Detecting Selected Network Covert Channels Using Machine Learning

Mehdi Chourib

Detecting Selected Network Covert Channels Using Machine Learning

Mehdi Chourib
Faculty of Mathematics and Computer Science
FernUniversität in Hagen
Hagen, Germany
mehdi.chourib@fernuni-hagen.de

***Abstract*—Network covert channels break a computer's security policy to establish a stealthy communication. They are a threat being increasingly used by malicious software. Most previous studies on detecting network covert channels using Machine Learning (ML) were tested with a dataset that was created using one single covert channel tool and also are ineffective at classifying covert channels into patterns. In this paper, selected ML methods are applied to detect popular network covert channels. The capacity of detecting and classifying covert channels with high precision is demonstrated. A dataset was created from nine standard covert channel tools and the covert channels are then accordingly classified into patterns and labelled. Half of the generated dataset is used to train three different ML algorithms. The remaining half is used to verify the algorithms' performance. The tested ML algorithms are Support Vector Machines (SVM), k-Nearest Neighbors (k-NN) and Deep Neural Networks (DNN). The k-NN model demonstrated the highest precision rate at 98% detection of a given covert channel and with a low false positive rate of 1%.**

***Index Terms*—Network steganography, Information hiding, Active warden, Data leakage protection.**

I. Introduction

Although most organizations and critical infrastructures [57] enforce their network perimeter defense by means of common security controls such as firewall or anti-malware tools, cyber attacks and particularly data breaches [3] continue to increase. In these situations, where information is either infiltrated or exfiltrated using techniques that are not supposed to be authorized by computer policy, security controls can become ineffective.

In computer security, these techniques are called covert channels [9]. They are considered as threats and defined as communication channel[s] that can be exploited by a process to transfer information in a manner that violates system security policy [5], [21]. Covert channels can be used maliciously to transfer data from one subject to another [15] and are commonly classified into storage, timing and hybrid channels [56]. Storage covert channels use a storage carrier to encode covert data [21]. Timing covert channels employ modulation on the shared resources by the sender, which impacts the response time observed by the receiver. Hybrid channels on the other hand refer to covert channels that use both timing and storage. Because our focus is on storage covert channels, this paper will not further discuss the other two.

Scrubbers, normalizers and wardens are used to defend a network from storage covert channels. Scrubbers and normalizers eliminate any ambiguities identified in the traffic [46]. The detection of a Network Covert Channel (NCC) is the primary objective of the warden. A warden is a single node in a network that intends to unveil, limit or eliminate any hidden communication [43]. To achieve these objectives, this warden network typically employs one of four types of approaches: passive, active, proactive or reactive [56], [44]. The passive approach relies on the monitoring of network traffic, yet the active approach influences the traffic through many ways to delete ambiguous data from TCP/IP [7] header fields. The proactive approach deliberately add crafted covert packets to figure whether any hidden communication is occurring. When the warden takes an action by only targeting covert communication, it is called reactive.

The fundamental deficiencies on the existent wardens are two. Firstly, the need of human intervention to insert rules that are capable of identifying ambiguities on the network traffic. Secondly, the warden is limited in the sense that it can only detect threats that are already known. A Network Anomaly Detection System (NADS) circumvents the mentioned deficiencies by encountering unusual patterns in the network traffic that are non-compliant with expected normal behavior. NADS can be statistically-based, classification-based, clustering-based or information theory-based. Currently, the anomalies can fall into three groups: collective, contextual and point. Point anomalies refer to any deviation of particular data from a normal pattern of a dataset. When anomalies occur in a particular context, they are designated as a contextual group. Collective anomalies are the correlation of similar anomalies within an entire dataset [54], [36]. The main limitations of NADS are:

1) The inherent need to define the notion of the traffic's "normality". Actually, an object is considered as anomalous if its rate of deviation within the defined profile of normal is adequately high.
2) NADS usually requires human intervention for analyzing, interpreting and acting on the generated alert by the infected systems.
3) The variation detection of network traffic with known anomalies by NADS has not been dealt with in depth.

> Indeed, one of the most powerful and cheapest tactics for a cyber-attacker to evade security countermeasure is to develop new variants from existing covert channel techniques.

