A Federated Network Online Network Traffics Analysis Engine for Cybersecurity

Shaoning Pang, Yiming Peng, Tao Ban, Daisuke Inoue, and Abdolhossein Sarrafzadeh

***Abstract*— Agent-oriented techniques are being increasingly used in a range of networking security applications. In this paper, we introduce FNTAE, a Federated Network Traffic Analysis Engine for real-time network intrusion detection. In FNTAE, each analysis engine is powered with an incremental learning agent, for capturing attack signatures in real-time, so that the abnormal traffics resulting from the new attacks are detected as soon as they occur. Owing to the effective knowledge sharing among multiple analysis engines, the integrated engine is theoretically guaranteed performing more effective than a centralized analysis system. We deployed and tested FNTAE in a real world network environment. The results demonstrate that FNTAE is a promising solution to improving system security through the identification of malicious network traffic.**

I. Introduction

ACCORDING to security threat report for 2013 and 2014 [1], [2], malicious codes, malware and related cyber security threats have grown and matured rapidly, and cyber criminals have started to impose online marketing as a way to promote and sell their services on the black market. It obviously shows that security problems are more complex than what we used to think of, and a formidable challenge of how to deal with these issues stands in front of us. In the field of intrusion detection, challenges are addressed by researchers as well as industry companies in their annual technical reports [3], [2], [4]. This includes,

1) *Real-time Data Analytics*: Deficiency of abilities to conduct real-time compliance analysis;
2) *Ever-changing Cyber Attacks*:The techniques advances in every single minute, so does the means of cyber attacks;
3) *Self Security*: As one of countermeasures to address those security issues, the security software/platform will be fain to be exposed to attacks. Thus, privacy and self-security for the platform must be treated as the first task on the agenda.

Collaborative Intrusion Detection (CID) emerges as a promising solution by using information from multiple sources to gain a better understanding of objective and impact of complex Internet attacks. CID utilizes multiple agents with each agent specializing on one regional threats analysis. Driven by the types of, the sources of, and the targets of future attacks, analysis agents with varied specialties will perform their own traffic analysis. As the failure of one agent will not seriously reduce the overall robustness of the integrated system, most of security incidents can be effectively detected. Fig. 1 gives an example of federated network traffic analysis. The federation is jointly established in a decentralized manner by 4 element networks that collaborate and share a global pool of information.

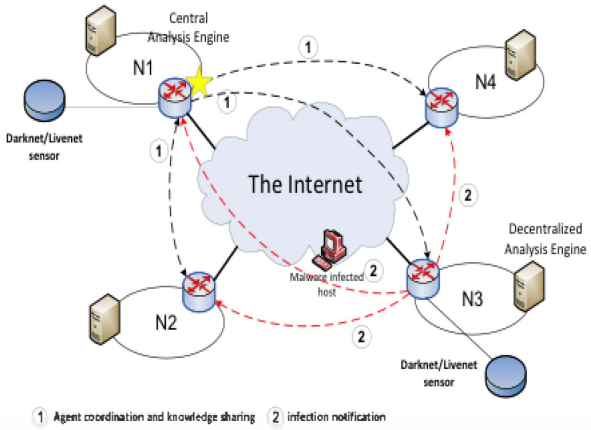

Fig. 1: An example of federated network traffic analysis, the central analysis engine N1 works cooperatively with the remaining 3 decentralized units

The benefits of CID include the scalability of solutions as well as robustness and availability, e.g. by means of the absence of the Single-Point-of-Failure. The second main advantage is teamwork: known from organizational studies, the team represents nowadays the atomic unit to solve complex problems. Tasks can be shared, e.g. load balancing, and can be solved by directing them to the most capable member. This approach can also compensate the shortcomings of individuals, e.g. an agent capable of misuse detection collaborates with an agent capable of anomaly detection. In addition, coordinated decision, e.g. voting or a joint detection status, as well as coordinated response for fast containment of malicious activity is enabled. Third, the Bigger Picture is realized by collaborating monitors. This allows the awareness for distributed attacks.

There have been many works in literature discussing and incorporating collaborative aspects in building security platforms [5], [6], [7]. These models are successfully applied however mostly limited to such a collaborative platform for solving one or another existing problems, e.g. single-point-

Shaoning Pang, Yiming Peng and Abdolhossein Sarrafzadeh are with the department of computing, Unitec Institute of Technology, New Zealand (email: ppang@unitec.ac.nz, ypeng@unitec.ac.nz & hsarrafzadeh@unitec.ac.nz.)

Tao Ban and Daisuke Inoue are with Cybersecurity Laboratory, National Institute of Information and Communications Technology (NICT), Japan (email: tao@nict.go.jp & daisuke@nict.go.jp)

