IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

O-ADPI: Online Adaptive Deep-Packet Inspector Using Mahalanobis Distance Map for Web Service Attacks Classification

MOHSEN KAKAVAND[1], (Member, IEEE), AIDA MUSTAPHA[2], ZHIYUAN TAN[3], (Member, IEEE), SEPIDEH FOROOZANA[4] AND LINGGES ARULSAMY[5]

[1]School of Science and Technology, Sunway University Malaysia (e-mail: mohsenk@sunway.edu.my)
[2]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Malaysia (e-mail: aidam@uthm.edu.my)
[3]School of Computing, Edinburgh Napier University, Edinburgh, United Kingdom (e-mail: z.tan@napier.ac.uk)
[4]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia (e-mail: foroozan.sepideh@gmail.com)
[5]School of Science and Technology, Sunway University Malaysia (e-mail: linggest90@gmail.com)

Corresponding author: Mohsen Kakavand (e-mail: mohsenk@sunway.edu.my)

ABSTRACT Most active research in Host and Network Intrusion Detection Systems are only able to detect attacks on the computer systems and at the network layer, which are not sufficient to counteract SOAP/REST or XML/JSON-related attacks. In dealing with the problem of anomaly detection in web service message datasets, this paper proposes an anomaly detection system called the Online Adaptive Deep-Packet Inspector (O-ADPI) for web service message attacks classification. The proposed approach relies on multiple statistical methods which use Unigram-based Weighting Scheme (UWS) that combines text mining techniques with a set of different statistical criteria for Feature Selection Engine (FSE) to effectively and efficiently explore optimal subspaces in detecting anomalies embedded deep in the high dimensional feature subspaces. We utilize a supervised intrusion detection algorithm based on Mahalanobis Distance Map classifier. As web service attacks can be classified into anomaly and normal, the task of anomaly detection can be modeled as a classification problem. The O-ADPI model was assessed for F-value, true positive rate (TPR), and false positive rate (FPR) in order to evaluate the detection performance of O-ADPI against different type of feature selections engines with corresponding PCs for each message-specific service. The experiments were performed using the REST-IDS Dataset 2015 and the results demonstrated that the proposed O-ADPI model achieved the best results in each message-specific service.

INDEX TERMS Anomaly Detection, Feature Selection Engine, Mahalanobis Distance Map, Web Service Attacks.

I. INTRODUCTION

THE use of computer networks has now become imperative to organizations. This fact broadens the scope for network attackers and increases the damage that these attacks may cause. These security dangers are particularly genuine for Web Service (WS) applications, where delicate and a huge sum of business-related confidential information are being transfered through the internet. Although the degree of security risk among traditional network intrusions from the highest to lowest is known as U2R> R2U> DoS> Probes, another causation risk factor has been the complexity of modern web service platforms, whereby web services intrusions are now having new features of vulnerabilities that make them unique. As a result, a substantial fraction of cyber-attacks against organizations with Internet presence targets their web-based services [1].

Web services are independent platforms that are introduced to cater interoperable standards, provide application flexibility, integration and data communication between systems [2]. This allows more and more software applications to be built on internet-enabled WS platforms, particularly Business Intelligence (BI) and E-commerce application. However, applications use XML or JSON message to communicate with each other via the application layer using the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

Hypertext Transfer Protocol (HTTP) standard [3]. Therefore, the associated port for these protocols (i.e., port 80) needs to be accessible at all times to support functionality and services. The application layer is therefore open to different types of threats such as XML injection, SON injection, Structured Query Language (SQL), Simple Object Access Protocol (SOAP), oversized payload, and recursive payload resulting in XML Denial-of-Service (DoS) attack.

Despite the risk, unfortunately all standard security appliances such as network firewalls, content filters, or network intrusion detection/prevention systems are not able to block the intruders from attacking the web services [4]. This is because current IDSs are incapable of deducing payload attacks inside the XML and JSON messages at the service level, hence they are insufficient for detecting and preventing the concealed threats in apparent constant network traffics [5]. In addition, although the WS-Security on WS offers confidentiality to sensible data, XML encryption can obscure message information from being inspected so the encrypted information can still conceal attacks such as oversized payload, coercive parsing, or XML injection [6].

For example, in Thakar et al. [7], the requests for WSs were simulated on honey-pots and a Support Vector Machine (SVM) classifier but they were only able to intercept SOAP requests by identifying few attacks such as the SQL Injection and XML DoS attacks. In another study, Chan et al., [8] focused only on XML-related attacks, where the authors derived fuzzy association rule models for segregating known XML data-based attack patterns, but only covered SQL injections, buffer overflows, and SOAP oversized payloads.

In effort to detect more attacks at the service layer, this paper proposes an Online Adaptive Deep-Packet Inspector (O-ADPI) model that is specifically targeted for web service message attacks in order to effectively counter tag injection attacks, XML injection, JSON injection, XML Denial-of-Service (DoS) attacks as well as HTTP Parameter Pollution (HPP) attacks. The goal of the proposed WS-IDS is to complement existing ID/IP systems, which are mainly network-based and do not address the SOAP/REST and XML/JSON-related payload attacks. The methodology utilizes multiple statistical methods which include Unigram-based Weighting Scheme (UWS) that combines text mining techniques and a supervised intrusion detection algorithm based on Mahalanobis Distance Map classifier for web service layer security. In summary, the main contributions of the research are:

- O-ADPI can handle high-dimensional information in SOAP/REST web services base on XML/JSON data payload, where most of the approaches that exist unable to. Moreover, most of the existing approaches can only handle HTTP payload as low application level.
- O-ADPI uses a Feature Selection Engine (FSE) that is able to uncover feature subspaces effectively and efficiently to detect anomalies embedded in subspaces where most of the high-dimensional information anomalies occur.
- O-ADPI is an online system for detecting anomalies that is computationally efficient to operate in real time, in environments with large bandwidth.