The related academic literature on detecting NCC with ML has mainly two limitations. Firstly, the proposed detection schemes were largely tested on very limited covert channel techniques. Secondly, little attention has been given on classifying covert channels into patterns.

The primary contributions of this paper are: (1) analyzing eleven popular NCC tools and classifying them accordingly into patterns [28]; (2) designing a proof of concept for the detection and classification of NCCs using three different ML algorithms; (3) comparative evaluation of the used ML algorithms; and (4) identification and discussion of the results and indicating possible future research directions.

We present our analysis of related work in Section II. In subsequent sections (Sections III-IV), we describe the detection scheme and discuss the experimental measurements. Our findings are discussed in Section V and conclusion in Section VI.

II. Fundamentals & Related Work

The traditional approach on detecting NCCs (i.e. signature-based, behavior-based, heuristic-based) relies on the manual definition of signatures. This approach often fails to detect novel threats. To circumvent this problem ML is widely used [14], which is capable of modelling the normal behaviour of a network traffic and consequently can detect any unexpected behavior within the network traffic without (or with minor) human intervention. ML aims at making computer systems adapt their actions so that these actions get more accurate [58] ML techniques are generally grouped into two categories: supervised and unsupervised. The supervised category learns from a set of labelled data and encodes that learning into a model to predict an attribute for new data. On the other hand, unsupervised ML is used to find patterns within data that are without a specified target variable [24]. Supervised ML, which is also called classification, is characterized by the use of a labelled dataset. Classification is defined by the creation of a model by the training the labelled dataset. The created model is then used to predict the label of a certain dataset with an unknown label. We distinguish three families of datasetss as follows [1]:

1) *Synthetic:* Created to fulfill special requirements in relation with real data circumstances.
2) *Benchmark:* Generated on a simulated environment along with network devices.
3) *Real life:* Prepared by collecting network traffic during a certain period of time.

The identification and detection of covert channels are distinct. The identification aims to determine a shared resource that could be utilized as covert carrier. However, the detection examines the event flow in order to reveal a covert channel in operation. To minimize any negative impact on performance, the detection mechanism should be implemented before the elimination one. There are mainly three strategies to detect network covert channels: signature-based, anomaly-based and specification-based [19]. The signature-based detection requisites the creation of a baseline of signatures that need regular maintenance and update (signatures are typically patterns that a warden should monitor). This type of detection is typically the pattern that a warden should monitor to detect covert channels. The anomaly-based detection approach identifies any deviation from the normal traffic. Lastly, the specification-based approach intends to match the predefined specifications of a protocol to verify any misuse or attacks. The capabilities of the listed detection approaches are limited to the known NCCs. These limitations could be circumvented by the use of ML, which is referred to as the studies of automatic techniques for learning to make accurate predictions based on past observations [45].

There are three types of Intrusion Detection System (IDS) using ML: single, hybrid and ensemble. When an IDS uses an individual ML algorithm it is called single, hybrid IDS uses several algorithms. However, ensemble IDS refers employs a combination of several weak ML algorithms [27].

Various ML algorithms have been proposed in covert channels related literature. For instance, Hidden Markov Model (HMM) was used to detect covert communication on the TCP stack [30]. The main pitfall of it though is the limitation of the algorithm on detecting covert communication in applications that use tunneling. Gilbert and Bhattacharya [50] suggested a twofold detection system that features both covert channel profiling and anomaly detection. The genetic algorithm was used in the IDS first in 1995 by applying a hybrid approach of multiple agents and genetic programming in order to detect anomalies [13], [33]. Some enhanced ML techniques include the Intelligent Heuristic Algorithm (IHA) based on Naive Bayes classifiers to detect covert in IPv6 [41]. Salih et al. [22] improved the detection rate and reached an accuracy of 94% by using enhanced decision trees C4.5 with a very low false negative rate. C4.5 was also applied to detect protocol switching covert channels (PSCC) in [55]. The main inconvenience of the supervised method is that it requires labelled information for efficient learning. Additionally, it can hardly deal with the relationship between consecutive variations of learning inputs without additional prepossessing.