978-1-4799-1959-8/15/$31.00 ©2015 IEEE

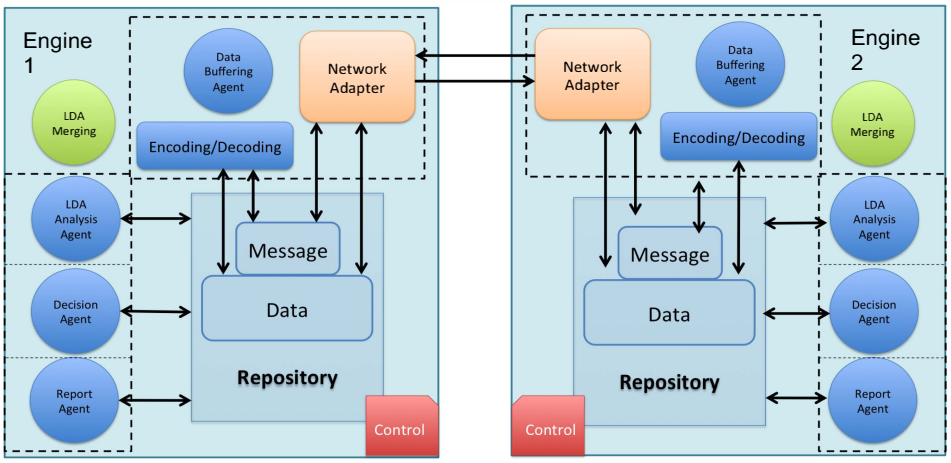

Fig. 2: The overall design of FNTAE

of-failure or Cyber attack changes. In this work, we consider FNTAE, a Federated Network Traffic Analysis Engine, for malware attacks analysis using software agents armed with swarm intelligence, and constitute the indispensable and essential parts of global analysis system that can be seamlessly integrated under a unified multi-agent paradigm. FNTAE turns the advantage of decentralized analysis into reality, having each of its decentralized analysis engines with strong capability of real-time data analytics, knowledge and information sharing, as well as self-privacy protection.

II. Related Works

In literature, there have been many works dis- cussing and incorporating collaborative aspects in building security platforms. In 2005, Benattou [5] built a dynamic, hierarchical, multiagent-based DIDS for detecting distributed attacks in communication networks. Chun-Hsin et al. [6] constructed in 2009 a collaborative network security platform in P2P networks, which has a efficient collaborative infrastructure due to the involvement of high speed p2p networks, and which also owns a high defensible services towards the specified type of attack - "TCP SYN flooding". Later, Xinming proposed another collaborative security platform named NetSecu [7]; and Yichi Zhang et. al. developed a distributed intrusion detection system in a multi-layer network architecture of smart grids. Also, there are many other similar distributed security systems using different technologies, like RFID. These models are successfully applied but limited to a collaborative platform for solving one or another existing problems, e.g. single-point-of-failure or scalability.

Further, when referring to real-time analysis, a huge shortage stands in front of most security experts whilst using those systems. In addressing this shortage issue, expert systems have been widely applied, yet they are usually expensive and time consuming due to the engagement of abundant human resources, also this manual task is slow and prone to errors [3]. In practice, a straightforward solution to real-time analytics is to integrate tentatively real-time analysis modules into existing security systems. However, because most of these systems are utilizing centralized solutions which rely mostly on the analysis of inbound network connections [8], it is always difficult to construe the extremely tremendous amount of information in a short period using a centralized mechanism [3], [8]. As seen, a demand of real-time analysis with capability of processing big data becomes more and more desperate.

Besides the insufficiency of distributed real-time analytics, as mentioned above, self-security, especially the information security, is the first thing of first whilst constructing a security platform. According to the definition of NIST (National Institute of Standards and Technology) [9] and the CIA triad [10], an information security model can be defined as confidentiality, integrity, and availability [11]. Here, information means all types of data, while referring to privacy, it usually means to prevent the personal information from being circulated to others [12]. This aspect usually is well-handled by most security oriented products by applying diverse mechanisms.

III. System Design

FANTAE is a decentralized analyzer that has a structure of two major components: multiple intelligent agents each of which specializes on the network traffic of a particular region, and a knowledge share mechanism which aggregates the condensed information offered by the intelligent agents for decision making. Aside of the research on the above learning schemes, another essential component of the system is a mechanism that support cost-effective and privacy-preserving communication between the agents. Fig. 2 gives the block diagram of FNTAE.

FNTAE has six main modules, the communication module, the analysis agent, the decision agent, the report agent, the knowledge sharing agent, and a control panel. The communication module is the communication interface to the network, responsible for acquiring packets from the network, which will be detailed in section IV-A. The analysis module performs an online linear discriminant analysis (LDA) on the incoming packet data for attack pattern analysis. The decision

module is responsible for judging and classifying attacks. The reporting module generates reports, logs, and update the 3D visualization. The knowledge sharing module enables analysis agents conducting extensive knowledge sharing on intrusion detection. The control panel allows users to control user accounts, and change accessibility options.

The decision agent starts working by requesting the feature data from its complementary LDA analysis agent. The decision agent then will call online classifier to judge if current acquired data is normal or involving a certain type of attack. In our system, a simple K Nearest Neighbor (K-NN) classifier is used for the decision making of intrusion detection. By the report agent, the result of classification is displayed as an instant message on the screen and a message to the central analysis engine.

An analysis engine incorporates the communication module, the LDA agent, the decision agent, and the report agent, so that the engine is able to independently conduct data acquiring, attack pattern analysis, decision making, results reporting as well as user account and system control. Every analysis engine has an LDA merging agent, but it performs normally inactive until integration is called in a way that a user logs in one of workstations as the root administrator and set on the signal of integration, then the analysis engine on that workstation becomes the central engine to rest of engines or workstations with an engine installed, and all analysis engines works as one integrated engine.

The integration of multiple analysis engines builds the collective intelligence of realtime attack pattern discovery and tracking by a Multi-agent framework described below.

IV. Multi-agent Framework

A. Communication Module

Network communication is the groundwork of a multi-agent system (MAS), which enables agents to interact and communicate with each other. The approach that we take for multi-agent communication is to allow one channel of communication between agents and to constrain communication to one language. We build FNTAE communication layer in Java Agent DEvelopment Framework (JADE) [13] in supporting agents for knowledge sharing as well as data transferring.

JADE is a p2p-based architecture, thus we simply address the transport layer, which is the Transmission Control Protocol (TCP) and the Internet Protocol suite (IP). As the most common core part of the IP, the TCP is often called TCP/IP. It is designed to provide reliable, ordered and error-checked delivery of octets stream between computers with connection to a local network, intranet or public Internet [14], [15], [16], [17]. A complete TCP operation consists of normally three stages: 1) connections establishment: relies on a multi-step handshake process; 2) data transmission: transfers information in bit streams; and 3) connection Termination: closes established virtual circuits and releases all allocated resources [14], [15], [16], [17].Connections must be properly established in a multi-step handshake process (connection establishment) before entering the data transfer phase. After data transmission is completed, the connection termination closes established virtual circuits and releases all allocated resources.

