Early Detection of LDDoS Attacks in IOT Utilizing Locality Sensitive Incremental TSVM Method

ZengGuang Liu*, XiaoChun Yin**, RuiPing Yang*, AnMing Dong***

*College of Computer Science and Engineering, ShanDong University of Science and Technology, QingDao, 266590 CN
**Blockchain Laboratory of Agricultural Vegetables, WeiFang University of Science & Technology, ShouGuang, 262700 CN
***School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), JiNan, 250353 CN

Corresponding Author: xiaochunyin@wfust.edu.cn, Tel: +86 18853688750

Abstract— The Mirai botnet and its variants has made Internet of Things (IOT) devices a powerful amplifying platform for Low Rate Distributed Denial-of-Service (LDDoS) attacks. In this paper, we investigate and develop a novel semi-supervised Locality Sensitive Incremental Transductive Support Vector Machine (LS-ITSVM) method. The proposed method maximizes the margins of different network flows by incorporating local frequency-domain features from the autocorrelation sequence of network flow into the regularization time-domain framework of TSVM. And it saves training and detecting time by incremental training support vectors and new added samples. The result of simulation proves the proposed method can distinguish abnormal network flows with higher detection accuracy, faster training and response time, and prevent abnormal network flow groups with less impaction.

Keywords— LDDoS, Internet of Things, Locality Sensitive Incremental TSVM, Frequency-Domain Features, Semi-Supervised Clustering

I. INTRODUCTION

Distributed Denial-of-Service (DDoS) attacks, characterized by brute-force, sustained high rate or specifically designed to explore the protocol limitations or software vulnerabilities in services, are well-known. The difficulty of traditional internet DDoS is how to control as many as desktop computers, which has strong password, updated anti-virus software regularly, and unexpected off-line. With the emergence of IOT, things change. IOT devices have weak security configurations, lacked computational capabilities to run anti-virus, and constantly connected to Internet. As a result, it becomes soon a powerful amplifying platform for DDoS attacks. The recent prominent examples are Mirai botnet. According to the French webhost and cloud service provider OVH, a DDoS attack using Mirai malware hit them with 1.1 Tbps at peak by about 145K hacked cameras in September 2016[1]. Moreover, hackers announced they could rent a Mirai botnet of 400K Bots only two months later[2]. Thus, researches tried to understand Mirai botnet operation and communication model[3], and proposed many feasible methods.

In order to dodge existing detection proposals, Mirai’s variants with low-rate attack mechanisms start to obfuscate their DDoS activity and are becoming a serious threat to Internet. As we know, a successful DDoS attack needs to tune a tuple (T, R, L) of TCP at least. T is an integer multiple coincides with minRTO value of TCP; R is the peak rate and L is the burst length. In order to cause TCP packet loss and begin slow-start, R should be large enough and L should be longer than RTT value. As FIGURE 1 shown, we simulated a LDDoS attacks of 4 Mirai Bots with the R of 50Kb/sec. For the purpose of drawing them clearly in FIGURE 1, we start to attack one by one with a delay of 10ms. Obviously, we can see that the LDDoS attack stream hides itself among normal traffic by making its peak rate even lower than the normal traffic rate. All in all, compared with traditional DDoS attacks, LDDoS ones have three mainly characteristics: hard to detect, because it has same flow classified features with normal flow; low-cost and multi-targets, because the attacks can be finished in a few nodes with small flow data, they can be orchestrated to attack multiple targets simultaneously; long term attacked-target insensitive attacks, because attacked-target has self-adapt mechanism to adjust network flow. Thus, the aim of this paper is to have a method with high detection accuracy, fast response time to detect LDDoS Attacks.
This paper is organized as follows: we firstly review the related state-of-the-art work in section II. Next, a novel semi-supervised locality sensitive incremental TSVM method is proposed and described to address the LDDoS in section III. And then, simulation results are presented to show the effectiveness of defending against LDDoS attacks in section IV. Finally, Section V concludes the paper with suggestive future enhancements.