There is a considerable number of works that have used Support Vector Machine (SVM) to classify network anomalies [24], [16], [2], [4], [10], [11], [12], [20], [23], [35]. Compared with other ML algorithms, SVM has faster processing and is capable of processing both supervised and unsupervised learning [26], [17], [6]. For instance, SVM was used in a passive warden to detect TCP anomalies within TCP ISN and IP ID [25] or IPv4 network anomaly detection [29], [8]. SVM could be used to classify patterns based on statistical learning techniques for the regression and the categorization [23], [4]. This algorithm aims to achieve the optimal separating hyperplane in a higher dimensional feature space by using a kernel function.

On the other hand, it has been demonstrated that k-NN is one of the simplest ML algorithms [31]. Firstly, it classifies the entire dataset into training and testing data points. Secondly, it evaluates the distance from all training points to the testing points. The point that has the lowest distance is named nearest neighbor. Tsai et al. [34] suggested a hybrid method based on a triangle area using the k-NNs approach to detect attacks. suggested a hybrid method based on a triangle area using the k-NNs approach to detect attacks. They extract the number of center clusters where each cluster center constitutes one specific type of attack. Then, the triangle area is calculated by two clusters chosen randomly and one data point from the dataset. Finally, the constituted triangle symbolizes one new feature for measuring similar attacks. This k-NN classifier is used based on the feature of triangle areas to detect intrusions. Most studies on the detection of storage covert channels were tested with a single popular tool (e.g. Covert-TCP), [47], [48], [51], [49] [42], [52], [53] with captured traffic [45], [46], [44], [48] or with a personalized developed tool for the purpose of research work [47]. The authors of this paper propose a detection concept that uses ML with 3 different algorithms, which are not based on own particular developed techniques but on popular tools instead.

III. Detection Approach

The proposed approach is based on three steps: (1) generating the datasetss containing network traffic, (2) training and feature extraction, and (3) testing the models with different tools.

A. Generating the datasets

In order to train the ML algorithms, datasets were created of a mixture of real life and benchmark network packets [1] (ge umhnerated in a lab environment). For the benchmark dataset we have collected a set of tools as listed on Tab. II and covert traffic was produced. Then, PCAP files were labelled according to the type of pattern the covert packets belonged to.
Wendzel et al. introduced a pattern-based classification of covert channels. 109 covert channels were categorized into 11 distinct patterns based on their similarities [28]. For example, the pattern P7 represents covert channels that encode data into a reserved or unused field. To train the ML models, large labelled datasets (supervised) that represent each type of pattern were used. As shown in Tab. I, this research work focuses only on the following patterns:

TABLE I
List of patterns

Pattern	Tittle	Description
P0	Normal packet	Non covert
P1	Size modulation	The covert channel uses the size of a header element or of a PDU to encode the hidden message.
P5	Random Value	The covert channel embeds hidden data in a header element containing a random value.
P7	Reserved/Unused	The covert channel encoded hidden data into a reserved or unused heaer/PDU element.
NP0157	Non-P0-P1-P5-P7	Different patterns from the above

Labels are saved in a CSV file, so that each packet of the PCAP file is numbered and labelled accordingly.

TABLE II
Classification of covert channels tools

Tool	Protocol	Field	Pattern
ishell	ICMP	Reserved	P7
icmp-covert-channel	ICMP	Echo reply	P7
		Echo request	P7
		Payload	P7
dns2tcp	DNS	TXT records	P7
dns-tunnel	DNS	Transaction id	P7
sean	IP	IP address	P5
	IP	source port	P5
	IP	TTL	P5
	TCP	WIN	P5
	TCP	TOS	P5
	TCP	ACK NUM	P5
spinfoo	TCP	SYN	P5
iodine	DNS	TXT	P1
	DNS	SRV	P1
	DNS	MX	P1
	DNS	CNAME	P1
	DNS	A records	P1
tunnelshell	UDP	Many fields	P1
	TCP		P1
	ICMP		P1
	IP		P1
mawi.wide	TCP/IP	Many fields	P0
Covert-tcp	TCP	Spaces in TCP	NP0157
Jordan marling	UDP	src & dst port	NP0157

B. Training and features extraction

As we use supervised ML, our training process requires a pair of files (PCAP and CSV) and uses the Pcap2scikit class from scikit-learn (Python ML library). Each time the model is to be trained, the script checks the previous training model and adds it to the new data. If there is no previous data, then the model is newly created (Fig. 1). At the same time, features are extracted from each packet.