In FNTAE, we build a hybrid communication protocol on top of the above traditional TCP/IP communication protocol as Fig. 3. As seen from the figure, the revised TCP/IP

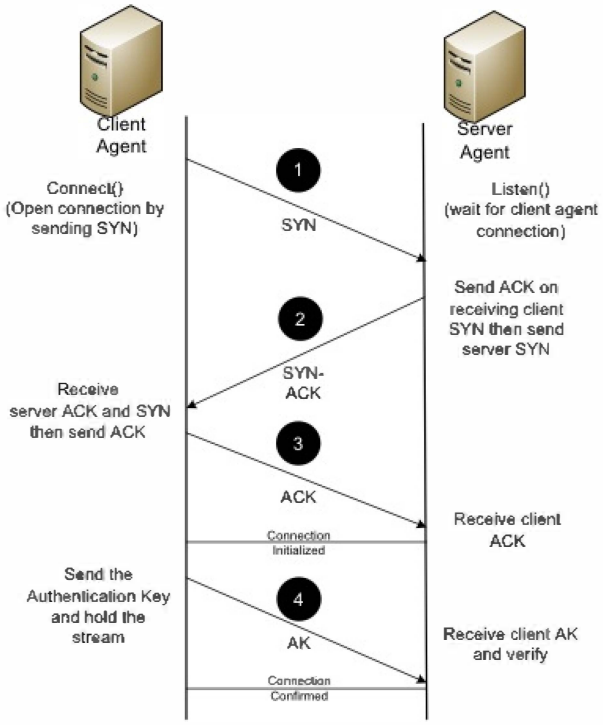

Fig. 3: The modified TCP/IP communication protocol

protocol consists of four steps to ensure an safe connection being established. The first three steps follows the traditional TCP three ways handshake for an initialization of a TCP connection. To ensure the safety of the connection, we define an embedded key for every stream being transferred. To confirm the connection, we let the client agent send the embedded key in first packet, and the server agent take the key and verify it with local key. If the key is authenticated by the server agent, then the connection is confirmed safe, and remains open for the transmission of the rest packets; otherwise, the server agent send a reject response to the client agent to terminate the TCP/IP connection.

B. LDA Agent

Consider the pattern of attack changes dynamically. An analysis engine thus is required to learn from incoming traffics the new signatures in real-time, so that the abnormal traffic resulting from the new attacks can be detected as soon as they occur.

In this purpose, we build an intelligent agent update its existing model at time t, $\Gamma(t)$, by incorporating the knowledge from newly presented a chunk of instances $\boldsymbol{Z}$

in class q. We can write the updating rule of the updating model formally as,

$$\Gamma(t+1) = \mathcal{F}(\Gamma(t), \boldsymbol{Z}, q). \tag{1}$$

where $\mathcal{F}$ is an incremental learning function.

In FNTAE, we use simply a Fisher type of linear discriminant analysis (LDA) for intrusion pattern distinction. Here, feature extraction method LDA is used instead of feature selection methods [18], this is because that we intend to build an online classifier by an action of separation, which is to force data points from normal traffic to one side, and points from abnormal traffic to another side in the feature space. Thus, $\mathcal{F}$ is implemented as Pang's incremental Fisher LDA (FIncLDA) [19], which has desirable one-pass and chunk data processing property.

Given a chunk of network traffic data $\boldsymbol{Z} = \{\boldsymbol{z}_1, .., \boldsymbol{z}_l\}$ are acquired at each time point t, where l is the number of samples in one chunk, which is a random positive integer and $l \geq 1$. an existing Fisher LDA (FLDA) $\boldsymbol{\Omega} = \{\boldsymbol{S}_w, \boldsymbol{S}_b, \bar{\boldsymbol{x}}\}$ on presented data $\boldsymbol{X}$. Here, $\bar{\boldsymbol{x}}$ is the general mean vector; $\boldsymbol{S}_w$ and $\boldsymbol{S}_b$ are the within and between class scatter matrix, respectively. The function of FIncLDA is to compute $\Phi = \{\boldsymbol{S'}_w, \boldsymbol{S'}_b, \bar{\boldsymbol{x}}'\}$ using $\boldsymbol{\Omega}$ and $\{\boldsymbol{Z}, q\}$.

When $q \in [1, K]$, we assume that l_k of l new samples all belong to class $k, k \in [1, K]$, thus $N'_k = N_k + l$, $N + l = \sum_{k=1}^{K} N'_k = \sum_{k=1}^{K} (N_k + L_k)$ and $\bar{\boldsymbol{x}}'_k = \frac{1}{N_k + l_k}(N_k \bar{\boldsymbol{x}}_k + L_k \bar{\boldsymbol{z}}_k)$, where $\bar{\boldsymbol{z}}_i$ is the mean of new samples in class i.