II. RELATED WORK

After Mirai botnet, IoT botnets had received spread attention. Including traditional detection methods, researchers began to protect IoT with the skill of machine learning (ML) and Artificial Intelligence (AI)\cite{17}. When LDDoS attacks came into the world, above methods were not applied to these new type attacks. As a result, researchers had to focus on new attacks, and a few defensive solutions had been existed in the literature. G. Kaur et al.\cite{8} worked out a variant of the Shiryaev-Roberts's procedure to detect and deter these stealthy low rate DoS attacks dynamically, using Checkpoint Anomaly Detection techniques. Z. Wu et al.\cite{9} studied complex multifractal structure for developing concise mathematical models, and proposed an algorithm of multifractal detrended fluctuation analysis (MF-DFA), which is used to explore the change in terms of multifractal characteristics over a small scale of network traffic due to LDDoS attacks. ML and AI also were applied in LDDoS. Z. G. Liu et al.\cite{10} proposed a new method to take frequency-domain characteristics from the autocorrelation sequence of network flow as clustering feature to group end-user flow data by BIRTH algorithm, and re-merge these clustering results into new groups by overcoming the deficiency of BIRTH algorithm. M. Yue et al.\cite{11} investigated a new identification approach based on wavelet transform and combined neural network to classify normal network traffic and LDDoS attack traffic. Z. J. Wu et al.\cite{12} used the propagation back (BP) model of neural network to establish the nonlinear model of network traffic, and proposed a method of LDDoS attacks detection based on particle filter according to the mechanism of LDDoS attack. In this method, the difference between the estimated value of the particle filter and the one step prediction is used as the detection basis, and a detection threshold is designed to determine the initiation and termination of the LDDoS attacks for the purpose of detecting LDDoS attacks. P. Nagar et al.\cite{13} provided a comparative study between the ANN and the optimizer-based ANN technology. They concluded that the convolution neural network with SVM show effective analysis providing accurate forms of IDS, thereby improving its detection based on individual class along with maintaining its results fundamentally. X. Peng et al.\cite{14} optimized a Mahalanobis distance by perturbing continuous features and discrete features of DDoS samples respectively, and proposed an improved boundary-based method to craft adversarial DDoS samples. M. Begli et al.\cite{15} designed an intrusion detection system (IDS) using SVM algorithm, and proved its efficiency. P. Rivas et al.\cite{16} implemented and trained a non-probabilistic binary linear attack pattern classifier, and trained a support vector machine and a convolutional neural network using a supervised learning model with labeled data sets. Experimental results suggested that the models can detect DDoS attacks with high accuracy rates. T. A. Tuan et al.\cite{17} performed an experimental analysis of the machine learning methods (including Support Vector Machine, Decision Tree, Naive Bayes, and ANN) for Botnet DDoS attack detection. The evaluation is done on the UNBS-NB 15 and KDD CUP-99 which are well-known publicity datasets for Botnet DDoS attack detection. Results showed SVM has higher accuracy. Although these methods have high accuracy, they need more training and detecting time when new samples were labeled.

Thus, we proposal and develop a novel semi-supervised locality sensitive incremental TSVM method for higher detecting accuracy and fast training and response time.

III. LOCALITY SENSITIVE INCR-TSVM METHOD

In LDDoS attacks scenarios, attack flows almost have same classified features with normal flows, and they can fire long term attacked-target insensitive attacks. As a result, the traffic volume analysis method cannot detect such a stealthy attack any more. Recently, some researchers begin to introduce the semi-supervised TSVM methods\cite{18-22}. The pros are that these methods can take use of the underlying geometric structure of unlabeled samples to train the classifier, while the cons are that they are not suitable for incremental learning and their training speed is too slow to meet IOT’s requirement. In order to address above issues, we propose a novel locality sensitive incremental TSVM method in this paper. “Locality sensitiveness” helps to quickly locate which cluster the unlabeled data belongs to and speed up training. “Incremental TSVM” can help to do incremental learning by support vectors and prior classified unlabeled data.

