Engineering Journal
issn: 0125-8281

Article

A New Incremental Decision Tree Learning for Cyber Security based on ILDA and Mahalanobis Distance

Saichon Jaiyen[a] and Ployphan Sornsuwit[b,*]

Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
E-mail: [a]kjsaicho@kmitl.ac.th, [b]ployphan.en@gmail.com (Corresponding author)

Abstract. A cyber-attack detection is currently essential for computer network protection. The fundamentals of protection are to detect cyber-attack effectively with the ability to combat it in various ways and with constant data learning such as internet traffic. With these functions, each cyber-attack can be memorized and protected effectively any time. This research will present procedures for a cyber-attack detection system Incremental Decision Tree Learning (IDTL) that use the principle through Incremental Linear Discriminant Analysis (ILDA) together with Mahalanobis distance for classification of the hierarchical tree by reducing data features that enhance classification of a variety of malicious data. The proposed model can learn a new incoming datum without involving the previous learned data and discard this datum after being learned. The results of the experiments revealed that the proposed method can improve classification accuracy as compare with other methods. They showed the highest accuracy when compared to other methods. If comparing with the effectiveness of each class, it was found that the proposed method can classify both intrusion datasets and other datasets efficiently.

Keywords: Cybersecurity, IDTL, incremental learning.

ENGINEERING JOURNAL Volume 23 Issue 5
Received 17 March 2019
Accepted 11 June 2019
Published 30 September 2019
Online at http://www.engj.org/
DOI:10.4186/ej.2019.23.5.71

DOI:10.4186/ej.2019.23.5.71

1. Introduction

At this current time, the internet has become an important part of people's routines and is utilized for business communication, online social activity, education, medicine, public sector support, etc. Once any of these aspects becomes significant, they are always prone to malicious activities and data theft because any important information stored in organizational networks can be a target of ill-intended individuals attempting to access and misuse these pieces of information. Cyber- attack detection systems are designed differently depending on system vulnerabilities and intrusion intentions such as phishing website, password guessing, spam, distributed denial of service (ddos) attacks, access to observe activities of a target, Elevating Privileges Access Attacks and so on. Currently, new technology is planned and designed to monitor network attacks, for example, cloud computing [1] or Internet of Things (IoT) [2], etc. Especially in current intrusion to communication by secure traffic there is an inadequate transmission of data to the communication, such as chat, file transfer protocol (ftp), p2p or tor traffic etc. It is difficult to detect when sending data over the network. And the intruders are now trying to develop a way to overcome the security of data traffic for commercial purposes, or maybe in an attempt to scramble and test the system's capabilities.

They can be classified into 2 major types – Anomaly-based IDS and Misuse-based IDS [3]. These 2 features have both pros and cons. Misuse-based IDS can detect intrusion accurately by memorizing patterns or dataset rules. Despite highly accurate detection, it is ineffective against newer intrusion due to its limited dataset or unfamiliarity to some systems. Anomaly-based IDS can detect intrusion through analyzing statistics of the normal behavior of users. In the case of any significantly unfamiliar activities contrasting from normal behavior, it can immediately detect them. Therefore, this feature can detect newer intrusions but holds a higher rate of false alarm.

Many previous studies attempted to use algorithms in order to function as anomaly-based IDS detection by presenting procedures to enhance detection features while lowering mistakes and speeding up the processor. In fact, it is compulsory to immediately notify of intrusion [3]–[7]. Linear Discriminant Analysis (LDA) is another important method of feature reduction that has been practiced for many years. This method is utilized effectively in IDS by combining preprocesses and classification of intrusion [8]–[12]. Many previous studies had applied LDA with other machine learning to enhance IDS together with newer intrusion datasets. Likewise, some studies developed and improved LDA algorithms [11], [13] to make the application more effective and more suitable such as Incremental Linear Discriminant Analysis (ILDA) [14] This method used incremental loading for processing. Similar to LDA generalization that models specific processes when finishing without reusing, it is considered as a method suitable for modern IDS. Some studies might classify data by using distance function for clustering as well [15]–[17] that benefits detection of a newer intrusion when a large number are hidden in other intrusion datasets. It's because of using distance function without forming a model resulting in newer intrusion detection.

And cyber-attack detection on the current network within the system should be able to incrementally learn behavior of the normal user and then continuously learn the types of invasion that can be detected immediately. This is a different point from traditional machine learning in traditional Anomaly-based IDS, which uses the model to detect abnormalities. Over time, the model will be updated to the new version. The learning model can't be detected immediately. It is a challenge to develop a network intrusion detection system that is currently in use. It can be incrementally learned behavior with usage and risk of invasion at any time through the data into the system and can be classified as the invasion types that are found in the current system effectively.

In relation to these previous studies, this study aimed to develop intrusion detection procedures through Incremental Decision Tree Learning (IDTL) to classify data in order to enhance the effectiveness and the suitability of the framework for detection on network traffic. To incrementally function, a framework is provided to build models and classify intrusion datasets of the classified structure in a binary hierarchy. Development and improvement of Incremental Linear Discriminant Analysis (ILDA) with using mahalanobis distance to measure distance to classify. Similar to the combination of supervised learning and unsupervised learning, it can detect intrusions that the model is familiar with and other unfamiliar intrusions. The objectives of this study are as follows:

1. Some features, out of the high number of features on network traffic, were selected through Pearson Correlation, as many features may not benefit the overall classification measurement.
2. Provide an IDTL structure for binary hierarchical cyber-attack detection and development and improvement from ILDA using mahalanobis distance. The procedure launches increment learning

of the data to form a constant, one-time, and immediate model without restoring the data to re-calculate with other data.

3. Effectiveness of other procedures of intrusion detection were also compared and analyzed.

This study, therefore, is broken down as follows: Section 2: The review of related studies, Section 3: Theory, Section 4: Materials and Methods, Section 5: Results, Section 6: Discussion and Section 7: Conclusions.

2. Relate Studies

Many researchers have tried to study machine learning used in a variety of algorithm-based intrusion detection. They may come in the form of unsupervised learning [15]–[17] clustering without target specification and supervised learning that is used for training to model in estimation before new data estimation. Semi-supervised learning [5], [13], [18]–[20] is another type involved in function estimation on labeled and unlabeled data, falling between unsupervised learning and supervised learning, and Ensemble Learning [21]–[23] which uses many classification models to vote on an estimation. However, even if the ensemble has to build multiple models for high-quality voting, it may not be suitable for intrusion detection on an always-available network.