The updated mean is:

$$\bar{\boldsymbol{x}}' = (N\bar{\boldsymbol{x}} + L\bar{\boldsymbol{z}})/(N + l) \tag{2}$$

where $\bar{\boldsymbol{z}} = \frac{1}{l}\sum_{i=1}^{L} \boldsymbol{z}_i$

The updated $\boldsymbol{S}_b$ matrix is:

$$\boldsymbol{S'}_b = \sum_{k=1}^{K} N'_k (\bar{\boldsymbol{x}}'_k - \bar{\boldsymbol{x}}')(\bar{\boldsymbol{x}}'_k - \bar{\boldsymbol{x}}')^T \tag{3}$$

The updated $\boldsymbol{S}_w$ matrix is,

$$\boldsymbol{S}'_w = \sum_{k=1}^{K} \boldsymbol{\Sigma}'_k \tag{4}$$

$$\begin{aligned} \boldsymbol{\Sigma}'_k = \quad & \Sigma_k + \tfrac{N_k L_k^2}{(N_i + L_k)^2}(\boldsymbol{D}_k) + \tfrac{N_k^2}{(N_k + l_k)^2}(\boldsymbol{E}_k) \\ & + \tfrac{L_k(L_c + 2N_k)}{(N_k + L_k)^2}(\boldsymbol{F}_k) \end{aligned} \tag{5}$$

where the second term $\boldsymbol{D_k}$ is the scatter matrix of the new sample class mean vector $\bar{\boldsymbol{z}}_k$ around the mean vector $\bar{\boldsymbol{x}}_k$ of class k

$$\boldsymbol{D_k} = (\bar{\boldsymbol{z}}_k - \bar{\boldsymbol{x}}_k)(\bar{\boldsymbol{z}}_k - \bar{\boldsymbol{x}}_k)^T. \tag{6}$$

The third term $\boldsymbol{E_k}$ is the scatter matrix of the new sample mean vector $\bar{\boldsymbol{z}}_k$ around the mean vector $\bar{\boldsymbol{x}}_k$ of class k.

$$\boldsymbol{E_k} = \sum_{i=1}^{L_k} (\boldsymbol{z}_{ki} - \bar{\boldsymbol{x}}_k)(\boldsymbol{z}_{ki} - \bar{\boldsymbol{x}}_k)^T. \tag{7}$$

The fourth term $\boldsymbol{F_i}$ is a within-class scatter matrix of the new samples,

$$\boldsymbol{F_k} = \sum_{i=1}^{L_k} (\bar{\boldsymbol{z}}_{ki} - \bar{\boldsymbol{z}}_k)(\bar{\boldsymbol{z}}_{ki} - \bar{\boldsymbol{z}}_k)^T. \tag{8}$$

When $q > K$, we can also assume L_{K+1} of L new samples belong to class $K + 1$ without loss of generality. In this case, the updated between-class matrix (3) can be rewritten as

$$\begin{aligned} \boldsymbol{S'}_b &= \textstyle\sum_{k=1}^{K} N'_k (\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}')(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}')^T \\ &+ L_{K+1}(\bar{\boldsymbol{z}} - \bar{\boldsymbol{x}}')(\bar{\boldsymbol{z}} - \bar{\boldsymbol{x}}')^T \\ &= \textstyle\sum_{k=1}^{K+1} N'_k (\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}')(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}')^T \end{aligned} \tag{9}$$

where N'_k is the number of samples in class k after $\boldsymbol{Z}$ is presented. If $q = K + 1$, then $N'_q = L_{K+1}$, else $N'_q = N_q + L_q$.

Also, the updated within-class matrix (4) can be rewritten as

$$\boldsymbol{S'}_w = \sum_{k=1}^{K} \boldsymbol{\Sigma}_k + \boldsymbol{\Sigma}_q = \sum_{k=1}^{K+1} \boldsymbol{\Sigma}'_k \tag{10}$$

where $\boldsymbol{\Sigma}_q = \sum_{y \in \{y_\bullet\}} (\boldsymbol{z} - \bar{\boldsymbol{z}}_q)(\boldsymbol{z} - \bar{\boldsymbol{z}}_q)^T$.

C. LDA Merging Agent

An individual analysis engine described above performs independent network traffic analysis and decision making by a KNN classifier. The integration of multiple analysis engines aims to form a collective intelligence of multiple analysis engines for network intrusion detection. In this case, learning of attack pattern is done over multiple learners each of which accesses data being acquired locally.

To obtain the global model of attack pattern, all data are required to be shared among all individual analysis engines. However, this causes often huge communication and computation costs especially when the number of analysis engines is large. In FNTAE, we combine multiple models learned/trained locally to capture the complete discriminant information of attack pattern. This paradigm is deterministically efficient under the following conditions:

1) As long as the individual learning models capture all the essential information from the data, sharing the knowledgeable models among learners is equivalent to sharing the samples in the sense of information gain;
2) There are cases that transmission of learning models among learners are more efficient than transmission of samples, since the knowledge can be represented with greater density – take the PCA representation of the dataset as an example;
3) In the paradigm of distributed learning, due to privacy or technical reasons, sharing samples among individual learners is often considered sensitive or impossible. In such a situation, knowledge sharing is only available in the form of statistical models interchanging.

In the following, we discuss how to adapt the FIncLDA to a learning model which supports easy merging operations between multiple learning models, without making explicit

use of the vectorial representation of the samples. Mathematically, we define the operation of LDA merging agent as

$$\mathbf{\Omega}'_x = \mathbf{\Omega}_x \oplus \mathbf{\Omega}_z \tag{11}$$

where $\mathbf{\Omega}_x$ and $\mathbf{\Omega}_z$ are independent models learned from datasets $(\boldsymbol{X}, \boldsymbol{Y})$ and $(\boldsymbol{Z}, \boldsymbol{Q})$, respectively.

1) LDA Merging: Following the same denotation as in FIncLDA, we here discuss the merging operation between two independent FLDA models $\mathbf{\Omega}_x$ and $\mathbf{\Omega}_z$. Merging between multiple models is realized by sequential binary operations.

