Comparison of SVMs with Radial-Basis function & Polynomial Kernels in Classification of Categories in Intrusion Detection

C. Obimbo, E. Nyakundi
School of Computer Science, University of Guelph, Guelph, Ontario, Canada
{cobimbo, enyakund}@uoguelph.ca

***Abstract*—Recent increase in hacks and computer network attacks around the world, including the bank heist in Japan of over $81 million (May 2016) and the hacking into the Democratic National Committee (DNC) emails, which were leaked to and subsequently published by WikiLeaks on July 22, 2016, just to name a few, give a compelling need to develop better Intrusion Detection and Prevention systems. Network intrusions have become larger and more pervasive in nature. However, most anomaly intrusion detection systems are plagued by large number of false positives thus limiting their use. In this paper as a contribution to building better Intrusion Detection Systems, we compare the classification ability of Support Vector Machines (SVMs) with two different Kernels, SVM with polynomial kernel (SVM-P) and SVM with radial basis function kernel (SVM-RBF). Experiments were done and the polynomial kernel (SVM-P) was found to be the kernel of choice between the two.**

Keywords: SVMs, Intrusion Detection Systems, ISCX Dataset.

I. Introduction

The influence of the Internet in business, education, science and technology, government and other services, and in many aspects of our lives has been tremendous. However, it is imperative that we have consumer confidence in all areas of its use. The protection of computer systems and networks is thus of great importance to keep information safe from hackers and malicious users. Recent victims of hackers and data breaches include Sony Corporation [1], Home depot [2], and Minecraft [3]. A list containing just a few attacks from 2015 to the beginning of 2017 with the entity's name, number of records compromised, the kind of organization attacked is contained in Table I. The goal of intrusion detection is to identify, preferably in real time, unauthorized use, misuse and abuse of computer systems by both system insiders and external penetrators [4]. The intrusion detection problem is becoming more challenging due to the great increase in computer networks connectivity, the thriving technology advancement and the ease of finding hackers for hire.

Table I: Data Breaches [5]

Entity	**Year**	**Type**	**Method of Leak**	**Stolen Rcds**	**DS**[1]
Waterly	2017	Israel-based app	vulnerability	1M	3
Clinton campaign	2016	E-mail hacked. Used to influence vote.	hacked	5M	20
ClixSense	2016	Info. stolen: full user data.	hacked	6.6M	50000

where:
3: Email address / Password, 20: SSN/Personal details, 50000: Full bank account details.

II. Background Knowledge

A. Intrusion Detection Systems

Intrusion detection systems (IDSs) are security systems used to monitor, recognize, and report malicious activities or policy violations in computer systems and networks. IDSs are based on the hypothesis that an intruder's behaviour will be noticeably different from that of a legitimate user and that many unauthorized actions are detectable. Some of the security violations that would create abnormal patterns of system usage include unauthorized users trying to get into the system, legitimate users doing illegal activities, trojan horses, viruses and denial of service [6]. IDSs are generally categorized as Signature-based (Misuse detection) systems, Behaviour-based (Anomaly detection) systems, or Hybrid systems.

There are a number methods that have been used for Intrusion Detection Systems some of which are:

1. Misuse Intrusion Detection
2. Anomaly Intrusion Detection
3. Hybrid Systems; and
4. Intrusion Detection using Machine Learning Models

Anomaly detection is based on an IDS gathering "normal behaviour" and revealing the behaviour that differs from this. Misuse detection is based on similarities between a certain payload to a previous known signature or pattern of an attack. Most IDSs, such as NIDES (Next-generation Intrusion Detection Expert System) [8] and SNORT (which is owned by CISCO through CTO) that are deployed on networks are

based on misuse detection. This is because they are robust and have low false alarm rates. However, they suffer from one major shortcoming, they are not able to detect new attacks. Current research is focused on the anomaly detection approach since it can detect new attacks. Anomaly detection approach suffer from high false positive rates that render them impractical to be implemented in live network settings. Thus it is important to find new learning algorithms that can accurately detect day-zero attacks, while reducing the false positives. Machine learning algorithms, and specifically support vector machines (SVM) has been found to learn and correctly classify data in different fields (for example in Cancer research). We have thus chosen it to attempt to classify the different payloads as normal and anomalous, and in the latter case, define which kind of attack.
In this paper we will only concern ourselves with Machine learning Models, and in particular Support Vector Machines.

