVM classifier parameter define was from 0.0001 to 10,0000.

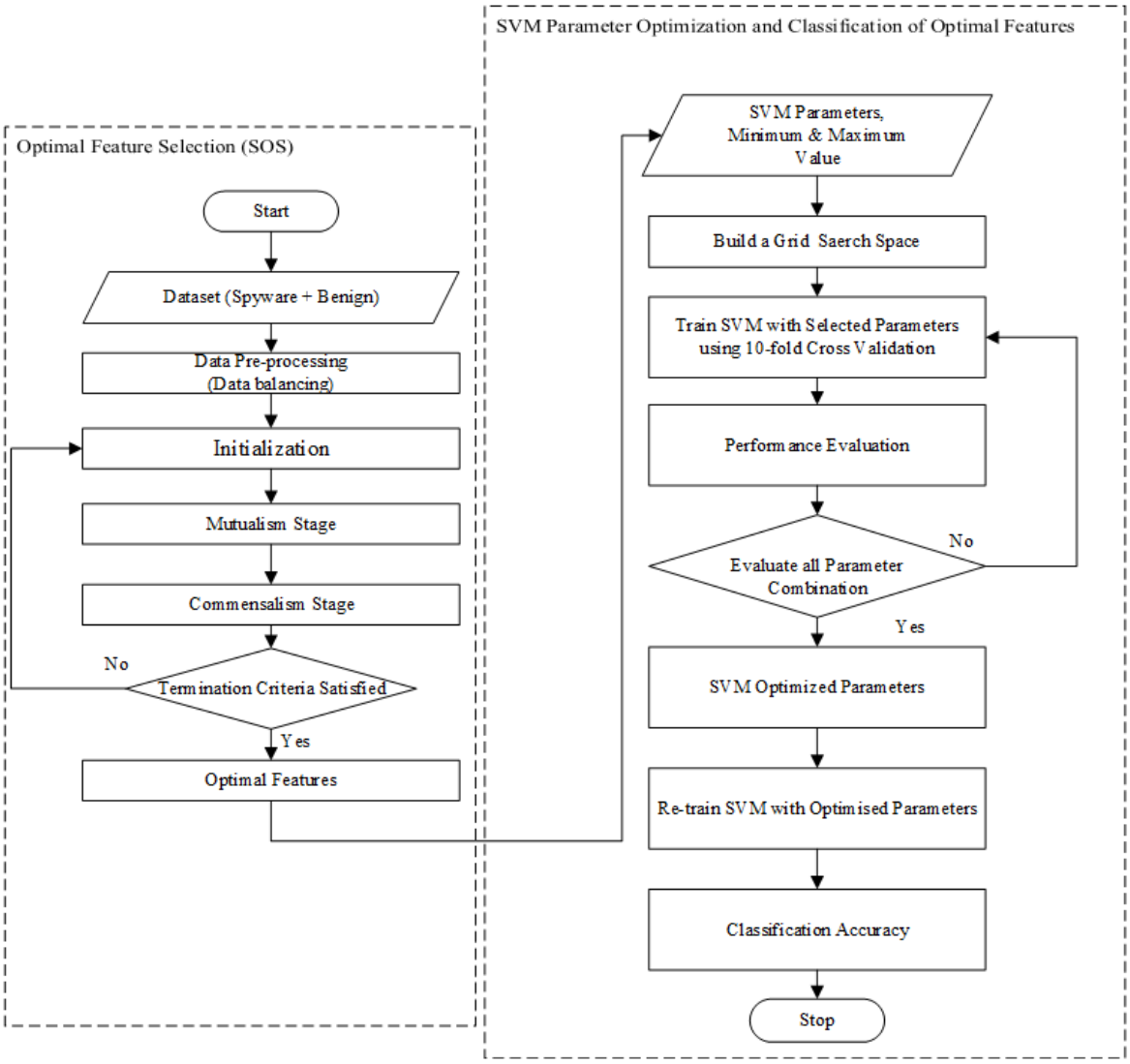

Figure 3.4. Flowchart of the Proposed Optimization of SVM Using Grid Search for Spyware Classification

