Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting

Ployphan Sornsuwit & Saichon Jaiyen

To cite this article: Ployphan Sornsuwit & Saichon Jaiyen (2019) A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting, Applied Artificial Intelligence, 33:5, 462-482, DOI: 10.1080/08839514.2019.1582861

To link to this article: https://doi.org/10.1080/08839514.2019.1582861

APPLIED ARTIFICIAL INTELLIGENCE
2019, VOL. 33, NO. 5, 462–482
https://doi.org/10.1080/08839514.2019.1582861

Taylor & Francis
Taylor & Francis Group

A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting

Ploypan Sornsuwit and Saichon Jaiyen

Advanced Artificial Intelligence Research Laboratory, Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

ABSTRACT

A hybrid machine learning is a combination of multiple types of machine learning algorithms for improving the performance of single classifiers. Currently, cyber intrusion detection systems require high-performance methods for classifications because attackers can develop invasive methods and evade the detection tools. In this paper, the cyber intrusion detection architecture based on new hybrid machine learning is proposed for multiple cyber intrusion detection. In addition, the correlation-based feature selection is adopted for reducing the irrelevant features and the weight vote of adaptive boosting that is adopted to combine multiple classifiers is concentrated. In the experiments, UNB-CICT or network traffic dataset is used for evaluating the performance of the proposed method. The results show that the proposed method can achieve higher efficiency in every attack type detection. Furthermore, the experiments with Phishing website dataset UNSW-NB 15 dataset NSL-KDD dataset and KDD Cup'99 dataset are also conducted, and the results show that the proposed method can produce higher efficiency as well.

Introduction

In recent years, many applications of computer and network technologies have been used and added in daily life, including the use of data privacy, government data, or business data. Cybersecurity has become more important to prevent intrusion into systems. In the past, setting up a security policy on a firewall may not have enough protection against these intrusions because the invasion of new forms has been developed, including using the weaknesses of the operating system, including the settings in communication between networks. However, we can detect the malfunction as well as prevent the intrusion by using the Intrusion Detection System (IDS) (Anand and Patel 2012). Presently, intrusion focuses on commercial interests because there are many activities that are security risks, and important activities such as transferring money online, sending important files through emails or social networks, etc. Although there is

CONTACT Ploypan Sornsuwit ploypan.en@gmail.com

the use of https protocol, that does not mean that it can be protected completely, for example, website phishing activities that trick individuals into registering personal information, DDoS attacks to stop services of target machines, etc. One major attack that evades firewalls and IDS/IPS is Tor or "The Onion Router" which is a distributed overlay network designed to anonymize TCP-based applications like web browsing, secure shell, and instant messaging (Dingledine, Mathewson, and Syverson 2004). This is a service created to allow people to surf the Internet without revealing themselves. The user will need to connect to a network of other middleware that will hide the IP address from the website visited as a private route, so no one can trace your usage, even with Tor users it can be hard to detect. Currently, this is a challenge for these securities.

Machine Learning has been made for practical usage to enhance the detection capabilities of IDS, but still cannot detect them all. There still exist some errors (AbdElrahman and Abraham 2014; Amer, Goldstein, and Abdennadher 2013; Mascaro, Nicholson, and Korb 2014; Sagha et al. 2013; Sheykhkanloo 2014). Moreover, hybrid and ensemble systems are also used to increase the capability of the traditional IDS. The research (Aburomman and Reaz 2017) on the detection of abnormalities found that the ensemble implementation of IDS detection enhancements has been developed in two ways: the homogeneous ensemble method and the heterogeneous ensemble method. In the homogeneous ensemble method, a weak learner is used in the same way, but heterogeneous ensemble method will choose a different weak learner. Both methods must boost the weak learner to combine decisions to achieve better final results than the single learner. This is achieved when testing by doing the classification in each research by using the different methods. Voting found that homogeneous ensembles can frequently classify some classes into which the difficult class is. For heterogeneous ensemble, it has low false alarm detection but both still have the same disadvantage which cannot detect the new irregularities.

Therefore, through these studies, we can see that machine learning is widely used in classification problems. The problem of multiple intrusion detection can be considered as a multiclass classification problem. So, the objective of our research is to develop new effective Adaboost algorithm to classify multiclass intrusions by using UNB-CIC Tor Network Traffic dataset (UNB, 2017). In addition, a new hybrid classifier is developed for IDS dataset in which features are collected by correlation-based selection. The selected features will be trained with multiple weak learners and build a strong hypothesis by voting.

The rest of this paper is arranged as follows. "Review of Related Work" is a section describing recent related researches. "Ensemble Learning" and "correlation-based feature selection" are described briefly. Concepts of the two algorithms are proposed. "Proposed Method" presents an algorithm

and proposed hybrid method. "Experimental Results" section shows our experiment. The final section is "Conclusion and Future Work".

Related Work

Based on the research in the past, many current researchers have developed various studies to detect malfunctions on network-based investigations and applied a variety of machine learning techniques in anomaly detection (AbdElrahman and Abraham 2014; Amer, Goldstein, and Abdennadher 2013; Mascaro, Nicholson, and Korb 2014; Sagha et al. 2013; Sheykhkanloo 2014), mostly to improve classification efficiency. For example, Hussain and Lalmuanawma tested various methods of hybrid systems with different feature selections with a 4.5 weak learner which was adapted to the Adaboost algorithm. The experiment showed that the wrapper method and Adaboost with decision tree with weak learners gave the best efficiency (Hussain and Lalmuanawma 2014). Wahba et al. proposed hybrid feature selection methods by combining correlation-based and information-gained in selecting relevant features and classification steps using Adaboost.M1 with Naïve Bayes weak learners, the result showed a good detection rate and a low false positive rate (Wahba, ElSalamouny, and ElTaweel 2015). Aburomman and Reaz proposed a novel combination of multiple experts (SVM, k-NN, PSO) into one ensemble algorithm, they combined all results from different experts by using a weighted majority vote. The result showed that the novel approach gave better accuracy than other methods (Aburomman and Reaz 2016). Michael et al. proposed supervised machine learning with meta-classifiers, the results showed that the bagging with REPTree weak learners was more capable in predicting than other meta-classifiers (Michael, Kumaravel, and Chandrasekar 2015). Nejad and Abadi developed a security system with IG and GR feature reduction and applied features in Adaboost methods, IG and Adaboost with random tree gave the better performance than other methods (Nejad and Abadi 2014).

