ARTICLE OPEN ACCESS

# A DISTRIBUTED APPROACH FOR PREDICTING MALICIOUS ACTIVITIES IN A NETWORK FROM A STREAMING DATA WITH SUPPORT VECTOR MACHINE AND EXPLICIT RANDOM FEATURE MAPPING

**Prabaharan Poornachandran [1], Premjith B[2], Soman K. P[2]**

[1]*Amrita Center for Cyber Security Systems and Networks, Amrita Vishwa Vidyapeetham, Kollam, INDIA*

[2]*Centre for Computational Engineering and Networking, Amrita Vishwa Vidyapeetha,, Coimbatore, INDIA*

## ABSTRACT

*Technology reduces human effort. However technological advancements always bring threat to personal as well as organizational security, mainly because we all are connected to the internet. Therefore, ensuring cyber security becomes the major topic of discussion. As the magnitude of activities over the internet is unimaginable, envisioning the characteristics of network activities whether it is malicious or good, coming from a stream of data in real time is really a tough task. To tackle this problem, in this paper, we propose a distributive approach based on Support Vector Machine (SVM) with explicit random feature mapping and features mapping is obtained using Compact random feature maps (CRAFTMaps) algorithm. Distributing the job achieves notable improvement in the total prediction time.*

**Received on:** 2nd-April-2016
**Revised on:** 27h-April-2016
**Accepted on:** 28th-April-2016
**Published on:** 26th-July-2016

KEY WORDS

*Cyber security, Streaming data, Support Vector Machine, Compact random feature maps, Distribute processing*

***Corresponding author: Email:** prem.jb@gmail.com **Tel:** +91-9597141816

## INTRODUCTION

As the new technological innovations are emerged, cyber-attacks are also changing the colours. The ubiquity of technology leads to the exponential growth in the cyber threats. Now cyber security has become one of the primary concerns of governments as well as private organizations. An important problem of cyber security is how to effectively monitor and predict threats in real time, i.e. detecting the threats from streaming data. A streaming data is nothing but a massive volume of data coming from various sources, such as videos, images, text etc. and may not be stored in a disk for analysis. Streaming data are considered as data in motion and the analysis is always single-pass, i.e. the data cannot be reanalyzed once it is streamed.

With the increase in network traffic, network logs have become huge and the detection of cyber-threats from these massive streaming logs is now a tedious task. Conventional machine learning algorithms are not suitable for the real time prediction of malicious activities in a network as they require storage of the data to predict whether it is a threat or not. But this is not possible for streaming data. As the data streams are one-pass and highly non-static, the decision has to be taken in quick time. This necessitates the requirement of a fast and scalable mechanism for real time detection of cyber threats from a data stream.

In cyber security, one of the problems is classification of network logs into malicious and benign. In machine learning, classification [1], a supervised learning approach, is used for detecting the malicious activities in a network. Support vector machine [2], Regularized least squares [3] etc. are common classification algorithms. Generally classification algorithms are linear in the sense; they are able to classify data which are linearly separable. But a streaming data or network traffic logs are never linearly separable and are often non-stationary. So offline storage and analysis is quite impossible [4]. Conventional classification algorithms are designed to work with offline data. There are certain other issues which make the classification of data streams makes tougher which are high speed nature of data streams, unbounded memory and hardware requirements, concept drifting, data visualization, challenges in distributed applications, modeling of mining results in real time, tradeoff between accuracy and efficiency etc. [9].

Nonlinear classification algorithms such as non-linear SVM with kernel methods give state-of-the-art accuracy in detecting cyber threats when the prediction is done offline and come with a huge computational cost for real time prediction. Generally kernel methods transform input data to a finite higher dimensional space where a linear separation is obtained. Kernel methods were widely used for classifying data when the data size is relatively small. But if the input data size is massive and if the data are not linearly separable, the number of support vectors (Special subset of input data) to be stored becomes large. Therefore as the size of the data increased, so does the computation time and storage. Explicit feature mapping methods alleviate this curse of support problem and thereby make the classification algorithms appropriate to deal with streaming data. An explicit feature mapping algorithm projects the input feature vectors to a higher dimensional vectors which are randomly generated from a standard normal distribution and then compute the dot products.