1: **Begin**
2. initialize $\boldsymbol{x}, \boldsymbol{y}$ **% Initialization Phase**
3: initialize sets of SVM parameters
4: form SVM using training spyware dataset
5: **While** (*Accuracy* < *MaxAccuracy*)
6: **For** $\boldsymbol{x = a_i \rightarrow a_n}$ **do**
7: **For** $\boldsymbol{y = b_i \rightarrow b_n}$ **do**
8: **Set** $\boldsymbol{C = 10^x}$
9: **Set** $\boldsymbol{Gamma = 10^y}$
10: Train SVM with $\boldsymbol{C}$ and $\boldsymbol{Gamma}$ on TrainingSet
11: Evaluate SVM Model on ValidationSet
12: **If** $\boldsymbol{Accuracy}$ is better than $\boldsymbol{MaxAccuracy}$ then**% Optimization Phase**
13: Save $\boldsymbol{OptimalC = C}$ and $\boldsymbol{OptimalGamma = Gamma}$
14: **End If**
15: **End For**
16: **End For**
17: **End While**
18: return $\boldsymbol{OptimalC}$ and $\boldsymbol{OptimalGamma}$
19: classify spyware on testing dataset with optimal SVM parameter
20: return perform metrics
21: **End**

Figure 3.5 Grid Search-SVM Optimization for Spyware Classification Pseudocode

3.6 Performance Evaluation Metrics

The metrics used in evaluating the performance of the system is include Accuracy, True Positive Rate and False Positive Rate:

The performance of each classifier is evaluated using the true positive rate, false positive rate and overall accuracy which are defined as follows:

a. **Accuracy (ACC)**: Percentage of measure of how correctly a model can identified benign from a spyware expressed in equation below.

$$Accuracy(ACC) = \frac{TP + TN}{(TP + FP + FN + TN)} \quad (3.1)$$

b. **True Positive Rate (TPR)**: Percentage of correctly identified benign instances by the classification model, expressed in equation below

$$TPR = \frac{TP}{(TP + FN)} \quad (3.2)$$

c. **False Positive Rate (FPR)**: Percentage of wrongly identified Spyware instances, expressed in equation below

$$FPR = \frac{FP}{(TN + FP)} \quad (3.3)$$

d. **Recall**: this is a measure of completeness (what percentage of positive tuples are labeled as such), Recall is the same as sensitivity, the measures can be computed as below

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{P} \quad (3.4)$$

Given the following expression;

True Positive (TP): Number of correctly identified benign instances.

False Positive (FP): Number of wrongly identified Spyware instances.

True Negative (TN): Number of correctly identified Spyware instances.

False Negative (FN): Number of wrongly identified benign instances.

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

This section presents the results of the research performed for the purpose of designing an optimize Support Vector Machine classification algorithm for the classification of spyware to achieve better performance metrics.

4.1 Optimized SVM Parameters

GridSearch optimization algorithms enhanced the parameters of SVM after a complex computational operation was carried out on the defined parameter ranges of gamma (γ) and cost (C) functions of SVM, Table 4.1 presents the obtained optimized parameters of both gamma (γ) and cost (C) based on the defined ranged as well as cross validation (CV) values. Table 4.2 depicts the default SVM parameters.

Table 4.1 **Optimized SVM Parameter Gamma (γ) and Cost (C)**

Cross Validation (CV)	Gamma (γ)	Cost (C)	Boundary
5	0.1	1000	$1e^{-05} - 1e^{05}$
10	0.1	1000	$1e^{-05} - 1e^{05}$

Table 4.2 **Default SVM Parameter Gamma (γ) and Cost (C)**

Gamma (γ)	Cost (C)
0.0	1

4.2 Result for SOS Based Feature Selection

Literatures have indicated that existing models for classification of spyware suffers a setback in terms of evaluation of accuracy metric; based on the fact of existing nature of irrelevant features in spyware dataset which can serve as impediment to better performance as it regards to classification algorithm. Feature selection is a critical aspect of data

preprocessing stage that tends to reduce redundancy in features of dataset, while ensuring that only the most relevant features are selected for model classification processes.

To support the optimization of SVM machine learning for spyware classification, SOS algorithm was implemented for feature selection through the three basic steps of mutualism, commensalism and parasitism, a population size of 15, 20, 25, 30 and 50 was define respectively and 10, 20, 30 and 50 iteration was employed, a total of 186 optimal feature was selected out of 189 features.

4.3 Spyware Classification Performance Based on Default and Optimized SVM Parameter

This section introduces the performance of spyware classification based on default SVM parameter and optimized SVM parameter with feature selection and without feature selection

4.3.1 Performance of spyware classification based on default SVM parameter

The default SVM parameters as reflected in Table 4.2 was used in the classification of spyware based on the balanced spyware data without feature selection of 189 and the balanced optimal feature spyware data generated by SOS which is 186 optimal features subset.

Figure 4.1, depicts the representation of the result obtain from the research. The following performance metrics was obtained; an accuracy of 0.776 and 0.783 was achieved using the default SVM and optimal parameters of SVM(GridSearch) as represented in table 4.2 and 4.1 respectively, False Positive Rate (FPR) of 0.234 and 0.227 for balanced spyware features without feature selection and balanced optimal spyware feature with feature selection, 0.776, 0.783 True Positive Rate (TPR) for balanced spyware features without

feature selection and balanced optimal spyware feature with feature selection respectively, 0.776, 0.787 recall respectively for balanced spyware feature without feature selection and balanced optimal spyware feature with feature selection .