Other machine learnings and hybrid methods were improved performance of Adaboost such as SVM (Ren 2014), Neuro-Fuzzy (Kumar and Selvakumar 2013) or new weight vote framework (Kuncheva and Rodríguez 2014). In addition, classification is conducted by using Adaboost.m1, which is a multiclass boosting tool which is used to improve classification methods and show satisfactory performance more than other ensemble methods. Most of this research tries to improve the performance of Adaboost methods with several weak learners, but most of these are not effective to detect multiclass intrusion (Zhang and Xie 2010). However, there are various intruding ways and behaviors that avoid network detection, and conceal or prevent communication in order to make it difficult to trace internet activity or fraudulent websites, etc. Thus, some studies

made efforts to develop detection algorithms: Hodo et al. presented the process of classification of Tor Traffic and Non-tor traffic to monitor the activity and security of the user's usage. The researchers have compared the quality of classification with Artificial Neural Network and Support Vector Machine using UNB-CIC TOR Network Traffic dataset and resulted in the usage of Correlation-based feature selection (CFS) and can select 10 features then classify them with Artificial Neural Network. The results of this study had an accuracy of 99.8% (Hodo et al. 2014). The research of Ghafir, Svoboda, and Prenosil also presented a methodology for detecting Tor by applying our methodology on campus live traffic and showed that it can automatically detect Tor connections (Ghafir, Svoboda, and Prenosil 2014b). Some studies use hybrid feature selection with Mbox2xml tools to extract features, then use Bays Net Algorithm as a Classifier to analyze whether this is phishing email or not. Results found that when select features are left to only eight features and accuracy in classifying is as high as 94% (Hamid, Abawajy, and Kim 2013) as with the research (Abdelhamid, Ayesh, and Thabtah 2014) developed a Multi-label Classifier based Associative Classification (MCAC). This was used to classify phishing using Chi-square feature selection method to select features for the test compared to other machine learning and found that using MCAC has a higher accuracy and can detect a new class called "Suspicious" that was not originally in the training data set.

Ensemble Learning

Boosting is an important method in ensemble learning, Boosting is the method which was involved with the creation of different ensembles from many weak learners that were combined these weak learners into a single strong learner. The idea of Boosting, when we have distribution from various weak learners may have the answer of class that is correct or incorrect. Boosting can combine them to achieve single strong learners which is the final correct answer, according to the following procedure (Zhou 2012)

Adaboost.M1 is an extension of the original adaptive boosting method. It is extended to multiclass boosting with a different weight changing mechanism (Galar et al. 2014). The key idea of Adaboost.M1 is that it will update the distribution weights of samples that are classified by the current hypothesis. In Adaboost.M1, the weak learner requires errors less than 0.5% before adding to the ensemble. Adaboost.M1 will concentrate on difficulty classified instances by increasing weights of incorrectly classified samples. The details of this algorithm are shown in algorithm 1.

Algorithm 1: Adaboost.M1

Input: sequence of m examples $(x_1, y_1), \ldots, (x_m, y_m)$, $x_i \in X$, with labels $y_i \in Y = \{1, \ldots, k\}$

Weak learning algorithm (**Weaklearn**)

Integer T specifying number of iterations

Initialize $D_1(i) = 1/m$ for all$i$

Do for $t = 1.2, \ldots, T$

1. Call **Weaklearn** and provide it with the distribution D_t
2. Get back a hypothesis $h_t : X \rightarrow Y$
3. Calculate the error of h_t: $\varepsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$ if ε_t> 1/2 then set $T = T - 1$ and abort loop
4. Set $\beta_t = \varepsilon_t/(1 - \varepsilon_t)$
5. Update distribution D_t : $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t \text{ if } h_t(x_i) = y_i \\ 1 \text{ otherwise} \end{cases}$

   Where Z_t is a normalization constant (Chosen so that D_{t+1} will be a distribution)

Output the final hypothesis: $h_{fin} = \underset{yY}{argmax} \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t}$

Correlation-Based Feature Selection

Correlation-based feature selection (CFS) is a principle of screening and ranking subgroups according to the relationship between features and classes by the good subgroups of features, it will have a high correlation with a class that will be selected for using in predicting the answer for class. In the case of features which have no correlation in redundant information should be eliminated as well.

Network Traffic dataset contains attributes that are correlated with each other, so we need to select only the high relationship features by using correlation-based feature selection. Correlation-based feature selection evaluates subsets of features by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred (Hall 1999).

The correlation can be calculated as

$$Merit_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}} \quad (1)$$

where

$Merit_s$ is the correlation between the summed components and the outside variable.

k is the number of components.

$\bar{r}_{cf}$ is the mean feature-class correlation $(f \epsilon S)$

$\bar{r}_{ff}$ is the average inter-correlation between components.

The heuristic metrics $\bar{r}_{zi}$ and $\bar{r}_{ii}$ are computed as the symmetrical uncertainty (SU)

$$SU = 2.0 \times \left[\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)}\right] \tag{2}$$

where $H(X)$ is defined as entropy that can be calculated as

$$H(X) = -\sum\nolimits_{x \epsilon X} p(x) log_2(p(x)) \tag{3}$$

Proposed Method

The proposed model will use five machine learning techniques as weak learners to build the model and combine the model to be the final hypothesis with Adaboost.M1 and then evaluate the efficiency of the algorithm and comparison. The processes of training the proposed model are consisting of four stages which are data preprocessing, hybrid weak classifier training, strong classifier training, and performance evaluation as shown in Figure 1. The first stage is the data pre-processing. Firstly, some symbolic features must be converted to numeric features such as Source IP and Destination I because they cannot be calculated by machine learning algorithms. Then, the correlation-based feature selection is applied for selecting the relevant features in the dataset in order to reduce the number of features. The second stage is to train various classifiers with the training set. In this stage, five classifiers including k-NN, C4.5, MLP, SVM, and LDA are adopted to build the weak classifiers. Each classifier is effective for detecting each type of intrusion. The third stage is to build a strong classifier by Adaboost.M1 (Freund and Schapire 1996). The final stage is to evaluate the performance of the classifier. The main idea is to build a strong classifier from various types of weak classifiers by adopting Adaboost.M1 (Galar et al. 2014). In Adaboost.M1 algorithms, the weak classifiers are the same type, and the strong classifier is built by the combination of the same weak classifiers. In our proposed method, the combination of the same type of weak classifiers is changed to the combination of various types of weak classifiers.