Many mentioned procedures attempted to enhance detection by increasing accuracy and reducing false alarm rates. Therefore, many studies applied many methods called Hybrids [24]–[26] which are a combination of methods to improve maximum effectiveness of intrusion detection.

The study [10] used LDA to implement feature reduction and NDL-KDD dataset before classification on neutral networks. It was found that it could reduce features resulting in lower training time and highly effective classification of intrusions. Likewise, the study [25] functions both PCA and LDA to run feature extraction by finding class-pair that both PCA and LDA could pinpoint the best feature value. The feature was later classified through SVM. The result of the test found that it could improve efficiency. In addition, there were more ideas to improve LDA effectiveness. As shown in the study [26], Direct LDA was developed by removing S_b eigenvectors, corresponding to the eigenvalues that were equal to zero or close to zero and kept the null space of S_w, to enhance effective detection rates and lower false alarm rates. Not only did LDA prevent DoS but also black hole attacks of self-driving communication and semi self-driving vehicles in VANETs [27]. LDA was more efficient than QDA. It has been stated that feature selection and dataset intrusion could enhance effective classifications [9], [28]. Many researchers emphasized this point, as it could lower data insignificant for calculation, enhancing effectiveness and reducing time consumption. To function in real life applications, a lot of data on networks running through IDS should be significantly feature-selected so that it could detect an intrusion immediately as it occurs.

Currently, the procedure suitable for intrusion detection that is similar to real life application conditions is incremental learning because it can learn from large-scale dynamic stream data and build up a knowledge base over time to benefit future learning and decision-making processes [29]. While, intrusion on incoming and outgoing network traffic is essential to have statistical calculation of constant timely changes to make a decision at a certain moment ensuring if it is a detected intrusion [30]. The study [31] presented Weight ILDA (WILDA) to function with online hand-written Chinese character recognition. WILDA is a method to recognize the issue of an uncertain number of incremental data through methods of weight of S_w and S_b calculation to reduce the problem of lower accuracy found in a small proportion of increment new samples. In the test, WILDA was found to solve this problem by increasing accuracy and holding higher efficiency than ILDA.

Besides this, there is another study [32] developing online system FNTAE that can detect real-time intrusion through FIncLDA as a learning model and k-NN as a decision agent to make decisions regarding intrusion detection. This system can utilize chunk LDA for online learning and can be applied to increase the effectiveness of intrusion detection. To increase effectiveness of intrusion detection, preprocessing is important. Many studies tested feature extraction or feature selection through different methods as found in the study [33] using Discrete Wavelet Transform (DWT) for effective enhancement and iPCA in conducting an interactive factor analysis. This study used them for visual comparison of various features and the researchers comparatively experimented data NSL-KDD projection during DWT and non-DWT usage. It was found that using DWT made a clear separation among the attack categories in some classes, for example, R2L which is unidentifiable with raw features. However, R2L can be identified with DWT and is effective with a machine learning test. Additionally, another study [14], [34] used Chi Squared Attribute Evaluator to select relevant features for classification through LDA and logistic regression (LR). It revealed that both methods could perform well on multiclass and binary classification. Despite higher accuracy than Naïve Bayes, it cannot be higher than SVM and C4.5 having a low computational overhead that is higher than SVM is considered more appropriate to the development of real-time network monitoring. Adding Tor dataset indicates the current significance [37] and discussed Pearson [35].

As shown in the previous studies, this study is interested in developing each procedure of cyber-attack detection to be effective through IDTL based on Sequential Incremental LDA (ILDA) and mahalanobis distance integrated with classification of Tor traffic datasets, which is an invasion of hidden services and other cyber-attack datasets. It aims to be more effective and function in Incremental Learning that can increment numbers of new datasets for modeling constantly as well as efficiently classify each attack type. The model is extended by Incremental learning which will probably be applied with real cyber-attack detection on a real network in the future.

3. Theory

3.1. Pearson Correlation

Pearson correlation coefficient is to determine and measure of the strength of the association between the variable X and variable Y based on the method of covariance of these two values, divided by the product of their standard deviations the following values are calculated in Eq. (1) [35].

$$\text{Let } X = \{x_1, x_2, \dots, x_n\} \text{ and } \mathrm{Y} = \{\mathrm{y}_1, \mathrm{y}_2, \dots, \mathrm{y}_\mathrm{n}\}$$

$$r_{XY} = \sum_{i=1}^{n} \frac{(X_i - \bar{X})}{\sqrt{\sum_{i=1}^{n} (X_i - X)^2}} \cdot \frac{Y_i - \bar{Y}}{\sqrt{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}} \tag{1}$$

where $\mathrm{r_{XY}}$ is correlation coefficient whose value is between -1 and 1.

3.2. Sequential Incremental Linear Discriminant (Sequential ILDA)

Let $X = \{x_1, x_2, \dots, x_N\}$ be a set of training samples with M classes and N be the number of training samples. Let y be the new incoming datum with the class label k, the new eigenspace model, $\Omega' = (Sw', Sb', \bar{x}, N+1)$, must be updated by using only the old Ω and the new incoming datum y. For the new mean parameter $\bar{x}'$, it can be calculated as Eq. (2) [14]:

$$\overline{x}' = \frac{(N\bar{x} + y)}{(N+1)} \tag{2}$$

For between-class scatter matrix Sb', if $k = M + 1$ representing a newly introduced class, are shown in Eq. (3) and Eq. (4) [14]:

$$Sb' = \sum_{c=1}^{M} n_c (\overline{x_c} - \bar{x}')(\overline{x_c} - \bar{x}')^T + (y - \bar{x}')(y - \bar{x}')^T \tag{3}$$

$$Sb' = \sum_{c=1}^{M+1} n_c' (\overline{x_c} - \bar{x}')(\overline{x_c} - \bar{x}')^T \tag{4}$$

where $\mathrm{n_c'}$ is the number of samples in class c after having data y appear, $\mathrm{n_c}'$=$\mathrm{n_c}$ when $1 \le \mathrm{c} \le \mathrm{M}$, $\mathrm{n_c'} = 1$ when $\mathrm{c} = \mathrm{M} + 1$, and $\overline{\mathrm{x_c}} = \mathrm{y}$ when $\mathrm{c} = \mathrm{M} + 1$.
When $1 \le c \le M$ then $\mathrm{Sb'}$ is updated using the Eq. (5) [14]

$$Sb' = \sum_{c=1}^{M} n_c' (\bar{x}_c - \bar{x}')(\bar{x}_c - \bar{x}')^T \tag{5}$$

where $\overline{\mathrm{x_c}} = (1/(\mathrm{n_c} + 1))(\mathrm{n_c}\overline{\mathrm{x_c}} + \mathrm{y})$ and $\mathrm{n_c}'$=$\mathrm{n_c} + 1$ if y equals class c; else $\overline{\mathrm{x_c}}' = \overline{\mathrm{x_c}}$ and $\mathrm{n_c'} = \mathrm{n_c}'$