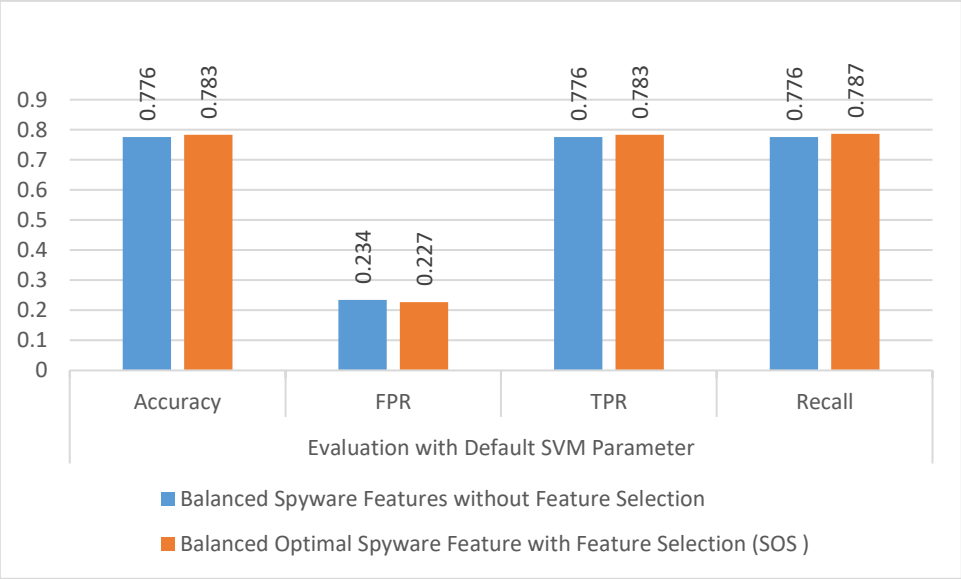

Figure 4.1 **Spyware Classification (Balanced Spyware Features without Feature Selection and Balanced Optimal Spyware Features with Feature Selection) with Default SVM Parameter**

4.3.2 Performance of spyware classification based on optimized SVM parameter

The Figure 4.2, represents the performance evaluation result obtained from the classification of spyware through employing the optimized SVM parameters depicted in table 4.1. An accuracy of 0.967, 0.974 was obtained for balanced spyware features without feature selection and balanced optimal spyware features with feature selection respectively, 0.0831, 0.023 FPR was obtained for balanced spyware features without feature selection and balanced optimal spyware features with feature selection respectively, 0.967, 0.974 TPR was equally obtained for balanced spyware features without feature selection and balanced optimal spyware features with feature selection respectively, while recall of 0.966,

0.974 was recorded for balanced spyware features without feature selection and balanced optimal spyware features with feature selection respectively.

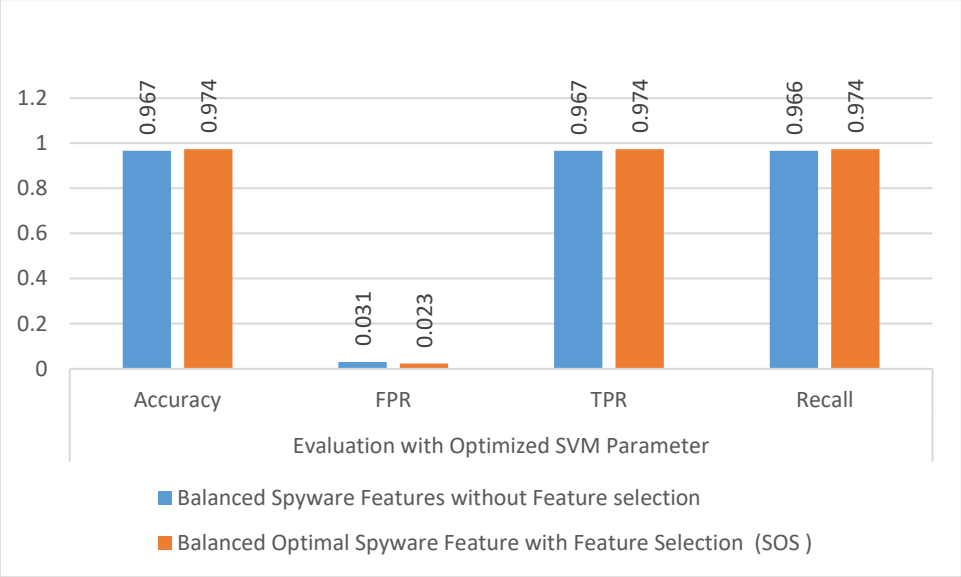

Figure 4.2 **Spyware Classification (Balanced Spyware Features without Feature Selection and Balanced Optimal Spyware Features with Feature Selection) with Default SVM Parameter**

In other to establish the optimized SVM parameter strength in classification of spyware in terms of enhance performance, its performance is compared against the existing spyware machine learning model classifiers based on relevant performance metrics found in literatures, metrics such as accuracy, TPR, FPR, and Recall.