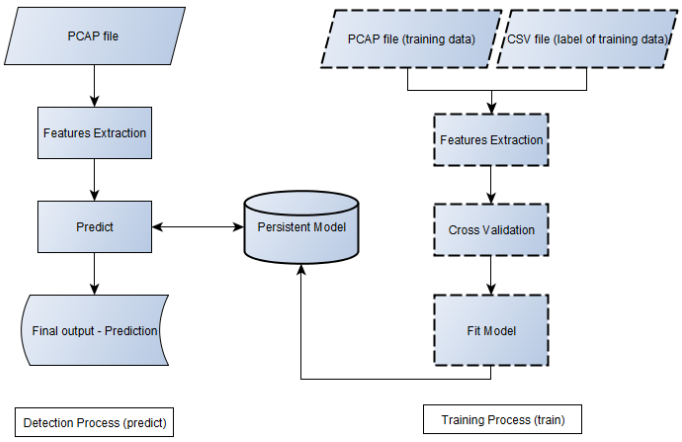

Fig. 1. Detection process

The features are extracted when preprocessing the packet data. For example, the TTL field can be preprocessed to determine whether the packet is involved in TTL value modulation. TTL values in packets sent by each sources address are compared to previous TTL values in packets sent by the same source. If the TTL value has changed, then the feature is the percentage of packets that have previously had modified TTLs out of the total packets sent from the same source address.

Both ML algorithms (k-NN and SVM) have a similar training process. Therefore, they are stored in a file. The cross-validation is executed using utility methods provided by SKLEARN. For DNN we used TEENSORFLOW which does not store models on a single file. Instead, a directory is used to store several meta data and graph files. Since TEENSORFLOW does not provide convenient methods for cross-validation or predictions, we have created a method to make these possible on both TEENSORFLOW and SKLEARN.

C. Testing

ML models are also called classifiers, and aim at learning by corresponding the classes with the inputs [38]. Classifiers are widely used to detect general network anomalies and also covert channels. By generating knowledge based on using the normal packets, the classifiers treat any activities that differ from the normal packet attribute as covert. Therefore, novel covert channel techniques can be detected with minimum effort (as they also deviate from normal packets). Selecting the appropriate classifier is a challenging task and generally based on the accuracy of the prediction. In this paper only supervised ML algorithms were used (k-NN, SVM and DNN) for the following reasons:

- k-NN is characterized to be one of the most straightforward instance-based learning algorithms [32].
- SVM belongs to the newest supervised machine techniques, it is pertinent with large number of features and it is very useful in insolvency analysis (when data are non regular) [37].
- DNN is capable of learning features automatically at any level of abstraction by mapping the input and the output directly from data with a negligible human-crafted feature [39], [40].

The detection process requires a pair of (PCAP, CSV) files and starts by first extracting the features from the packets similarly to the training process. Secondly, it loads the model from a pkl file. Thirdly, it calculates the accuracy. Lastly, it creates both metrics and confusion matrix. An output file is then produced which contains metrics values such as the number of packets and the prediction rate (whether a packet is normal or covert), as well as the classification of a certain covert packet into covert channel patterns.

To train the ML models we used large labelled datasets (supervised) that represent each type of pattern Tab.II. The following 20 features are extracted from preprocessing the packet data:

TABLE III
LIST OF FEATURES

Feature	Description
TCP.RESERVED	The reserved field of TCP
TCP.URGPTR	The urgent pointer field of TCP
TCP.ACK	TCP acknowledgment number
TCP.FLAGS	The IP flags field
TCP.SPORT	The source port for TCP
UDP.SPORT	The source port of UDP
IP.TTL	TTL field value of IP
IP.TOS	TOS of IP
IP.SRC	The IP source address
IP.ID	IP ID field
IP.FLAGS	IP flags field
IP.NUM.OPTS	Number of IP options
ICMP.TYPE	ICMP type number
ICMP.UNUSED	Unused ICMP fields
ICMP.RS.RESERVED	Reserved field of ICMP router solicitation
ICMP.SEQ	The ICMP sequence field
IGMP.TYPE	Type field of IGMP
IGMP.MRTIME	The IGMP Max Resp Time field
IGMP.LOAD	The IGMP Source Address field
IGMP.LOADLEN	The length of IGMP payload

D. Metrics

The ML model is first evaluated using the 3-fold cross-validation via the cross_val_score function in scikit-learns tool.