According to [19], the models that represent the extracted knowledge from the dataset can be noted as the following 3-tuples,

$$\mathbf{\Omega}_x = (\{\mathbf{\Sigma}_k\}, \{\bar{\boldsymbol{x}}_k\}, \{N_k\}), \tag{12}$$

$$\mathbf{\Omega}_z = (\{\boldsymbol{E}_k\}, \{\bar{\boldsymbol{z}}_k\}, \{L_k\}), \tag{13}$$

where $\mathbf{\Sigma}_k$ and $\boldsymbol{E}_k$ are the class covariance matrices, $\bar{\boldsymbol{x}}_k$ and $\bar{\boldsymbol{z}}_k$ the class mean vectors, and N_j and L_j the class cardinal numbers of the two datasets, respectively.

Pang et al. [20] developed LDA merging model, which can be summarized as follows: Let $\mathbf{\Omega}'_x = (\{\mathbf{\Sigma}'_k\}, \{\bar{\boldsymbol{x}}'\}, \{N_k + L_k\})$, then the updated all-instance mean vector $\bar{\boldsymbol{x}}'$, class mean vector $\bar{\boldsymbol{x}}'_k$, within-class scatter matrix $\boldsymbol{S}'_w$, and between-class scatter matrix $\boldsymbol{S}'_b$, can be calculated as:

$$\begin{aligned}
\bar{\boldsymbol{x}}' &= (N\bar{\boldsymbol{x}} + L\bar{\boldsymbol{z}})/(N+l), \\
\bar{\boldsymbol{x}}'_k &= (N_k\bar{\boldsymbol{x}}_k + L_k\bar{\boldsymbol{z}}_k)/(N_k+L_k), \quad \text{for } k = 1, \ldots, K, \\
\boldsymbol{S}'_b &= \sum\nolimits_k (N_j + L_j)(\bar{\boldsymbol{x}}'_k - \bar{\boldsymbol{x}}')(\bar{\boldsymbol{x}}'_k - \bar{\boldsymbol{x}}')^T, \\
\boldsymbol{S}'_w &= \sum\nolimits_k \mathbf{\Sigma}'_k.
\end{aligned} \tag{14}$$

To prevent the explicit use of the vectorial forms of samples to update $\boldsymbol{S_w}$, we have

$$\mathbf{\Sigma}'_k = \mathbf{\Sigma}_k + \boldsymbol{E}_k + \frac{N_k L_k}{N_k + L_j}(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{z}}_k)(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{z}}_k)^T. \tag{15}$$

Proof: For class ω_j in the merged class set $\boldsymbol{X}'$, the class covariance matrix is,

Following (14), we have

$$\begin{aligned}
\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}'_k &= \tfrac{L_k}{N_k+L_k}(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{z}}_k), \\
\bar{\boldsymbol{z}}_j - \bar{\boldsymbol{x}}'_k &= \tfrac{N_k}{N_k+L_k}(\bar{\boldsymbol{z}}_k - \bar{\boldsymbol{x}}_k).
\end{aligned} \tag{16}$$

Substituting (16) into (16) gives

$$\begin{aligned}
\mathbf{\Sigma}'_k &= \mathbf{\Sigma}_k + N_k L_k^2 \frac{(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{z}}_k)(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{z}}_k)^T}{(N_k+L_k)^2} + \boldsymbol{E}_k \quad (17) \\
&\quad + N_k^2 L_k \frac{(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{z}}_k)(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{z}}_k)^T}{(N_k+L_k)^2} \\
&= \mathbf{\Sigma}_k + \boldsymbol{E}_k + \frac{N_k L_k}{N_k + L_k}(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{z}}_k)(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{z}}_k)^T \quad (18)
\end{aligned}$$

□

Note that in (14), only the two independent models in (12) and (13) are employed to update the learning model. Thus, we can summarize the updating rule of *FLDA merging* as Given $\mathbf{\Omega}_x$ and $\mathbf{\Omega}_z$ are two independent FLDA models learned respectively from datasets $(\boldsymbol{X}, \boldsymbol{Y})$ and $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}})$, then Γ_z can be appended to $\mathbf{\Omega}_x$ without loss discriminative information as,

$$\mathbf{\Omega}_x(t+1) = \mathcal{F}^+(\mathbf{\Omega}_x(t), \mathbf{\Omega}_z) = \mathcal{F}^+(\mathbf{\Omega}_x(t), \{\boldsymbol{E}_k\}, \{\bar{\boldsymbol{z}}_k\}, \{L_k\}). \tag{19}$$

In another word, (19) indicates that an integrated learning model can be obtained from two (or more) existing models without access to the underlying data.

2) Condensed Knowledge-Sharing: The above FLDA merging makes use of the class covariance matrices to share knowledge among multiple models. However, when the dimension of the input patterns is greater than the number of samples in the subset, the covariance matrix does not support condensed knowledge representation.

It is interesting to note that the class covariance matrix $\mathbf{\Sigma}$ is Hermitian, and can be uniquely decomposed as $\mathbf{\Sigma} = \boldsymbol{V}\mathbf{\Lambda}\boldsymbol{V}^T$, where $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}$, and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues, λ_i. And therefore

$$\mathbf{\Sigma} = \Sigma_i \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T, \tag{20}$$

where $\boldsymbol{v}_i$ are the eigenvalues associated with λ_i. Thus by sharing the most significant eigenvalues and the associated eigenvectors of the class covariance matrices, we can enable condensed knowledge-sharing among multiple learners.

When a condensed eigenspace model is acquired, an efficient approach to updating the acceptor's existing FLDA model is to make use of the incremental eigenanalysis method in an iterative mode. Incremental eigenanalysis is essential for high dimensional data where retraining the eigenspace model is computationally expensive. Thus to make the performance evaluation clearer, we use just principle component analysis (PCA) for data dimension reduction, and stick to batch FLDA for computing the eigen-space models and refrain from other advanced algorithms that could possibly bring non-deterministic factors due to different implementations.