After learning processes, $\beta_t$will be obtained from every weak leaner as $\beta_1 - \beta_5$. After that, β_t is sent for calculation in the testing process. Testing data will be exploited to classify it with five weak learners to get a hypothesis h_t from $h_1 - h_5$. In this process, β_t and h_t will be employed to vote with the method of Adaboost.M1. The proposed hybrid machine learning for detecting cybersecurity intrusions is shown in Figure 2. This new model is designed

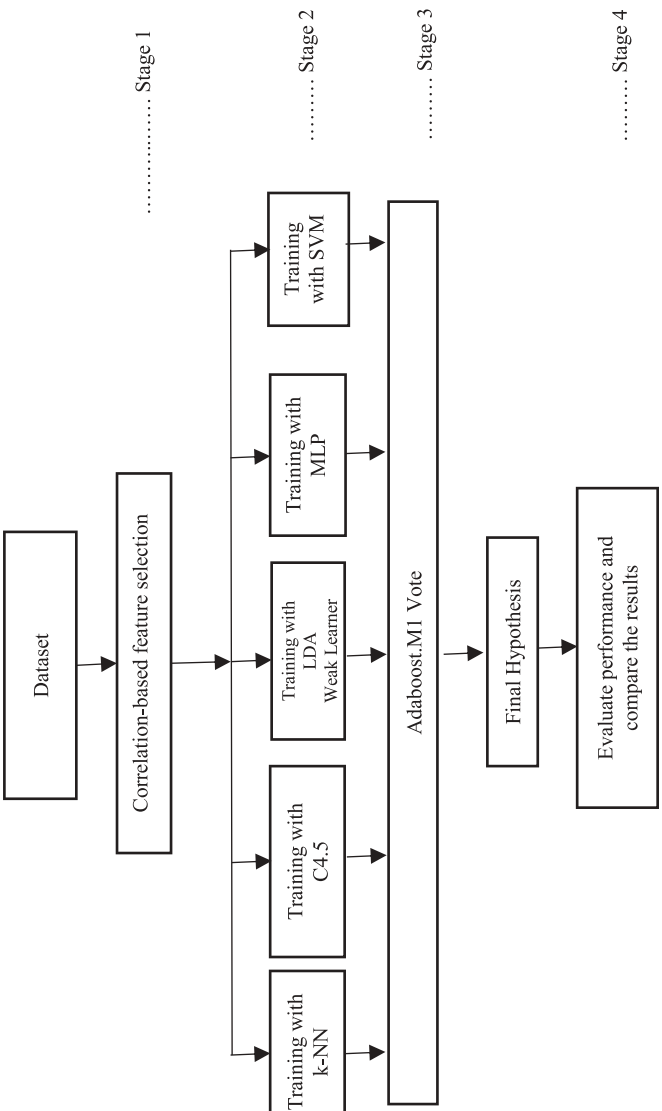

Figure 1. The learning processes of proposed hybrid machine learning.

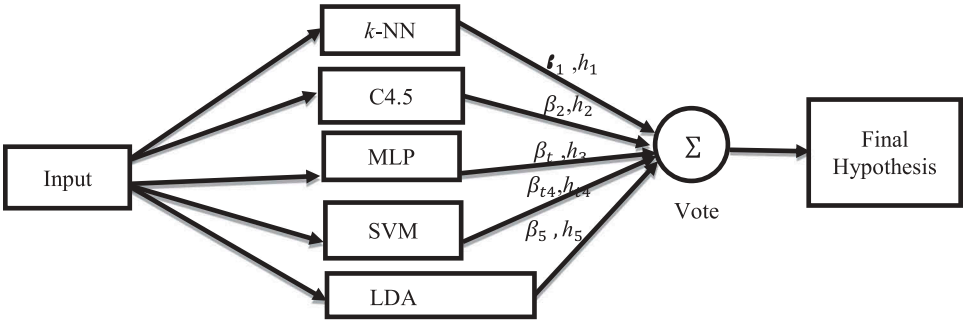

Figure 2. The proposed hybrid machine learning model.

to increase the efficiency of cybersecurity threat detection according to the ability of hybrid machine learning algorithms and correlation-based feature selection as mentioned above.

Experimental Results

Our research dataset was created from real-world traffic, defined as a set of tasks that created users, including Alice and Bob, to use different applications such as Skype, Facebook, and so forth to capture the traffic that occurs during eight service communications, which includes audio, browsing, chat, file transfer, mail, P2P, video, and VOIP (Lashkari et al. 2017) used in the dataset for experiments which contain two scenarios: Scenario A and Scenario B, in which both scenarios are different. Scenario A has two classes which are interested in classifying normal traffic and Tor traffic, but Scenario B is interested in classifying characterization of all eight services of Tor traffic as mentioned above. The details of the two scenarios can be found in Tables 1 and 2 as follows.

In addition, experiments will use the cybersecurity dataset to confirm the performance of our proposed algorithms, with a variety of attacks and dataset with traditional and existing intrusion including Phishing website, UNSW-NB15, NSL-KDD and KDD Cup'99.

Data Preprocess and Feature Selection

In the pre-processing step, the text features that are converted into numeric features are proto, service and state. In this process, the correlation-based

Table 1. Number of data in Scenario A.

Class	No. of data	Description
Tor	8,044	Tor Traffic
Nontor	59,790	Normal Traffic
Total	67,843	Tor and Normal Traffic

Table 2. Number of data in Scenario B.

