Machine Learning Approach for detection of nonTor Traffic

Elike Hodo[1],Xavier Bellekens[2], Ephraim Iorkyase[1], Andrew Hamilton[1], Christos Tachtatzis[1], Robert Atkinson[1]

Department of Electronic&Electrical Engineering

University of Strathclyde

Division of Computing and Mathematics

University of Abertay Dundee

E-mail[1]: {firstname.surname@strath.ac.uk}

E-mail[2]: {x.bellekens@abertay.ac.uk}

ABSTRACT

Intrusion detection has attracted a considerable interest from researchers and industries. After many years of research the community still faces the problem of building reliable and efficient intrusion detection systems (IDS) capable of handling large quantities of data with changing patterns in real time situations. The Tor network is popular in providing privacy and security to end user by anonymising the identity of internet users connecting through a series of tunnels and nodes. This work focuses on the classification of Tor traffic and nonTor traffic to expose the activities within Tor traffic that minimizes the protection of users. A study to compare the reliability and efficiency of Artificial Neural Network and Support vector machine in detecting nonTor traffic in UNB-CIC Tor Network Traffic dataset is presented in this paper. The results are analysed based on the overall accuracy, detection rate and false positive rate of the two algorithms. Experimental results show that both algorithms could detect nonTor traffic in the dataset. Artificial neural network proved a better classifier than SVM in detecting nonTor traffic in UNB-CIC Tor Network Traffic dataset.

Keywords

Artificial neural network, support vector machines, intrusion detection systems Tor and nonTor, UNB-CIC Tor Network Traffic dataset.

1. INTRODUCTION

The computing world has changed over the past decade due to the rapid development of internet and the network systems, network and computer systems traffic are susceptible to intrusion and this has been a great concern to the research community and industries.

Over the last decade, traffic classification has advanced in its applications in systems like quality of service (QoS) tools or Security information and Event management(SIEM) [1]. A considerable interest have been attracted from researchers and the industries to the study of these technologies and developing classification techniques [2][1].

The Tor networks help to provide privacy and anonymity over the internet but have been an obstacle in the classification of internet traffic. It operates by anonymising the identity of users connecting through a series of tunnels and nodes. Tor networks do not detect anomalies in traffic flow and do not have the ability to raise an alarm when an anomaly occurs.

To this effect intrusion detection system (IDS) plays an important role in Tor networks. Intrusion

Detection Systems are placed on the networks and computer systems to monitor and detect anomalies. In general IDS can be categorised into two components, based on the detection technique. Signature based and Outlier based IDS. Most IDs employ a signature based detection approach where the network traffic is monitored and compared against database rules or signature of known anomaly in network traffic [3][4]. An alarm is raised on detection of a mismatch. Signature based is the most common as they do not necessarily have to learn the network traffic's behaviour. Although it is effective in detecting known anomalies, it cannot detect unknown anomalies unless the signature and rules are updated with new signatures [5][6]. Signature based is known to have a significant time lapse between anomaly detection and activation of its corresponding signature [4]. Signature based techniques are mainly human-dependent in creating, testing and deploying signatures.

The outlier technique is a behavioural based detection system. It observes changes in normal activity of network traffic and builds a profile of the network traffic being monitored [7][8]. An alarm is raised whenever a deviation from the normal behaviour is detected. It has the ability to detect unknown anomalies. However outlier detection based IDS have the disadvantage of being computational expensive because the profile generated over a period needs to be updated against each system activity [9] [4]. Machine learning techniques have the ability to learn the normal and anomalous patterns automatically by training a dataset to predict an anomaly in network traffic. One important characteristic defining the effectiveness of machine learning techniques is the features extracted from raw data for classification and detection. Features are the important information extracted from raw data. The underlying factor in selecting the best features lies in a trade-off between detection accuracy and false alarm rates. The use of all features on the other hand will lead to a significant overhead and thus reducing the risk of removing important features. Although the importance of feature selection cannot be overlooked, intuitive understanding of the problem is mostly used in the selection of features [10]. A study by A.Lashkari *et al*.[1] Showed Decision Tree C4.5 classifies Tor and nonTor traffic on UNB-CIC Tor Network Traffic dataset [11] with a high precision and recall. In the study by [1] ,23 time based features were extracted from the traffic and was reduced to 5 for training and testing of the machine learning algorithms.