Several explicit feature mapping algorithms such as Random Kitchen Sink [5-6], [13], Fastfood [7], Compact random feature maps [8] etc. are generally used for mapping input data explicitly to a higher dimensional space. In this paper we investigate a recently introduced explicit mapping method - Compact random feature maps. Compact random feature map algorithm is a polynomial kernel approximation which resolves the rank deficiency (underutilization of projected space) problem [8] that commonly appears in random mapping.

In this paper we investigate the feasibility of distributing the abnormal activity detection in a streaming data with compact random feature mapping algorithm and Support Vector Machine classification algorithm. Experimental study showed that, like other explicit feature mapping algorithms, compact random feature map algorithm proposed by Hamid, et.al [8] computes only one dot product in the higher dimensional space and hence the storage requirement is very less. This algorithm also manages the underutilized space in the featured space by down projecting the obtained features. Utilizing this advantage of explicit feature mapping algorithm, this paper proposes a parallel implementation of real time prediction of streaming data. Four machines run in parallel implements the decision function and result from all the machines combined to attain the overall prediction time and accuracy.

## METHODS

Even though kernel methods are successful for offline prediction, it is found to be difficult to use typical kernel methods for the prediction at real time. This is because of the fact that, the storage and offline analysis of data streams is impractical. So classification algorithms like Support Vector Machine works poorly for streaming data. In order to fix this issue, we utilize compact random feature map algorithm, an explicit way of projecting feature vectors to a higher dimensional space, in a distributed way. Parallelism is achieved by dividing the projected features uniformly and each subset of features is passed to different processors. For each processor, define weight vectors $\omega$ . Linear combination of features and weight is computed at each processor and final prediction is computed by combining the results obtained at each processor. **Figure-1** shows the block diagram of how prediction is implemented distributive after projecting the input feature vectors to a finite higher dimensional space.
Mapping of features to another dimension is achieved by kernel trick and is discussed in the next subsection.

### *Kernel trick*

The heart of a classification algorithm is kernel trick. Using kernel trick, the decision function can be computed as,

$$g(x) = \omega^T \psi(x) \tag{1}$$

Here $\psi(.)$ is a feature mapping operator. Unfortunately this feature mapping may lead to infinite dimensionality and make the computation very expensive. In order to nullify this problem, a dual representation was introduced to compute the decision function,

$$g(x) = \omega^T \psi(x) = \sum_{i=1}^{d} a_i \langle \psi(x_1), \psi(x) \rangle \tag{2}$$

Where $H(x, x') = \langle \psi(x), \psi(x') \rangle$

This representation is called kernel function. Classification algorithms such as Support Vector Machine, a linear classifier in nature, uncover the non-linear relationship among data using the dual representation. This kernel trick reduces computational cost of evaluating the feature mapping function $\psi(.)$.

One disadvantage of the typical kernel methods is scaling problem. That is, these methods often fail with large data set or real time prediction. A fast learning approach is required to deal with this problem. Explicit random feature mapping algorithms are one of the solutions to this problem. Numerous algorithms such as Random Kitchen Sink, Fastfood etc. have been devised for the explicit mapping of input feature vectors to a manageable higher dimensional space. Compact random feature map is one such algorithm which approximates polynomial kernels. The background of all explicit random feature mapping is the method of random Fourier features proposed by Rahimi et al. [5].