For within-class scatter matrix Sw, if y is a new class which means k is the $(M+1)$ class. Therefore, updating within-class scatter matrix is not changing as in the Eq. (6).

$$Sw' = \sum_{c=1}^{M} \sum_c + \sum_k = \sum_{c=1}^{M+1} \sum_c = \sum_{c=1}^{M} \sum_c \tag{6}$$

In case that $1 \leq c \leq M$ will update s_w as in the equation shown at the proof in the Appendix.

$$Sw' = \sum_{c=1,c\neq k}^{M} \sum_c + \sum_k{}' \tag{7}$$

$$\Sigma_k' = \Sigma_k + \frac{n_k}{n_k+1}(y-\bar{x}_k)(y-\bar{x}_k)^T \tag{8}$$

3.3. Mahalanobis Distance

Mahalanobis distance is another interesting measure of the distance between two points in multivariate space as defined in Eq. (9) where d is mahalanobis distance, x is the observation and μ is the mean of samples. S is the covariance, it can be displayed as Eq. (9) [36].

$$d(x,\mu) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{9}$$

4. Materials and Methods

4.1. The Proposed Method

This research proposes IDTL which is a new incremental hierarchical learning based on Incremental LDA and mahalanobis distance. The proposed method adopts an Incremental LDA method that can learn several attack types through binary classification. This method is different from traditional LDA algorithms and intrusion detection procedures from other studies. Incremental LDA learns to perform intrusion detection through tree-diagram node forming. Each node can classify different classes of the attack types. IDTL can specify new data through calculation as a one-time process. Briefly, each new data is calculated to form a model once before being completely discarded. Therefore, the learned data can be discarded after being learned. There is no need to store the old data to learn the new incoming data. Additionally, classification is enhanced by mahalanobis distance to increase accuracy. This proposed method is suitable for a modern model of cyber-attack detection on an online computer network that is prone to cyber-attack all the time without attack type identification and damage protection during the application. The process of doing this research is presented as Fig. 1.

We used Tor dataset [37] which is a dataset that has hidden services in traffic network. It has developed a tool to use tor widely and is difficult to detect because it uses multiple protocols. Tor will protect or obscure the personal privacy of its users, as well as their freedom and ability from Internet activities. Therefore, it is a vulnerability for attackers to use Tor as a channel to avoid detection when attacking the network as show in Fig. 1.

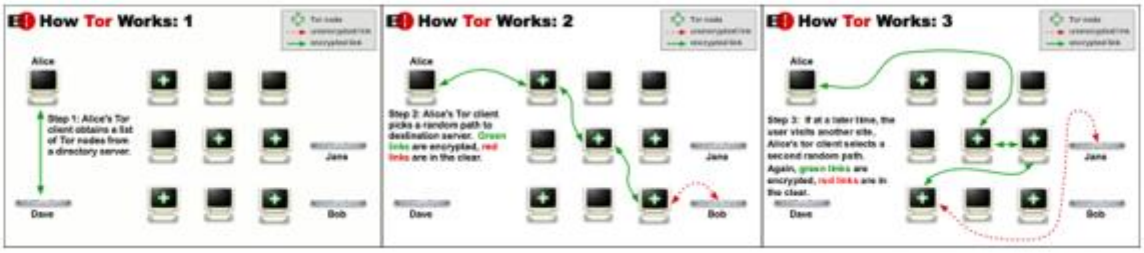
How Tor Works: 1
Alice
Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.
Jane
Dave
Bob
How Tor Works: 2
Alice
Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.
Jane
Dave
Bob
How Tor Works: 3
Alice
Step 3: If at a later time, the user visits another site, Alice's tor client selects a second random path. Again, green links are encrypted, red links are in the clear.
Jane
Dave
Bob

Fig. 1. Attack process of Tor [37].

DOI:10.4186/ej.2019.23.5.71

Our research was conducted with the Tor dataset in two scenarios: scenario A represents of the implementation classification binary class and scenario B represent the implementation of multiclass classification. It is also tested with other datasets that looks like an invasion, for example, NSL-KDD as a dataset revised from KDD Cup'99 [38], spam dataset [39], phishing dataset [40] and SAME is the dataset of Android system invasion [41]. All stages of the experiment are shown in Fig. 1. They consists of the stages of classification with prior stage of data preprocessing for data availability. Then, features were selected to obtain only related features. Next stages are training and testing along with performance measurement. All stages of the experiment can be discussed as follows:

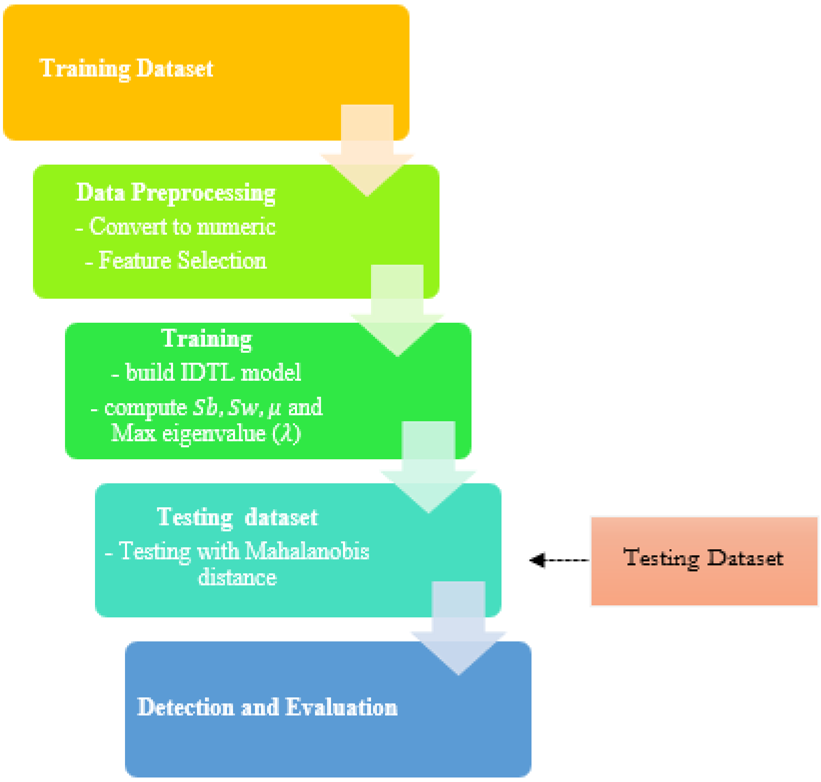

Fig. 2. Proposed cyber-attack detection model based on IDTL.