This paper analyses the performance of Artificial neural network (ANN) and Support vector machines (SVM) in terms of overall accuracy in detecting nonTor traffic in a Tor network traffic dataset data from the University of New Brunswick (UNB), Canadian Institute for cyber security (CIC) using an anomaly based approach with all features in the dataset. As part of the work, the results are compared with the results of [1] being the only study published to the best of our knowledge using the UNB-CIC Tor Network Traffic dataset. In the proposed approach 10 features are selected out of the 28 features of the dataset using Correlation based feature selection (CFS) for training and testing the detection algorithms.

The rest of the paper is organised as follows: section 2 describes intrusion detection systems, section 3 describes the UNB-CIC Tor Network Traffic dataset, section 4 introduces Artificial neural network and Support vector machines algorithms used in the experiment respectively, section 5 analysis experimental results, conclusion and future works are presented in section VI.

2. INTRUSION DETECTION SYSTEMS

Intrusion detection system is a software application or a device placed at strategic places on a network to monitor and detect anomalies in network traffic [12][13] as shown in Figure 1. The main features of IDS are to raise an alarm when an anomaly is detected. A complementary approach is to take corrective measures when anomalies are detected, such an approach is referred to as an intrusion Prevention System (IPS) [14]. Based on the interactivity property of IDS, it can be designed to work either on-line or off-line. On-line IDS operates on a network in real time by analysing traffic packets and applying rules to classify normal and analogous traffic. Off-line IDS operates by storing data and after processing to classify normal and anomaly.

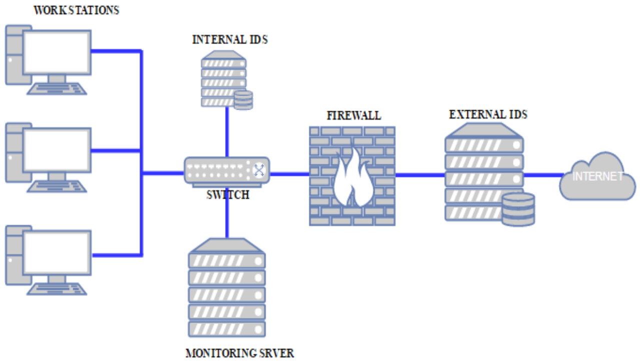


Figure 1. Intrusion detection system model

3. UNB-CIC TOR NETWORK TRAFFIC DATASET

3.1 UNB-CIC tor network traffic dataset description

UNB-CIC Tor Network Traffic dataset [8] is a representative dataset of real-world traffic defined as a set of task. Three users were set up for browser traffic collection and two users for the communication parts such as chat, mail, p2p etc. from more than 18 representative applications such as Facebook, skype, Spotify, Gmail etc. The dataset contains 8 types of Tor traffic as shown in table I and non-Tor traffic. The dataset contains 8044 (11.86%) records of Tor traffic and 59790 (88.14%) records of nonTor traffic. The non-Tor traffic captured in the dataset contains unique characteristics differentiating it from the Tor traffic. These characteristics are called features. The UNB-CIC Tor Network Traffic dataset contains a total of 28 features listed in tables II. The features were generated by a sequence of packets having the same values for {source IP, source Port, destination port and protocol (TCP and UDP)}. All Tor traffic was TCP since the flow does not support UDP. The generation of flows was done by a new application, the ISCXFlowMeter which generates bidirectional flows [7].