The rest of this paper is organized as follows; Section 2 reviews the related works in Host and Network Intrusion Detection (ID) and Intrusion Prevention (IP) systems. Section 3 describes sample attacks of web services, mainly focused on different types of REST-based web service attacks. Section 4 presents the proposed Online Adaptive Deep-Packet Inspector (O-ADPI) model along with the anomaly detection techniques. Section 5 presents the experimental design, evaluation and results respectively with the simulation of REST-IDS dataset 2015 including the computational and speed scalability tests. Finally, Section 6 concludes and discusses future work.

II. RELATED WORK

In general, there are two major detection methods among the Intrusion Detection Systems (IDSs), which are the signature-based and anomaly-based [9] and [10]. The signature-based intrusion detection (SID) includes a database of defined signatures for matching strings against the attacks. Hackers would then craft attack variants to beat the signature strings or enhance the attacks to exploit new vulnerabilities. Whenever SID falls short, anomaly-based intrusion detection (AID) attempts to close the holes. AID is a newer approach as compared to SID in the fight against misuse and exploits. However, AID is not a cure-all. When it is used along with an influential SID solution, it becomes an influential tool for network protection.

Reviews on related works in IDSs within the context of web service security as well as a branch of network security, along with data mining/machine learning and statistical models concerning the location of attacks, IDSs and common protocols, the works can be generally categorized into two categories; IDSs for (1) web applications and (2) web services. However, since both categories contain overlaps and high similarities, understanding their differences is crucial in recognizing the importance of intrusion detection in web services.

Within the first category of IDSs for web applications, most research focuses on monitoring the HTTP packets and methods to infer the web-based attacks over payload of network traffic [11], [12], [13], [14] and [15]. The payload request and response are part of web application behaviors, hence there are relationships between its features. The research deeply inspects web applications to comprehend a web application's normality and abnormality by its critical points. The abstraction emerges from web servers and the underlying protocols in this category as well.

Works throughout this category include the use of a chunk as a small payload to detect content-based threats on application-level protocols, such as HTTP and FTP [16]. In another study, Juvonen et al. [17] proposed an online anomaly detection system that could detect web server log attacks using three different techniques including random

TABLE 1. IDS Research in Web Applications

IDS Model/source	Techniques
TMAD Kakavand et al [13]	**TMAD** uses **n-gram text categorization** (TC) which serve as the commonly term weighting schemes.
N-gram Analysis [14]	**N-gram Analysis** applies three n-gram techniques such as χ^2 **distance statistical test, Ad-hoc n-gram distance, and pattern counting** to the problem of HTTP attack detection.
Automatic Generation [15]	This model relies on the use of a **service-specific, semantic-aware anomaly detection scheme** that combines stochastic learning with a model structure based on the protocol specification.
Online Anomaly Detection Juvonen [17]	**Online Anomaly Detection** uses three different techniques including **random projection, principle component analysis** and **diffusion maps** for anomaly detection.
Auto-encoders (AEs) [18]	**AEs model** using **deep auto-encoder** as the automatic feature learning and an **unsupervised feature learning** approach for both malware classification and network-based anomaly intrusion detection.
A multi-classifier [19]	**Multiple classifier systems** use a **Bayesian detection method** which applies N-grams to extract feature patterns of both the normal and attack payload from the HTTP traffics.

projection (RP), principle component analysis (PCA), and diffusion maps (DM). The results from the three methods showed that the approach can be used to reduce dimensionality before detecting anomalies. However, before logging, the system could not detect attacks that jeopardize the safety of a web application. [18] proposed an Autoencoder-based feature learning called Auto-encoders (AEs), where it produces a model for studying the hidden representation of various function sets with an unsupervised malware classification learning strategy. In a similar research work of malicious payload detection, [19] examined a multi-classifier malicious payload detection system, where the System utilizes two additional techniques of detection, based on Bayesian inference to extract features and a multi-feature-vector to predict whether the incoming payload is anomalous or normal. These researchers focused on the packet payload anomaly detection from HTTP traffic, but the main drawback of these systems is that they are not suited for high dimensional data as attempted in this paper. Table 1 summarizes the research works for the first category which is web applications. Further references to surveys on anomaly-based detection system are available from [20] and [21].

The second category of IDS research is for the web services. Although this category is smaller than the research in web applications, few researches that fit within this category includes works by Thakar et al., [7] and Chan et al., [8]. However, both studys are not considered as a defense-in-depth web service security to detect bad payload embedded in a web service. Other works such as Yee et al., [22] and Wang and Iacono [23] proposed IDS for web services from the theoretical perspective. Lee and Mehta [24] examined the JavaScript Object Notation (JSON) attack in REST web services and proposed a basic input validation method for JSON message in order to prevent input injection attacks. This implies that the message would still be discarded if a hacker registers as a legitimate user and performs assaults through data input. While all these methods can be effective in stopping the aforementioned attacks at that stage in time, there are no statistics in both instances to demonstrate that their methods are effectively significant in terms of detection speed and performance of applications. At present, there is no known anomaly detection system available and tailored to detect XML and JSON attack payloads against web service applications.