As represented in Table 4.4, the proposed optimized SVM model for spyware classification outperformed other techniques in respect to accuracy, TPR, FPR and Recall. In addition, deducing from Table 4.4, accuracy performance metric is the most used in the area of spyware classification and machine learning based on the facts that it is most common in the baseline literatures. The proposed optimization of SVM model achieved a better accuracy of 97.40%, which is followed by method in the research by Javaheri et al., (2018);

Sai et al., (2019) with a wide margin in accuracy of 95.45%, 93.00% respectively, while Kumar *et al*., (2019) has the least accuracy of 86.93%.

Table 4.4**: Comparison of Proposed Optimize SVM Classifier with Baseline Literatures**

Reference	Approach	Accuracy	TPR	FPR	Sensitivity or Recall
Proposed OptimizedModel	**GridSearchSVM Model**	**97.40**	**97.40**	**2.30**	**97.40**
Kumar *et al*., (2019)	J48 Decision Tree	86.93	86.69	3.30	-
Sai et al., (2019)	MACI- I	95.45	-	-	-
	Logistics Regression	78.78	-	-	91.17
	SVM	77.27	-	-	85.28
	Decision Tree	89.89	-	-	97.05
Javaheri *et al*., (2018)	Linear Regression+JRip+J48	93.00	-	7.00	92.77

N.B: (-) means the metric value is not reported in the reference.

The better performance record by the proposed optimized SVM reflects how correctly the model can classify a spyware attack from benign, a low FPR of 2.30% was record in the proposed optimization of SVM for spyware classification against Javaheri *et al*., (2018) which record a FPR as high as 7.00%, followed by Kumar *et al*., (2019) which recorded 3.30% FPR both of which are on a high side FPR, this proof that the proposed optimized SVM have an efficient capability in classification of spyware from benign

The recall which measures the completeness of the performance of spyware classification recorded 97.40% which is higher than the research by Sai *et al*., (2019) and Javaheri *et al*., (2018) which record 97.05 and 92.77% respectively.

4.3.3 Analysis of Baseline Literature Techniques with Proposed Spyware Dataset

In other to have a balance view and avoid bias in performance evaluation analysis against baseline literature, the proposed spyware dataset employed in this study is used in the model techniques employed by baseline literatures. This is as a result of non-disclosure of the dataset used by the baseline literatures. The following machine learning model was used by the baseline literatures; J48 Decision Tree and Logistic Regression, JRip, the results obtained are represented in table 4.5, JRip, Logistics Regression and J48 Decision Tree recorded an accuracy of 87.36, 68.96, 87.36 respectively, while True Positive Rate 87.74, 69.00, and 87.40 was recorded respectively as well, 80.00, 30.10, and 45.10 False Positive Rate was achieved respectively and Recall recorded 87.40, 69.00, 87.40 respectively, throughout the recorded performance by JRip, Logistics Regression and J48 Decision Tree machine learning techniques by baseline literatures. It is clear that none of the performance metric supersedes the proposed optimized Support Vector Machine, proving that the optimization of Support Vector Machine Classifier actually outperformed the baseline literatures by far when compared in terms of performance metric deployed on relevant literatures.

Table 4.5 **Performance of Baseline Techniques with Proposed Spyware Dataset**

Baseline Techniques	Accuracy	TPR	FPR	Sensitivity or Recall
JRip	87.36	87.74	80.00	87.40
Logistics Regression	68.96	69.00	30.10	69.00
J48	87.36	87.40	45.10	87.40

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Spyware is a wide spreading stealthy threat to computing environment and devices. The large collect of data from the computing devices may be a challenging factor in terms of redundancy and unbalances of data, which in turn have a great impact and setback to performance analysis evaluation in terms of classification of spyware by machine learning model, not ruling out the tuning of classification model for better performance. Hence, this research proposed an enhanced SVM classifier for spyware classification based on Gridsearch optimization algorithm, tuning and determining the optimal SVM parameter, further supported by optimal feature selection based on SOS algorithm in order to enhance spyware classification with a better accuracy and low FPR which is a setback in existing classification model for spyware classification.