Table I. Description of UNB-CIC Tor Network Traffic

No.	Type of Traffic	Description
1	Browsing	HTTP and HTTPS traffic generated by users while using Firefox and chrome
2	Email	Traffic samples generated using a Thunderbird client and two other accounts holders. Mails were delivered through SMTP/S and received using POP3/SSL in client 1 and IMAP/SSL in client 2.
3	Chat	Instant messaging applications were identified under the chat label. The label was associated with Facebook and hangouts through web browser, skype and IAM and ICQ using an application called pidgin.
4	Audio-Streaming	Traffic was captured from Spotify identifying audio applications that require a continuous and steady stream of data.
5	Video-Streaming	Traffic was captured from YouTube and Vimeo services using Chrome and Firefox identifying video applications that require a continuous and steady stream of data.
6	File Transfer	This traffic was generated from skype file transfers, FTP over SSH (SFTP) and FTP over SSL (FTPS) traffic sessions identifying the traffic applications sending or receiving file documents.
7	Voice over Internet Protocol (Voip)	This is the traffic generated by voice applications using Facebook, Hangouts and Skype.

8	Peer-peer (p2p)	This label identifies file sharing protocols like Bit torrent. The traffic was generated by downloading different torrent files from the Kali Linus distribution and captured traffic using Vuze application on different combinations of upload and download speeds.

Table II. Description of captured features

No.	Feature Name	Feature description
1	Source IP	IP address sending packets from source to destination
2	Source Port	Port sending packets from source
3	Destination IP	IP address receiving packets from source
4	Destination Port	Port receiving packets
5	Protocol	Type of the protocol used, e.g. udp,tcp, etc
6	Flow Duration	Length of connection in seconds
7	Flow Bytes/s	Number of data bytes
8	Flow Packets/s	Number of data packets
9	Flow IAT Mean	Packets flow inter arrival time Mean
10	Flow IAT Std	Packets flow inter arrival time Standard deviation
11	Flow IAT Max	Packets flow inter arrival time Max.
12	Flow IAT Min	Packets flow inter arrival time Min.
13	Fwd IAT Mean	Forward inter arrival time, the time between two packets Sent forward direction Mean.
14	Fwd IAT Std	Forward inter arrival time, the time between two packets sent forward direction Standard deviation.
15	Fwd IAT Max	Forward inter arrival time, the time between two packets sent forward direction Max.
16	Fwd IAT Min	Forward inter arrival time, the time between two packets sent forward direction Min.
17	Bwd IAT Mean	Backward inter time, the time between two packets sent backward Mean.
18	Bwd IAT Std	Backward inter arrival time, the time between two packets sent backward Standard deviation.
19	Bwd IAT Max	Backward inter arrival time, the time between two packets sent backward Max.
20	Bwd IAT Min	Backward inter arrival time, the time between two packets sent backward Min.
21	Active Mean	The amount of time a flow was active before becoming idle mean.
22	Active Std	The amount of time a flow was active before becoming idle Standard deviation.
23	Active Max	The amount of time a flow was active before becoming idle Max.
24	Active Min	The amount of time a flow was active before becoming idle Min.
25	Idle Mean	The amount of time a flow was idle before becoming active Mean.
26	Idle Std	The amount of time a flow was idle before becoming active Standard deviation.
27	Idle Max	The amount of time a flow was idle before becoming active Max.
28	Idle Min	The amount of time a flow was idle before becoming active Min.

4. DETECTION ALGORITHMS

4.1 Artificial Neural Network

Artificial neural network (ANN) consists of information processing elements known to mimic neurons of the brain.

In this experiment, the neural network which is a Multilayer perceptron (MLP) is provided with a set labelled training set which learns a mapping from input features listed in table II represented as x in Figure 2. to outputs (Tor and NonTor) as y in Figure 2. given a labelled set of inputs-output pairs

$$d = \{(x_i, y_i)\}_{i=1}^{N} \quad (4)$$

Where, d is called the training set and N is the number of training examples. It is assumed that y_i is a categorical variable from some infinite set, $y_i \in \{1 \dots C\}$ [15].The technique used to train the MLP neural network is the Back Propagation hence the name MLP-BP. The construction of the MLP-BP neural network is by putting layers of non-linear elements to form complex hypotheses. Each node takes an element of a feature vector. The structure of the ANN consists of three layers feed-forward neural network as shown in Figure 2. Nodes labelled $x_1 \dots \dots x_n$ have been used to represent the input feature vectors to the ANN.