To fill these gaps, this paper presents a new IDS model called the Online Adaptive Deep-Packet Inspector (O-ADPI) to detect web service anomalies from high-dimensional data. The proposed O-ADPI model constructs a Unigram-based Weighting Scheme (UWS), a set of Payload Feature Construction (PFC), a Payload Weighting Scheme (PWS), and uses a dimensionality reduction technique based on the Principle Component Analysis (PCA) to enhance the performance of IDSs with multiple criteria. This is to allow feature selection from all subspace areas where there are subspace anomalies to get all PWS. Models are built by mean of supervised approach, namely, Mahalanobis Distance Map (MDM) classifier.

III. WEB SERVICE ATTACKS

A web service is a software system intended to promote machine-to-machine communication over a network, according to W3C [25]. There are two popular existing web service technologies; SOAP-based services and REST-based services [26]. Most of the web services are using traditional simple object access protocol (SOAP), based on the Extensible Markup Language (XML) protocol which is responsible for information transfer; Web Services Description Language (WSDL) that describes the functions of a web service, how it communicates and where it can be found, including the web service registry, which is the Universal Description, Discovery and Integration (UDDI) [27].

As compared to SOAP, which is the Web service's heavyweight preference despite the limits mentioned, REST-based web services are easier to use along with many other advantages [28]. REST-based web services requires no expensive tools, have a smaller learning curve, are effective by using smaller message formats, and are quick because there is no need for a comprehensive processing.

In many practices, REST uses the Uniform Resource Identifiers (URIs) to communicate additional metadata as per the nature of an HTTP request; and uses GET, PUT, POST, DELETE methods (verbs) to remotely select, update, insert, and delete the resources through web services. In addition,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

representation of REST is a resource serialization in a given format, such as the XML and JSON. Theoretically, a resource representation reflects almost any details about the state of a resource. This paper focuses on the REST services to cater the migration of platforms among many service providers from the SOAP-based services to REST-based services [26].

At present, there is inadequate evaluation to countermeasure on REST payload attacks. Consider an e-commerce web service scenario used in online banking, whereby a web service collects the account information of the client and authenticates at the same moment. Next, from the database the client receives his account credit. To acknowledge the targeted and characteristic features of attacks by web services is crucial for an IDS to be able to identify the REST-based web service attacks. This is because REST is basically just a web application that employs a structured set of principles, there are many potential vulnerabilities that an intruder could attack such as via HTTP parameter pollution (HPP), JSON injection attack, XML content attack, and Tag injection attack.

HTTP Parameter Pollution (HPP): HPP attack was discovered in 2009 [29]. This attack relies on the throughout discrepancy by replicating request parameters so as to override program-specific default values in the URLs. Common attacks use the & character to mislead backend services in accepting controlled request variables from the attacker. **Example:** To test for HPP vulnerabilities, the same parameter can be appended to the GET or POST information but assigned with distinct values. Suppose that a HPP attack code is given in Fig. 1. The response page is analyzed to determine which value(s) was/were parsed. In the example below, the owner results may show 987, 456, and some combination of both (987, 456 or 987 456 or ['987','456']). It may also give an empty result, or error page.

FIGURE 1. A JSON injection attack

JSON Injection Attack: JavaScript Object Notation (JSON) is language independent whereby it uses text-based data interchange structure. It is a good alternative for transporting / exchanging messages respectively services between consumers and servers [30]. A JSON injection attack could lead in sensitive data being disclosed [31]. **Example:** The risk is derived from a registered attacker who made a client user profile as a customer with location data through the website. In this case, the attacker attempts to manipulate the user information such as the UserId, Username and Email in order to inject a mimic customer information as shown in Fig. 2. In essence, the service is fooled by fake customer profile. This is a potential safety defect in many REST-based services at the application level.

FIGURE 2. A JSON injection attack

XML Content Attack: XML injection is a method of attack used to manipulate or compromise the XML request or service logic. XML injection can cause the insertion of malicious content into the resulting message/document [6]. **Example:** The attacker inserts characters (metacharacters such as (’), (<), (&), (<![CDATA[ ]]>)) within the XML structures. When the user registers for service; in a standard request, the application gets the user data. In this case, the attacker attempts to insert malicious elements into the concomitant message/document. Thus,the resulting XML document is not well formed or may be invalid, as shown in Fig. 3.

FIGURE 3. XML content attacks with metacharacters

Tag Injection attack: A tag injection attack is a type of XML-content-driven threat that has the ability to pressure the REST-based service hosts into running malicious codes. This will lead to an unwanted condition of the system, such as a XDoS or a straight crash [32]. **Example:** A tag injection attack attempts to falsely terminate a tag. In this instance, an attack is injected as parameter content such as in the following message: <empID>98765</empID>. After the attack was injected, the attacker would properly terminate and force processing. It is simple to imagine a situation in which unintended service conduct can result in access to restricted data as shown in Fig. 4.

FIGURE 4. XML content attacks with metacharacters

Given such attacks in web service layers, current standards are still insufficient to fulfill all REST security needs. Also,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

the REST payload attacks cannot be detected using the traditional IDSs, hence they should be treated as new threats too. This calls for a defense mechanism within the web service layer to protect REST against many new unknown vulnerabilities.Thus the proposed REST mechanism must be sensitive to such attack payloads.

IV. ONLINE ADAPTIVE DEEP-PACKET INSPECTOR MODEL

Fig. 5 presents a new anomaly detection model for web service messages, called the Online Adaptive Deep-Packet Inspector (O-ADPI). The model is trained using current information in the first section. The architecture of the model requires four distinct service payloads (REGISTER, WITHDRAW, UPGRADE and TRANSFER) as input. These might be any structured text files theoretically.