4.1.1. Data preprocessing

At this stage, we will convert every text feature into numeric features and select a feature using Pearson Correlation. Because of the algorithm that selects the relationship of each feature, the research uses a variety of fields [42], [43] efficiently. Data is divided into two parts. The first part is for training stage and the second one is for testing stage. By doing so, data for testing stage is not found in data for training stage. This way is similar to authentic intrusion detection in the network.

4.1.2. Training stage

After completing the preprocessing of the training, datasets of the training procedure can be divided into two main structures. First, training structure for the binary class and this is for the case normal and abnormal classification is required. Second, multiclass structure is for classifying multi classes that are both normal and several attack types. Both structures use similar IDTL method.

The binary structure is shown in Fig. 3. This demonstrates a learning process of hierarchical visualization of the IDTL, which is classified as normal and abnormal intrusion detection.

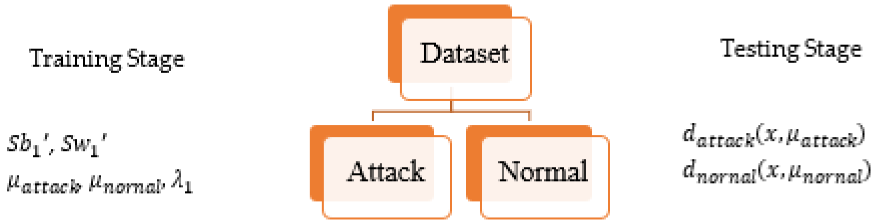
Training Stage
Sb_1', Sw_1'
$\mu_{attack}, \mu_{normal}, \lambda_1$
Dataset
Attack
Normal
Testing Stage
$d_{attack}(x, \mu_{attack})$
$d_{nornal}(x, \mu_{nornal})$

Fig. 3. Procedures of training the proposed model based on binary classes.

Figure 4 demonstrates a learning process of hierarchical visualization of the IDTL which is incremental learning structures: the multiclass classifier structure classifies intrusion detection. The procedure launches increment learning of the data to form a constant, one-time, and immediate model without restoring the data to re-calculate with other data.

All stages of training process of hierarchical structure based on IDTL in Fig. 4 are described as follows:

First, data are divided into Normal class and Attack class as the first stage of classification. When cyber-attack occurs, detection operates to analyze whether data are Normal class or Attack class. At this stage, all data are incremented in the learning of Sb_1' , Sw_1' between Normal Class and Attack Class, and the mean (μ) of two classes. Then, the first maximum eigenvalue λ_1 is solved to prepare testing stages.

Next, Attack type 1 and Other attack#1 types are classified. Attack type 1 has cyber-attack behaviors that can be clearly distinguished from other attack types of intrusion due to its various attempts to cause attack in services that can be found in network disturbance.

It processes through the learning of Sb_2', Sw_2', mean of two classes (μ) and the second maximum eigenvalue λ_2. Next, Sb_3', Sw_3', mean of two classes (μ) and the third maximum eigen value λ_3. IDTL will increment learn in attack type classes in sequence up to the last stage, Attack type n-1 and Attack type n are classified as separate from each other as shown in Fig. 4.

Our research has improved the equation of the Sequential ILDA, which is multiclass into a binary class equation for binary learning in each tree hierarchy.

$$Sb' = \sum_{c=1}^{2}(\bar{x}'_1 - \bar{x}'_2)(\bar{x}'_1 - \bar{x}'_2)^T \tag{10}$$

In the training process, we will read the sequential data one record at a time to calculate and update the virtual model to read the internet traffic to calculate one record. After calculating and updating the model, we do not take that information back to calculations.

That means that any data will be calculated only once, then the model will represent all data. This research has focused on data stream reading, which is used for outlier detection [44].

The incremental learning algorithm is highly effective and is consistent with outlier detection with network-based intrusion detection data streams.

In the training stage, every learning n class hierarchy of IDTL, the eigenspace is updated in every 1 record that is currently taining as shown in Fig. 5.

4.1.3. Testing stage

During the testing stage, mahalanobis distance is used to find the distance for identifying the class of test data in the same sense as real network detection. If the test data is classified to be attack type, these data must be submitted to calculate the distance until they can be identified into the class they belong. The procedure repeats eventually till all data classes have been identified.

$$d(x, \mu) = \sqrt{(x-\mu)^T \lambda^{-1} (x-\mu)} \tag{11}$$

To calculate the distance, we use the Max eigenvalue (λ) and mean (μ) obtained by increment learning in each class of the class to calculate in the equation.