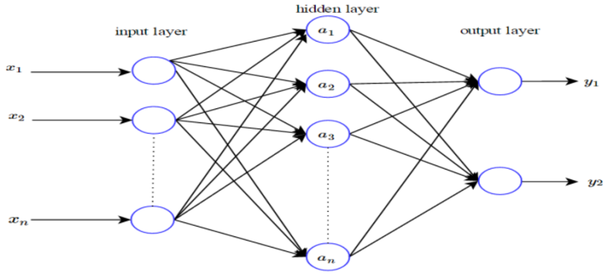


Figure 2. ANN model used in experiment

Hidden inner nodes $a_1 \dots \dots a_n$ make up the hidden layer with an output layer of y_1 *and* y_2 nodes denoting different output classes (Tor and NonTor). The interconnection between the nodes is associated with scalar weights with an initial weight assigned to the connection. During training, the weights are adjusted. Evaluating the hypotheses is done by setting the input modes in a feed-back process and the values are propagated through the network to the output. At this stage the gradient descent is used so as to push the error in the output node back through the network by a back propagation process in order to estimate the error in the hidden nodes. The gradient of the cost – function is then calculated [16].

4.2 Support Vector Machines

Support Vector Machines (SVM) is a machine learning algorithm that learns to classify data using points labelled training examples falling into one or two classes. The SVM algorithm builds a model that can predict if a new example falls into one category or the other [17]. The model is constructed by constructing k Models of SVM, where k denotes the number of classes (Tor and NonTor). x_1 *and* y_2 SVM represented as lth SVM is trained with all the examples in the lth class labelled 1 and the other labelled 0.

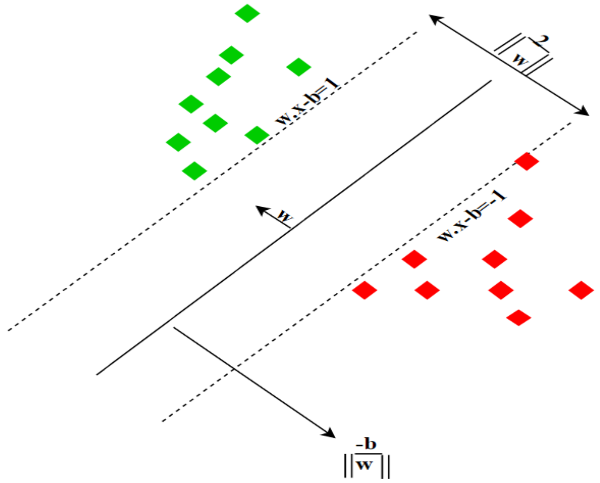


Figure 3. Maximum-margin hyper plane and margins for an SVM trained with samples from two classes.

Where,$x_i \in R^d$, $y_i \in \{1,0\}$,$i = 1 \ldots . . N$ and $y_i \in \{1 \ldots \ldots k\}$ is a class of x_i. Introducing a slack of positive variables ξ_i, that measures the extent of constraint in a non-linear situation. The prima Optimisation problem becomes [18]:

$$\min_{w^l, b^l, \xi^l} \ \frac{1}{2}(w^l)^T w^l + C\sum_{i=1}^{N} \xi_i^l$$

$$(w^l)^T \phi(x)_i + b^l \geq 1 - \xi_i^l, \quad if\ y_i = N,$$

(6)