In the second segment, the Unigram-based Weighting Scheme (UWS) is built using either unlabeled training data and/or labeled domain expert anomaly instances. UWS is an assembly of Payload Feature Construction (PFC) and Payload Weighting Scheme (PWS). Using the n-gram, PFC converts the web service payloads into a series of feature vectors while PWS examines the vector space model and serves as a weighting scheme [33] to enhance the feature vectors performance. The weights in the systems refer to the significance of a function in a chosen collection of unique payloads. The training UWS profile is then saved for later.

In the third segment, the PCA algorithm, which is defined as an orthogonal linear transformation, maps information into a reduced dimensionality coordinate structure where sub-space anomalies are likely to occur in high-dimensional space. Next, the O-ADPI uses multiple criteria, called Feature Selection Engine (FSE), to select for the subspaces where subspace anomalies exist in order to obtain PWS. Mathematical solutions such as Guttman-Kaiser Criterion (KC) and FSE applies the Cumulative Criterion (CC) separately to determine the amount of dominant Principle Components (PCs), the outcomes of the dimensionality reduction segment should be maintained according to the assessment.

Finally, for web service attack classification, this research proposes Mahalanobis Distance Map classifier. The model is used for estimating an anomaly score for each message in detection mode. Such a score is a measure of the difference between the message observed and the model's anticipated conduct. The identified anomalies are then presented to the user.

The model is used for estimating an anomaly score for each message in detection mode. Such a score is a measure of the difference between the message being observed and the model's anticipated conduct.

A. SYSTEM ARCHITECTURE AND OPERATION OF O-ADPI

Fig. 6 shows the O-ADPI model associated with high level application web service. The model consists of two main modules: data capture from XML, JSON or a parameter as

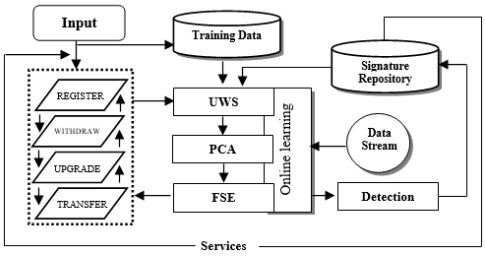


FIGURE 5. An overview of O-ADPI

payload data, as well as text-based knowledge discovery. In addition, the structure of the O-ADPI model is also based on both anomaly and misuse detection techniques, which essentially what drives the machine.

In this research, the O-ADPI model takes in various kinds of data format entries produced through the web service by web applications. As observed in Fig. 6, this component exposes a web service to receive both XML and JSON payload from the application services. From the text-based knowledge discovery point of view, the text mining engine is designed for payload pre-processing operation. There is a payload repository for storing all the payload message services handled by a text categorizer and a weighting scheme.

From the unigram-based weighting scheme point of view, two categories (or labels) were defined; normal and attack for the payload messages to be handled by this module. In order for a statistical model algorithm (MDM) to learn the classification models of the normal and attack categories, it needs to be previously supplied with a set of training instances corresponding to payload messages labelled normal or attack. If an intrusion analyzer detects a possible intrusion attempt, the adequate action will be taken, which generally includes raising a serious alarm with the system administrator and blocking the last action requested by the dubious user, as well as updating the inputs and payload emails to the database of the attack signature. Moreover, in accepted payload messages, it will duplicate inputs onto normal profile database. Next, the operation of the proposed anomaly detector for web service message attacks is described by considering the transaction flow during a web service message attack.

B. O-ADPI FOR WEB SERVICE MESSAGE ATTACKS

In order to illustrate web service message attacks, Fig. 7 shows an instance of payload transaction flow based on specific services for a particular site in order to compute its own "site-specific" service payload anomaly detector. Within this context, the set of four service messages are:

- S_W as the WITHDRAW service message
- S_R as the REGISTER service message
- S_U as the UPGRADE service message
- S_T as the TRANSFER service message

As shown in Fig. 7, by analyzing the weight scores of S_W, replicate request parameters/variables and parameter pollution attacks can be detected. In this service, attacker uses

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

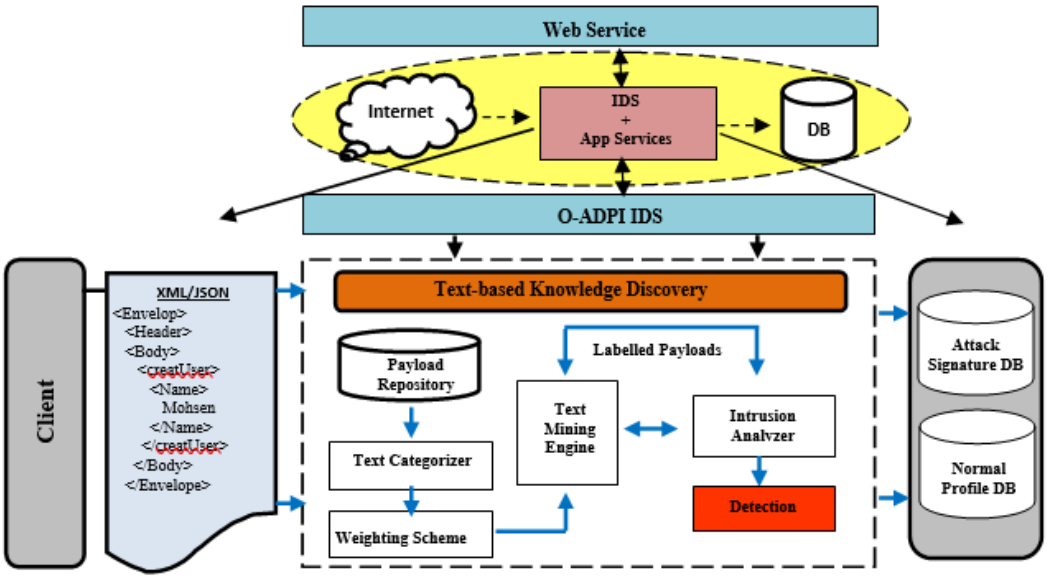


FIGURE 6. An overview of the online adaptive deep-packet inspector model

the "&" character to mislead backend services in accepting attacker-controlled request variables.

