TACKLING THE MENACE OF CURSE OF DIMENSIONALITY IN INTRUSION DETECTION SYSTEMS: MEMBRANE COMPUTING APPROACH

Rufai K. I.
Computer Science Department,
College of Science & Information Technology,
Ijagun, Ogun-State,
Nigeria.

E-mail: rufaiki@tasued.edu.ng

Ravie C. M.
Centre for Software Technology & Management,
Faculty of Information Science & Technology,
National University Malaysia (UKM),
Malaysia.

E-mail: ravie@ukm.edu.my

Usman O. L.
Computer Science Department,
College of Science & Information Technology,
Ijagun, Ogun-State,
Nigeria.

E-mail: usmanol@tasued.edu.ng

Abstract

That our Network and Information Systems (NISs) are still being hacked and compromised is not unconnected to the simple fact that a completely perfect system is yet unborn! The introduction of Intrusion Detection System (IDS) was to stem the tide of the heinous activities being perpetuated by intruders on the NISs. However, researches have shown that one of the banes of IDS is the poor handling of high dimensional features in traffic data leading to what is commonly referred to as 'Curse of dimensionality'. When this is not professionally attended to, an IDS then experiences general performance problems including poor quality (high false alarm and low detection rates) and inefficiency (low processing speed and throughput). Meanwhile, Membrane Computing (MC) which is an emerging field in Computer Science is a versatile, non-deterministic and maximally parallel computing device. The benefits of MC have been (and are still being) harnessed in various fields such as Medicine, Image processing, Linguistics and Optimization. Therefore, the aim of this article is to take a deeper look at the concept of feature selection (FS) in IDS and to introduce a MC-inspired algorithm towards ameliorating the challenge of 'Curse of dimensionality'.

Key words: Membrane computing, P system, Curse of dimensionality, Feature selection

INTRODUCTION

Intrusion detection system (IDS), is indisputably an arm of network/computer security which has been acknowledged as *sine qua non* in the area of the protection of our NISs. Therefore, a variety of techniques and models have been introduced in the past in order to develop efficient and reliable IDSs which have very high quality.

Meanwhile, in line with the views of Toosi & Kahani (2007), Zuech, et al. (2015) and Acker (2015) every detection technique in a network environment, is basically confronted with the following major challenges among others:

(i) Big data problem resulting from continuous access to internet leading to high-bandwidth networks.

(ii) Data to be processed are mostly unevenly distributed.

Expectedly, a typical IDS should perform three fundamental, but major roles which are; (a) monitoring of a network environment, (b) determining which traffic is intrusive or not and (c) reporting same to a security station (Cavusoglu, et al. 2005; Anand & Patel 2012). So, since its evolution, there has been constant need to expand and upgrade the different types of available IDSs mainly due to:

(i) change in the sophistication in the methods adopted by mischief makers who find solace in committing the heinous crime of intrusion.

(ii) the need to be ahead of the attackers and intruders in their game.

(iii) increase in the bandwidth resulting in higher processing/detection power.

Sequel to the above, different approaches have been applied to improve IDS on various platforms such as sequential machine, FPGA, Multi-core and GPU. Of Primary concern in almost all these approaches are the issues of selecting appropriate features, determining the precise boundaries between intrusive and anomaly behaviour, effective and efficient processing of huge traffics, among others. Going by the existing literature, these problems remain partially solved and hence require deeper investigation and application of novel approaches like P system.

The subject of feature selection in intrusion detection and indeed some other engineering field has attracted attention because of its problems. It is generally believed that the existence of useless and redundant feature in network data unnecessarily prolongs detection time and thereby negatively affecting the intrusion detection accuracy (Mukkamala & Sung 2003; Kaur &Kaur 2014; Ayman et al. 2014). Succinctly, according to Sung & Mukkamala (2005), problems of feature selection include:

(a) Presence of high number of input variables $z = \{z_1, z_2, \mathrm{K}, z_n\}$ where some elements of z are important and others are unimportant.