V. System Implementation and Deployment

We implemented FNTAE at our research lab in building 183 of Mt. Albert campus, Unitec Institute of Technology New Zealand. We deployed six workstations for experiments, even though FNTAE support in principle a big number of workstations to the maximum communication band and capacity of our campus network. Table I shows the hardware and software configuration profile of each workstation used in our FNTAE implementation.

Fig. 4 presents the network topology of the FNTAE implementation. As seen, the system is designed supporting a two-switchable-mode architecture, which enables FNTAE three running mode: 1) decentralized mode only, 2) centralized mode, and 3) two modes coexistence.

Among the six workstations, we use one workstation as the data distributor distributing data to each individual analysis engine. A principal component analysis is conducted by the distributor before dispatching data, because Fisher

$$
\begin{aligned}
\mathbf{\Sigma}'_k &= \sum_{\boldsymbol{x}'_i \in \omega_k} (\boldsymbol{x}'_i - \bar{\boldsymbol{x}}'_k)(\boldsymbol{x}'_i - \bar{\boldsymbol{x}}'_k)^T = \sum_{\boldsymbol{x}_i \in \omega_k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}'_k)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}'_k)^T + \sum_{\boldsymbol{z}_i \in \omega_k} (\boldsymbol{z}_i - \bar{\boldsymbol{x}}'_k)(\boldsymbol{z}_i - \bar{\boldsymbol{x}}'_k)^T \\
&= \sum_{\boldsymbol{x}_i \in \omega_k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k + \bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}'_k)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k + \bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}'_k)^T + \sum_{\boldsymbol{z}_i \in \omega_k} (\boldsymbol{z}_i - \bar{\boldsymbol{z}}_k + \bar{\boldsymbol{z}}_k - \bar{\boldsymbol{x}}'_k)(\boldsymbol{z}_i - \bar{\boldsymbol{z}}_k + \bar{\boldsymbol{z}}_k - \bar{\boldsymbol{x}}'_k)^T \\
&= \sum_{\boldsymbol{x}_i \in \omega_k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)^T + N_k(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}'_k)(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}'_k)^T + \sum_{\boldsymbol{z}_i \in \omega_k} (\boldsymbol{z}_i - \bar{\boldsymbol{z}}_k)(\boldsymbol{z}_i - \bar{\boldsymbol{z}}_k)^T + L_k(\bar{\boldsymbol{z}}_k - \bar{\boldsymbol{x}}'_k)(\bar{\boldsymbol{z}}_k - \bar{\boldsymbol{x}}'_k)^T
\end{aligned}
$$

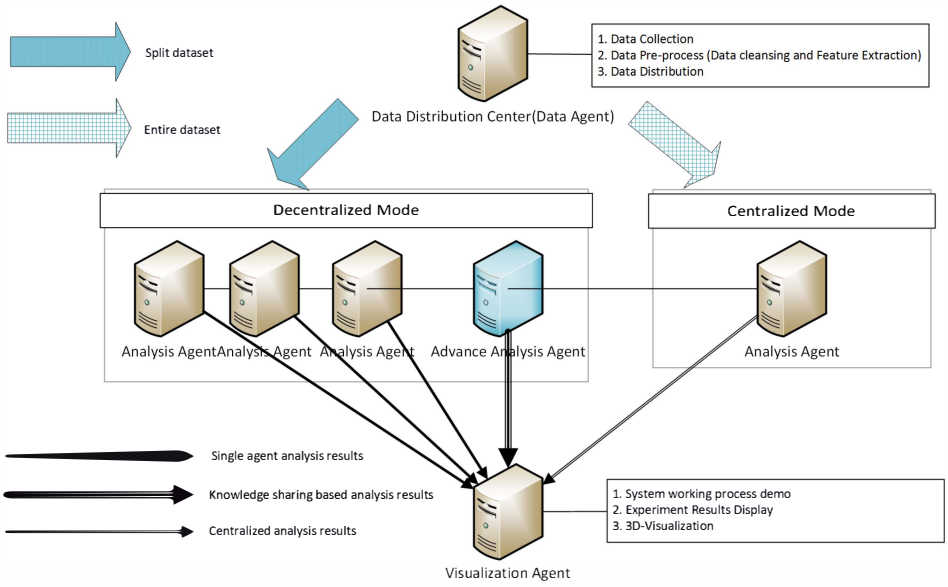

Fig. 4: FNTAE network topology

HardwareSoftware	Description
CPU	Intel(R) Core(TM) i7-2600 CPU @ 3.40Ghz
Memory	8 GB
Hard drive	500GB
Video Adapter	Nvidia Quadro 400/PCIe 512MB
Network Adapter	Intel(R) PRO/1000 Network connection
Operating System	CentOS 5.8 (Linux v2.6.18-308.el5PAE) 32bit
JRE Version	Java(TM) SE Runtime Environment
JVM Heap	512MB - 2000MB

TABLE I: Hardware & Software configuration profiles

LDA is vulnerable to the singularity difficulty (when eigen-decomposition is conducted on a sparse matrix, the obtained eigen-matrix and eigenvalue is meaningless). Here, PCA works as a pre-processing step that removes the singularity problem, also speeds up the system by reducing significantly the number of data sample dimensions.

Another workstation is used as the central analysis engine, named advance analysis agent in Fig. 4. Whenever the global signal of "integration" is set on by the administrator, all other analysis engines are required to start reporting regularly its status and results including local attack pattern and intrusion detection accuracies to the central engine. The central engine will have LDA merging agent to summarize the collected data/knowledge, and call report agent for results visualization on its screen. The remaining three workstations are used as distributed analysis engines accomplishing tasks of attack discrimination analysis, intrusion classification, and result reporting. Note that to reduce the burden of central analysis engine, we normally use a separate workstation for the full operation of visualization.