$$(w^l)^T \phi(x)_i + b^l \leq -1 + \xi_i^l, \ if\ y_i = N,$$

$$\xi_i^l \geq 0, \ i = 1 \ldots \ldots N,$$

Where the training set x_i are mapped into higher dimensional space by the function ϕ and C, where C is a parameter which trades off wide margin with small number of margin failures . Minimisation of $\frac{1}{2}(w^l)^T w^l$ implies maximising $\frac{2}{\|w^l\|}$, which is the margin between the two data points. The SVM then searches for a balance between the regularisation term $\frac{1}{2}(w^l)^T w^l$ and the errors in training the dataset. Solving (6) gives k decision functions:

$$(w)^{1^T} \phi(x) + b^1$$

$$\vdots$$

$$(w)^{k^T} \phi(x) + b^k$$

where x is the class having the largest value of the decision function:

$$x \equiv argmax_{l\equiv 1 \ldots k}((w^l)^T \phi(x) + b^l) \qquad (7)$$

The dual problem of (6) having the same number of variables as the number of data in (6). Thus k N-variable quadratic programming problems are solved.

5. EXPERIMENTAL RESULTS ANALYSIS

5.1 Results Evaluation Metrics

The effectiveness of IDS requires high accuracy, high detection rate (Recall) and high Positive Predictive value (Precision) as well as low false positive rate. The performance of IDS in general is evaluated in terms of overall accuracy, detection rate and false positive rate. The confusion metrics shown in table III is used to evaluate these parameters.

Table III. Confusion Metrics

		Predicted Class	
		Tor traffic	Nontor traffic
Actual Class	Tor traffic	TN	FP
	Nontor traffic	FN	TP

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

Detection Rate (DR)= $\frac{TP}{TP+FP}$

False Positive rate (FPR)= $\frac{FP}{FP+TN}$

Positive Predictive Value (PPV)= $\frac{TP}{TP+FP}$

Where, True Negative (TN): a measure of the number of normal events rightly classified normal.

True Positive (TP): a measure of attacks classified rightly as attack.

False Positive (FP): a measure of normal events misclassified as attacks.

False Negative (FN): a measure of attacks misclassified as normal.

5.2 Feature Selection

This paper proposes correlation based feature selection (CFS) to select the relevant features out of the 28 features.

The CFS is an algorithm that is a heuristic evaluating with an appropriate correlation measure. A good set of features are highly correlated with the target (class) and at the same time uncorrelated to each other. The CFS algorithm reduces the dimensionality of the dataset, reduces overfitting and gives a shorter training time. Table IV shows the 10 selected features based on the appropriate correlation measure and heuristic search strategy.

Table IV. CFS features selection

No.	**Feature Name**	**No.**	**Feature Name**
1	Destination Port	6	Idle Min
2	Bwd IAT Mean	7	Flow Bytes/s
3	Idle Max	8	Flow IAT Std
4	Fwd IAT Min	9	Source IP
5	Source Port	10	Destination IP

5.3 Experimental Results

Neural network and Support vector machine classification involves two phases: the classification phase and training phase as shown in Figure 4. In the training phase, the algorithm learns the distribution of the features with corresponding classes. During the classification phase, the learned model is applied to a test set which has not been previously seen by the training phase.