E. Model fitting and persistence

When the model is full with data, then it is stored to the models directory. The scores from cross-validation and the label encoder are also added. The original input PCAP and CSV files are saved to the model's directory. The model must be saved to a disk to be reloaded and used for the testing (prediction). The training data is saved to be combined with additional training data in the future.

IV. EXPERIMENTS

A. Experimental setup

1) The dataset: The covert dataset was created using ten popular tools and the normal dataset from http://mawi.wide.ad.jp. Afterwards, they were classified into four patterns. Network packets were generated with each tool

and were systematically labelled as belonging to one of four patterns as described in Tab. I. The global dataset is made up of a consolidation of all packets created and labelled. The distribution of normal and covert network packets (dataset) for the classification of training and testing is summarized in Tabs. IV.

TABLE IV
SAMPLE DISTRIBUTIONS OF THE VALIDATION DATASET

Patterns	Number of samples	Sample percentage (%)
P7	28,969	10.6
P5	40,331	14.9
P1	47,603	17.6
P0	72,857	27.3
NP0157	80,092	29.6
	270,579	100

2) Evaluation methodology: To measure the performance of the NCC's, the confusion matrix Tab. X, the accuracy, false positive, detection and precision Tab. XI were calculated using the following metrics:

- False Negative (FN): Incorrect negative classification.
- True Negative (TN): Correct negative classification.
- False Positive (FP): Incorrect positive classification.
- True Positive (TP): Correct positive classification.

B. Experimental results

1) SVM: The classification results of the NCC using SVM with the degree 2 of the five patterns. The measurement revealed a detection rate of 87%, an accuracy of 7%, an average precision of 87% and false positive rate of 3%.

2) k-NN: After having tested many values of k, k=4 was identified as being the best value. As shown in Tab. VI, the rate of detection, accuracy, precision and false positive with SVM are 89%, 90%, 96% and 1%, respectively.

3) DNN: The detection rate is high (92%) with a precision of 85%, the accuracy rate is 67% and false positive 4%.

TABLE V
CLASSIFICATION RESULTS WITH TESTING NCC DATASET USING K-NN

		Predicted	
		Normal	Covert
Actual	Normal	87,461	943
	Covert	283	60,705

TABLE VI
CONFUSION MATRIX RESULTS BY TRAINING DATASET USING K-NN

		Predicted				
		P0	P1	P5	P7	NP0157
Actual	P0	60,705	86	311	64	473
	P1	47	14,434	0	1	14
	P5	119	0	24,416	0	0
	P7	10	0	0	16,882	5
	NP0157	107	0	72	0	31,637

TABLE VII
PERFORMANCE COMPARISON OF THE MODELS BY TRAINING DATASETS

	Accuracy	Precision
P0	98%	99%
P1	99%	99%
P5	99%	98%
P7	99%	99%
NP0157	99%	98%

TABLE VIII
CONFUSION MATRIX RESULTS BY TESTING DATASET USING K-NN

		Predicted				
		P0	P1	P5	P7	NP0157
Actual	P0	28,139	334	247	317	229
	P1	536	32,489	0	7	73
	P5	843	1519	9891	0	3543
	P7	1579	0	0	10,220	0
	NP0157	261	0	30	0	47985

TABLE IX
PERFORMANCE COMPARISON BY TESTING DATASET USING K-NN

	Accuracy	False Positive	Precision
P0	0.96	0.01	0.89
P1	0.98	0	0.95
P5	0.63	0.05	0.97
P7	0.87	0.01	0.97
NP0157	0.99	0	0.93

V. DISCUSSION

This section provides a comparison between the different classifiers based on their metrics performance: accuracy, detection rate, precision, TP, TN, FP and FN. This comparative provides a basis of evaluation in order to identify the best ML algorithm to detect NCC's. Tab. X shows the performance of the training dataset. k-NN performs the best given the highest rates of detection, precision, accuracy, TP and lowest rates of FN and FP. Tab. IV provides the detection capabilities for the different patterns over the training dataset. The results reveal that k-NN is capable of classifying NCC's into patterns with high accuracy and precision.