A. Locality Sensitive Features Extraction

The idea of locality sensitive is for solving the approximate in high dimensional spaces. It can help to map similar items to the same buckets with high probability. With it, the problem of “searching the similar samples in a huge data set” is turned to “searching them in a small bucket”. Thus, the searching can speed up. In order to use locality sensitive, we need to satisfy the following conditions for any \(x_1, x_2 \in M\):

- if \(d(x_1, x_2) \leq d_1\), then \(h(x_1) = h(x_2)\) with probability at least \(P_1\);
- if \(d(x_1, x_2) \geq d_2\), then \(h(x_1) = h(x_2)\) with probability at most \(P_2\);

Here, \(M\) is data set space; \(d\) is a distant function in \(M\); \(x_1, x_2\) is vectors in \(M\); \(d_1\) and \(d_2\) is the threshold of distance; \(h\) is a function used to map elements from \(M\) to a bucket \(s \in S\).

As to \(x_1, x_2\), we should choose them from training dataset well. In our case, if we review the FIGURE 1 carefully, we find that there are always periodic low-rate rectangle shape waves. Generally, what exactly happens is that more power of the autocorrelation function is distributed in the lower frequency band if there is shrew stream contained in the traffic. Discrete Fourier Transform (DFT) and Fast Fourier Transformation (FFT) is exactly the mathematics method to convert a finite

![Figure 1. LDDoS Attacks of Mirai 4 Bots](image-url)
sequence of equally-spaced samples of a function into a same-length sequence of equally-spaced samples of the discrete-time Fourier transform. Thus, we add frequency-domain representations of time-domain features Power Spectrum Density (PSD) converted by DFT/FFT as new features.

After that, features subset is chosen by enhanced information entropy. Information entropy is the average rate at which information is produced by a stochastic source of data. The measure of information entropy associated with each possible data value is the negative logarithm of the probability mass function for the value:

\[ S = -\sum_i P_i \log P_i \]  \hspace{1cm} (1)

When the data source produces a low-probability value, the event carries more “information” than when the source data produces a high-probability value. In order to use entropy in feature extraction, conditional probability is introduced into entropy definition. Now entropy is calculated based on event probability. Every feature value is treated as event. Thus, event entropy means the distribution of value. The re-definition of (1) is as following:

\[ S(i, l) = \int P(x|\omega_i) \log P(x|\omega_i) \, dx \]  \hspace{1cm} (2)

Here, \( S(i, l) \) is the entropy of \( i \)th feature value in single classification of \( l \). Thus, the expectation of all classification would be \( S(l) \), it is a reference point of different features. The formula can be defined as

\[ S(i) = E(S(i, l)) = \sum_l S(i, l) P(\omega_i) \]  \hspace{1cm} (3)

As to \( h \), which aims to maximize the probability of a “collision” for similar items, there are several open-sourced algorithms. We can use any one of them directly. Here, we use minimal distance algorithm family in Hamming space. Since it has a huge improvement compared to other algorithms.

**Algorithm 1: Locality Sensitive Features Extraction Algorithm**

**Input:** NetworkDataSet = mat(labeled dump network items); EntropyThresh;
**Output:** priComps = array(features). HashTable = array(hashBuckets)

**Procedure:**
1. Initialized FeatureList based on NetworkDataSet;
2. Set well-tuned EntropyThresh;
3. Calculate frequency-domain features based on time-domain features (PSD) based on DFT/FFT, and add it into FeatureList;
4. for features in FeatureList:
5. Calculate its entropy based on (3);
6. end for
7. Extract features in ascending order by probability value based on EntropyThresh;
8. for vector in NetworkDataSet:
9. Cast vector with extracted features by Hamming min-Distance algorithm, and generate h(vector);
10. end for
11. Generate hash buckets with same hash values;
12. Generate HashTable based on buckets.