By validating weight scores S_R, It is possible to detect malicious content and buffer overflow attacks. Promptly register request is prevented (Fig. 7, [S_R]). By validating the service payloads S_R such as userid, password, email, and username, all XML-related attacks can be detected and prevented (Fig. 7, [S_R]).

In the case of JSON injection attack, consider an intruder who uses the client user profile as a customer and locations data through the S_U (refer Fig. 7, [S_U]). In this case, the attacker attempts to re-enter the user information such as the Accountnumber, Username and Email in order to inject mimic information customer. In the essence, the service is fooled into a fake customer profile, which is a potential security defect in REST-based services at the application level.

Next, in S_T, (Fig. 7, [S_T]) a Tag injection attempt to falsely terminate a tag. For example, an attack is injected as parameter content such as in the following message: <empID>98765</empID>. After the attack is injected, the attacker would properly terminate and force processing. It is simpler to imagine a situation where this can lead to unintended service behaviour, e.g. restricted data access. As each request for service comes with a particular weight score, when a different request is entered, the weight score then varies according to service request regardless of whether the inputs are benign or malicious.

Based on this example, the steps of a payload transaction flow with specific services can be described as follows:

Step 1: The service message is delivered from client site.
- In **S_R** case, client submits userid, password, email, username .

Step 2: The service message values are transmitted to web application server.

Step 3: The weighting score is assigned to service message.

Step 4: Evaluating the weighting value to accept or reject the message.
- If the weight value is valid, the normal transaction is allowed to proceed.
- If the weight value is invalid (malicious), then the transaction is rejected.

Step 5: The malicious message rejected is considered as anomaly for prevention step.
- Malicious transaction information onto attack signature database.

This scenario will be repeated for different web service messages (**S_W**, **S_R**, **S_U** and **S_T**) following the sequence in Fig. 7.

C. DETECTION TECHNIQUES

The intention of this study is to develop and execute a precise and effective method to safeguard web service message from suspecting messages. The approach is based on content analysis of service messages captured on the web service. The O-ADPI model uses four techniques for constructing models of normal web service messages, all of which rely on content-based unigram weighting scheme analysis by utilizing principle component analysis, feature selection engines and a supervised statistical model, called Mahalanobis Distance Map.

1) Unigram-based weighting scheme (UWS)

The weight of a function f in a class reflects f bigoted capacity to standard and attack classes. The greater the weight,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

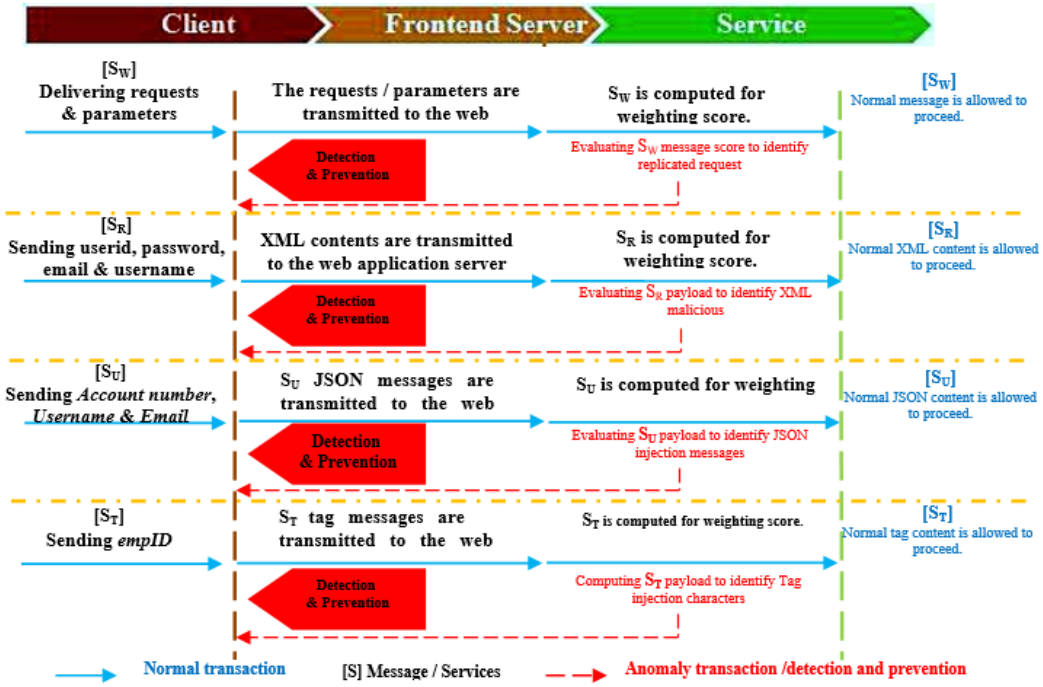


FIGURE 7. An overview of instance of payload transactions flow with specific services

the higher this feature's ability in discriminating the strength to identify the cases of anomaly. Because of its nature, $Payload - FeatureVectorf$ is a stack of characters used as a unigram, the $tf - idf$ (term frequency-inverse document frequency) weighting method, a commonly used technique in the domain of text mining, is used to measure the weight of feature for each class. The $tf - idf$ weight is a statistical measure used to evaluate the importance of a term to a document in a collection or corpus. The importance increases proportionally to the number of times a term appears in the document but is offset by the frequency of the term in the whole corpus.