Also, in the past, there have been varied classifications of attacks used by IDSs. In this paper, we use the classifications used by the ***Information Security Centre of Excellence*** (ISCX) 2012 dataset [7], i.e. besides the Normal data, we have L2L (Local to Local), SSH, Botnet and DoS.

B. Machine Learning Models

Machine learning techniques generate models based on a provided training dataset with instances that are labeled normal or anomalous. Some datasets label the anomalous instances with the specific attack type. For example, probe, DoS, U2R or R2L. The labeling of training datasets used in machine learning is performed manually by human experts which makes it expensive to obtain an accurate labeled dataset.

Machine learning based techniques used in anomaly detection operate in three learning modes; Supervised, semi-supervised and unsupervised techniques. Supervised methods (classification methods), require a training dataset that contains both normal and anomalous labeled instances to generate the predictive model. Semi-supervised methods use a combination of unlabeled data and small amounts of labeled data. This reduces the labeling cost and also harnesses the good performance achieved by supervised methods. Unsupervised learning techniques (clustering methods) do not require training data. Unsupervised approach assumes that most of the network connection instances are normal traffic and that only a very small number of the traffic instances are anomalous. The approach also assumes that there is a statistical difference between the anomalous traffic and the normal traffic [9]. Using the assumptions above, data instances are clustered into groups of similar instances. The cluster with frequent data instances represent normal traffic while cluster with less frequent instances are considered anomalous. Research on machine learning has been carried out using the following techniques; Fuzzy logic [10], and Theory, Support Vector Machine (SVM) [11], Evolutionary computation, Association rules, Clustering and Artificial Neural Networks (ANN) [12].

C. Hybrid Systems

Hybrid intrusion detection systems combine both misuse detection and anomaly detection approaches to get the advantages of both approaches. The misuse detection system detects known attacks and anomaly detection approach is utilized for novel or unknown attacks. Hwang et al. [13] proposed to combine a signature based IDS and an anomaly detection system to get the advantages of low false positive rate and also an ability to detect novel attacks. Their anomaly detection system mined anomalous traffic from internet connections and used them to supplement the known signature base of the SNORT [14]. They compared their hybrid detection system with SNORT and BRO [15] systems and achieved 60% positive detection rate compared to 30% and 22% of the SNORT and BRO IDS respectively.

Zhang and Zulkernine [16] proposed a serial hybrid detection system that uses misuse detection method followed by the anomaly detection method. They use random forest technique for the misuse detection system to detect known intrusions and the outliers that result from the random forest algorithm are then utilized by the anomaly detection system. Their hybrid system achieved an overall detection rate of 94.7% and a small false positive rate of 2%.

A novel hybrid system was proposed by Depren et al. [17] that utilized both misuse and anomaly detection approaches. They added a decision system that combined the results of the two approaches. Their anomaly detection system used the Self-organizing Map (SOM) to model user behaviour and the misuse detection system used J.48 decision tree for classifying the various attack types. Their experiments showed they achieved better performance from the hybrid system compared to using each of the other systems individually.

Kim et al. [18] proposed a hybrid system that integrated misuse detection system hierarchically with an anomaly detection system. Their misuse detection system is based on the C4.5 decision tree algorithm. The misuse detection system is used to decompose the training dataset for the anomaly detection system into smaller subsets. The anomaly detection model is then created using the one class SVM on every decomposed region of the training subset. They showed that the decomposed training of the anomaly detection model took 50% -60% less time than the conventional models.

D. Other Classification

There are a number of other concepts used to classify IDSs [19] as seen on Figure 1. *behaviour on detection* describes the response of the IDS during an attack; active systems take active or proactive measures to counter the attack while passive systems merely generate alarms. Most of the IDSs are passive. The few active systems implement measures such as cutting connections that carry attacks, blocking traffic from attacking host, throttling bandwidth, or reconfiguring equipment such as routers or firewalls.