**B. Incremental TSVM Algorithm**

TSVM extend SVMs in that they could treat partially labeled data in semi-supervised learning by following the principles of transduction. The key idea of TSVM is that it begins with a labeling of the test/predict data based on the classification of an inductive SVM. Then it improves the solution by switching the labels of test/predict examples so that the objective function decreases. In other words, TSVM can be treated as seeking an optimal solution in given labeled dataset: \( \{(x_i, y_i)\}, i = 1, 2, ..., n; x_i \in \mathbb{R}^n; y_i \in \{+1, -1\} \) and unlabeled dataset: \( x_1, x_2, ..., x_k \), and it can be described as following:

\[
\min V(y_1^*, ..., y_k^*, \omega, \epsilon_1, ..., \epsilon_k) = \frac{1}{2} ||\omega||^2 + C \sum_i^n \epsilon_i + C^* \sum_j^k \epsilon_j^*
\]

s.t. \( \forall i=1^n: y_i [\omega \cdot x_i + b] \geq 1 - \epsilon_i \) \hspace{1cm} (4)
\( \forall j=1^k: y_j^* [\omega \cdot x_j^* + b] \geq 1 - \epsilon_j^* \)
\( \forall i=1^n: \epsilon_i \geq 0 \)
\( \forall j=1^k: \epsilon_j^* \geq 0 \)

In (4), parameter \( C \) is the impact factors of hyperplane of labeled samples, while \( C^* \) is the impact factors of unlabeled samples; \( \epsilon_i \) and \( \epsilon_j^* \) are slack variables; \( <\omega, b> \) is the hyperplane. For obtaining the optimal solution of (4), the first step is to set well-tuned \( C \) and \( C^* \) by historical experiment, and to learn the initialized SVM classifier based on labeled samples through induction method. The second step is to use this SVM classifier to classify all unlabeled samples, and mark only pairwise labels of positive and negative samples in support vectors nearby. The last step is to re-train SVM based on new chosen samples and labeled samples. Training procedure is repeated between step 2 and 3, until all unlabeled samples have their own classification. From above procedure, there are two disadvantages of transduction method. The one is that TSVM builds no predictive model. And the other one is if a previously unknown point is added to the set, the entire transductive algorithm would need to be repeated with all of the points in order to predict a label.

In real IOT environment, we need not only train a predictive model on limited labeled samples, but also adjust the model by new coming packages in real-time. Obviously, TSVM can be computationally expensive in this kind of case. Further, this might cause the predictions of some of the old points to change, which is unexpected in early warning system and is treated as false positive. After an in-depth study of TSVM, we find that “not all samples” play the same important role during learning the optimal solution of (4). These samples that meets the support vector conditions contribute a lot to find out optimal hyperplanes and decision functions. In general, the \( x_i \) whose lagrange multiplier’s value \( a_i \) is between 0 and \( C \) (\( 0 < a_i < C \)) is defined as normal vector, while the \( x_i \) whose lagrange multiplier’s value \( a_i \) equals \( C \) (\( a_i = C \)) is defined as support vector. The later represents classified features of most samples, and can be learned to get the final classification. In other words, the set of support vectors can fully describe the features of the whole training set, learning in the set of support vectors is equivalent to learning in whole training set. Obviously, the scale of support vectors set is far smaller than training set. In every iterative training, only the support vectors are mentained and combined with the new added samples for new classifier. With this, TSVM has the capacity of incremental learning. The incremental TSVM algorithm can be described as following:
From algorithm 2, the biggest distinguished difference between incremental TSVM and SVM is that the learning samples of incremental SVM consist of only the support vectors of the learned samples and the new learning samples, while the SVM consists of all the learned samples and the new learning samples. Incremental TSVM discards some samples without losing classification accuracy. At the same time, its training time is speeded up.

C. Integrated with Locality Sensitive and ITSVM

Now that we have worked out locality sensitive features extraction method for locating and pre-labeling examples, at the same time, we have incremental TSVM for quickly incremental learning. Thus, we further integrate them together for semi-supervised learning quickly and trying to improve classification accuracy by a small margin.