Class	No. of data	Description
Browsing	1,604	HTTP and HTTPS traffic is generated by the user with two web browsers; Firefox and Chrome.
Email	282	Traffic sample created with Thunderbird client, both Alice and Bob accounts using Gmail in which the clients will send of SMTP/S and received with POP3/SSL.
Chat	323	Send Instant Message and specify a chat label for applications such as ICQ, Skype, IAM as well as Facebook and Hangout (on the web browser).
Audio	721	The traffic of the streaming data that is labelled as audio is stored in Spotify.
File Transfer	864	The traffic used to receive and send files, FTP over SSH (SFTP) and FTP over SSL (FTPS).
P2P	1,085	The traffic sharing using a protocol like Bit Torrent, which in order to generate traffic download the .torrent file and then capture the session traffic.
Video	874	The traffic of the video data that is labelled Video, which will be collected from YouTube and Vimeo Services.
VOIP	2,291	It is a VOIP application that is stored from Voice Call from Facebook, Skype and Hangout Application.
Total	8,044	All classes ofTor in Scenario B

feature selection is applied to select the relevant features, because it is an effective way to detect intrusion (Bahl and Sharma 2015; Eid et al. 2013; Nguyen, Franke, and Petrovic 2014; Shahbaz et al. 2016; Zhang et al. 2017). After pre-processing, we divide the UNB-CIC Tor Network Traffic dataset into Scenario A and Scenario B to the training dataset and testing dataset of 70:30, which gives the amount of data for both scenarios shown in Table 3. There are six features selected from this process, to be found in Table 4.

Performance Evaluation

In the experimental results, the comparative efficiency between single classifiers and the proposed multiple classifiers are done by using efficiency: precision, detection rate, specificity, FPR, f-Measure, and accuracy. Tables 5 and 6 shows confusion matrix of Scenario A and scenario B. Tables 7 and 8 show the efficiency analysis value from the confusion matrix from both Scenario A and scenario B. Based on the analysis, the analysis yielded 100% efficiency for Scenario B, which means that it is very efficient to classify Tor traffic. Comparing performance with other machine learning methods has been made for regular classification finds that the method we offer for detector intrusion efficiency is higher than Scenario A and Scenario B between weak learners before a vote and model and after a vote, as shown in Tables 9 and 10, which is most efficient when compared to other methods of Scenario A and scenario B.

In addition, our research offers comparisons with other Intrusion Datasets. The results show a comparison of Phishing web performance (Abdelhamid, Ayesh, and Thabtah 2014) in Table 11, UNSW-NB15 (Moustafa and Slay

Table 3. The details of the dataset for the Training dataset and the Testing dataset.

	Scenario A		Scenario B	
Class	Class	No. of data	Class	จำนวน
Training	Tor Traffic	5631	Audio	505
	Normal Traffic	1853		
			Browsing	1,123
			Chat	226
			File-Transfer	605
			Mail	197
			P2P	760
			Video	612
			VOIP	1,604
Testing	Tor Traffic	2413	Audio	216
	Normal Traffic	17937		
			Browsing	481
			Chat	97
			File-Transfer	259
			Mail	85
			P2P	325
			Video	262
			VOIP	687
	Total	67,843	Total	8,044

Table 4. The six features from the feature selection process.

Dataset	Number of features in dataset	Features name	Dataset	Number of features in dataset	Features name
Scenario A	2	Source Port	Scenario B	1	Source IP
	5	Protocol		2	Source Port
	7	Flow Bytes/s		7	Flow Bytes/s
	14	Fwd IAT Std		11	Flow IAT Max
	17	Bwd IAT Mean		16	Fwd IAT Min
				20	Bwd IAT Min

Table 5. The confusion matrix of our proposed method Scenario A.

	Predict Class	
Actual Classes	Normal	Tor
Normal	17937	0
Tor	5	2408

Table 6. The confusion matrix of our proposed method from Scenario B.

	Predict Class							
Actual Classes	Audio	Browsing	Chat	File-transfer	Mail	P2P	Video	VOIP
Audio	216	0	0	0	0	0	0	0
Browsing	0	481	0	0	0	0	0	0
Chat	0	0	97	0	0	0	0	0
File-transfer	0	0	0	259	0	0	0	0
Mail	0	0	0	0	85	0	0	0
P2P	0	0	0	0	0	325	0	0
Video	0	0	0	0	0	0	262	0
VOIP	0	0	0	0	0	0	0	687

Table 7. The efficiency of all classifiers when training with UNB-CIC Tor Network Traffic dataset for Scenario A form this table between weak learner before vote and model after vote.

Classifier		TP	FP	FN	TN	Precision	Detection Rate	Specificity	FPR	f-Measure	Accuracy
k-NN	Normal	17,759	178	167	2,246	99.01	99.07	92.66	7.34	99.04	98.03
	Tor	2,246	167	178	17,759	93.08	92.66	99.07	0.93	92.87	
C4.5	Normal	17,916	21	32	2,381	99.88	99.82	99.13	0.87	99.85	99.74
	Tor	2,381	32	21	17,916	98.67	99.13	99.82	0.18	98.90	
LDA	Normal	17,100	837	1391	1,022	95.33	92.48	54.98	45.02	93.88	89.05
	Tor	1,022	1391	837	17,100	42.35	54.98	92.48	7.52	47.85	
MLP	Normal	17,536	401	742	1,671	97.76	95.94	80.65	19.35	96.84	94.38
	Tor	1,671	742	401	17,536	69.25	80.65	95.94	4.06	74.52	
SVM	Normal	16,591	1346	53	2,360	92.5	99.68	63.68	36.32	95.96	93.19
	Tor	2,360	53	1346	16,591	97.8	63.68	99.68	0.32	77.14	
Our Approach	**Normal**	**17,937**	**0**	**5**	**2,408**	**100**	**99.97**	**100**	**0**	**99.98**	**99.98**
	Tor	**2408**	**5**	**0**	**17937**	**99**	**100**	**99.97**	**0.03**	**99.49**	

Table 8. The efficiency of all classifiers when training with UNB-CIC Tor Network Traffic dataset in scenario B from this table between weak learner before vote and model after vote.