For a feature, the term frequency (tf) is just the number of times it appears in each payload packet. This count is standardized to give the packet payload a measure of significance for the function. Standardization is performed to avoid a bias towards the class with a larger number of features that may have a higher term frequency regardless of the actual significance of those features in the class.The tf for feature f_i in payload p_j is as defined in Eq. (1):

$$tf_{ij} = \frac{N\left(p_j, f_i\right)}{N\left(p_j\right)} \tag{1}$$

where N(p_j, f_i) denotes the number of occurrences of feature f_i in payload p_j and N(p_j) is the number of occurrences of all features in payload p_j.

The inverse document frequency (idf) is an indication of the overall significance of the term. The idf for feature f_i in payload p_j is described as the reverse proportion of the classes included f_i. The logarithmic shape of this proportion is normally used for scaling purposes as defined in Eq. (2):

$$idf_{i,j} = log\ \frac{N}{|\{p_j,\ \text{where}\ f_i \in p_j\}|} \tag{2}$$

where N corresponds to the total number of payloads and $|\{p_j, where f_i \in p_j\}|$ is the number of packet payloads that contain f_i. Finally, the $t_f - i\,df$ weight of feature f_i with regard to payload p_j is the product of $tf_{i,j}$ and $idf_{i,j}$, as defined in Eq. (3):

$$w_{f_i,\,p_j} = tf_{i,j}.idf_{i,j} \tag{3}$$

2) Principle Component Analysis (PCA)

One of the most common dimensionality reduction methods is the primary component analysis. The objective is to depict the data contained in the initial corresponding variables using a narrower amount of independent variable named principal components [34].

For the initial data matrix to conduct PCA $X = [x_1 x_2 \ldots x_n]$, the matrix is first centered to form the matrix $_{zm} = [(x_1 - \underline{x})\,(x_2 - x) \ldots (x_n - x)]$. Covariance matrix $C_x = \frac{1}{n-1} X_{zm}\, X_{zm}^T$, then the oriented data is calculated. The following decomposition can be used for real-valued matrices as described in Equation (4):

$$\Lambda\, \mathrm{W} = C_x W \tag{4}$$

Next, C_x is then broken down into a matrix W and a diagonal matrix Λ. Consequently, Λ and W are usually categorized throughout climbing down order against the variance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

contributed for every element.The columns of the matrix W indicate the particular eigenvectors (i.e., the principal components) from the covariance matrix C_x, along with the factors across the diagonal of the matrix Λ including the placed eigenvalues connected with the corresponding eigenvectors inside the matrix W. Like a linear mathematical method, PCA can be enhanced depending on eigenvector-based multivariate evaluation.

3) Feature Selection Engine (FSE)

Feature selection techniques provide a way to reduce computational time, increase efficiency of predictions and information comprehension in machine learning applications. The focus of feature selection (variable elimination) is to pick a subset of variables from the input that can describe the input information effectively while decreasing noise impacts or irrelevant variables and still provide excellent predictive outcomes [35]. The O-ADPI model uses two mathematical solutions, which are Cumulative Criterion and Guttman-Kaiser Criterion to be applied independently during the PCA stage to determine the number of dominant principle components (PCs).

Component selection based on Guttman-Kaiser Criterion: The Guttman-Kaiser criterion [36] that is related to every single factor is displayed by the corresponding eigenvalues. Principal components associated with eigenvalues are extracted from a covariance matrix. The rules propose to hold only principal components as they are the eigenvalues larger than 1. While 1 might be considered as the average variance for the standardized data, the rule has been modified in order to select PCs derived from the covariance matrix as follows in Eq. (5):

$$KC = \sum_{i=1}^{256} \frac{\sigma_i^2}{256} \tag{5}$$

Nonetheless, the components which are larger in magnitude than the average of the eigenvalues are preserved. In the case of eigenvalues extracted from a covariance matrix, the average is determined using above equation.

Component selection based on Cumulative Criterion: The respective own value represents an energy connected with a component. The higher the individual value, the greater the energy of the corresponding component (eigenvector). Suppose $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{256}\}$ that are eigenvalues that decomposed from the covariance matrix C_x . Cumulative Criterion (CC) is used exclusively and to sample corresponding k_2 principal components (the eigenvectors of matrix W). The cumulative criterion is described by the amount of energies between the first element and the k_2 element, and it is computed using as defined in Eq. (6):

$$CC = \sum_{i=1}^{k_2} \lambda_i \quad where\ k_2 \in \{1,\ 2, \ldots 256\} \tag{6}$$

TABLE 2. Performance metrics for classification evaluations

IDS Model/source	Techniques	Formula
Precision	It is the fraction of data instances predicted as positive that are actually positive	$\frac{TP}{TP+FP}$
True Positive Rate Sensitivity (Recall)	Intrusions that are successfully detected by the IDS.	$\frac{TP}{TP+FN}$
False Positive Rate	Normal behavior that is wrongly classified as intrusive by the IDS.	$\frac{FP}{FP+TN}$
Accuracy	It is computed as the ratio between the number of correctly detected attacks and the total number of attacks.	$\frac{TP+TN}{TP+TN+FP+FN}$
F-value	For a given threshold, the F-value is the harmonic mean of the precision and recall at that threshold.	$2 \cdot \frac{Recall \cdot Precison}{Recall+Precision}$

where CC can be determined subject to the objective function given in Eq. (7):

$$CC \geq \ \alpha \ . \sum_{i=1}^{k_2} \lambda_i \tag{7}$$

Note that is the subspace variation ratio to the original space's total variation. This objective function aims to achieve a value of k_2 as low as possible while at the same time attaining a fairly elevated percentage-based value of CC. For brevity, the steps of the Feature Selection Engine (FSE) is illustrated in Fig. 8.