Assumed that there is a small labeled dataset \( \Omega_o \), the new sequenced added unlabeled dataset \( \Omega_k (k = 1, \ldots, n) \). The support vector set correlated \( \Omega_k \) is \( \Omega_k^{sv} \). The hash table of \( \Omega_k \) is \( H_k \). \( h(\Omega_k) \) is used to represent hashed values vector. Samples set filtered by locality sensitive hash function is \( \Omega_k^{lish} \), \( f(x) \) is the classifier of ITSVM. With these, integrated algorithm of locality sensitive and ITSVM is detailed described as following:

**Algorithm 3: Integrated Algorithm of Locality Sensitive and ITSVM**

**Input:** \( \Omega_o, \Omega_k \),
**Output:** classifier \( f(x) \), predictedResults

**Procedure:**
1. create HashTable \( H_0 \) and \( \Omega_{sv} \) for \( \Omega_o \) per Algorithm 2 & 3;
2. while (\( \Omega_i \) != NULL & & i<=k):
   3. calculate hash vectors \( h(\Omega_i) \);
   4. for each hash vectors \( h(\Omega_i) \):
      5. if (vector exists in HashTable \( H_{i-1})
      6. pre-mark the vector per the correlated label of HashTable \( H_{i-1} \);
   7. end for
3. merge \( \Omega_{sv} \) into \( \Omega_{lish} \) for new\( \Omega_{lish} \);  
4. do
   5. train classifier \( f(x) \) based on \( \Omega_{lish} \) per Algorithm 3;
   6. classify \( \Omega_{lish} \) based on \( f(x) \);
   7. if (pre-marked values != predicted values)
      8. unmark these values, and treat them as unlabeled samples;
   9. end if
10. while (unlabeled samples != NULL)
   11. report predictedResults for \( \Omega_i \);
12. end while

IV. Experiments and Results

The proposed method should be a general and universal method, which has high detection accuracy and fast training and response time. In order to verify these, we carried out the following experiments, and made the comparison with Label propagation algorithm (LPA), and ISVM. In the experiment, we apply our algorithm in KDD CUP 1999 dataset, which can prove the algorithm is general and universal method, and it has a good performance than others.

A. Features extraction

**FIGURE 2. Time and Frequency Distribution of dst_bytes**

KDD CUP-99 dataset was provided by Lincoln Labs, who set up an environment to acquire nine weeks of raw TCP dump data for a local-area network simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks. The raw training data was about four gigabytes of compressed binary TCP dump data. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. In order to verify the above algorithm quickly, we use kddcup.data_10_percent as data
source. We randomly choose 50,000 data items as test set. We split the to-be-trained data set into labeled and unlabeled parts by 20%:80%.

Furthermore, each of those samples in “KDD CUP-99 10% dataset” has 41 features. 38 features of them are character and number related features. Thus, we pre-process them by reflecting and normalization. E.g. there are 3 choices in protocol_type: tcp, udp, icmp, which are reflected into 0, 1, 2. And then, we generate PSD by FFT and add them as new dimensions. E.g. in FIGURE 2, the normalized cumulative amplitude spectrum (NCAS) value of dst_bytes are shown. What exactly happens is that more power of the autocorrelation function of dst_bytes is distributed in the lower frequency band when there is LDDoS contained in the traffic. After above operations, dimensions of features are bumped to 110.

In order to avoid the influence of the dimension of feature attribute on the experiment, it is necessary to unify the dimension of the experimental data. Thus, we use (5) to normalize the feature attributes to $[0, 1]$. In (5), $x$ is the feature vectors. $\text{MIN}$ is the minimal value of $x$, while $\text{MAX}$ is its maximal value.

$$x = \frac{x - \text{MIN}}{\text{MAX} - \text{MIN}}$$  \hspace{1cm} (5)

After above necessary dimensionality expand and normalize. The entropies of them are calculated for feature extraction. We well tune the threshold to 50, whose entropy values are bigger than 0.1, and chose them as training features.

### B. Evaluation Indicators Analysis

In order to explain that the proposed method is better than others, we compare two indicators: detection accuracy score, training and response CPU time.

Detection accuracy refers to how closely a prediction comes to measuring a "true value", since prediction are always subject to error. Here, we use the ratio between predicted value and the corresponding true value detailed as following, where $1(x)$ is the indicator function.