The measurement results of k-NN on testing dataset demonstrates a difference on classification capabilities of the NCC into patterns. (Tab. IX). While NP0157, P1 and P0 allowed for the highest classification accuracy and precision rates, P5 and P7 resulted in the lowest values. On average, compared with DNN and SVM, k-NN provides the best accuracy and precision rates and lowest FP value. DNN has the highest detection and FP rates.

TABLE X
COMPARISONS OF THE MODELS BY THE TRAINING DATASET

	True Positive	True Negative	False Positive	False Negative
SVM	55,568	84,356	6071	3388
k-NN	60,705	87,461	934	283
DNN	55.174	82.319	6465	5425

	Detection Rate	False Positive	Accuracy	Precision
SVM	94%	6%	94%	90%
k-NN	99%	1%	99%	98%
DNN	91%	0.07	98%	90%

TABLE XI
COMPARISON OF THE MODELS BY TESTING DATASETS

	True Positive	True Negative	False Positive	False Negative
SVM	25,534	105,893	3732	3085
k-NN	28,139	105,757	1127	3221
DNN	24,960	106,727	4306	2251

	Detection Rate	False Positive	Accuracy	Precision
SVM	89%	3%	7%	87%
k-NN	89%	1%	90%	96%
DNN	92%	4%	67%	85%

The results indicate that k-NN has a significant level of difference with DNN and SVM over both training and testing datasets. Therefore, a reliable conclusion can be drawn that k-NN perform the best to detect and classify NCC by a considerable margin.

Our work is limited in different ways. First, the dataset of the study was restricted to some selected NCC tools. Second, we obtained the training data from the tools and we believe that this is a problem for real world applications of machine learning algorithms. Third, the work was limited to using 3 machine learning algorithms (i.e. SVM, k-NN, DNN)

VI. CONCLUSION

The rapid growth of computer networks has driven forward the need to acquire security policy that ensure confidentiality, integrity and availability of information. This has led cyber-attackers to find ways to break security policy and infiltrate or exfiltrate information using network covert channel techniques. Detection mechanisms to detect covert channels are based on identifying any deviation of nonstandard or abnormal behavior.

In this paper, selected ML methods were applied to detect popular network covert channels. The capacity of not only detecting, but classifying covert channels with high precision is also demonstrated. A dataset was created from eleven standard covert channel tools and the covert channels are then accordingly classified into patterns and labelled. Half of the generated dataset is used to train three different ML algorithms. The remaining half is used to verify the algorithms precision. The tested ML algorithms are Support Vector Machines (SVM), k-Nearest Neighbors (k-NN) and Deep Neural Networks (DNN). The k-NN model demonstrated the highest precision rate at 98% detection of a given covert channel and with a low false positive rate of 1%. DNN has the highest rate of FP and SVM has the lowest precision with testing dataset.

The findings of the research results suggest several areas of future work. Firstly, the possibility of additional covert channel patterns to be tested through the detection scheme. Secondly, further investigation with other ML algorithms could be considered. Most importantly however, research is needed to study how the discussed classifiers impact legitimate network communications while detecting and classifying covert channels on a large scale.

REFERENCES

[1] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network anomaly detection: methods, systems and tools. IEEE communications surveys and tutorials, 16(1):303336, 2014.

[2] H. Byun and S.W. Lee. A survey on pattern recognition applications of support vector machines. International Journal of Pattern Recognition and Artificial Intelligence, 17(03):459486, 2003

[3] I.T. R. Center . Data breach reports, 2017.

[4] C. Cortes, and V. Vapnik "Support-vector networks." Machine learning 20.3 (1995): 273-297.

[5] C. Latham Donald "Department of defense trusted computer system evaluation criteria." Department of Defense (1986).

[6] S. Dumais and H. Chen Hierarchical classification of web content. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 256263, ACM, 2000.

[7] K. R. Fall and W. R. Stevens. TCP/IP illustrated, volume 1: The protocols. addison-Wesley, 2011.

[8] P. A. Gilbert and P. Bhattacharya. An approach towards anomaly based detection and profiling covert tcp/ip channels. In Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on, pages 15. IEEE.

[9] C. G. Girling Covert channels in lans. IEEE Transactions on software engineering, 13(2):292, 1987.

[10] K. A. Heller, K. M Svore, A. D. Keromytis and S. J. Stolfo, One class support vector machines for detecting anomalous windows registry accesses. In Proc. of the workshop on Data Mining for Computer Security,volume 9, 2003.