To achieve an optimal SVM model for spyware classification, a Gridsearch optimization based algorithm was implemented in identifying optimal parameters based on defined range of computation for the parameters gamma **(γ)** and cost (C) function respectively for SVM, while a cross validation was deployed for validate the trained spyware features while obtaining optimal accuracy which defines the optimal SVM parameters of gamma **(γ)** and cost (C), furthermore, SOS based metaheuristic algorithm selected optimal spyware features used for classification.

The recorded research proves that optimal SVM classification algorithm for spyware classification outperformed the traditional SVM classifier and most of the baseline

techniques for spyware classification. The supremacy of the performance of the proposed optimal SVM classification algorithm for spyware classification is based on the fact that parameters was tuned, spyware dataset was balanced based on SMOTE, and optimal spyware features was obtained based on SOS metaheuristic algorithm which climax the performance of SVM classification accuracy.

Based on the finding of this research, the research establishes that optimal SVM classification algorithm for spyware classification is an effective model for spyware classification while feature selection-based SOS metaheuristic algorithm serves as better algorithm aids in mitigating the challenges of data overfitting by dropping redundant spyware features.

5.2 Contributions to Knowledge

This research contributed the following to knowledge;

i. Design of SOS metaheuristic algorithm for spyware feature selection which obtain optimal features
ii. Optimized SVM classifier, that demonstrated a more superior classification performance over default SVM parameter which recorded **97.40%** each for accuracy, TPR and recall respectively, and a FPR of **2.30%,** against the baseline literature by Javaheri *et al*., (2018); Kumar *et al*., (2019); Sai *et al*., (2019) with the accuracy of 93.00%, 86.93 and 95.45% respectively, FPR of 3.30% and 7.00% by Kumar *et al*., (2019) and Javaheri *et al*., (2018) respectively.

5.3 Recommendations

From the finding of this research, the following recommendation were made;

i. Optimization of Support Vector Machine with other existing optimization algorithms to achieve a higher accuracy value is still attainable.

ii. Exploring other feature selection algorithms as well as algorithms that can balance unbalanced data set to resolve the challenge of overfitting.

iii. Hybridizing multiple algorithm classifier machine learning model to enhance performance of same in spyware classification.

5.4 Published Article

N. N. Gana and S. M. Abdulhamid, "Machine Learning Classification Algorithms for Phishing Detection: A Comparative Appraisal and Analysis," *2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)*, Zaria, Nigeria, 2019, pp. 1-8.

REFERENCES

Abdullahi, M., Abdulhamid, M., Dishing, S. I., & Usman, M. J. (2019). Variable neighborhood search-based symbiotic organisms search algorithm for energy-efficient scheduling of virtual machine in cloud data center. *Advances on Computational Intelligence in Energy, Green Energy and Technology*, 77-97

Abdullahi, M., Ngadi, A., & Abdulhamid, M. (2015). Symbiotic organism search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems*, 15, 167-259, doi:10.1016/j.future.2015.08.006

Akbarifard, S., & Radmanesh, F. (2018). Predicting sea wave height using symbiotic organisms search algorithm. *Ocean Engineering*, 167, 348–356, doi:10.1016/j.oceaneng.2018.04.092

Alwan, H. B., & Ku-Mahamud, K. R. (2017). Integrated incremental ant colony optimization , ant colony optimization mixed varable-support vector machine algorithm. *International Journal of Computer and Information Engineering*, 11, 1309-1313, doi:1307-6892/10008369

Arya, M., & Bedi, C. S. S. (2018). Survey on support vector machine and their application in image classification. *International Journal of Information Technology*, 10, 1-11, doi:10.1007/s41870-017-0080-1

Bahraminikoo, P., Samiei, M., & Babu, G. P. (2012). Utilization data mining to detect spyware. *Journal of Computer Engineering*, *4*, 1–4. Retrieved from http://www.iosrjournals.org/iosr-jce/papers/Vol4-issue3/A0430104.pdf

Boldt, M., Jacobsson, A., Lavesson, N., & Davidsson, P. (2008). Automated spyware detection using end user license agreements. *International Conference on Information Security and Assurance* (pp. 445-452). Korea:IEEE

Boswell, D. (2002). *Introduction to support vector machines.* California, San Diego: Departement of Computer Science and Engineering University of California Press

Bustamante, F., Fuertes, W., Tulkeredis, T., & Ron, M. (2018). Situational status of global cybersecurity and cyber defense according to global indicators. Adaptation of a model for Ecuador. *International Conference of Research Applied to Defense and Security* (pp. 12-26). Ecuador:Springer

Çelik, E., & Öztürk, N. (2018). First application of symbiotic organisms search algorithm to off-line optimization of PI parameters for DSP-based DC motor drives. *Neural Computing and Applications*, *30*(5), 1689-1699.