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} 1(y_i = \hat{y}_i)$$  \hspace{1cm} (6)

As FIGURE 3 shown, the proposed locality sensitive ITSVM almost has the same accuracy with ISVM, even if we bump the semi-supervised increase set size to 100k. but locality sensitive ITSVM method has far better detection accuracy than LPA method. The accuracy of LPA is stick to 80% or so. And even worst, LPA is a huge memory-consumed method. It reports memory is not enough when it calculates label. Thus, there is not statistical data when size is bigger than 15k.

The training and detecting CPU time is the other critical parameter to evaluate the performance of locality sensitive ITSVM method. We define it as the time when method trains with semi-supervised incremental data and detects whether malicious flows exist or not. As FIGURE 4 shown, the CPU time of ISVM is increased continually with the growth of increase set size, and bumps to almost 100s from 60s. Same to accuracy, LPA method is stopped when size is bigger than 15k. Compared with them, our proposed method training and detecting time is stable in 50s.

Thus, take both detection accuracy score and CPU time into account, locality sensitive ITSVM, which has better performance, can detect DDoS with higher accuracy and shorter CPU time.

### V. CONCLUSION

In the paper, we develop a novel semi-supervised locality sensitive incremental TSVM method for resolving LDDoS issue based on Mirai botnet and its variants issues. The results based on KDD-CUP prove LS-ITSVM has better performance, and it can differ bots from normal users. But during experiment, we find the KDD-CUP dataset was not developed for LDDoS attacks. Thus, in the near future, firstly, we will focus on setting up an IOT LDDoS test bed to collect data and test these indicators in simulated environment and integrate above algorithms into real data center. Secondly, we will try to investigate Generative Adversarial Networks (GAN) or Federated Learning methods for generating better models.

### ACKNOWLEDGMENT

This research was supported by Doctoral Scientific Fund Project of Weifang University of Science & Technology of China (Grant number: 2017BS17) and was also supported by the Scientific Fund Project of Facility Horticulture Laboratory.
of Universities in Shandong of China (Grant number: 2018YY016).

REFERENCES

ZengGuang Liu was born in ShouGuan, China in 1992. He received the B.S. and M.S. degrees in computer science from the University of ShangHai for Science and Technology in 2005 and 2008 respectively. He is currently pursuing the Ph.D. degree in computer science at the University of Shandong for Science and Technology. From 2008 to 2016, he was a Senior Architecture Engineer in Core Network Department of Alcatel-Lucent. He had been a Principal AI Engineer of Hewlett Packard Enterprise since 2016. His research interests include big data, AI and network security areas. He was a recipient of the Innovation Award of Shandong for Excellence in 2011.

XiaoChun Yin (M’12) was born in ShouGuang, China in 1980. She received the B.S. degree in Education and Technology from QuFu Normal University in 2004, and received the M.S. degree in Education and Technology from Nanjing Normal University in 2007, and received the Ph.D. from Dongseo University in 2015.

From 2007 to now, she is an associate professor in Weifang University of Science & Technology. Her research interests include network security, IoT security, blockchain and AI areas.

Professor Yin was a recipient of the Science Research Innovation Award of Weifang University of Science & Technology in 2017 and 2018.

RuPiing Yang was born in Shandong, China in 1990. She received her B.Sc. (2014) degree and M.Sc. (2017) degree in mathematics from QuFu Normal University. She is currently a Ph.D. candidate in the College of Computer Science and Engineering, Shandong University of Science and Technology. Her current research interests include machine learning, deep learning and neural networks.

Anning Dong received the B.E. degree in electronic information science and technology from Liaocheng University in 2004, the M.E. degree in communications and information systems from Lanzhou University in 2007, and the Ph.D. degree in communications and information systems from Shandong University in 2016. He is currently an Associate Professor with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences). His research interests include MIMO techniques, wireless communications, wireless machine learning, and application of optimization techniques for signal processing and wireless communications.

He was a recipient of the Excellent Doctoral Dissertation Awards of Shandong Province, in 2017.