Classifier		TP	FP	FN	TN	Precision	Detection Rate	Specificity	FPR	f-Measure	Accuracy
k-NN	AUDIO	166	50	37	2,159	76.85	81.77	97.74	2.26	79.23	91.83
	BROWSING	425	56	78	1,835	88.36	84.49	97.07	2.93	86.38	
	CHAT	84	13	9	2,306	86.6	90.32	99.44	0.56	88.42	
	FILE-TRANSFER	236	23	31	2,122	91.12	88.39	98.93	1.07	89.73	
	MAIL	61	24	8	2,319	71.76	88.41	98.98	1.02	79.22	
	P2P	317	8	7	2,080	97.54	97.84	99.62	0.38	97.69	
	VIDEO	240	22	23	2,127	91.6	91.25	98.98	1.02	91.42	
	VOIP	686	1	4	1,721	99.82	99.42	99.94	0.06	99.62	
C4.5	AUDIO	212	4	0	2,196	98.15	100	99.82	0.18	99.07	99.46
	BROWSING	478	3	5	1,926	99.38	98.96	99.84	0.16	99.17	
	CHAT	97	0	1	2,314	100	98.98	100	0	99.49	
	FILE-TRANSFER	259	0	0	2,153	100	100	100	0	100.00	
	MAIL	84	1	3	2,324	98.82	96.55	99.96	0.04	97.67	
	P2P	324	1	0	2,087	99.69	100	99.95	0.05	99.84	
	VIDEO	262	0	4	2,146	100	98	100	0	98.99	
	VOIP	683	4	0	1,725	99.42	100	99.77	0.23	99.71	
LDA	AUDIO	107	109	120	2,076	49.54	47.14	95.01	4.99	48.31	74.72
	BROWSING	267	214	119	1,812	55.51	69.17	89.44	10.56	61.59	
	CHAT	97	0	25	2,290	100	79	100	0	88.27	
	FILE-TRANSFER	248	11	64	2,089	95.75	79.49	99.48	0.52	86.87	
	MAIL	53	32	28	2,299	62.35	65.43	98.63	1.37	63.85	
	P2P	324	1	7	2,080	99.69	97.89	99.95	0.05	98.78	
	VIDEO	12	250	0	2,150	4.58	100	89.58	10.42	8.76	
	VOIP	687	0	254	1,471	100	73	100	0	84.39	

(*Continued*)

Table 8. (Continued).

Classifier		TP	FP	FN	TN	Precision	Detection Rate	Specificity	FPR	f-Measure	Accuracy
MLP	AUDIO	190	26	144	2,052	87.96	56.89	98.75	1.25	69.09	89.97
	BROWSING	332	149	24	1,907	69.02	93.26	92.75	7.25	79.33	
	CHAT	97	0	0	2,315	100	100	100	0	100.00	
	FILE-TRANSFER	259	0	4	2,149	100	98	100	0	98.99	
	MAIL	79	6	2	2,325	92.94	97.53	99.74	0.26	95.18	
	P2P	323	2	19	2,068	99.38	94.44	99.9	0.1	96.85	
	VIDEO	208	54	5	2,145	79.39	97.65	97.54	2.46	87.58	
	VOIP	682	5	44	1,681	99.27	93.94	99.7	0.3	96.53	
SVM	AUDIO	216	0	324	1,872	100	40	100	0	57.14	81.97
	BROWSING	167	314	28	1,903	34.72	85.64	85.84	14.16	49.41	
	CHAT	97	0	3	2,312	100	97	100	0	98.48	
	FILE-TRANSFER	235	24	23	2,130	90.73	91.09	98.89	1.11	90.91	
	MAIL	5	80	3	2,324	5.88	62.5	96.67	3.33	10.75	
	P2P	322	3	2	2,085	99.08	99.38	99.86	0.14	99.23	
	VIDEO	260	2	12	2,138	99.24	95.59	99.91	0.09	97.38	
	VOIP	675	12	40	1,685	98.25	94.41	99.29	0.71	96.29	
Our Approach	**AUDIO**	**216**	**0**	**0**	**2,196**	**100**	**100**	**100**	**0**	**100.00**	**100**
	BROWSING	**481**	**0**	**0**	**1,931**	**100**	**100**	**100**	**0**	**100.00**	
	CHAT	**97**	**0**	**0**	**2,315**	**100**	**100**	**100**	**0**	**100.00**	
	FILE-TRANSFER	**259**	**0**	**0**	**2,153**	**100**	**100**	**100**	**0**	**100.00**	
	MAIL	**85**	**0**	**0**	**2,327**	**100**	**100**	**100**	**0**	**100.00**	
	P2P	**325**	**0**	**0**	**2,087**	**100**	**100**	**100**	**0**	**100.00**	
	VIDEO	**262**	**0**	**0**	**2,150**	**100**	**100**	**100**	**0**	**100.00**	
	VOIP	**687**	**0**	**0**	**1,725**	**100**	**100**	**100**	**0**	**100.00**	

Table 9. The performance of the proposed algorithm compared to doing Classification with other methods of Scenario A.

Classifier	Intrusion type	Detection Rate	False Positive Rate	Accuracy
k-NN	Tor	15.64	1.13	37.36
	Non Tor	98.87	84.36	
C4.5	Tor	11.87	0	11.95
	Non Tor	100.00	88.13	
LDA	Tor	17.43	0	43.85
	Non Tor	100.00	82.57	
MLP	Tor	43.07	0	84.32
	Non Tor	100.00	56.93	
SVM	Tor	8.62	12.02	84.13
	Non Tor	87.98	91.38	
Our Approach	Tor	**99.97**	**0**	**99.98**
	Non Tor	**100**	**0.03**	

2015) in Table 12, NSL-KDD (UNB 2018) and KDD Cup'99 (KDD Cup 99 1999) Table 13 shows that the algorithms we present are effective in classification. In the case of the UNSW-NB15 dataset, the NSL-KDD dataset, and the KDD Cup'99 dataset, we will use the database for training and for testing the original file created by the developer.

In Table 11, classification with the website phishing dataset showed that the proposed algorithm had the highest efficiency of 97.54%, with a detection rate of phishing class of 100% as well as the UNSW-NB15 dataset. As shown in Table 12, even with the classification of many classes and invasions were different, the way we present still gives the highest performance in the NSL-KDD and KDD Cup'99, there are five types of invasions that are similar, even NSL-KDD is a modified version of the KDD Cup'99as shown in Table 13. The algorithm we are presenting can also detect both dataset intrusion and capture well on every dataset presented.

Furthermore, we tested the procedure with the ensemble method to test 50 models by applying C4.5 as a weak learner; it was found that Adaboost.M1 had an accuracy of 76%. According to the analysis, it was found that our proposed method was regarded as an incorporation of effective procedures, but it required only 1 model to vote with the method of Adaboost.M1. This resulted in the most effective experimental result, and it was substantially different from other methods.

Conclusions

In this paper, the new hybrid machine learning for cybersecurity threat detection is proposed. This new hybrid classifier is the combination of C4.5, MLP, SVM and LDA based on adaptive boosting. The UNB-CIC Tor Network Traffic datasets are used in the experiments for evaluating the performance of the proposed model. In addition, the experiments, correlation-based feature selection method is applied to all datasets in order to reduce redundant features.