DOI:10.4186/ej.2019.23.5.71

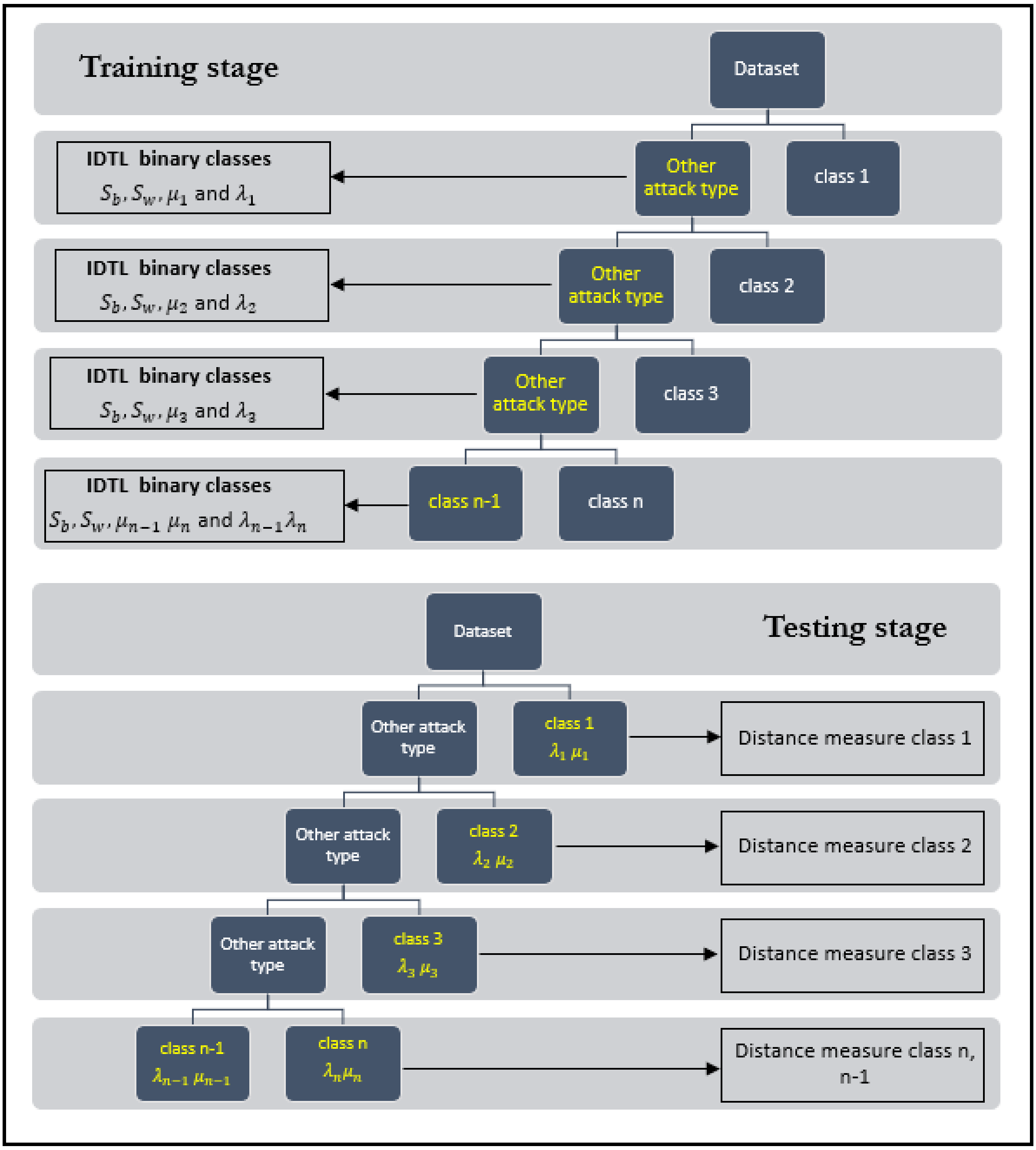

Fig. 4. Procedures of training the proposed model based on multiclass classes.

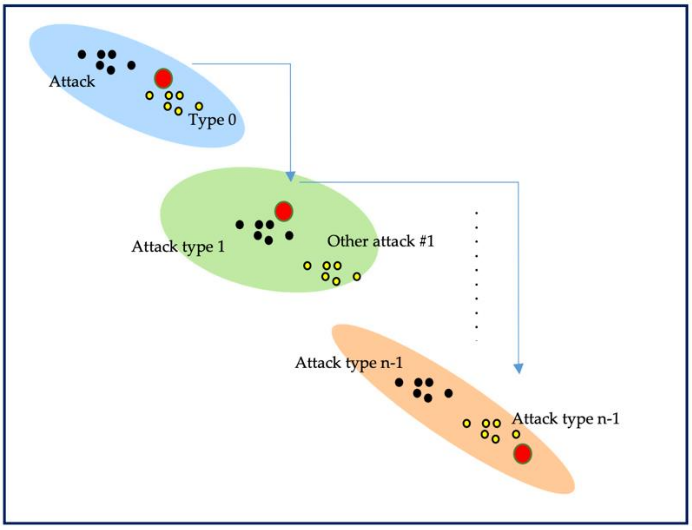

Fig. 5. The eigenspace update in each IDTL hierarchy.

4.2. Evaluation

This research used various methods of performance measurement to ensure analysis accuracy. In classification, class could be predicted through all test data that underwent performance measurement of values as shown in Table 1. Then, the following values' performance was measured.

Table 1. Confusion Matrix

Predict Value	Actual Value	
	Positive	Negative
Positive	True Positive: TP	False Positive: FP
Negative	False Negative: FN	True Negative: TN

Precision: The amount of data predicted from the prediction of considering class as shown in Eq. (10).

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

Recall or Sensitivity or Detection Rate is a proportion of True Positive cases that are correctly predicted as positive as shown in Eq. (11).

$$Recall = \frac{TP}{TP+FN} \tag{11}$$

F-measure is an overall measure of Precision and Recall as shown in Eq. (12).

$$f - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

Accuracy is the number of correct data prediction from classes as shown in Eq. (13).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{13}$$

4.3. Implementation

This research was conducted by using a personal computer Intel Core, i5-4258U CPU @2.4GHz, 8 GB memory without GPU acceleration and our algorithm was implemented in MATLAB R2017. In the step of preprocess it was used to select important features before the training and testing of data that was previously presented. In classification, class of attack types training and testing was done to measure efficiency.

Next, this study's method was compared with other methods of various machine's learning. For example, Naïve Bayes, Decision Tree, k-NN, Multi-layer Perceptron (MLP) and SVM. It was also tested with other attack types that the dataset currently represents an intruder to confirm the effectiveness of our approach.

4.4. Algorithm

The proposed incremental decision tree learning (IDTL) algorithm can be described as follows:

```
Algorithm IDTL Algorithm

Input: Training set = $\{(x_{1,}y_1), (x_{2,}y_2), \dots, (x_n y_n)\}$ , with  class label $y_i \in \{1, .., k\}$ of training dataset
Output: IDTL Model
For $i = 1, .., n$ // $n$ is number of samples
    For $y = 1, \dots, k$ // $k$ is number of class label
        If class label belong to any $k$ class. Then
            Update $\bar{\mu}'$, $Sb'$ and $Sw'$
        Else
            Update $\bar{\mu}'$, $Sb'$ and $Sw'$  for other $k$ class
            Calculate max eigenvalue for any $k$ class // for binary classification
            If class label belong to any $k - 1$ class
                Update $\bar{\mu}'$, $Sb'$ and $Sw'$
        Else
                Update $\bar{\mu}'$, $Sb'$ and $Sw'$  for  $k$ class
                Calculate max eigenvalue for any $k$ class // for binary classification
            End
        End
    End
    IDTL model with max eigenvalue of every class label for hierarchical distance measure

                    $d(x, \mu) = \sqrt{(x - \mu)^T \lambda^{-1} (x - \mu)}$
End
```

5. Results and Discussion

5.1. Results

We will test with Tor the data stored in a real-world model. The view is divided into 2 scenarios. Scenario A tests the binary classes Tor and non –Tor. In Scenario B, it tests multiclass include Browsing, Email, Chat, Audio, Video, FTP, VoIP and P2P. Each class has different numbers and types of hidden services. We have similar and smaller group classes as follows, Browsing's label as class 1, FTP + P2P label as class 2, Audio + VOIP label as class 3 and Chat + Email + Video label as class 4

The experiment will use all data from all developers, divided into 70 percent for training datasets and 30 percent for testing datasets. Data for testing will never appear in the training dataset as with cyber attack detection in computer networks.