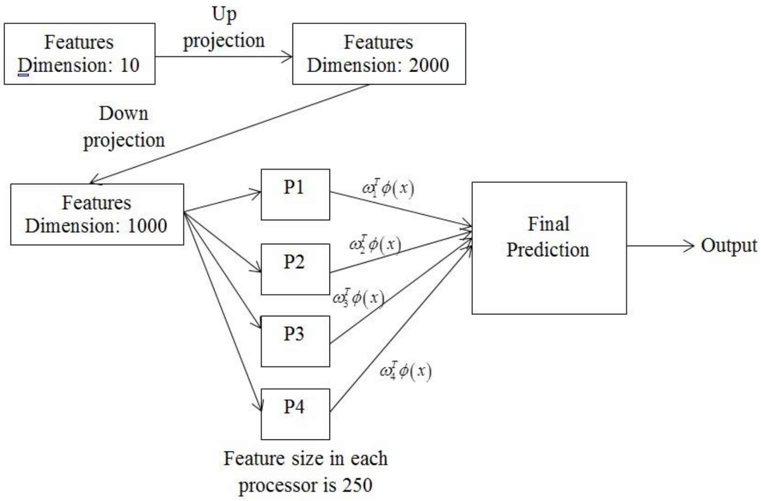

**Fig: 1. Block diagram for how the prediction is computed parallel with explicit feature mapping**

## *Random Fourier Features*

Rahimi et.al proposed an alternate approach, Random Kitchen Sink, to the kernel trick. This approach accelerates the training as well as testing by projecting the feature vector to a manageable random higher dimensional space. And the obtained inner product of the transformed features is approximately equal to the inner product in feature space.

The key theorem behind Random Kitchen Sink (RKS) is Bochner's theorem [10]. According to Bochner's theorem, any shift invariant, continuous function is positive definite if and only if it is the Fourier transform of a positive measure. Mathematically, we can write this theorem as,

$$P(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(z) e^{-jz^T \omega} dz \qquad (3)$$

Where $z = x - y$.

So as a consequence of Bochner's theorem, the inverse Fourier transform is interpreted as the expectation in probability theory.

$$H(\omega) = \int_{\mathbb{R}^d} P(\omega)\, e^{j\langle \omega, (x-y)\rangle} d\omega \qquad (4)$$

Where $P(\omega)$ is a probability distribution if the kernel is properly scaled [11].

Therefore we can write as,

$$H(x-y) = E\left[ e^{j\omega^T x} e^{-j\omega^T y} \right] \qquad (5)$$

Where $\omega$ is sampled from $P(\omega)$.

The estimate can be improved by drawing random samples from the distribution $\omega_1, \omega_2, \ldots, \omega_D \sim P(\omega)$ [11]. Now the estimate is computed as the expectation of mean of samples.

$$H(x-y) = E\left[\frac{1}{D}\sum_{i=1}^{D} e^{j\langle \omega_i, (x-y)\rangle}\right] \quad (6)$$

Since both the probability distribution and kernel function are real, we can omit the imaginary part in the expansion of $e^{j\langle \omega_i, (x-y)\rangle}$. Hence, the kernel function can now be written as,

$$H(x-y) = E\left[\frac{1}{D}\sum_{i=1}^{D} \cos\left(\omega_i^T (x-y)\right)\right] \quad (7)$$

Since $\cos\left(\omega_i^T (x-y)\right) = \cos\left(\omega_i^T x\right)\cos\left(\omega_i^T y\right) + \sin\left(\omega_i^T x\right)\sin\left(\omega_i^T y\right)$

$$\cos\left(\omega_i^T (x-y)\right) = \begin{bmatrix} \cos\left(\omega_i^T x\right) \\ \sin\left(\omega_i^T x\right) \end{bmatrix} \bullet \begin{bmatrix} \cos\left(\omega_i^T y\right) \\ \sin\left(\omega_i^T y\right) \end{bmatrix} \quad (8)$$

And now the kernel function can be written as,

$$H(x-y) = \frac{1}{D}\begin{bmatrix} \cos\left(\omega_i^T x\right) \\ \sin\left(\omega_i^T x\right) \end{bmatrix} \bullet \begin{bmatrix} \cos\left(\omega_i^T y\right) \\ \sin\left(\omega_i^T y\right) \end{bmatrix} = \frac{1}{D} J(x) \bullet J(y) \quad (9)$$

Where, $J(x) = \dfrac{1}{\sqrt{D}}\begin{bmatrix} \cos\left(\omega_i^T x\right) \\ \sin\left(\omega_i^T x\right) \end{bmatrix}$ and $J(y) = \dfrac{1}{\sqrt{D}}\begin{bmatrix} \cos\left(\omega_i^T y\right) \\ \sin\left(\omega_i^T y\right) \end{bmatrix}$

RKS algorithm defines the maps feature vector to a higher dimensional space with this feature mapping operator.