Table 10. The performance of the proposed algorithm compared to doing classification with other methods of Scenario B.

Classifier	Intrusion type	Detection Rate	False Positive Rate	Accuracy
k-NN	AUDIO	66.24	2.8	66.79
	BROWSING	57.84	10.72	
	CHAT	16.05	3.6	
	FILE-TRANSFER	42.42	8.41	
	MAIL	27.69	2.85	
	P2P	87.22	0.54	
	VIDEO	43.42	7.46	
	VOIP	82.90	1.12	
C4.5	AUDIO	57.41	3.02	73.51
	BROWSING	72.40	14.65	
	CHAT	53.53	0.27	
	FILE-TRANSFER	100.00	7.99	
	MAIL	100.00	0.47	
	P2P	86.79	0.15	
	VIDEO	49.55	2.23	
	VOIP	85.52	0.12	
MLP	AUDIO	67.02	6.37	60.98
	BROWSING	52.00	1.89	
	CHAT	6.45	3.99	
	FILE-TRANSFER	61.46	8.65	
	MAIL	53.85	2.8	
	P2P	80.60	0.05	
	VIDEO	36.28	3.86	
	VOIP	98.91	16.09	
SVM	AUDIO	32.90	3.24	74.17
	BROWSING	83.28	10.15	
	CHAT	39.53	2.71	
	FILE-TRANSFER	100.00	8.15	
	MAIL	97.44	1.98	
	P2P	99.38	0.19	
	VIDEO	99.13	1.56	
	VOIP	77.22	0.59	
LDA	AUDIO	43.45	6.75	26.87
	BROWSING	57.99	15.17	
	CHAT	59.32	2.63	
	FILE-TRANSFER	28.27	8.83	
	MAIL	21.43	3.42	
	P2P	23.98	0.09	
	VIDEO	0.00	11.65	
	VOIP	0.00	30.68	
Our Approach	AUDIO	**100**	**0**	100
	BROWSING	**100**	**0**	
	CHAT	**100**	**0**	
	FILE-TRANSFER	**100**	**0**	
	MAIL	**100**	**0**	
	P2P	**100**	**0**	
	VIDEO	**100**	**0**	
	VOIP	**100**	**0**	

Tables 5–6 show the confusion matrix and Tables 7–8 show the efficiency of our proposed model including precision, detection rate, false positive rate, f-measure, and accuracy. It was found that the algorithm we offer has a high

Table 11. The efficiency of all classifiers when training with Website Phishing Dataset in case of weak learner before vote and after vote model.

Classifier		Detection Rate	FPR	f-Measure	Accuracy	Classifier		Detection Rate	FPR	f-Measure	Accuracy
k-NN	Normal	87.23	0	93.18	80.3	MLP	Normal	85.99	11.65	84.11	78.08
	Suspicious	25.33	3.63	35.85			Suspicious	7.14	7.71	9.19	
	Phishing	100	25.86	80.79			Phishing	92.23	15.49	88.12	
C4.5	Normal	89.53	4.27	91.66	89.66	SVM	Normal	85.64	1.38	91.48	82.02
	Suspicious	61.11	5.15	44.89			Suspicious	26.67	4.34	35.17	
	Phishing	92.13	6.32	93.21			Phishing	98.73	22.18	84.55	
LDA	Normal	80.9	8.77	84.21	82.51	**Our Approach**	**Normal**	**94.25**	**0**	**97.04**	**97.54**
	Suspicious	0	7.71	0.00			**Suspicious**	**100**	**1.06**	**93.11**	
	Phishing	85.27	10.99	87.82			**Phishing**	**100**	**2.99**	**98.56**	

Table 12. The efficiency of all classifiers when training with UNSW-NB15 dataset in case of weak learner before vote and after the vote model.

Classifier		Detection Rate	FPR	f-Measure	Accuracy	Classifier		Detection Rate	FPR	f-Measure	Accuracy
k-NN	Analysis	14.02	0.2	23.71	90.97	MLP	Analysis	0	0.82	0.00	50.19
	Backdoor	34.19	0.61	19.58			Backdoor	0	0.71	0.00	
	DOS	63.03	2.28	59.42			DOS	35.71	4.96	0.48	
	Exploit	94.31	3.82	83.42			Exploit	46.31	9.47	41.55	
	Fuzzers	76.94	1.65	78.08			Fuzzers	2.19	7.87	2.41	
	Generic	99.96	0.06	99.87			Generic	39.25	3.38	55.25	
	Normal	99.69	1.27	99.06			Normal	93.47	28.64	67.05	
	Reconnaissance	82.52	0.38	86.81			Reconnaissance	1.76	4.25	0.17	
	Shellcode	98.95	0	99.21			Shellcode	0	0.46	0.00	
	Worms	100	0	98.85			Worms	0	0.05	0.00	
C4.5	Analysis	85.25	0.76	14.09	89.28	SVM	Analysis	0	0.82	0.00	27.59
	Backdoor	57.45	0.68	8.57			Backdoor	0	0.71	0.00	
	DOS	51.32	1.53	59.70			DOS	0	4.97	0.00	
	Exploit	73.98	3.63	75.41			Exploit	0	13.52	0.00	
	Fuzzers	75.6	2.64	70.75			Fuzzers	16.57	1.11	28.03	
	Generic	99.52	0.51	98.90			Generic	90.59	21.65	13.49	
	Normal	96.18	1.95	96.90			Normal	90.71	32.67	58.04	
	Reconnaissance	96.35	0.65	90.44			Reconnaissance	0	4.26	0.00	
	Shellcode	77.22	0.16	70.32			Shellcode	0	0.46	0.00	
	Worms	75	0.02	63.16			Worms	0.11	0.02	0.22	
LDA	Analysis	0	0.82	0.00	66.51	**Our Approach**	**Analysis**	**27.53**	**0**	**43.17**	**97.84**
	Backdoor	0	0.71	0.00			**Backdoor**	**100**	**0.61**	**24.13**	
	DOS	13.86	4.33	15.92			**DOS**	**100**	**0.58**	**94.07**	
	Exploit	47.6	9.38	42.36			**Exploit**	**100**	**0.65**	**97.85**	
	Fuzzers	27.36	7.29	2.55			**Fuzzers**	**100**	**0.04**	**99.77**	
	Generic	92.57	4.11	89.10			**Generic**	**100**	**0.04**	**99.92**	
	Normal	66.84	10.97	76.87			**Normal**	**100**	**0.02**	**99.98**	
	Reconnaissance	0	4.25	0.00			**Reconnaissance**	**100**	**0.36**	**95.72**	
	Shellcode	0	0.46	0.00			**Shellcode**	**100**	**0**	**100.00**	
	Worms	0	0.05	0.00			**Worms**	**100**	**0**	**100.00**	

Table 13. The efficiency of all classifiers when training with NSL-KDD dataset and KDD Cup'99 in case of weak learner before vote and after vote model.