Chandra, M. A., & Bedi, S. S. (2018). Survey on support vector mechine and their application in image classification. *International Journal of Information Technology*, 18, 1-11, doi:10.1007/S41870-017-0080-1

Cheng, M. Y., & Prayogo, D. (2014). Symbiotic organisms search: a new metaheuristic optimization algorithm. *Computers & Structures*, *139*, 98-112. doi:10.1016/j.compstruc.2014.03.007

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321-357. doi:https://doi.org/10.1613/jair.953

Demidova, L. (2016). Big data classification using the support vector mechine classifiers with the modified particle swarm optimization and the support vector mechine Ensembles. *International Journal of Advanced Computer Science and Applications*, *7*(5), 294–312. doi: 10.14569/IJACSA.2016.070541

Guha, D., Roy, P. K., & Banerjee, S. (2018). Symbiotic organism search algorithm applied to load frequency control of multi-area power system. *Energy Systems*, *9*(2), 439-468. doi:10.1007/s12667-017-0232-1

Huang, C. L., & Wang, C. J. (2006). A gentic algorithm-based feature selection and parameters optimization for support vector machines. *Expert Systems With Applications*, *31*(2), 231-240.doi:10.1016/j.eswa.2005.09.024

Jaffel, Z., & Farah, M. (2018). A symbiotic organisms search algorithm for feature selection in satellite image classification. *International Conference on Advanced Technologies for Signal and Image Processing* (pp. 1-5).Tunisia:IEEE

Javaheri, D., Hosseinzadeh, M., & Rahmani, A. M. (2018). Detection and elimination of spyware and ransomware by intercepting kernel-level system routines. *IEEE Access*, *6*, 78321-78332. doi: 10.1109/ACCESS.2018.2884964

Kremic, E., & Subasi, A. (2016). Performance of random forest and support vector mechine in face recognition. *Internatioal Arab Journal of Information Technology*, *13*(2), 287-293. Retrieved from http://iajit.org/PDF/Vol.13,%20No.2/8468.pdf

Kulkarni, S. R., & Harman, G. (2011). Statistical learning theory: a tutorial. *Wiley Interdisciplinary Reviews: Computational Statistics*, *3*(6), 543-556. doi: https://doi.org/10.1002/wics.179

Kumar, D. A., Kapat, S. K., Das, S. K., & Tripathy, S. N. (2019). Classification of spyware affected files using data mining techniques. *International Journal of Recent Technology and Engineering*, *8*(2), 462–466. doi: https://doi.org/10.35940/ijrte.B1088.0782S619

Lin, J., & Zhang, J. (2013). A fast parameters selection method of support vector machine based on coarse grid search and pattern search. *Fourth Global Congress on Intelligent Systems.* (pp. 77-81). Hong Kong, China:IEEE

Liao, T. W., & Kuo, R. J. (2018). Five discrete symbiotic organisms search algorithms for simultaneous optimization of feature subset and neighborhood size of k-nearest neighbor classification models. *Applied Soft Computing*, *64*, 581-595. doi:10.1016/j.asoc.2017.12.039

Liu, R., Liu, E., Yang, J., Li, M., & Wang, F. (2006). Optimizing the hyper-parameters for support vector mechine by combining evolution strategies with a grid search. *Intelligent Control and Automation*, 344, 712-721, doi: https://doi.org/10.1007/978-3-540-37256-1_87

Lv, D., Ma, Z., Yang, S., Li, X., Ma, Z., & Jiang, F. (2018). The application of smote algorithm for unbalanced data. *Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality* (pp. 10-13). New York, USA:ACM

Maigida, A. M., Olalere, M., & Ismaila, I. (2019). An intelligent crypto-locker ransomware detection technique using support vector machine classification and grey wolf optimization algorithms. *i-Manager's Journal on Software Engineering*, *13*(3), 15-23, doi: 10.26634/jse.13.3.15685

Nanda, S. J., & Jonwal, N. (2017). Robust nonlinear channel equalization using wavelet neural network trained by symbiotic organism search algorithm. *Applied Soft Computing*, *57*, 197-209, doi: https://doi.org/10.1016/j.asoc.2017.03.029

Nawfal, T. O., & Wesam, B. (2016). Review of data mining techniques for malicious detection. *Research Journal of Applied Sciences*, *11*(10), 942–947, doi: 10.36478/rjasci.2016.942.947

Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques*. Berlin:Springer Science & Business Media.