Fig. 5 shows the FNTAE 3D interface. As seen, the 3D visualization follows the physical architecture of the whole building, as well as the status of desk deployment in each room of the building. This enables us to trace the threats into each room of the building. For example, whenever an attack is detected by FNTAE, and through the 3D interface we will be able to find quickly the physical machine causes the attack and the room where the machine is installed. For networking part, the system describes every networked desktop computers in staff office, research lab as well as lecture room. The status of each computer is in trace, especially when this computer is used as a FNTAE workstation, the network traffics are tracked from raw data transferring, knowledge sharing in-between stations, to control panel message delivery. The blue streaming symbol represents normal network traffics, and red color means attack detected, yellow means knowledge sharing in-between stations occurring.

VI. Network Traffic Analysis Experiments

A. Data

For the purpose of testifying the concepts of our system, we simply employ the benchmark KDD99 data in security fields for experimental attempts. The dataset was used for The Third International Knowledge Discovery and Data Mining Tools Competition in conjunction with The Fifth

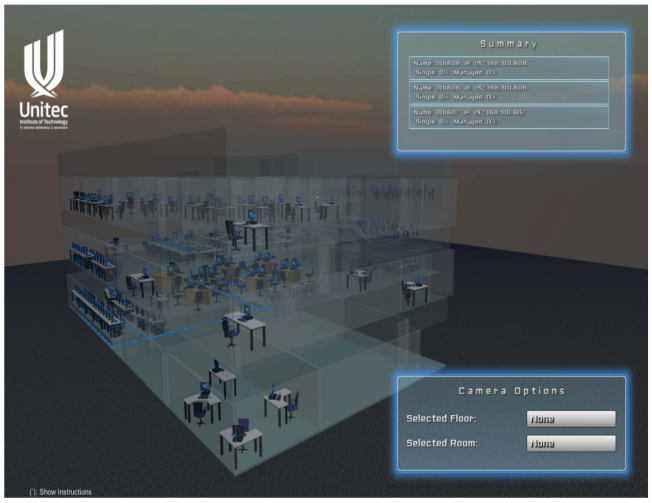

Fig. 5: FNTAE 3D visual reality interface

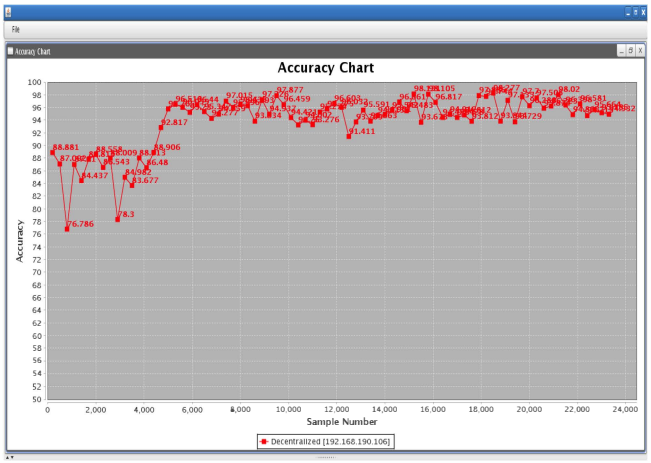

Fig. 6: Centralized Network Analysis Engine Performance in terms of Classification Accuracy

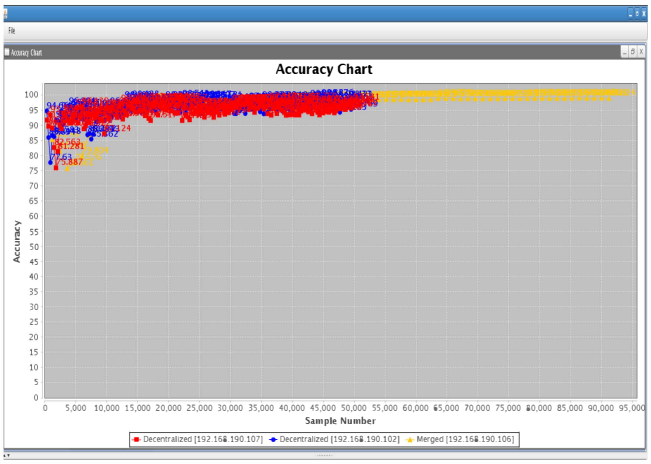

Fig. 7: Integrated Engine Performance in terms of Classification Accuracy

International Conference on Knowledge Discovery and Data Mining (KDD99). The task is to construct a predictive model capable of distinguishing between "bad" connections, called intrusions or attacks, and "good" normal connections [21]. The database contains a standard set of data to be audited, which includes a wide variety of simulated intrusions. It has total 494020 samples with 233 features located in 23 categories. Each sample corresponds to either a normal TCP connection or an attack.

B. Experimental Setup

In our experiments, we split the entire dataset into two parts: 90% of whole dataset for training and the rest 10% for testing (i.e., 444618 training samples and 49402 testing samples).

To examine FNTAE on decentralized network traffic analysis in a comparison to that of centralized analysis for real time attack pattern discovery and tracking, we feed at one stage a chunk of training samples to a centralized analysis engine, meanwhile distribute the same amount of data equally to three decentralized engines. The attack pattern discovered by the joint efforts from three decentralized engines is compared to that from single centralized engine, in terms of the resulting intrusion detection accuracy (measured in FNTAE as K-NN Leave-One-Out (LOO) classification rate). We observe the relationship of intrusion detection accuracy against the number of samples presented so far.