## *Compact random feature map (CRAFTMaps)*

CRAFTMaps is another explicit random feature mapping algorithm whose key idea is to apprehend the Eigen structure of the exact kernel space completely, and then represent it in a more compact form. Unlike RKS, CRAFTMaps algorithm approximates polynomial kernels of the form $\left(x^T y + q\right)^r$ with $q \in \mathbb{R}^+$ and $r \in \mathbb{N}_0$. CRAFTMaps algorithm computes the feature mapping in the following two steps: up projection and down projection.

In the up projection step, feature mapping function is defined as, $J : \mathbb{R}^d \to \mathbb{R}^D$, where $d$ is input feature dimension and $D$ is the dimension of projected space, $d < D$ and for all $x, y \in \mathbb{R}^D$, $\langle J(x), J(y)\rangle = H(x, y)$. Feature mapping is obtained by projecting $x$ onto a set of random $d$ dimensional vectors from standard Gaussian distribution, and then compute the dot product at the projected space. This up projection can be achieved using any of the well-known explicit feature mapping algorithms.

One of the main disadvantages of random feature mapping is rank deficiency. In order to nullify this effect, CRAFTMaps down projects the resultant feature vectors to a relatively lower dimension. Now the feature mapping operation is defined as, $F : \mathbb{R}^D \to \mathbb{R}^E$, $E < D$ and $\langle F(J(x)), F(J(y))\rangle \approx \langle J(x), J(y)\rangle$.

## RESULTS

We conducted our experiment with 1999 KDD cup data set [12]. Data set contains 494021 data samples and 41 features. Among these data, 449785 are abnormal and remaining data is normal. The objective of our experiment is to predict whether the incoming data is abnormal or not. Since the distribution of data is not uniform, (i.e. 91.05% data are abnormal) we can employ One-class SVM algorithm for the classification. In the preprocessing step, all data are normalized to the range [-3, +3] and also we assume that training data are not linearly separable. Therefore, input feature vectors are mapped to a higher dimension where we can draw a classifier which can separate data linearly.

After normalizing the training data, CRAFTMaps algorithm is used for the explicit random projection and features are projected to a dimension 4 times the input feature dimension. Random projection of features is obtained by multiplying each feature vector with the product of a Hadamard matrix and a diagonal matrix whose elements are coming from {+1, -1} with equal probability. Dimension of space to which the features are projected, input feature dimension and the degree of the polynomial kernel determines the number of such multiplications for each feature. Here, the input feature size is $d = 41$, the dimension to which features are projected is taken as $D = 164$ and the degree of the polynomial kernel is fixed to $r = 2$. So 8 multiplications are required in this case ($T = (r * D / d)$).

In order to avoid the rank deficiency problem, resultant feature vectors are down projected to a lower dimension. This dimension depends on the degree of the polynomial kernel. After shuffling elements in resultant feature vectors randomly, we combine two adjacent elements from each feature vectors to generate new feature vectors. Now the dimension of features has reduced to 82. These obtained features are given as input to SVM for classification. Training model is created by adjusting certain parameters in the toolbox. Test data also first normalized and then mapped to a relatively manageable higher dimension.

These computations are performed on 4 parallel machines and the final output is obtained by assembling the results from the parallel machines. Total data (80000 samples) were divided into 4 batches and each batch of 20000 data was given to each processor for prediction. Prediction times from all four machines are observed and total prediction time is fixed as the maximum time taken by a single processor. Even though four machines are of same configuration, speed of prediction depends on many external as well as internal factors.

Despite implicit mapping of features gives 100% prediction accuracy, the time taken to predict the malicious activities in the network is huge. Explicit random feature mapping improves the time complexity to a great extent by allowing a small percentage of error. Application of CRAFTMaps algorithm also reduces computational complexity by avoiding the underutilized spaces in the featured space.