[11] W. Hu, Y. Liao, and V. R. Vemuri, Robust support vector machines for anomaly detection in computer security. In ICMLA, pages 168174, 2003.

[12] T. Joachims Estimating the generalization performance of an svm efficiently. In Proc. 17th International Conf. on Machine Learning, pages 431438. Citeseer, 2000.

[13] P. Jongsuebsuk, N. Wattanapongsakorn and C. Charnsripinyo. Network intrusion detection with fuzzy genetic algorithm for unknown attacks. In Information Networking (ICOIN), International Conference on, pages 15. IEEE, 2013.

[14] M. I. Jordan, and T. M. Mitchell . Machine learning: Trends, perspectives, and prospects. Science, 349(6245):255260, 2015.

[15] R. A. Kemmerer. Shared resource matrix methodology: An approach to identifying storage and timing channels. ACM Transactions on Computer Systems (TOCS), 1(3):256277, 1983.

[16] A. Konar U.K. Chakraborty, and P. P. Wang. Supervised learning on a fuzzy petri net. Information Sciences, 172(3):397 416, 2005.

[17] A. Konar and J. Lakhmi "Supervised Learning by a Fuzzy Petri Net." Cognitive Engineering: A Distributed Approach to Machine Intelligence (2005): 233-255.

[18] B. Lampson. A note on the confinement problem. Communication of the ACM, 16(10):613615, 1973.

[19] H. J. Liao, C. H. R. Lin, Y. C. Lin, and K. Y. Tung. Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16-24, 2013.

[20] M. Pontil, and A. Verri. Properties of support vector machines. Neural Computation, 10(4):955974, 1998.

[21] C. H. Rowland Covert channels in the tcp/ip protocol suite. First Monday, 2(5). 1997.

[22] A. Salih, X. Ma and E. Peytchev, New intelligent heuristic algorithm to mitigate security vulnerabilities in ipv6. IJIS International Journal of Information Security, 4, 2015.

[23] B. Schlkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. Neural computation, 13(7):14431471, 2001.

[24] T. Shon and J. Moon. A hybrid machine learning approach to network anomaly detection. Information Sciences, 177(18):3799 3821, 2007.

[25] T. Sohn, J. Moon, S. Lee, D. H. Lee, and J. Lim Covert channel detection in the icmp payload using support vector machine. In International Symposium on Computer and Information Sciences, pages 828835. Springer, 2003.

[26] A. H. Sung and S. Mukkamala Identifying important features for intrusion detection using support vector machines and neural networks. In Applications and the Internet, 2003. Proceedings. 2003 Symposium on, pages 209 216. IEEE, 2003.

[27] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin. Intrusion detection by machine learning: A review. Expert Systems with Applications, 36(10):1199412000, 2009.

[28] S. Wendzel, S. Zander, B. Fechner and C. Herdin. Pattern based survey and categorization of network covert channel techniques. ACM Computing Surveys (CSUR), 47(3):50, 2015.

[29] S. Zander, G. Armitage and P. Branch. A survey of covert channels and countermeasures in computer network protocols. IEEE Communications Surveys and Tutorials, 9(3):4457, 2007.

[30] J. Zhai, G. Liu and Y. Dai Detection of tcp covert channel based on markov model. Telecommunication Systems, 54(3):333343, 2013.

[31] D. Ucci, L. Aniello and R. Baldoni. Survey on the usage of machine learning techniques for malware analysis. arXiv preprint arXiv:1710.08189, 2017.

[32] D. W. Aha, D. Kibler, M. K. Albert. Instance-based Learning Algorithms. Machine Learning 6(1), 3766 (1991).

[33] P. Laskov, P. Dssel, C. Schfer and K. Rieck Learning intrusion detection: supervised or unsupervised. In International Conference on Image Analysis and Processing (pp. 50-57). Springer, Berlin, Heidelberg (2005, September).

[34] C. F. Tsai and C. Y. Lin A triangle area based nearest neighbors approach to intrusion detection Pattern recognition, 43(1), 222-229, 2010.

[35] P. L. Shrestha, M. Hempel, F. Rezaei and H. Sharif, "A Support Vector Machine-Based Framework for Detection of Covert Timing Channels" in IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 274-283, 1 March-April 2016.