5.1.1. Feature selection

In scenario A: 9 features were selected out of a total of 28 features using as shown in Table 2. And scenario B: 11 features were selected out of a total of 28 features as shown in Table 3.

Table 2. Selected features from the feature selection stage and its explanation of scenario A.

Feature No.	Feature Name
8	Flow Packets/s
7	Flow Bytes/s
6	Flow Duration
5	Protocol
19	Bwd IAT Max
1	Source IP
2	Source Port
12	Flow IAT Min
9	Flow IAT Mean

Table 3. Selected features from the feature selection stage and its explanation of scenario B.

Feature No.	Feature Name
2	Source Port
11	Flow IAT Max
15	Fwd IAT Max
19	Bwd IAT Max
18	Bwd IAT Std
14	Fwd IAT Std
4	Destination Port
10	Flow IAT Std
6	Flow Duration
1	Source IP
9	Flow IAT Mean

5.1.2. Results for IDTL

When tested, IDTL is most effective when compared to traditional ILDA algorithms, and better than without feature selection is shown in Table 4 - 5. When we compared the IDTL with feature selection, we proposed the choice of feature, or no feature being chosen. In addition, compared with traditional ILDA with mahalanobis the distance would have a classification procedure similar to the one we proposed. The difference is based on the original Sb'value of the traditional ILDA algorithm and without the feature selection.

The results show that the methods we proposed are most effective overall. And to determine the f-measure and accuracy, the sub-class was found to be the most effective as well.

In the case of a binary class, it is classified separately between Tor traffic and non Tor, which is classified as normal and abnormal. In the case of a multi class, it distinguishes difficult and some illegal services in some countries. Like P2P. Other services are also unobtrusive services, such as submitting malware via ftp. The experiments show that IDTL can detect these services and performs better than algorithms others have compared.

Table 4 - Table 5. Compare the efficiency between the algorithms we proposed in the feature selection and without the feature selection and the traditional ILDA approach with the mahalanobis distance of scenario A and scenario B.

Table 4. Comparison of performance between algorithms we proposed in feature selection and not feature selection and conventional traditional ILDA with mahalanobis distance of scenario A.

Traditional ILDA with mahalanobis distance	Tor	Non-Tor
Precision	14.27	100
Sensitivity	100	19.15
Specificity	19.15	100
f-Measure	24.98	32.14
Accuracy	28.74	
IDTL without features selection	Tor	Non-Tor
Precision	22.3	99.99
Sensitivity	99.96	53.16
Specificity	53.16	99.96
f-Measure	36.47	69.42
Accuracy	58.71	
IDTL with feature selection	Tor	Non-Tor
Precision	44.68	97.42
Sensitivity	82.52	86.62
Specificity	86.62	82.52
f-Measure	57.97	91.70
Accuracy	86.14	

Table 5. Comparison of performance between algorithms we proposed in feature selection and not feature selection and conventional traditional ILDA with mahalanobis distance of scenario B.

Methods	Class 1	Class 2	Class 3	Class 4
Traditional ILDA with mahalanobis distance				
Precision	50.31	87.58	70.21	45.32
Sensitivity	51.35	66.44	77.52	49.1
Specificity	87.36	96.99	80.32	86.64
f-Measure	50.82	75.56	73.68	47.13
Accuracy	64.39			
IDTL with full features	Class 1	Class 2	Class 3	Class 4
Precision	50.31	85.49	71.4	48.13
Sensitivity	51.35	66.61	77.96	52.03
Specificity	87.36	96.39	81.31	87.35
f-Measure	50.82	74.88	74.54	50.00
Accuracy	65.13			
IDTL with feature selection	Class 1	Class 2	Class 3	Class 4
Precision	55.6	98.55	97.72	75.37
Sensitivity	78.38	93.32	75.97	81.31
Specificity	84.41	99.56	98.94	94
f-Measure	65.05	95.86	85.48	78.23
Accuracy	81.63			

Then, when comparing the IDTL with other machine learning methods, the results are shown in Table 6 and Table 7. Even though IDTL is an incremental learning course, the overall classification performance is far superior to any other learning machine.

Table 6. IDTL performance vs. machine learning of scenario A.

Method	Efficiency	Tor		Non-Tor
IDTL	Precision	44.68		97.42
	Sensitivity	82.52		86.62
	Specificity	86.62		82.52
	f-Measure	57.97		91.70
	Accuracy		86.14	
Tree	Precision	11.86		100
	Sensitivity	100		0.07
	Specificity	0.07		100
	f-Measure	21.21		0.14
	Accuracy		11.92	
Naïve Bayes	Precision	10.86		87.49
	Sensitivity	36.3		59.9
	Specificity	59.9		36.3
	f-Measure	16.72		71.11
	Accuracy		57.11	
k-NN	Precision	15.9		98.5
	Sensitivity	96.44		31.4
	Specificity	31.4		96.44
	f-Measure	27.30		47.62
	Accuracy		39.11	
MLP	Precision	16.02		89.48
	Sensitivity	32.78		76.87
	Specificity	76.87		32.78
	f-Measure	21.52		82.70
	Accuracy		71.65	
SVM	Precision	15.1		100
	Sensitivity	100		24.36
	Specificity	24.36		100
	f-Measure	26.24		39.18
	Accuracy		33.33	

Considering Table 6, it was found that when tested scenario A: IDTL had an accuracy of 74.66 and had the highest class of f-measure. When considering the class, it was found that IDTL could best classify two classes. As in the Table 7 scenario B: the highest accuracy is 81.63 and this classification is most effective.

Table 7. IDTL performance compared with machine learning of scenario B.