Based on Fig. 8, once the eigenvalues are collected from the covariance matrix, two different feature selection techniques will be applied, which are the Kaiser Criterion on one hand, and the Cumulative Criterion on the other hand. Next, the feature module is used for refinement and assessment. In the phase of refinement, the range of the selected principal components is extended and the discriminative power of the subsets of principal components to represent massages is observed. Lastly, the final PCs $\in k_1,\ k_2$ are selected through a reiterative process of normal training model using the F-value as defined in Table 2.

In addition to Table 2, Precision indicates how many events an IDS predicts as anomaly, but they are the genuine attacks. A low accuracy value implies a greater level of false positives and vice versa. Recall measures the missing part of the Precision, namely the percentage of the actual intrusions the classifier is covering. A reduced recall value is a greater degree of false negatives and vice versa.

The metrics utilized are True Positive (TP) when the number of actual attack is classified as an attack, True Negative (TN) when the number of actual normal is classified as normal, False Positive (FP) when the number of actual

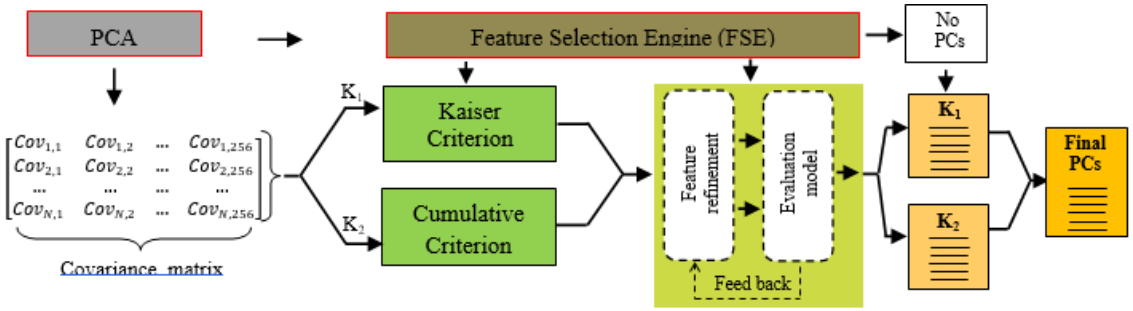


FIGURE 8. Schema of the proposed feature selection engine (FSE)

TABLE 3. Confusion matrix

		Predicted Class	
		Normal	**Attack**
Actual Class	Normal	True Negative	False Positive
	Attack	False Negative	True positive

normal is classified as attack and False Negative (FN) when the number of actual attack is classified as normal. Table 3 represents the definition of a confusion matrix. Subsequently, detection and false positive rate can be estimated as shown in Table 3.

The selected PCs are the one which facilitates the classifier to achieve the greatest F-value among the candidates k_1 and k_2. Then the selected PCs are used in the Mahalanobis Distance Map (MDM) classifier.

4) Mahalanobis Distance Map (MDM)

The Mahalanobis Distance Map (MDM) is used to measure the separation of two groups of objects. In this research, MDM is used to find similarity between an anomaly and normal instances. The Mahalanobis distance between the particular point x and the mean μ of the normal data is computed as defined in Eq. (8):

$$d_M \leftarrow \sqrt{(x-\mu)^T \,.\, \Sigma^{-1} .\, (x-\mu)} \qquad (8)$$

where, $X = [x_1 x_2 \ldots x_n]$ is set of a given dataset, μ refers to the average of each feature and Σ^{-1} is the inverse of the covariance matrix.

V. EXPERIMENTAL DESIGN AND EVALUATION OF THE MODEL

A. DATASET

One significant issue when assessing proposals linked to IDS technology is the absence of well-established frameworks for proper evaluation of experimental outcomes. Many tests are therefore not reproducible, leading to problems in assessing the results validity. The DARPA / MIT Lincoln Lab Framework for 1998 and 1999 [37] was an early attempt in this direction and was used for years to evaluate fresh ideas, both in the field of IDS and in global contests such as Knowledge Discovery and Data Mining (KDD). The DARPA dataset has now been considered outdated and has been widely criticized [38] due to the nature of the simulation environment that created the data [39]. Moreover, the features of both the attacks and ordinary traffic included in such datasets are currently mostly outdated. Researchers are urged to set up their own assessment structure in this scenario, motivating the experimental design and sharing it to promote the reproducibility of experiments where it is feasible.

B. SIMULATED REST-IDS DATASET 2015

At this point, there is no open access IDS dataset that is derived specifically from web services, this research generated a new set of data called the REST-IDS Dataset 2015. This dataset consists of five days network activities capturing 12,600 packets of data messages. It also contains 4,200 web service attacks to allow evaluation of detection systems on high level web service applications. The dataset was implemented based on open-source code proof-of-concept version. In order to identify vulnerabilities and to verify the REST-IDS 2015 dataset, a free and open source cross-platform automated security testing solution called the soapUI [40] was employed. One of the features of soapUI is the ability to automate security testing on attacks using scripts to work with REST-based web services. The verification process involved:

- Verifying that the attack instance has been correctly executed
- Determining labels for each normal, and attack
- Classifying each attack instance over specific REST services

The process of generating the new dataset was based on findings by Shiravi et al. [41], whereby there are four important factors that should be considered in order to define a set of requirements for a new IDS-dataset; whereby the dataset (1) should not exhibit any unintended property due to post-insertion of data, (2) must be adequately labeled, (3) should cover all network interactions, and (4) must be entirely non-anonymized. Fig. 9 shows the architecture for generating the REST-IDS Dataset 2015 followed by description of each component.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