[36] A. Mohiuddin, A. N. Mahmood, and J. Hu. "A survey of network anomaly detection techniques." Journal of Network and Computer Applications 60 (2016): 19-31.

[37] L. Auria and A. M. Rouslan "Support vector machines (SVM) as a technique for solvency analysis." (2008).

[38] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. rndi, P. Laskov and F. Roli Evasion attacks against machine learning at test time. In Joint European conference on machine learning and knowledge discovery in databases (pp. 387-402). Springer, Berlin, Heidelberg, (2013, September).

[39] D. Yu, M. L. Seltzer, J. Li, J. T. Huang, and F. Seide, Feature learning in deep neural networks-studies on speech recognition tasks, arXiv preprint arXiv:1301.3605, 2013.

[40] Y. Bengio, A. Courville, and P. Vincent, Representation learning: A review and new perspectives, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 17981828, 2013.

[41] V. V. Kumar, and S. Kumar. "Detection of TCP/IP Covert Channel based on Nave-Bayesian Classifier." International Journal Of Engineering And Computer Science ISSN: 2319-7242, 2014.

[42] S. J. Murdoch, and S. Lewis. "Embedding covert channels into TCP/IP." International Workshop on Information Hiding. Springer, Berlin, Heidelberg, 2005.

[43] S. Cabuk, C. E. Brodley, and C. Shields. "IP covert channel detection." ACM Transactions on Information and System Security (TISSEC) 12.4 (2009): 22.

[44] M. Wojciech, S. Wendzel, M. Chourib, J. Keller Countering Adaptive Network Covert Communication with Dynamic Wardens, Future Generation Computer Systems (FGCS), Vol. 94, pp. 712-725, Elsevier, 2019.

[45] R. E. Schapire "The boosting approach to machine learning: An overview." Nonlinear estimation and classification. Springer, New York, NY, 2003. 149-171, 2003.

[46] N. B. Lucena, G. Lewandowski, and S. J. Chapin "Covert channels in IPv6." International Workshop on Privacy Enhancing Technologies. Springer, Berlin, Heidelberg, 2005.

[47] H. Zhao and Y. Q. Shi. "Detecting covert channels in computer networks based on chaos theory." IEEE transactions on information forensics and security 8.2 (2013): 273-282.

[48] Y. F. Huang, S. Tang, and Y. Zhang. "Detection of covert voice-over Internet protocol communications using sliding window-based steganalysis." IET communications 5.7 (2011): 929-936.

[49] J. Zhai, G. Liu, and Y. Dai. "A covert channel detection algorithm based on TCP Markov model." 2010 International Conference on Multimedia Information Networking and Security. IEEE, 2010.

[50] P. A. Gilbert and P. Bhattacharya. "An approach towards anomaly based detection and profiling covert TCP/IP channels." 2009 7th International Conference on Information, Communications and Signal Processing (ICICS). IEEE, 2009.

[51] S. Hammouda, L. Maalej, and Z. Trabelsi. "Towards optimized TCP/IP covert channels detection, IDS and firewall integration." 2008 New Technologies, Mobility and Security. IEEE, 2008.

[52] N. Nagatou and T. Watanabe. "Run-time detection of covert channels." First International Conference on Availability, Reliability and Security (ARES'06). IEEE, 2006.

[53] S. Zander, G. Armitage, and P. Branch. "Covert channels in the IP time to live field.", 2006.

[54] S. J. Murdoch, and S. Lewis. "Embedding covert channels into TCP/IP." International Workshop on Information Hiding. Springer, Berlin, Heidelberg, 2005.

[55] S. Wendzel, S. Zander: Detecting Protocol Switching Covert Channels, in Proc. 37th IEEE Conf. on Local Computer Networks (LCN), pp. 280-283, IEEE, 2012.

[56] W. Mazurczyk, S. Wendzel, S. Zander et al.: Information Hiding in Communication Networks, Wiley-IEEE, 2016.

[57] H. Okhravi, B. Stanley, and S. T. King "Design, implementation and evaluation of covert channel attacks." 2010 IEEE International Conference on Technologies for Homeland Security (HST). IEEE, 2010.

[58] S. Marsland, Machine learning: an algorithmic perspective. Chapman and Hall/CRC, 2011.