C. Results

Fig. 6 presents FNTAE output on the variation of intrusion detection accuracy for centralized online network traffic analysis, in which one analysis engine is receiving all data presented so far. The X axis represents number of samples received so far, which increases with the time going on. The Y axis shows the accuracy in terms of the percentage of data that are correctly classified by the system. As seen from the figure, the accuracy increases in general with the number of data samples, but the convergence of system seems not so smooth, and slight performance drop happens at the final stage of the experiment. The variation of the accuracy is in the range of [76.786%, 98.198%].

In comparison, Fig. 7 shows the FNTAE output for decentralized mode network traffic analysis, in which three analysis engines are receiving an equal partition of the data presented so far. The blue and red curve represent the results from two individual analysis engine, and the yellow curve plot the central analysis performance. As seen in the figure, the red and blue plot covers mostly the yellow plot at the beginning, but the yellow plot grows out of the cover after the stage of 55,000, and occupies the screen never been dominated again by other two colors. This indicates that FNTAE has an overwhelming superiority to the traditional centralized network traffic analysis.

VII. Conclusion

CID have been discussed intensively to understand the underlying research problems that may be solved with existing approaches or require new solutions to confront the adversary. Yet it is not part of a real world intrusion detection system. The paper introduces FNTAE, a complete Java-based multi-agent integrated network traffic analysis engine. FNTAE, in a distributed networking environment, conducts real-time multi-agent cooperative network traffic analysis. Compared to existing centralized ID systems and CID prototypes, FNTAE is summarized with the merit of (1) real time network traffic analysis powered by incremental linear discriminant analysis system; (2) advanced attack pattern discovery owing to condensed knowledge share through LDA merging; and (3) the high self-security of agent to agent communication because of knowledge share among agents is terms of eigen-matrix, which avoids the transfer of low level data at high frequencies.

FNTAE was tested in a real-world local network environment, in which the complete KDD99 data is used for system tests real time attack pattern discovery and tracking. The results show that FNTAE presents an evolving discriminant analysis in a dynamic network environment by incorporating knowledge collected from multiple agent, and the system converges consistent. As future work, we will connect FNTAE with darknet and livenet traffic monitoring and analysis for real time intrusion detection experiments.

Acknowledgment The authors wish to thank National Institute of Information and Communications Technology Japan for funding the presented work through a commissioned research titled "Research and Development on Decentralized Analytical Methods for Network Traffics with Regional Information" to the Department of Computing, Unitec Institute of Technology, New Zealand.

References

[1] SOPHOS, "Security threat report 2013," SOPHOS, Report, 2012. [Online]. Available: http://www.sophos.com/en-us/medialibrary/PDFs/other/sophossecuritythreatreport2013.pdf

[2] ——, "Security threat report 2014," SOPHOS, Report, 2013. [Online]. Available: http://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-security-threat-report-2014.pdf

[3] TIBCO, "Tibco cyber security platform," TIBCO, Report, 2012. [Online]. Available: http://www.tibco.com/multimedia/wp-cyber-security-platform_tcm8-16777.pdf

[4] D. A. Fisher, J. M. McCune, and A. D. Andrews, "Trust and trusted computing platforms," DTIC Document, Report, 2011.

[5] B. M. and T. K., "Intelligent agents for distributed intrusion detection system," in *in Proc. World Academy Science, Engineering and Technology*, June, 2005, Conference Proceedings.

[6] W. Chun-Hsin and H. Chun-Wei, "A collaborative network security platform in p2p networks," in *New Trends in Information and Service Science, 2009. NISS '09. International Conference on*, 2009, Conference Proceedings, pp. 1251–1256.

[7] C. Xinming, M. Beipeng, and Z. Chen, "Netsecu: A collaborative network security platform for in-network security," in *Communications and Mobile Computing (CMC), 2011 Third International Conference on*, 2011, Conference Proceedings, pp. 59–64.

[8] O. Flauzac, F. Nolot, C. Rabat, and L. A. Steffenel, "Grid of security: A new approach of the network security," in *Network and System Security, 2009. NSS '09. Third International Conference on*, 2009, Conference Proceedings, pp. 67–72.

[9] G. Stoneburner, C. Hayden, and A. Feringa, "Engineering principles for information technology security (a baseline for achieving security)," DTIC Document, Report, 2001.

[10] C. Perrin, "The cia triad," *Dostopno na: http://www.techrepublic.com/blog/security/the-cia-triad/488*, 2008.

[11] M. Whitman and H. Mattord, *Principles of information security*. Cengage Learning, 2011.

[12] M. Prakash and G. Singaravel, "A new model for privacy preserving sensitive data mining," in *Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on*, 2012, Conference Proceedings, pp. 1–8.

[13] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE*. John Wiley & Sons, 2007, vol. 7.

[14] J. Postel, "Transmission control protocol," 1981.

[15] D. E. Comer, "Internetworking with tcp/ip: Principles, protocols and architecture, vol. 1," 2000.

[16] V. G. Cerf and R. E. Icahn, "A protocol for packet network intercommunication," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 2, pp. 71–82, 2005.

[17] R. Y. Zaghal and J. I. Khan, "Efsm/sdl modeling of the original tcp standard (rfc793) and the congestion control mechanism of tcp reno," *URL: http://www. medianet. kent. edu/technicalreports. html*, 2005.

[18] Z.-H. A. N. Kayacik H. G. and H. M. I., "Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets," in *Proceedings of the Third Annual Conference on Privacy, Security and Trust (PST-2005)*, October, 2005, Conference Proceedings.

[19] S. Pang, S. Ozawa, and N. Kasabov, "Incremental linear discriminant analysis for classification of data streams," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, no. 5, pp. 905–914, 2005.

[20] S. Pang, T. Ban, K. Youki, and K. N., "Lda merging and splitting with applications to multiagent cooperative learning and system alteration," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 552–564, 2012.

[21] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the jam project," in *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, vol. 2. IEEE, 2000, pp. 130–144.