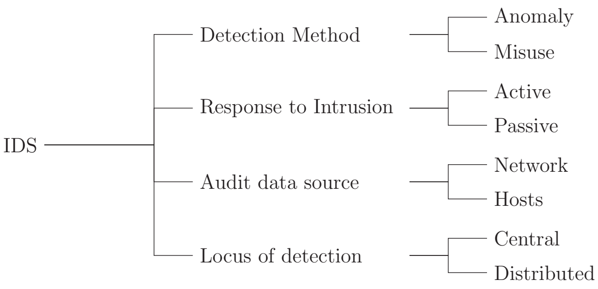

Figure 1: Characteristics of Intrusion Detection Systems.

Audit source location distinguishes IDSs based on the kind of information that is analyzed. Host-based IDSs reside on a single system and monitor activity on that machine using audit trails or system logs to detect possible attacks. Research on Host-based intrusion detection include [20], [21]. Network-based IDSs are placed at an important point or points within the network to analyze passing network traffic for signs of intrusion.

Locus of detection describes where the monitoring, detection, and reporting are controlled from. In centralized IDSs, monitoring, detection, and reporting are controlled directly from a central location. In distributed IDSs, monitoring and detection are controlled from a local control node with hierarchical reporting to one or more central location(s). The IDSs can also be off-line or on-line based. Off-line IDSs analyze the data off-line at a later time while the on-line systems analyze the data when the system is still working.

E. Support Vector Machines

Support Vector Machines (SVMs) were introduced by Vapnik [22]. SVMs have strong theoretical foundations and they have shown excellent empirical successes in classification tasks such as text classification and digit recognition.

SVM separates data into different classes by a hyperplane or hyperplanes since it has the ability to handle multidimensional data. SVMs minimizes empirical classification error and maximizes the margin. It also known as maximum margin classifier.

Suppose we have N training data points$\{(x_1, y_1), \ldots, (x_N, y_N)\}$, where $x_i \in R^d$ and $y_i \in \{+1, -1\}$. The hyperplane equation in d dimensions becomes:

$$(w^T \cdot x) + b = 0 \tag{1}$$

where $w \in R^n$ is weight vector, $b \in R$ is a bias value and x is an input vector. The decision function becomes

$$f(x) = sign(w^T \cdot x) + b \tag{2}$$

From the structural risk minimization principle, the optimal separating hyperplane of a linear classification is constructed by solving equation (3)

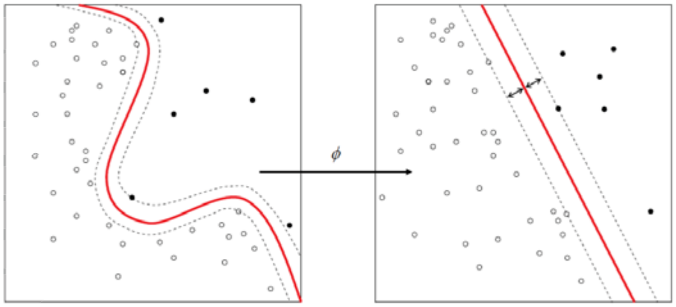

Figure 2: Nonlinear Transformation into Higher Dimension Linear Separable Function Using Kernel Trick [23]

$$\text{Min } \frac{1}{2}||w||^2, \tag{3}$$

subject to

$$y_i(w^T \cdot x_i + b) \geq 1, \quad i = 1, \ldots, N \tag{4}$$

The soft margin SVMs are used to reduce the effects of outliers and mislabeled examples. The method introduces a non-negative slack variable to (3) as shown below

$$\text{Min } \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i, \tag{5}$$

subject to

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N \tag{6}$$

where ξ_i is the slack variable and C is a penalty parameter that controls the trade-off between the cost of misclassification error and the classification margin. The dual form of the optimization problem becomes

$$\text{Max } \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j), \tag{7}$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N \tag{8}$$

where $K(x_i, x_j)$ is the kernel function and α_i are Lagrange Multipliers. The kernel function is one of the important elements attributed to the success of SVMs. The 'kernel trick' which transforms a nonlinear form of SVM to linear form as shown in Figure 2 without explicitly computing the products in the high-dimensional feature spaces.
There are three common kernel functions used with SVMs:

1. Polynomial Kernel function:

$$K(x_i, x_j) = [(x_i \cdot x_j) + 1]^p$$

where p is the dimension and $p \geq 1$.