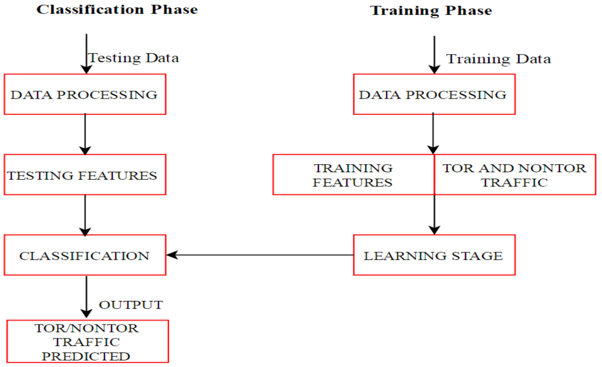


Figure 4. Experimental Model

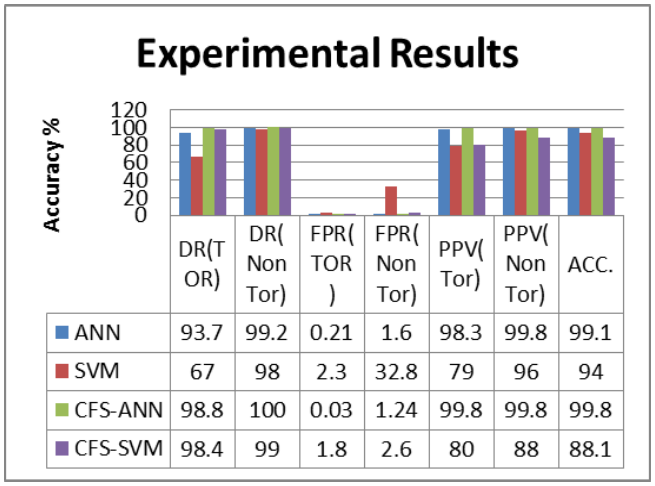
In this work, experiment was performed by training ANN and SVM with UNB-CIC Tor Network Traffic dataset to detect nonTor Traffic.

In the first set of experiment, ANN was trained with all 28 features of the dataset with 20-hidden neurons and with 10 features selected using CFS with 6-hidden nodes. The ANN uses Levenberg-Marquardt training function (trainlm) for learning.

In the second set of the experiment SVM was trained with all 28 features in the dataset and with 10 features selected using CFS.

The performance of ANN and SVM was evaluated on train (70%) dataset, test (15%) dataset and validation 20% dataset.

Figure 5 shows the results after training ANN and SVM with all 28 features and 10 features selected by CFS. A.Lashkari *et al.*[1] Proposed C 4.5 in classifying Tor Traffic and nonTor Traffic using only the time based features of the dataset.



	DR(TOR)	DR(Non Tor)	FPR(TOR)	FPR(Non Tor)	PPV(Tor)	PPV(Non Tor)	ACC.
🟦 ANN	93.7	99.2	0.21	1.6	98.3	99.8	99.1
🟥 SVM	67	98	2.3	32.8	79	96	94
🟩 CFS-ANN	98.8	100	0.03	1.24	99.8	99.8	99.8
🟪 CFS-SVM	98.4	99	1.8	2.6	80	88	88.1

Figure 5 Experimental results of SVM and ANN

Table V. Experimental results compared to A.Lashkari *et al.*[1]

	DETECTION ALGORITHM				
PERFORMANCE	**ANN**	**CFS-ANN**	**SVM**	**CFS-SVM**	**C4.5 [1]**
DR (Tor) %	93.7	98.8	67	98.4	93.4
FPR (Tor) %	0.21	0.03	2.3	1.8	-
PPV (Tor) %	98.3	99.8	79	80	94.8
DR (nonTor) %	99.2	100	98	99	99.2
FPR (nonTor) %	1.6	1.2	32.8	2.6	-
PPV(nonTor) %	99.8	99.8	96	88	99.4
Overall ACC. %	**99.1**	**99.8**	**94**	**88.1**	-

The experimental results as compared to the results in [1] show CFS-ANN performs with an overall accuracy of 99.8% in the classification of Tor and nonTor using only 10 features in the dataset. On the other hand the DR for Tor and nonTor Traffic in CFS-ANN recorded 98.9% and 100% respectively which performed better than C 4.5. The best values in detection accuracy, detection rate with a low false positive rate in the classification of Tor and nonTor traffic were recorded by CFS-ANN making it a promising detection system for nonTor traffic. A comparison of experimental results with results from [1] are shown in table V.