(b) Absence of mathematical formula to define input-output relationship in precise terms. This is not unconnected to the non-availability of acceptable analytical model.

So, because of these identified problems, researchers in the field of information security have been adopting empirical methods for FS by analyzing the dependency and correlation between two or more input variables. However, this approach, apart from being cost ineffective, it may be time-consuming and may be entirely unattainable if the input variables are very large because 2^n experiments may be required.

Consequent on the above therefore, and based on the fact that dimensionality reduction is now being considered very crucial in data mining techniques for intrusion detection, this paper discusses the concept of feature selection in IDS vis-à-vis its types and dimensionality problem. It further proposes a Membrane Computing-inspired algorithm as a panacea.

feature selection in inTRUSION DETECTION SYSTEM

The primary goal of any IDS is to flag intrusive packet in a stream of traffic promptly with high level of accuracy. However, this goal is being hampered by high dimensionality, i.e. the presence of redundant and irrelevant features in the traffic to be analyzed. This problem does not only affect the classification accuracy and detection rate adversely, but it invariably engenders increase in false positives/negatives. It implies then that to achieve good classification accuracy in IDS, there must be means to select relevant/appropriate features and discarding the irrelevant/redundant ones. Many approaches which could be grouped under feature ranking, feature selection and gradual feature removal alongside some other learning algorithms have been used to solve the highlighted problem to some extent and also for achieving efficiency in intrusion detection. However, in accomplishing this, a network administrator should be mindful of appropriate method to be deployed so as to minimize the cardinality of the set of selected features, without compromising the potential indicators of intrusive traffics (Nguyen et al. 2010; Sung & Mukkamala 2005; Hofmann et al. 2004).

FS is the process of identifying and selecting important feature subset in a set of features based on pre-defined criteria for the primary purposes of training and classification. Generally therefore, it may be said to be a technique of producing a reduced subset feature x_m from a set of complex and highly populated features x_n in a dataset for a classifier without losing the integrity of the dataset. This implies that $x_m < x_n$.

In intrusion detection systems generally, FS solves the following problems, among others;

(i) It allows for concise and precise description of dataset thereby eliminating the problem of confusion which the system may have suffered due to large and uncoordinated data.
(ii) It considerably reduces the training time thereby allowing the classifier to detect attacks quickly.
(iii) It invariably enhances the classification accuracy of the system since noisy features must have been discarded.

Feature selection approaches

Although many types of FS models exist (such as Correlation-Feature-Selection (CFS), Support Vector Machine (SVM), minimal-Redundancy-Maximal-Relevance (mRMR) measure, Classification & Regression Trees (CART) algorithms, Generic-Feature-Selection (*GeFS*) method etc.), but they are broadly grouped into: ***filter*** and ***wrapper*** approaches.

Wrapper feature selection approach: The wrapper model is accomplished through the performance of learning algorithm. Here, the selection of feature subset is totally dependent on the classifier deployed. It is however believed that time expended and computational resources usage in wrapper approach are high.

Filter feature selection approach: Unlike the wrapper method, filter approach solely depends on statistical information such as distance, dependency and consistency measures to select relevant features. Most often, filter model is preferred by many because of its computational efficiency. Filter method is atimes used in fields whereby thousands of cues would have to be recognized because it has the ability to analyze intrinsic properties of data.

Curse of dimensionality

One of the banes of IDS is the poor handling of high dimensional features in traffic data leading to what is commonly referred to as *'Curse of dimensionality'*

'Curse of dimensionality' refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of features) that do not occur in low-dimensional settings such as the three-dimensional physical space of

everyday experience. Although, network traffics consist of high-dimensional data, but a good IDS should be efficient in tracking and analyzing such data at very high speed. It must also be noted that certain learning algorithms perform poorly with high-dimensional data.