2. Radial-Basis Function (RBF) Kernel function:

$$K(x_i, x_j) = exp(-\frac{||x - x_i||^2}{\sigma^2})$$

where σ is the kernel width.

3. Sigmoid kernel function:

$$K(x_i, x_j) = tanh[v(x_i \cdot x_j) + c]$$

The decision function of the non linear SVM is given by the following function

$$f(x) = sign(\sum_{i}^{N} \alpha_i y_i (x_i \cdot x_j) + b) \qquad (9)$$

Figure 3 below, shows the support vectors, maximum margin, hyperplane and the slack variables.

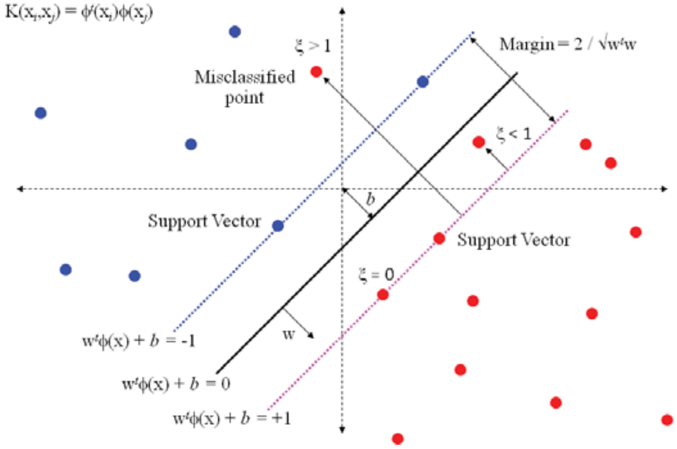

Figure 3: Maximum Margin Hyperplanes [24]

F. Intrusion Detection Datasets

We have chosen to use the ISCX 2012 Dataset created by Ghorbane et al [7] at the University of New Brunswick. This is due to the fact that previously used datasets such as the 1998 and 1999 DARPA intrusion detection systems [25][26] have been found to have had inherent problems. A full report of this may be found in McHugh's 2000 paper [26]. A summary of popularly used datasets may be found in Wu's 2010 paper [27].

Tavallaee, Stakhanova and Ghorbani [28] in 2010 reviewed the state of experimental practice in the anomaly intrusion detection area and they looked at 276 studies published between the year 2000 and 2008. They found out the most prevalent approach to evaluation of anomaly IDSs was based on fully or partially synthetic datasets. 70% of the studies used publicly available datasets, 32% created their own datasets, 9% used simulation tools and only 7% attempted to test their proposed systems on a real network. They pointed out that privacy was among the main issues in the criticism of existing publicly available datasets since most of them have been sanitized and anonymized to protect privacy.

Real traffic on the other hand is usually not guaranteed to be reliable. The other issues pointed out about the data sets include the definition of anomaly, data normalization, feature omission and data reduction.

III. Methodology and Implementation

A. ISCX2012 Dataset

The dataset used in this research to test the SVM classifiers is the Information Security Centre of Excellence (ISCX 2012) dataset created by Shiravi Ali, Shiravi Hadi, Tavallaee Mahbod and Ghorbani Ali from University of Brunswick [7]. The dataset was designed specifically for the purpose of developing, testing and evaluation of network intrusion and anomaly detection algorithms. It is among the few datasets that are public and not anonymized for privacy issues. It reflects the current trend of network data. The dataset was generated to address the shortfalls of most of the datasets used in anomaly Intrusion detection [29]. Most of the datasets used to test, evaluate and compare IDSs are internal and cannot be released to other researchers, or are outdated, or suffer from statistical inefficiencies. The dataset contains 18 features and the tag value indicates whether the flow is normal or attack. The features are shown in the Table II.

Table II: List of Dataset Features.