Method	Efficiency	Class 1	Class 2	Class 3	Class 4
IDTL	Precision	55.6	98.55	97.72	75.37
	Sensitivity	78.38	93.32	75.97	81.31
	Specificity	84.41	99.56	98.94	94
	f-Measure	65.05	95.86	85.48	78.23
	Accuracy	81.63			
Tree	Precision	60.53	72.48	78.44	74.4
	Sensitivity	76.51	67.64	80.18	56.31
	Specificity	87.57	91.79	86.81	95.63
	f-Measure	67.59	69.98	79.30	64.10
	Accuracy	72.01			
Naïve Bayes	Precision	60	73.41	78.13	73.14
	Sensitivity	74.84	67.12	80.29	57.66
	Specificity	87.57	92.23	86.55	95.22
	f-Measure	66.60	70.12	79.20	64.48
	Accuracy	71.85			
k-NN	Precision	55.13	66.87	75.98	51.14
	Sensitivity	59.25	57.02	87.93	40.54
	Specificity	87.99	90.97	83.37	91.26
	f-Measure	57.12	61.55	81.52	45.23
	Accuracy	66.00			
MLP	Precision	52.17	42.92	66.89	0
	Sensitivity	2.49	99.66	76.52	0
	Specificity	99.43	57.66	77.34	100
	f-Measure	4.75	60.00	71.38	0
	Accuracy	53.28			
SVM	Precision	60.23	97.7	8.33	26.34
	Sensitivity	33.06	94.35	0.11	93.24
	Specificity	94.56	99.29	99.27	41.16
	f-Measure	42.69	96.00	0.22	41.08
	Accuracy	46.64			

5.1.3. Other datasets

When testing a cyber-attack dataset set, it was found that, initially, the IDTL method we proposed was highly effective at classifying different attacking behaviors. And many systems such as NSL-KDD are the popular datasets for detecting abnormalities. There are 5 classes including Normal, Dos, Probe, R2L, and U2R. Next is the Phishing website, a dataset that used phishing sites.

There are two classes, Phishing and Non-Phishing. Next, SAME which is an invasion on the Android operating system. There are two classes of smartphone applications: benign and malicious. Next, the spam base which is a spam-infested dataset. There are two classes, spam and non-spam. The dataset is still present, and IDTL is tested. In the case of NSL-KDD, we used the KDDTrain + _20Percent dataset for the training dataset and KDDTest + for testing datasets. It's 100% used by developer's other datasets and uses 70% for training and 30% for testing. Testing data is new in the training process the results of the experiment show the efficiency as shown in Table 8.

Table 8. The comparison of accuracy machines learning between IDTL with other datasets.

Dataset	Method/ Accuracy					
	IDTL	C4.5	Naïve Bayes	k-NN	MLP	SVM
NSL-KDD	75.71	74.73	68.12	71.38	70.16	70.86
SAME	96.04	95.38	93.23	95.38	52.15	92.57
Phishing	91.05	89.93	87.13	83.54	49.23	71.30
Spambase	85	82.61	43.36	72.83	34.28	84.86

5.2. Discussion

The results show that IDTL is highly effective in classifying cyber-attacks. The structure of the IDTL is an incremental learning model that updates the model in sequential training. The equation we think is that the value of $Sb'$is computed in binary class in each class to identify any two nodes, thus obtaining the appropriate value of λ to find the distance to classify for the two nodes. If using ILDA's traditional Sb', the equation is to find the value between any class by the number of classes, for example, to classify 4 classes. Each class of the binary tree computes the other classes by taking the mean of all $\bar{x}'$ together with this there are some distortions in the calculation.

The IDTL focuses on only one layer, two layers, as a layered layer. The final class is the traditional, incremental learning model that is being updated to detect other types of cyber-attacks. And IDTL classified with distance in the testing phase by recognizing the values of the training phase, similar to the research of Aborujilah and Musa [45] which has the same high efficiency.

6. Conclusion

In this paper, IDTL developed a cyber- attack detection algorithm. Based on the ILDA algorithm, our research has tested on the main dataset, the Tor dataset, which is a dataset of hidden services. Since the current invasion is difficult to detect on the computer network, the goal of this research is to develop algorithms that can incrementally learn in hierarchical order. The proposed algorithms have feature selection to select only the most important features and classify them by using Pearson correlation as the algorithm for selecting the feature and developing the IDTL. Some enhancements have been made to optimize the IDTL structure. The results showed that IDTL was the most effective when compared to other methods, both binary and multiclass, as well as when tested with other cyber-attack datasets. It's high performance as well in a variety of ways regarding system intrusion. Future research will develop an incremental learning system based on current research. IoT or smart devices must be able to detect real-time intrusions at all times, such as in a factory or smart farmer.

7. Appendix

When new sample y in the kth class; $k\epsilon[1, M]$ as $\Sigma_{x\epsilon\{x_k\}}(x - \bar{x}_k) = 0$ Then, covariance matrix is equally updated as in the equation.

$$\Sigma_k' = \Sigma_k + \frac{{n_k}^2 + n_k}{n_k + 1^2}(y - \bar{x}_k)(y - \bar{x}_k)^T$$

$$= \Sigma_k + \frac{n_k}{n_k + 1}(y - \bar{x}_k)(y - \bar{x}_k)^T$$

DOI:10.4186/ej.2019.23.5.71

References

[1] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18–47, 2017.

[2] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[4] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 1690–1700, Mar. 2014.

[5] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484–497, Feb. 2017.

[6] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[7] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 70–91, Jan. 2015.

[8] Z. Tan, A. Jamdagni, X. He, and P. Nanda, "Network intrusion detection based on LDA for payload feature selection," *2010 IEEE Globecom Workshop on Web and Pervasive Security:* 6-10 December 2010, Miami, Florida, pp. 1545–1549, 2010.

[9] S. S. Sathya, R. G. Ramani, and K. Sivaselvi, "Discriminant analysis based feature selection in KDD intrusion dataset," *International Journal of Computer Applications*, vol. 31, pp. 1-7, 2011.

[10] R. Datti and B. Verma, "Feature reduction for intrusion detection using linear discriminant analysis," *International Journal on Engineering Science and Technology*, vol. 2, no. 4, pp. 1072–1078.

[11] S. Singh and S. Silakari, "Generalized discriminant analysis algorithm for feature reduction in Cyber Attack Detection System," *International Journal of Computer Science and Information Security*, vol. 6, no. 1, pp. 173-180, 2009.

[12] P. G. Jeya, M. Ravichandran, and C. S. Ravichandran, "Efficient classifier for R2L and U2R attacks," International Journal of Computer Applications, vol. 45, no. 21, pp. 28-32, 2012.

[13] Q. Gao, Y. Huang, X. Gao, W. Shen, and H. Zhang, "A novel semi-supervised learning for face recognition," *Neurocomputing*, vol. 152, pp. 69–76, Mar. 2015.

[14] S. Pang, S. Ozawa, and N. Kasabov, "Incremental linear discriminant analysis for Classification of data streams," *IEEE transactions on Systems, Man, and Cybernetics, part B (Cybernetics)*, vol. 35, no. 5, pp. 905-914, 2005.