CONCEPT OF MEMBRANE COMPUTING

The idea and possibility of computing with membranes was first conceived by Gheorge Păun over a decade ago (Păun 2000). MC, otherwise called *P* system is a non-deterministic, distributed and maximally parallel computing model. (Păun & Mario 2006). This model has successfully been applied in various fields including, image segmentation, fault diagnosis, NP-complete combinatorial optimization problem, cryptography system and digital filtering. There are three basic variants of MC, namely: Tissue-like *P* system, Cell-like *P* system and Neural-like *P* system. Whichever form, MC model has three main constituents. These are: membrane structure, multiset of objects and rules.

Membrane Structure

In MC configuration, the membrane structure has been considered as a very essential constituent. A membrane structure therefore may either be in hierarchical form or in form of network layout with channels of communication among the membranes. A membrane structural arrangement which has no other membrane within it is called *elementary* or *primary* membrane. On the other hand, a membrane hosting some other membrane(s) within it and which serves as the point of intersection between the membranes and the environment is termed *skin* membrane. Regions/compartments represent the space within the membrane where computation takes place.

Multiset of objects

Multiset of objects are chemicals inside liquid of compartmentalized section of a typical membrane, which to some extent, signify the multiple (or number of) copies of such objects that could be found in that region. Multiset of objects are parts of the main elements which are situated within membranes. When multiset of objects are combined with membrane structure, the transition P system's configuration is obtained at any given time during computation. It is on this multiset of objects that rules are applied during computation.

Also, these objects may as well be found in the environment outside the compartment of membrane after computation. Multiset of objects could be denoted using letters or strings if they have atomic or complex structures respectively.

Rules

Rules are ordinarily applied non-deterministically without any special consideration on the membrane objects. They could also be applied in a maximally parallel manner where all applicable rules are deployed as such at every step. Basically, all rules in a membrane system can be grouped into two; rewriting and communication rules.

INTRODUCING THE MS-B ALGORITHM

The hybridized Membrane System-Bee Algorithm (MS-BA) herein proposed improves the existing Bee Algorithm in solving *Curse of dimensionality* problem in IDS.

MS-BA consists of the following components:

(i) A membrane structure of the form: $[[]_1\ []_2\ []_3\ ]_0$; where *1,2* and *3* are elementary membranes situated within *0* which is the skin membrane.

(ii) An alphabet that consists of all possible individuals in the solution space $B_i^{j,k} \in [B_U]$ where B_U is the universal set of all bees, $i = 1,...,n$; $j = 1,...,BSS_k$; $k = 1,2,3$;which is the number of elementary membranes, n is the dimension of individual bees; BSS_k is the number of bees in each membrane.

(iii) A set of terminal symbols, where $B_{fittest}$ that represents the bee with the highest fitness function.

(iv) $i_0 = 0$; which denotes the output membrane through which the results of *maximum external iterations* ($Itr_{ext}^{\max}$) is released into the environment.

(v) Initial multisets $\omega_0 = \lambda$ or empty, $\omega_1 = B^{1,1}\,B^{2,1}...B^{N_1,1}$, $\omega_2 = B^{1,2}\,B^{2,2}...B^{N_2,2}$, . . . , $\omega_m = B^{1,m}\,B^{2,m}...B^{N_3 m,m}$; where $B^{i,j}$ is the i-th Bee in the j-th membrane. Population number in membrane i is N_i, with N, total number of bees: $\sum_{i=1}^{m} N_i = N$.

(vi) The *evolution* and *communication* rules.

Membrane structure of MS-BA feature subset selection process

The membrane configuration of the MS-BA feature subset selection process as depicted in Figure 1 is made up of three elementary membranes $[]_1, []_2$ and $[]_3$ and a skin membrane $[]_0$

Within each elementary membrane are; (i) evolution rules which oversee the transformation of the features and ensure that bees with 'good' fitness are generated (ii) communication rules which are responsible for transportation, i.e these rules send the 'good' bees (features) outwardly to the skin membrane. (iii) number of bees (features) N_i , which in this instance are 41 (based on the features of KDD cup dataset). So, depending on the number of available cores on a computer machine, the membranes are built to be worked upon by each core.