Attributes	
appName	sourceTCPFlagsDescription
totalSourceBytes	destinationTCPFlagsDescription
totalDestinationBytes	source
totalDestinationPackets	protocolName
totalSourcePackets	sourcePort
sourcePayloadAsBase64	destination
destinationPayloadAsBase64	destinationPort
direction	startDateTime
Tag	stopDateTime

B. Data Preparation

The size of the data means it has to be preprocessed to reduce the size and change the data types for it to work with the chosen algorithm. The dataset is inclusive of a labeled flow file that supports the use of supervised machine learning algorithms. The flow file is labeled with a 'Normal' and 'Attack' tag. The intrusions were carried out on specific days which enables us to rename the attack instances to the specific attack types which are Botnet attacks, Denial of Service attacks, brute force SSH attacks and internal infiltration (L2L) attacks.

The labeled flow is in XML format and it had to be transformed to a format that could be used to train the SVMs. Due to the large size of the dataset we picked 10% of all the labeled data as the test set. The training set is 1% of the entire labeled dataset. The split of 'Normal' and 'Attack' of the whole dataset is as shown on Table III chosen from the

Table III: Normal vs Attack Distribution

Flow	% Normal	% Attack
37870	100	0
13320	98.45	1.55
27555	92.61	7.39
17138	97.8	2.2
57170	93.46	6.54
52226	100	0
39760	98.7	1.3

Table IV: Dataset Attributes Statistics

	Training set	Test Set
Normal	1227	12285
L2L	60	605
SSH	46	463
Botnet	3	2
DoS	4	36
Total	1340	13391

specific days that had attack files. For example, Day 1 had all normal data, whereas Day 2 had 98.45% normal data. Table IV shows that 1% of the dataset in each category was used as the training data. High frequency attacks like L2L and SSH were also randomly chosen. Due to the low frequency of Botnet and DoS attacks in the dataset all of them were picked from the dataset and occur in both the training and testing dataset.

Table IV shows the distribution in categories of Intrusion, between the Training Set and the Test Set of the data we have selected to use.

The L2L attack is an infiltration of the network from the inside of the network. It entails a combination of attacks including a probing attack, a buffer overflow attack and SQL injection. The DoS attack was carried out on the web server to deny HTTP service. The Botnet is a DDoS attack and it was carried out using an IRC botnet. The SSH labeled attack is a probe attack that brute forces the main server using a dictionary with the goal of acquiring a SSH account. The different attack scenarios are further explained in [7].

Further adjustments had to be made to make the data fit for use. Reduction of the number of attributes from all the possible attributes had to be carried out. The following attributes were chosen for the experiment: 1) appName, 2) totalSourceBytes, 3) destinationTCPFlagsDescription, 4) totalDestinationBytes, 5) totalDestinationPackets, 6) protocolName, 7) totalSourcePackets, 8) sourcePort, 9) destinationPort, 10) destination, 11) direction, and 12) Tag.

C. Confusion Matrix

We use the common Confusion Matrix as part of our evaluating method. The matrix size is dependent on the number of distinct classes that are to be classified. It is used to compare the information visually about the actual class labels against the predicted class labels [30] from classifier. The confusion matrix displays the four values, that is

1. True Negative (TN), 2. True Positive (TP),
3. False Negative (FN), 4. False Positive (FP)

in a manner the relationship between them is easily comprehensible as shown in Table V.

Table V: Confusion Matrix Table.

		Predicted class	
		Normal	Abnormal
Actual class	Normal	TN	FP
	Abnormal	FN	TP

The other metrics used are precision and recall which are calculated using TP, FN, FP. F-measure is calculated from precision and recall. ROC curves are used to visualize the relation between the TP and FP rates [30].

D. Precision

Precision is calculated with respect to the intrusion class. It shows how many intrusions predicted by an IDS are actual intrusions [30]. A practical IDS should aim for high precision. A high precision means false alarms are minimized.

$$Precision = \frac{TP}{TP + FP} \quad where\ precision \in [0, 1]. \tag{10}$$

Precision cannot be used as the only metric because it does not express the percentage of predicted intrusions compared to all the intrusions in the present in the whole dataset.