Dataset	Classifier		Detection Rate	FPR	f-Measure	Accuracy	Dataset	Classifier	Precision	Specificity	FPR	f-Measure	Accuracy
KDD-Cup'99	k-NN	Normal	79.96	1.84	85.85	91.13	NSL-KDD	k-NN	98.71	99.03	0.97	99.06	99.17
		DoS	99.96	17.36	96.13				99.79	99.89	0.11	99.85	
		Probe	29.74	0.09	45.10				99.71	99.97	0.03	99.83	
		R2L	100	0	93.94				98.72	99.81	0.19	97.30	
		U2R	71.19	1.97	67.54				98.51	100	0	99.25	
	C4.5	Normal	96.01	1.9	94.00	97.64		C4.5	99.23	99.41	0.59	98.77	98.8
		DoS	99.93	0.05	99.95				99.71	99.85	0.15	99.79	
		Probe	97.99	0.14	93.82				98.76	99.85	0.15	98.82	
		R2L	80	0.01	77.04				95.36	99.32	0.68	96.58	
		U2R	74.95	0.7	80.71				86.57	99.96	0.04	87.22	
	LDA	Normal	78.58	6.63	75.17	86.21		LDA	89.01	89.52	10.48	78.32	72.91
		DoS	88.57	11.43	92.62				68.89	86.08	13.92	77.13	
		Probe	39.95	0.93	35.13				70.22	96.48	3.52	76.27	
		R2L	45.16	0.01	42.42				32.1	90.36	9.64	36.25	
		U2R	19.34	5.24	0.57				41.79	99.83	0.17	45.90	
	MLP	Normal	94.18	6.27	82.03	86.53		MLP	77.14	84.96	15.04	85.51	71.72
		DoS	98	9.25	97.29				89.26	94.01	5.99	80.21	
		Probe	8.46	0.4	15.17				83.4	97.63	2.37	50.45	
		R2L	0.38	0.02	0.73				0	87.19	12.81	0.00	
		U2R	100	5.26	0.02				0	99.7	0.3	0.00	
	SVM	Normal	81.27	0.23	89.30	94.46		SVM	94.72	95.88	4.12	92.91	66.02
		DoS	99.69	4.61	99.00				15.38	70.48	29.52	26.63	
		Probe	78.27	0.33	77.08				97.6	99.71	0.29	92.00	
		R2L	3.55	0.01	6.64				74.73	96.35	3.65	79.02	
		U2R	84.65	3.84	42.32				29.85	99.72	0.28	0.65	
	Our Approach	**Normal**	**99.78**	**0**	**99.89**	**99.96**		**Our Approach**	**100**	**100**	**0**	**99.99**	**99.99**
		DoS	**100**	**0**	**100.00**				**100**	**100**	**0**	**100.00**	
		Probe	**100**	**0.01**	**99.66**				**100**	**100**	**0**	**100.00**	
		R2L	**100**	**0**	**95.52**				**99.97**	**99.99**	**0.01**	**99.98**	
		U2R	**100**	**0.03**	**99.69**				**98.51**	**100**	**0**	**99.25**	

performance for classifying different types of Tor in scenario B dataset results up to 100%.

However, the overall efficiency was satisfactorily high. The experimental result, compared with efficiency between machine learning as weak learner five methods: k-NN, C4.5, MLP, SVM, was LDA and our approach before voting and after voting with adaboost.M1 found that our proposed model had the highest efficiency. This means that high detection accuracy and low false positives are ideal for further development for real-time intrusion detection. In addition, compared with other intrusion databases such as Phishing website dataset UNSW-NB15 dataset NSL-KDD dataset and KDD Cup'99 dataset, it was found that our proposed model still had the highest efficiency in detecting errors compared with other methods. Additionally, it was compared with other research (Hodo et al. 2014) studies that employed UNB-CIC Tor Network Traffic datasets and the result found was that our research had higher efficiency.

Efficiency compared with the experimental work presented in the report, CFS-ANN was used for 99.8% accuracy. However, the research was 100% for Scenario B dataset.

According to all experimental results, it could be confirmed that our proposed model not only had higher efficiency in detecting intrusion than other methods, but it also had efficiency in detecting new intrusions that have never been found in the system. It is suitable for detecting abnormalities in the current situations where new abnormalities are hidden in the network and they are harmful to implementation.

Funding

This project is supported by the Thailand Research Fund (TRF) under grant number RTA6080013.

References

Abdelhamid, N., A. Ayesh, and F. Habtahb. 2014. Phishing detection based associative classification data mining. *Expert Systems with Applications* 41 (13):5948–59. doi:10.1016/j.eswa.2014.03.019.

Abdelhamid, N., A. Ayesh, and F. Thabtah. 2014. Phishing detection based associative classification data mining. *Expert Systems With Applications (ESWA)* 41 (13):5948–59. doi:10.1016/j.eswa.2014.03.019.

AbdElrahman, S. M., and A. Abraham. 2014. Intrusion detection using error correcting output code based ensemble. In 14th International Conference on Hybrid Intelligent System, 181–86. Kuwait: IEEE.

Aburomman, A. A., and M. B. I. Reaz. 2016. A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing* 38 (C):360–72. doi:10.1016/j.asoc.2015.10.011.

Aburomman, A. A., and M. B. I. Reaz. 2017. A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Computers & Security* 65:135–52. doi:10.1016/j.cose.2016.11.004.