However, while the improvement iterations inside the primary membrane is known as internal maximum iterations ($Itr_{\text{int}}^{\max}$), that of the skin membrane is referred to as maximum external iterations ($Itr_{ext}^{\max}$). Ultimately, after the completion of both internal and external improvements, the bees (features) with highest fitness are released to the environment using the appropriate communication rule via the skin membrane (which in this instance, serves as the output membrane ($i_{out} = 0$)).

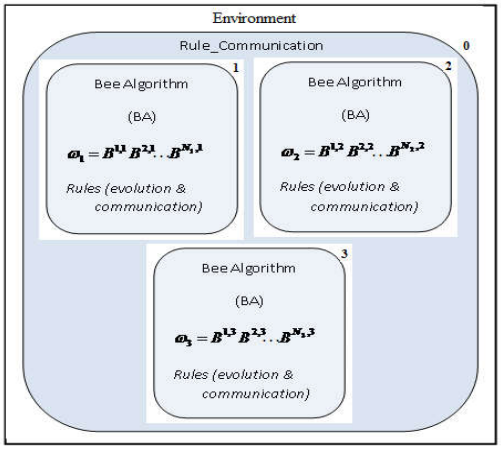

Figure 1: Membrane structure of MS-BA feature subset selection.

Definition of MS-BA's rules and their applications

Basically, five distinct rules of evolution (for transformation) and communication (for transportation) types were applied in the membrane systems used in this work. It must however be noted these rules were executed in all membranes concurrently. The following conditions suffice for the rules: $i = 1,...,n; j = 1,...,BSS_k; k = 1,...,m$; and BSS_k is the size of scout bees in *k-th* membrane. The *i-th* path of the *j-th* bee in$BS(B_i^{j,k})$. While x_i and y_i are initialized and new objects respectively, (B^{bfit}) denotes bee with better fitness.

$$\textbf{\textit{Rule}}_{\textbf{\textit{one}}} = r_{1,(i,j,k)} : x_i B_i^{j,k} \rightarrow x_i\, y_i B_i^{j,k} B_i^{new,k} \quad (1.1)$$

When the rules of type $Rule_{one}$; ($r_{1,(i,j,k)}$) are applied, they are meant to choose the patch visited by new bees from the population of scout bees. With this application and selection of ($B_i^{j,k}$), so a new path $B_i^{new,k}$ with an object y_i evolve.

$$\textbf{\textit{Rule}}_{\textbf{\textit{two}}} = r_{2,(i,k)} : x_i B_i^{rand,k} \rightarrow x_i\, B_i^{new,k} \quad (1.2)$$

With the type of $Rule_{two}$; Rules ($r_{2,(i,k)}$)assign the randomly generating values for patches ($B_i^{rand,k}$) from the possible range of values to $i-th$ patch of a new bee.

$$\textbf{\textit{Rule}}_{\textbf{\textit{three}}} = r_{3,(i,k)} : y_i B_i^{new,k} \rightarrow B_i^{new(adjust),k} \quad (1.3)$$

With $Rule_{three}$ $r_{3,(i,k)}$, the $i-th$ patch of a new bee ($B_i^{new(adjust),k}$) is adjusted using the object in r_2.

$$\textbf{\textit{Rule}}_{\textbf{\textit{four}}} = r_{5,k} : [B^{bfit}]_k \rightarrow []_k B^{bfit}; k = 1,...,m \quad (1.4)$$

This rule guarantees that bees with better fitness inside each of the elementary membranes are transported to the skin membrane.

$$\textbf{\textit{Rule}}_{\textbf{\textit{five}}} = r_5 : [B^{fittest}]_0 \rightarrow []_0 B^{fittest} \quad (1.5)$$

With $Rule_{five}$, bee with highest fitness is sent outwardly from the skin membrane (i.e. membrane *0*) to the environment.

Pseudo code for MS-BA

The pseudo code of MS-BA presented afterwards (Figure 2) is subject to the initialization of:

(a) # of bees in the elementary membranes defined as; $N_i = \lambda_i \times N_1; i = 2,...,m$.