E. Recall

Recall is a metric that shows the percentage of predicted intrusions versus all intrusions present. An IDS classifier should have a high recall value for it to be practical [31], [30].

$$Recall = \frac{TP}{TP + FN} \quad where\ recall \in [0, 1]. \tag{11}$$

Recall too has a disadvantage as it does not take into consideration false alarms. So an IDS might have a high recall value and a high false alarm rate.

F. F- Measure

F-Measure is a metric that gives a better measure of accuracy of an IDS. It uses a combination of precision and recall. It is the harmonic mean of precision and recall [31]. An IDS classifier's F-Measure is desired to be high, which implies high precision and high recall values [30]. If P is Precision and R is Recall, then F-Measure F_M is determined by the formula:

$$F_M = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{R + P} \quad \text{where} \quad F_M \in [0, 1]. \tag{12}$$

IV. Experiments and Results

Tests were done to compare the performance of the Support Vector Machines (SVM) using two different Kernels, the one classifier using a *polynomial kernel*, and another using a *radial basis function kernel*.

The tests were carried on the ISCX2012 data. 1% of the data randomly sampled from the whole data set was used as the training set. The test set contains 10% of the entire data set.

A. Objective of the Tests

The tests evaluates the performance of the RBF kernel function and polynomial kernel function to determine the better of the two functions when the SVM is applied to the train and test dataset.

B. General Steps of the Tests

The following steps were followed in the tests that were performed. The training dataset was loaded into the software suite. Depending on the classifier the data was preprocessed accordingly and will be stated in each tests description. The classifier is then applied to the dataset and tuning of the parameters to obtain the best detection rate was carried out. The models are then tested on the testing dataset. All the tests were carried out on the same training and testing dataset.

Test 1: Intrusion Detection Using SVM-Poly

This test is used to demonstrate intrusion detection for each class of scenario using a SVM classifier with polynomial kernel on a subset of the ISCX dataset. The effect of data normalization is also inspected on the classifier performance.

Test 2: Intrusion Detection Using SVM-RBF

This test is used to demonstrate intrusion detection each class of scenario using SVM classifier that uses a RBF kernel on a subset of the ISCX2012 labeled dataset. The effect of data normalization is also inspected.

C. SVM Algorithm Results

An experiment to find the performance of the SVM classifier was ran for both the polynomial kernel and the radial-basis function kernel. Tables VI and VII below shows the performance of the two kernels.

Table VI: SVM-P Confusion Matrix Table

Actual Class	**Predicted class**		
	Normal	L2L	SSH
Normal	12233	30	22
L2L	31	574	0
SSH	0	0	463

Table VII: SVM-RBF Confusion Matrix Table

Actual Class	**Predicted class**		
	Normal	L2L	SSH
Normal	12164	37	84
L2L	527	78	0
SSH	0	0	463

Using the Precision, Recall, and F-value parameters, we can see the comparison of the performances of the two Kernels in Figure 4 below.

As can be seen from Figure VII above, though there is a relatively similar performance in the SVMs with both Polynomial and RBF kernels in classifying the Normal data, with a slightly better performance using the P-kernel, there is a significant improved classification using the P-kernel with respect to the L2L and SSH data. In fact, the Polynomial-kernel shows about 7-basis points improvement in the F-value of L2L data (95% as compared with 88.3%), whereas it has 6% better value over the F-value of SSH (97.7% as compared to 91.7%).