[15] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in *Networked Digital Technologies*, R. Benlamri, Ed. Berlin, Heidelberg: Springer, 2012, vol. 293, pp. 135–145.

[16] K. S. A. Kumar and A. M. M. O. Chacko, "Clustering algorithms for intrusion detection: A broad visualization," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, Udaipur, India, 2016, pp. 135:1–135:4.

[17] A. Bohara, U. Thakore, and W. H. Sanders, "Intrusion detection in enterprise systems by combining and clustering diverse monitor data," in *Proceedings of the Symposium and Bootcamp on the Science of Security*, Pittsburgh, Pennsylvania, 2016, pp. 7–16.

[18] M. Wurzenberger, F. Skopik, M. Landauer, P. Greitbauer, R. Fiedler, and W. Kastner, "Incremental clustering for semi-supervised anomaly detection applied on log data," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, Reggio Calabria, Italy, 2017, pp. 31:1–31:6.

[19] Z. Xue, Y. Shang, and A. Feng, "Semi-supervised outlier detection based on fuzzy rough C-means clustering," *Mathematics and Computers in Simulation*, vol. 80, no. 9, pp. 1911–1921, May 2010.

[20] Y. Yuan, G. Kaklamanos, and D. Hogrefe, "A novel semi-supervised adaboost technique for network anomaly detection," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Malta, Malta, 2016, pp. 111–114.

[21] M. A. Jabbar, R. Aluvalu, and S. S. S. Reddy, "Cluster based ensemble classification for intrusion detection system," in *Proceedings of the 9th International Conference on Machine Learning and Computing*, Singapore, Singapore, 2017, pp. 253–257.

[22] P. Arun Raj Kumar and S. Selvakumar, "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems," *Computer Communications*, vol. 36, no. 3, pp. 303–319, Feb. 2013.

[23] S. T. Miller and C. Busby-Earle, "Multi-Perspective Machine Learning a Classifier Ensemble Method for Intrusion Detection," in *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing*, Ho Chi Minh City, Vietnam, 2017, pp. 7–12.

[24] J.-G. Yang, J.-K. Kim, U.-G. Kang, and Y.-H. Lee, "Coronary heart disease optimization system on adaptive-network-based fuzzy inference system and linear discriminant analysis (ANFIS—LDA)," *Personal Ubiquitous Comput.*, vol. 18, no. 6, pp. 1351–1362, Aug. 2014.

[25] A. A. Aburomman and M. B. I. Reaz, "Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection," in *Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2016*, Xi'an, China, 2017, pp. 636–640.

[26] A. Saad, C. Khalid, and J. Mohamed, "Network intrusion detection system based on Direct LDA," in *2015 Third World Conference on Complex Systems (WCCS)*, Marrakech, Morocco, 2015, pp. 1–6.

[27] K. M. A. Alheeti, A. Gruebler, and K. McDonald-Maier, "Using discriminant analysis to detect intrusions in external communication for self-driving vehicles," *Digital Communications and Networks*, vol. 3, no. 3, pp. 180–187, Aug. 2017.

[28] R. Datti and S. Lakhina, "Performance comparison of features reduction techniques for intrusion detection system," *International Journal of Computer Science and Technology*, vol. 3, no. 1, pp. 332-335, 2012.

[29] H. He, S. Chen, K. Li, and X. Xu, "Incremental learning from stream data," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1901–1914, Dec. 2011.

[30] D. Bhosale and R. Ade, "Intrusion detection using incremental learning from streaming imbalanced data," *International Journal of Managing Public Sector Information and Communication Technologies*, vol. 6, no. 1, pp. 09–20, Mar. 2015.

[31] L. Jin, K. Ding, and Z. Huang, "Incremental learning of LDA model for Chinese writer adaptation," *Neurocomputing*, vol. 73, no. 10, pp. 1614–1623, Jun. 2010.

[32] S. Pang, Y. Peng, T. Ban, D. Inoue, and A. Sarrafzadeh, "A federated network online network traffics analysis engine for cybersecurity," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 2015, pp. 1–8.

[33] S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," *Journal of Network and Computer Applications*, vol. 62, pp. 9–17, Feb. 2016.

[34] B. Subba, S. Biswas, and S. Karmakar, "Intrusion detection systems using linear discriminant analysis and logistic regression," in *2015 Annual IEEE India Conference (INDICON)*, New Delhi, India, 2015, pp. 1–6.

[35] P. Di Lena and L. Margara, "Optimal global alignment of signals by maximization of Pearson Correlation," *Inf. Process. Lett.*, vol. 110, no. 16, pp. 679–686, Jul. 2010.

[36] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The Mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, Jan. 2000.

[37] A. Habibi Lashkari, G. Draper Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor traffic using time based features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, Porto, Portugal, 2017, pp. 253–262.

[38] L. Dhanabal and D. S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446-452, 2015.

[39] M. Hopkins, E. Reeber, G. Forman and J. Suermondt. (1999). *UCI Machine Learning Repository: Spambase Data Set* [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase. [Accessed: 20 Aug 2018].

[40] R. M. A. Mohammad and L. McCluskey. (2015). *UCI Machine Learning Repository: Phishing Websites Data Set* [Online]. Available: https://archive.ics.uci.edu/ml/datasets/phishing+websites. [Accessed: 20 Aug 2018].

[41] K. Demertzis and L. Iliadis, "SAME: An intelligent anti-malware extension for Android ART Virtual Machine," in *Computational Collective Intelligence*, 2015, pp. 235–245.

[42] H. F. Eid, A. E. Hassanien, T. Kim, and S. Banerjee, "Linear correlation-based feature selection for network intrusion detection model," in *International Conference on Security of Information and Communication Networks*, Berlin, Heidelberg, 2013, pp. 240–248.

[43] D.-J. Chang, A. H. Desoky, M. Ouyang, and E. C. Rouchka, “Compute pairwise manhattan distance and Pearson correlation coefficient of data points with GPU,” in *2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*, Daegu, Korea, 2009, pp. 501–506.

[44] H. Yao, X. Fu, Y. Yang, and O. Postolache, “An incremental local outlier detection method in the data stream,” *Applied Sciences*, vol. 8, no. 8, p. 1248, Jul. 2018.

[45] A. Aborujilah and S. Musa, “Cloud-Based DDoS HTTP attack detection using covariance matrix approach,” *Journal of Computer Networks and Communications*, vol. 2017, pp. 1-8, 2017