Amer, M., M. Goldstein, and S. Abdennadher. 2013. Enhancing one-class support vector machines for unsupervised anomaly detection. In Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, 8–15. Chicago, Illinois: ACM. doi:10.1177/1753193412444401

Anand, A., and B. Patel. 2012. An overview on intrusion detection system and types of attacks it can detect considering different protocols. *International Journal of Advanced Research in Computer Science and Software Engineering* 38 (1):94–98.

Bahl, S., and S. K. Sharma. 2015. Detection rate analysis for user to root attack class using correlation feature selection. In International Conference on Computing, Communication & Automation, 66–71. Noida, India: IEEE. doi:10.3389/fmed.2015.00066.

Dingledine, R., N. Mathewson, and P. Syverson. 2004. Tor: The second-generation onion router. In Proceedings of the 13th USENIX Security Symposium, 21–21. San Diego, CA: USENIX Association. doi:10.1186/1476-0711-3-21.

Eid, H. F., A. E. Hassanien, T. Kim, and S. Banerjee. 2013. Linear correlation-based feature selection for network intrusion detection model. In International Conference on Security of Information and Communication Networks, 240-248. Berlin, Heidelberg: Springer.

Freund, Y., and R. E. Schapire. 1996. Experiments with a new boosting algorithm, machine learning. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, 148–56. San Francisco: Morgan Kaufmann Publishers Inc.

Galar, M., A. Fernandez, E. Barrenechea, and H. Bustince. 2014. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42 (4):463–84. doi:10.1109/TSMCC.2011.2161285.

Ghafir, I., J. Svoboda, and V. Prenosil. 2014b. Tor-based malware and tor connection detection. In International Conference on Frontiers of Communications, Networks and Applications, 1–6. Malaysia: IEEE Xplore Digital Library.

Hall, M. A. 1999. Correlation-based feature selection for machine learning. Phd. Diss., University of Waikato.

Hamid, I. R. A., J. Abawajy 1, and T. H. Kim. 2013. Using feature selection and classification scheme for automating phishing email detection. *Studies in Informatics and Control* 22 (1):61–70. doi:10.24846/v22i2y101307.

Hodo, E., X. Bellekens, E. Iorkyase, A. Hamilton, C. Tachtatzis, and R. Atkinson. 2014. Machine learning approach for detection of nontor traffic. In Proceedings of the 12th International Conference on Availability, Reliability and Security, 85: 1–85:6. Reggio Calabria, Italy: ACM.

Hussain, J., and S. Lalmuanawma. 2014. A hybrid approach for determining the efficient network intrusion detection system. *The IUP Journal of Computer Sciences* 8 (3):34–36.

KDD Cup 1999. 1999. *UCI machine learning repository*. Irvine: University of California, School of Information and Computer Science. Accessed February 2018. https://archive.ics.uci.edu/ml/machine-learning-databases/kddcup99-mld/.

Kumar, P. A. R., and S. Selvakumar. 2013. Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. *Computer Communications* 36 (3):303–19. doi:10.1016/j.comcom.2012.09.010.

Kuncheva, L. I., and J. J. Rodríguez. 2014. A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems* 38 (2):259–75. doi:10.1007/s10115-012-0586-6.

Lashkari, A. H., G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani. 2017. Characterization of tor traffic using time based features. In Proceedings of the 3rd International Conference on Information Systems Security and Privacy, 253–62. Porto, Portugal: SCITEPRESS.

Mascaro, S., A. E. Nicholson, and K. B. Korb. 2014. Anomaly detection in vessel tracks using Bayesian networks. *International Journal of Approximate Reasoning* 55 (1):84–98. doi:10.1016/j.ijar.2013.03.012.

Michael, G., A. Kumaravel, and A. Chandrasekar. 2015. Detection of malicious attacks by meta classification algorithms. *International Journal of Advanced Networking and Applications* 6 (5):2455.

Moustafa, N., and J. Slay. 2015. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In MilCIS-IEEE Stream, Military Communications and Information Systems Conference, 1–6. Canberra, ACT, Australia: IEEE.

Nejad, T. R., and M. S. A. Abadi. 2014. Intrusion detection in computer networks through a hybrid approach of data mining and decision trees. *Walia Journal* 30 (S1):233–37.

Nguyen, H. Y., K. Franke, and S. Petrovic. 2014. Improving effectiveness of intrusion detection by correlation feature selection. In 2010 International Conference on Availability, Reliability and Security, 17–24. Krakow, Poland: IEEE.

Ren, Y. 2014. An integrated intrusion detection system by combining SVM with adaboost. *Journal of Software Engineering and Applications* 7 (12):1031–38. doi:10.4236/jsea.2014.712090.

Sagha, H., H. Bayati, J. R. Millán, and R. Chavarriaga. 2013. On-line anomaly detection and resilience in classifier ensembles. *Pattern Recognition Letters* 34 (15):1916–27. doi:10.1016/j.patrec.2013.02.014.

Shahbaz, M. B., X. Wang, A. Behnad, and J. Samarabandu. 2016. On efficiency enhancement of the correlation-based feature selection for intrusion detection systems. In Proceedings of the 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference, 1–7. Vancouver, BC, Canada: IEEE.

Sheykhkanloo, N. M. 2014. Employing neural networks for the detection of sql injection attack. In Proceedings of the 7th International Conference on Security of Information and Networks, 318–23. Glasgow, Scotland, UK: ACM. doi:10.1177/1753193413517839.

UNB, University of New Brunswick. 2018. http://www.unb.ca/cic/datasets/tor.html.

University of New Brunswick (UNB). 2017. http://www.unb.ca/cic/datasets/tor.html.

Wahba, Y., E. ElSalamouny, and G. ElTaweel. 2015. Improving the performance of multi-class intrusion detection systems using feature reduction. *IJCSI International Journal of Computer Science* 12 (3):255–62.

Zhang, H., Z. Xie, Y. Yang, Y. Zhao, B. Zhang, and J. Fang. 2017. The correlation-base-selection algorithm for diagnostic schizophrenia based on blood-based gene expression signatures. *BioMed Research International* 2017:7860506.

Zhang, Z., and X. Xie. 2010. Research on adaboost. m1 with random forest. In Proceedings of the 2nd International Conference on Computer Engineering and Technology, 647–52. Chengdu, China: IEEE.

Zhou, Z.-H. 2012. *Ensemble methods: Foundations and algorithms*. Boca Raton, FL: CRC Press.