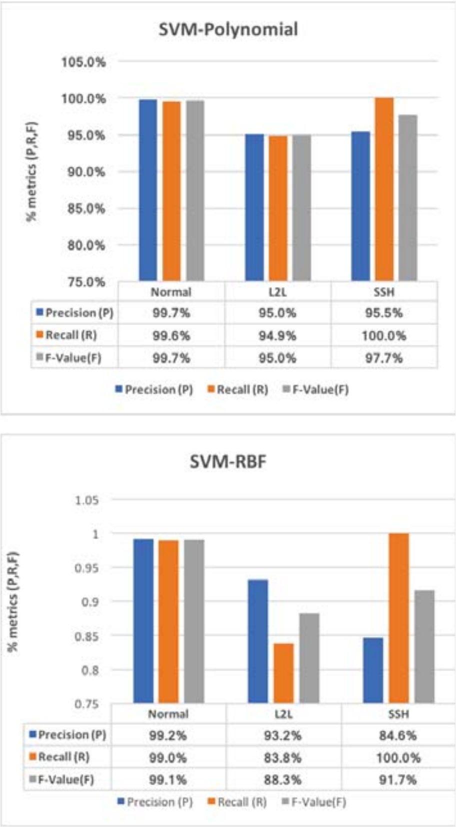

Figure 4: Comparison of SVM-P and SVM-RBF

The average performance of both the SVM with polynomial kernel (SVM-P) and the SVM with radial basis function kernel (SVM-RBF) in all three aspects, Precision, Recall, and F-value are shown on Table 6 below:

Table 6: SVM-P and SVM-RBF Performance Comparison.

	SVM-P	SVM-RBF
Precision	96.7	92.3
Recall	98.2	94.3
F-value	97.4	93.3

As can be seen, the average performance of the SVM with polynomial kernel (SVM-P) is better than the average performance of the SVM with radial basis function kernel (SVM-RBF). The SVM-P had an average precision of 96.7%, an average recall of 98.2%, and an average F-value of 97.4%, whereas the SVM-RBF classifier had average precision of 92.3%, an average recall of 94.3%, and an average F-value of 93.3% across all classes.

1) SVM-Polynomial Algorithm Results: After reviewing results the SVM with polynomial kernel (SVM-P) is found to be the better performing kernel. The process of tuning the classifier is carried out. Different values of the cost function were tested and a value of 1.0 was chosen. Normalization of the data sometimes improves the performance of some classifiers. Normalization was found not to have an effect on the SVM-P classifier.

2) SVM-RBF Algorithm Results: Support vector machine with the radial basis function kernel (SVM-RBF) is also implemented and all the parameters tuned. The same cost function of 1.0 chosen for the SVM-P is also used here. Different values of gamma are also tested to vary the effect of under-fitting and over-fitting. A gamma value of 0.1 is used.

V. Conclusion

SVM-P Comparison to SVM-RBF Results

After conducting the experiments it is evident that the SVM with polynomial kernel (SVM-P) has better F-values than support vector machine with the radial basis function kernel (SVM-RBF) on all classes of the data. SVM-RBF does not detect the low frequency classes of Botnet and DoS. The polynomial kernel becomes the clear kernel choice between the two (Radial-Basis function & Polynomial Kernels). Albeit, it would not be expedient to use it, as it does poor classification in the DoS attacks.

References

[1] T. Gara and C. Warzel, “Credit card breach at home depot,” http://www.buzzfeed.com/tomgara/sony-hack, [Online; accessed 1-April-2015].

[2] B. Krebs, “Credit card breach at home depot,” http://krebsonsecurity.com/2014/09/banks-credit-card-breach-at-home-depot/, [Online; accessed 10-April-2015].

[3] A. Hernandez, “Minecraft data breach affects users,” http://techaeris.com/2015/01/20/reports-minecraft-data-breach-affects-users/, [Online; accessed 21-April-2015].

[4] B. Mukherjee, L. Heberlein, and K. Levitt, “Network intrusion detection,” *Network, IEEE*, vol. 8, no. 3, pp. 26–41, 1994.

[5] M. Quick, E. Hollowood, C. Miles, and D. Hampson, “World’s biggest data breaches,” http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/, [Online; accessed 30- March-2015].

[6] D. Denning, “An intrusion-detection model,” *Journal of Graph Theory*, vol. SE-13, no. 2, pp. 222–232, 1987. [Online]. Available: http://dx.doi.org/10.1002/jgt.3190040204

[7] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, “Toward developing a systematic approach to generate benchmark datasets for intrusion detection,” *Computers and Security*, vol. 31, no. 3, pp. 357 – 374, 2012.

[8] D. Anderson, T. Frivold, and A. Valdes, *Next-generation intrusion detection expert system (NIDES): A summary*. SRI International, Computer Science Laboratory, 1995.