(b) maximum number of iteration in elementary membrane (i.e. maximum internal iteration ($Itr_{\text{int}}^{\max}$)) and maximum number of iteration in the skin membrane (i.e. maximum external iteration ($Itr_{ext}^{\max}$)).

(c) structure of membrane *m* by assigning membrane according to available machine cores *p*.

(d) membrane objects.

```
Begin Algorithm
Itr_ext ← 0; Itr_int ← 0
While (Itr_ext < Itr_ext^max)
    While (Itr_int < Itr_int^max)
        Execute Rule_one, Rule_two, Rule_three; // performed on p cores
        Execute Rule_four; // performed on p cores
        Itr_int = Itr_int + 1
    End while
    Execute Rule_one, Rule_two, Rule_three; // performed on master core
    Itr_ext = Itr_ext + 1
End while
Execute Rule_five; // performed on master core
End of the Algorithm
```

Figure 2: Pseudo code for MS-BA

CONCLUSION

In this work, a novel Membrane System-Bee Algorithm was introduced towards the reduction of the negative impact of high dimension features, which has been considered the bane of IDS. Invariably, if our approach is well harnessed, it could help engender high classification accuracy rate in IDS. It is recommended that more research works which improve other learning algorithms be further explored so as to increase attack detection rate as well as decrease false alarm rate of intrusion detection systems.

REFERENCES

Acker, F., 2015: Use of entropy for feature selection with intrusion detection system parameters. Unpublished PhD Thesis. Nova Southeastern University, USA.

Anand, A., & Patel, B. 2012: An Overview on Intrusion Detection System and Types of Attacks It Can Detect Considering Different Protocols. *International Journal of Advanced Research in Computer Science and Software Engineering, IJARCSSE*, *8*, 94-98.

Ayman I. Madbouly, Amr M. Gody & Tamer M. Barakat 2014: Relevant Feature Selection Model Using Data mining for Intrusion detection system. *International Journal of Engineering Trends and Technology (IJETT)*– Vol. 9(10) pp: 501-512.

Cavusoglu, H., Mishra, B., & Raghunathan, S. 2005: The value of intrusion detection systems in information technology security architecture. *Information Systems Research*, *16*(1), 28-46.

Dash S. K., Rawat S., & Pujari, A. K. 2007: Use of dimensionality reduction for intrusion detection. In *Information Systems Security* (pp. 306-320). Springer Berlin Heidelberg.

Hofmann, A., Horeis, T., & Sick, B.2004: Feature selection for intrusion detection: an evolutionary wrapper approach. In *Neural Networks, 2004.Proceedings.2004 IEEE International Joint Conference on* (Vol. 2, pp. 1563-1568).IEEE.

Kaur, P., & Kaur, R. 2014: An Optimized Approach for Feature Selection using Membrane Computing to Classify Web Pages. International Journal of Current Engineering and Technology Vol.4, No.5.

Mukkamala, S. & Sung, A.H. 2003: Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines. Journal of the Transportation Research Board of the National Academics, Transportation Research Record No 1822., 33-39.

Nguyen, H. T., Petrović, S., & Franke, K. 2010: A comparison of feature-selection methods for intrusion detection. In *Computer Network Security* (pp. 242-255).Springer Berlin Heidelberg.

Păun, G. 2000: Computing with membranes. *Journal of Computer and System Sciences*, *61*(1), 108-143.

Păun G. & Mario J. Perez-Jimenez. 2006: Membrane computing: Brief introduction, recent results and applications. BioSystems 85 pp. 11–22.

Sung, A. H., & Mukkamala, S. 2005: The feature selection and intrusion detection problems. In *Advances in Computer Science-ASIAN 2004.Higher-Level Decision Making* (pp. 468-482).Springer Berlin Heidelberg.

Toosi, A. N., & Kahani, M. 2007: A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. *Computer communications*, *30*(10), 2201-2212.

Zuech, R., Khoshgoftaar, T. M., & Wald, R. 2015: Intrusion detection and Big Heterogeneous Data: a Survey. *Journal of Big Data*, 2(1), 1-41.