Pal, S. S., Samui, S., & Kar, S. (2020). A new technique for time series forecasting by using symbiotic organisms search. *Neural Computing and Applications*, *32*(7), 2365-2381, https://doi.org/10.1007/s00521-019-04134-8

Patel, D. H. (2015). Spyware triggering system by particular string value. *International Journal of Engineering Research and Development*, *11*(09), 2278–67. doi:https://doi.org/10.13140/RG.2.2.28265.95847

Sai, K. N., Thanudas, B., Sreelal, S., Chakraborty, A., & Manoj, B. S. (2019). A malware detection technique using memory management application programing interface call mining. *Region 10 Conference* (pp. 527-532). Kochi, India:IEEE

Shahzad, R. K., Lavesson, N., & Johnson, H. (2011, August). Accurate adware detection using opcode sequence extraction. *Sixth International Conference on Availability, Reliability and Security* (pp. 189-195). Vienna, Austria:IEEE

Shahzad, R. K., Haider, S. I., & Lavesson, N. (2010). Detection of spyware by mining executable files. In *2010 International Conference on Availability, Reliability and Security* (pp. 295-302). Krakow, Poland:IEEE

Sheta, M. A., Zaki, M., & El Hadad, K. A. E. S. (2016). Anti-spyware security design patterns. *Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control* (pp. 465-470). Harbin, China:IEEE

Sheta, M. A., Zaki, M., & El Hadad, K. A. E. S. (2016). Spyware detection by extracting and selecting features in executable files. *International Conference on Electrical Engineering* (pp. 1-20). Egypt : Military Technical College

Stafford, T. F., Urbaczewski, A., & Stafford, T. F. (2004). Spyware : The ghost in the machine. *Communications of the Association for Information Systems*, 14(1), 49 doi:https://doi.org/10.17705/1CAIS.01415

Sukhanov, S., Merentitis, A., Debes, C., Hahn, J., & Zoubir, A. M. (2018). Combining support vector machines for classification on class imbalanced data. *Statistical Signal Processing Workshop* (pp. 90-94). Freiburg Breisgau, Germany: IEEE

Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). Support vector machine parameter optimization using grid search and genetic algorithm to improve classification performance. *Telkomnika*, *14*(4), 1502-1509, doi: 10.12928/telkomnika.v14i4.3956

Takeuchi, Y., Sakai, K., & Fukumoto, S. (2018). Detecting ransomware using support vector machines. *Proceedings of the 47th International Conference on Parallel Processing Companion* (pp. 1-6). Eugene, USA: ACM

Tripathy, Satya & Kapat, Sisira & Das, Susanta & Panda, Binayak. (2016). A Spyware detection system with a comparative study of spywares using classification rule mining. *International Journal of Scientific and Engineering Research*, 7, 179-184, doi: https://doi.org/10.1145/3229710.3229726

Wang, T. Y., Horng, S. J., Su, M. Y., Wu, C. H., Wang, P. C., & Su, W. Z. (2006). A surveillance spyware detection system based on data mining methods. *International Conference on Evolutionary Computation*, (pp. 3236-3241). Vancouver, BC, Canada: IEEE

Wang, X., & Chen, J. (2009). Interests-based spyware detection. *International Forum on Computer Science-Technology and Applications* (Vol. 2, pp. 175-178). Chongqing, China: IEEE

Wazid, M., Sharma, R., Katal, A., Goudar, R. H., Bhakuni, P., & Tyagi, A. (2013). Implementation and embellishment of prevention of keylogger spyware attacks. *International Symposium on Security in Computing and Communication* (pp. 262-271). Berlin: Springer

Wu, M. W., Wang, Y. M., Kuo, S. Y., & Huang, Y. (2007). Self-healing spyware: detection, and remediation. *IEEE Transactions on Reliability*, *56*(4), 588-596. doi: 10.1109/TR.2007.909755

Xu, H., Zhou, Y., Gao, C., Kang, Y., & Lyu, M. R. (2015). SpyAware: Investigating the privacy leakage signatures in app execution traces. *26th International Symposium on Software Reliability Engineering* (pp. 348-358). Gaithersbury, USA:IEEE

Yuanyuan, S., Yongming, W., Lili, G., Zhongsong, M., & Shan, J. (2017). The comparison of optimizing support vector machine by genetic algorithm and grid search. *International Conference on Electronic Measurement & Instruments* (pp. 354-360). Yangzhou, China: IEEE

Zhang, X., Wang, W., Zheng, X., Ma, Y., Wei, Y., Li, M., & Zhang, Y. (2019). A Clutter suppression method based on self organising map-sythetic minority oversampling technique random forest. *Radar Conference* (pp. 1-4). Boston, MA, USA:IEEE

Zhou, Y., Wu, H., Luo, Q., & Abdel-Baset, M. (2019). Automatic data clustering using nature-inspired symbiotic organism search algorithm. *Knowledge-Based Systems*, *163*, 546-557, doi: https://doi.org/10.1016/j.knosys.2018.09.013