**[Table-1]** explains the time taken by each machine for the prediction. Four machines gave different prediction time and the time taken for detecting the abnormal activities in the network is taken as 0.32 sec, which is the maximum time taken by a single processor. Total prediction accuracy is taken as the average of accuracies obtained from each machine. Here we get 98% accuracy in detecting malicious activities from a set of 80000 data, which is equivalent to the state of the art accuracy.

**Table: 1. Prediction time of parallel machines and prediction accuracy**

| Machine | Prediction time | Prediction accuracy |
|---|---|---|
| P1 | 0.29 sec | 98% |
| P2 | 0.32 sec | 98% |
| P3 | 0.29 sec | 100% |
| P4 | 0.31 sec | 96% |

## CONCLUSIONS

Identification and detection of malicious activities in a network in real time is a difficult task because time and storage requirement for real time prediction is huge. So detection of abnormal activities from a streaming network data with SVM and explicit random feature mapping algorithms (say CRAFTMaps) reduces the time requirement. Also when the prediction is done in parallel, the speed can be improved significantly. CRAFTMaps algorithm was used for explicit mapping and obtained 98% accuracy in 0.32 seconds for 80000 data. Distributive processing of data improves the prediction of malicious activities from streaming network logs by a great margin.

## CONFLICT OF INTEREST

None

## ACKNOWLEDGEMENT

None

## FINANCIAL DISCLOSURE

None

## REFERENCES

[1] Hand, David J, Heikki Mannila, and Padhraic Smyth. [2001]Principles of data mining. MIT press.

[2] Cortes, Corinna, and Vladimir Vapnik. [1995]Support-vector networks. Machine learning 20(3): 273-297.

[3] Li Wenye, Kin-Hong Lee, and Kwong-Sak Leung. [2006] Generalized regularized least-squares learning with predefined features in a Hilbert space, Advances in neural information processing systems.

[4] Angelov, Plamen P, and Xiaowei Zhou. [2008] Evolving fuzzy-rule-based classifiers from data streams." Fuzzy Systems, *IEEE Transactions on* 16.6: 1462-1475.

[5] Rahimi, Ali, and Benjamin Recht.[ 2007] Random features for large-scale kernel machines. Advances in neural information processing systems.

[6] Rahimi Ali, and Benjamin Recht.[ 2009] Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. Advances in neural information processing systems.

[7] Le, Quoc Viet, Tamas Sarlos, and Alexander Johannes Smola. [2014] Fastfood: Approximate Kernel Expansions in Loglinear Time. arXiv preprint arXiv:1408.3060 .

[8] Hamid, Raffay, et al. [2013]Compact random feature maps." arXiv preprint arXiv:1312.4626 .

[9] Charu C Aggarwal. [2006] Data Streams: Models and Algorithms (Advances in Database Systems). Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[10] W Rudin. [1994]Fourier analysis on Groups. Wiley Classics Library. Wiley-Interscience, New York, reprint edition.

[11] von Tangen Sivertsen, Johan.[2014] Scalable learning through linearithmic time kernel approximation techniques.

[12] Lichman M. [2013] UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[13] Kumar S, Sachin B, Premjith M, Anand Kumar, and KP Soman.[2015] AMRITA_CEN-NLP@ SAIL2015: Sentiment Analysis in Indian Language Using Regularized Least Square Approach with Randomized Feature Learning." In Mining Intelligence and Knowledge Exploration, pp. 671-683. Springer International Publishing.

## ABOUT AUTHORS

*__Prabaharan Poornachandran__ has two decades of experience in Academia and Industry, Currently serves as Assistant Professor at Amrita University and Principal Investigator for large scale Security projects. His current area of research includes, Big-data Security Intelligence, Cyber-Physical systems security, Machine learning for Security, Complex Binary analysis, IoT, SCADA and Hardware security, Application & Network security, Advanced Forensics and Incident handling etc.*

*__Premjith B__ is currently pursuing PhD in Center for Computational Engineering and Networking, Amrita University. His research area includes Natural Language Processing and Machine Learning.*

*__K P Soman__ currently serves as Head and Professor at Center for Computational Engineering and Networking (CEN), Amrita Vishwa Vidyapeetham, Coimbatore Campus. Further info on his homepage: https://www.amrita.edu/faculty/soman.*