6. Conclusions

This paper presents experimental study using two algorithms to detect nonTor traffic in UNB-CIC Tor Network Traffic dataset. The research mainly focuses on detecting nonTor traffic in a representative dataset of real-world traffic to expose the activities within the Tor-traffic that downgrades the privacy of users. The work proposes CFS-ANN hybrid classifier in the detection of nonTor traffic in UNB-CIC Tor Network Traffic dataset. Experimental results show the proposed algorithm detects nonTor with an accuracy of 99.8%, detection rate of 100% and false positive rate of 1.2%. The proposed algorithm performed better than SVM and C4.5 proposed in [1] as shown in table IV. The proposed hybrid classifier reduces the dimensionality of the data size by 65% removing the less effective features thereby lowering computational cost and training time.

In the future, the performance of the proposed algorithm and deep neural networks will be analysed in the classification of the 8 different types of traffic in the UNB-CIC Tor Network Traffic dataset.

7. ACKNOWLEDGMENTS

The authors would like to thank the Canadian Institute of Cyber Security, University of Brunswick for providing us with the dataset to carry out this work.

8. REFERENCES

[1] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, “Characterization of Tor Traffic using Time based Features,” pp. 253–262, 2017.

[2] T. Nguyen and G. Armitage, “A survey of techniques for internet traffic classification using machine learning,” *IEEE Commun. Surv. Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[3] Roesch and Martin, “Snort - Lightweight Intrusion Detection for Networks,” in *Proceedings of the 13th USENIX conference on System administration*, 1999, pp.

229–238.

[4] B. Subba, S. Biswas, and S. Karmakar, “A Neural Network based system for Intrusion Detection and attack classification,” in *2016 Twenty Second National Conference on Communication (NCC)*, 2016, pp. 1–6.

[5] G. A. Haidar and C. Boustany, “High Perception Intrusion Detection System Using Neural Networks,” in *2015 Ninth International Conference on Complex,*

Intelligent, and Software Intensive Systems, 2015, pp. 497–501.

[6] X. J. A. Bellekens, C. Tachtatzis, R. C. Atkinson, C. Renfrew, and T. Kirkham, “A Highly-Efficient Memory-Compression Scheme for GPU-Accelerated Intrusion Detection Systems,” *Proc. 7th Int. Conf. Secur. Inf. Networks - SIN ’14*, pp. 302–309, 2014.

[7] N. K. Mittal, “A survey on Wireless Sensor Network for Community Intrusion Detection Systems,” in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, pp. 107–111.

[8] J. Shun and H. a. Malki, “Network Intrusion Detection System Using Neural Networks,” *2008 Fourth Int. Conf. Nat. Comput.*, vol. 5, pp. 242–246, 2008.

[9] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, “Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System,” in *2016 3rd International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, pp. 1–6.

[10] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova, and A. A. Ghorbani, “Towards effective feature selection in machine learning-based botnet detection approaches,” in *2014 IEEE Conference on Communications and Network Security*, 2014, pp. 247–255.

[11] “Canadian Institute for Cybersecurity | Research | Datasets | UNB.” [Online]. Available: http://www.unb.ca/cic/research/datasets/tor.html. [Accessed: 27-Mar-2017].

[12] D. Rozenblum, “Understanding Intrusion Detection Systems,” *SANS Inst.*, no. 122, pp. 11–15, 2001.

[13] E. Hodo, X. Bellekens, A. Hamilton, and C. Tachtatzis, “Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey.” [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1701/1701.02145.pdf. [Accessed: 31-Mar-2017].

[14] “What it is Network intrusion detection system? | COMBOFIX.” [Online]. Available: http://www.combofix.org/what-it-is-network-intrusion-detection-system.php. [Accessed: 10-Dec-2015].

[15] K. Murphy, “Machine learning: a probabilistic perspective,” *Chance encounters: Probability in …*, 2012. [Online]. Available: http://link.springer.com/chapter/10.1007/978-94-011-3532-0_2. [Accessed: 06-Jan-2015].

[16] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, “Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[17] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167.

[18] W. Hu, Y. Liao, and V. R. Vemuri, “Robust Support Vector Machines for Anomaly Detection in Computer Security.”