APPENDIX A

Table 2.2 **Summary of Reviewed Literature**

References	Problem	Method	Parameter	Result	Limitation(s)
(Kumar *et al.*, 2019)	Spyware classification	J48 Decision Tree	Accuracy, True Positive Rate, False Positive rate	86.93%, 86.69% and 3.3% respectively	Imbalanced dataset, while focus was on API system call
(Sai *et al.*, 2019)	API call detection	MACA-I, Linear Regression (LR), SVM, Decision Tree(DT)	Accuracy, Sensitivity	95.45%, 78.78%, 77.27%, 89.89% respectively, while Sensitivity for LR, SVM, and DT are 91.17%, 85.28%, and 97.05% respectively	A low performance in accuracy and research was major on API call
(Javaheri *et al.*, 2018)	Detection and elimination of Spyware and Ransom ware	Intercepting kernel level routine using Linear Regression, JRip and J48 decision Tree	Accuracy and False Positive Rate (FPR)	93% and 7% respectively	Low accuracy rate and high FPR
(Tripathy *et al.*, 2016)	Spyware Detection System	Data mining (ZeroR, Decision Tree, JRip, J48 and Naïve Bayes algorithms)	Accuracy, kappa statistics, ROC	95.77%, 97.79%, 97.69%, 97.83%, 96.36% accuracy respectively, 0, 0.723, 0.7074, 0.7307, 0.379 respectively for kappa statistics, 0.485, 0.9356, 0.971, 0.932, 0.9647 respectively	Lack strong capability to detect unknown spyware

				for ROC area	
(Sheta *et al*., 2016)	Building a data mining based anti-spyware	CFFBE, J48	Accuracy, TPR, FPR, AUC	99.98% and 99.91% for new spyware and unknown spyware respectively, 99.9%, 0.3%, 0.99 respectively	
(Xu *et al*., 2015)	Determining the privacy leakage signatures in app execution traces	SVM, Naïve Bayes	Accuracy	67.4% and 64.2% respectively	Focus was based on Android OS smartphone with a very high undisclosed FPR
(Wazid *et al*., 2013)	Prevention mechanism against key logger	Key logger attack, honeypot based detection and prevention of key logger and generation of spyware attack		Tackle key logger attack	Focus based on key logger alone
(Shahzad, 2011)	Adware detection using opcode sequence extraction	Naïve Bayes, SVM, IBk, J48, JRip	AUC, FNR, FAR	0.838, 0.939, 0.949, 0.885 respectively for AUC	Focus was based on adware and little size of dataset was use in evaluation, unclear details about FNR and FAR
(Shahzad *et al*., 2010)	Spyware detection through mining executable files	ZeroR, Naïve Bayes, SVM, **J48**, Random Forest, JRip	Accuracy and Area Under ROC Curve (AUC)	86.92%, 89.80%, 89.65%, **90.5%**, 89.48%, 89.45% respectively for accuracy and 0.50, 0.62, 0.71, 0.65, 0.83 and 0.66 respectively for AUC	Study was designated on Windows executable and indicates a low accuracy rate
(Wu *et al*., 2007)	Detection and Remediation of Self-Healing Spyware	Stateful Threat Aware Removal System (STARS)		Capability to detect self-modifying spywares	Lack the capacity to detect hidden registry entries

(Wang *et al.*, 2006)	detection of Surveillance spyware	SVM	FP,TP, Accuracy	0.68%,95.33 % and 97.9% respectively	High FPR
(Wang and Chen, 2009)	Spyware detection based on interest	Abstract characterization through data mining		Detection of unknown and known spyware	Theoretically based result without implementation
(Bahraminikoo *et al.*, 2012)	Spyware detection	Data mining(*Breadth-First Search (BFS)*), Random Forest, Naïve Bayes, and Support Vector Machine (SVM)	Accuracy and FPR	90.5%, 89.49%, 88.21% , 88.02% for accuracy respectively, while FPR is 0.731, 0.665, 0.730 and 0.655 respectively	High false positive rate and low accuracy as well as unclear details about training and testing dataset
(Boldt *et al.*, 2010)	Spyware Classification based on EULA	Application of SVM, Muti-nominal Naïve Bayes and Voter Perception	AUC, Accuracy, False Positive Rate, True Positive Rate	SVM: AUC=0.84 Accuracy=83 .53 FPR=0.11 TPR=0.78 Muti-nominal Naïve Bayes Voter: AUC=0.80 Accuracy=87 .94 FPR=0.12 TPR=0.88 Perception: AUC=0.87 Accuracy=81 .47 FPR=0.22 TPR=0.85	High FPR and Low Accuracy