[9] K. Leung and C. Leckie, “Unsupervised anomaly detection in network intrusion detection using clusters,” in *Proceedings of the Twenty-eighth Australasian conference on Computer Science - Volume 38*, ser. ACSC ’05, 2005, pp. 333–342. [Online]. Available: http://dl.acm.org/citation.cfm?id=1082161.1082198

[10] J. Dickerson and J. Dickerson, “Fuzzy network profiling for intrusion detection,” in *Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American*, 2000, pp. 301 –306.

[11] S. Mukkamala, A. Sung, and B. Ribeiro, “Model Selection for Kernel Based Intrusion Detection Systems,” in *Adaptive and Natural Computing Algorithms*, 2005, pp. 458–461.

[12] R. P. Lippmann and R. K. Cunningham, “Improving intrusion detection performance using keyword selection and neural networks,” *Computer Networks*, vol. 34, no. 4, pp. 597 – 603, 2000, recent Advances in Intrusion Detection Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128600001407

[13] K. Hwang, M. Cai, Y. Chen, and M. Qin, “Hybrid intrusion detection with weighted signature generation over anomalous internet episodes,” *Dependable and Secure Computing, IEEE Transactions on*, vol. 4, no. 1, pp. 41–55, Jan 2007.

[14] M. Roesch *et al.*, “Snort: Lightweight intrusion detection for networks.” in *LISA*, vol. 99, no. 1, 1999, pp. 229–238.

[15] V. Paxson, “Bro: a system for detecting network intruders in real-time,” *Computer networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[16] J. Zhang and M. Zulkernine, “A hybrid network intrusion detection technique using random forests,” in *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, April 2006, pp. 8 pp.–.

[17] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, “An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks,” *Expert Systems with Applications*, vol. 29, no. 4, pp. 713 – 722, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417405000989

[18] G. Kim, S. Lee, and S. Kim, “A novel hybrid intrusion detection method integrating anomaly detection with misuse detection,” *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 1690 – 1700, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417413006878

[19] H. Debar, M. Dacier, and A. Wespi, “Towards a taxonomy of intrusion-detection systems,” *Journal of Graph Theory*, vol. 31, no. 8, pp. 805–822, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128698000176

[20] D.-Y. Yeung and Y. Ding, “Host-based intrusion detection using dynamic and static behavioral models,” *Pattern Recognition*, vol. 36, no. 1, pp. 229 – 243, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320302000262

[21] L. Ying, Z. Yan, and O. Yang-jia, “The design and implementation of host-based intrusion detection system,” in *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on*, April 2010, pp. 595 –598.

[22] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[23] Wikipedia, “Kernel machine. http://commons.wikimedia.org/wiki/file:kernel_ / machine.png,” [Online; accessed 21-Feb-2015].

[24] P. Anandan, M. Varma, and J. Joy, “Multiple kernel learning,” Available at http://research.microsoft.com/en-us/groups/vgv/.

[25] 1998 DARPA Intrusion Detection Evaluation Data Set, Available at http://www.ll.mit.edu/ideval/data/1998data.html, [Online; accessed 13-May-2014].

[26] J. McHugh, “Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory,” *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, Nov. 2000. [Online]. Available: http://doi.acm.org/10.1145/382912.382923

[27] S. X. Wu and W. Banzhaf, “The use of computational intelligence in intrusion detection systems: A review,” *Applied Soft Computing*, vol. 10, no. 1, pp. 1 – 35, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494609000908

[28] M. Tavallaee, N. Stakhanova, and A. Ghorbani, “Toward credible evaluation of anomaly-based intrusion-detection methods,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, no. 5, pp. 516–524, Sept 2010.

[29] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, july 2009, pp. 1 –6.

[30] A. A. Ghorbani, W. Lu, and M. Tavallaee, *Network Intrusion Detection and Prevention - Concepts and Techniques*, ser. Advances in Information Security. Springer, 2010, vol. 47. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-88771-5

[31] N. Ye, *The Handbook of Data Mining. http://search.ebscohost.com.subzero.lib.uoguelph.ca/login.aspx?direct=truedb=nlebkAN=83855site=ehost-livescope=site*, ser. Human Factors and Ergonomics. CRC Press, 2003.