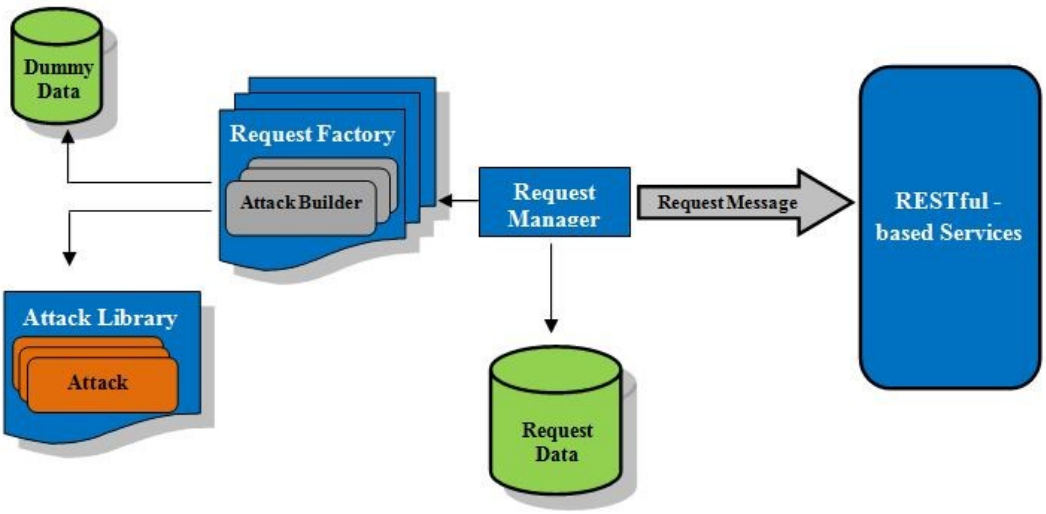


FIGURE 9. Testbed Architecture fo REST-IDS Dataset 2015

- **Dummy Data:** This is an XML file having a total of 3,000 unique records with basic raw data such as first and last names, addresses, phone and emails. The data was generated for free from an online website (http://www.generatedata.com/). The request factory and builders may use this database in order to create different sets of normal or attack request.
- **Services:** Based on the given requirements, few RESTful web services have been created as the targets for normal and malicious requests. These services are connected to a database and in practice are there to receive the data and perform simple operations like deserialization and CRUD to database.
- **Request Manager:** This is the first trigger point from client-side application. It receives the required service arguments and the attack type, and then works with associated request factory to get the request. After that, it creates an actual HTTP request and sends it to the server and at the same time logs the request into the request database as well.
- **Request Factory:** For each service, there should be a specific instance from the request factory to handle creation of the request and its payload. Based on the given attack type, this component internally calls for an attack request builder to build the request accordingly. Therefore, a request factory may need to use multiple request builders. For example, for this service, the builder for attack of type A might be different than the builder for attack of type B.
- **Request Builder:** This component receives a preliminary request from a factory as an input and tries to update and modify it in order to build the preferred and expected request/attack type.
- **Attack Library:** This is a library in which the logic for creating predefined attacks has been coded.
- **Attack:** An Attack is an implementation of request builder type and can be called directly by the request factories (such as the request builders) or they can be extended by custom request from the builder components. Each attack type has its own request builder implementation in the Attack Library. A complete description can be found at the Open Web Application Security Project [42]. In this dataset, the following are the simulated attacks:
  - HPP (HTTP Parameter Pollution
  - JSON Injection Attack
  - XML Injection Attack (malicious content)
  - XML Tag injection
- **Request Dataset:** This is a file storage (in Microsoft Excel format) to keep all normal and attack requests per each service. The request manager is responsible to insert new requests after necessary validations.

Next, based on the requested schema, the dataset keeps only few and required fields of a given HTTP request instead of all of them. This implementation stores a complete request data in a separate XML file and uses a GUID as key to link each of the requests from Excel to the corresponding element within the XML file. This allows the experiment to ask for other fields whenever necessary. Table 4 provides four major attack categories that were chosen with regards to the actions over specific REST-based services against the number of simulated attacks and number of data packet messages.

We developed REST-IDS Dataset 2015 by building the above components with the following technologies: .Net Framework 4, C# 5, Entity Framework 6.1, Single database file- SqlServerCompact, WCF (RESTful) and XML.

TABLE 4. REST-IDS Dataset 2015

NO	Service names	Simulated attacks	No. of data packets	No. of attacks	Total
1	WITHDRAW	HPP (HTTP Parameter Pollution)	2680	1200	3880
2	UPGRADE	JSON Injection Attack	1520	600	2120
3	REGISTER	XML Injection Attack	2800	1800	4600
4	TRANSFER	XML Tag injection	1400	600	2000
Total	All Service	All attacks	8400	4200	12600

C. EVALUATION ON DIFFERENT WEIGHTING SCHEME SERVICES

Note that the weight of a feature score assigned to each pattern is used to determine whether the feature is anomalous or a normal instance. Table 5 lists all the weighting scheme values or all data (normal and attack) from different web services. The features of four service message data are computed tf-idf weighting method in the original high-dimensional space. With regards the quality of the features shown in Table 5, most of the features derived from different web services will be assigned different weight scores for each packet payload. The weight scores, however, are very difficult to assess because different services produce different set of weights or scores values. The results showed that the specific scores obtained by each of the web service data are independent from one another. The following are different combination strategies used to compare the scores:

1) Select the minimum weight score for each feature
2) Select the maximum weight score for each feature
3) Average the weight score of all features

Table 6 shows that all three proposed weight score strategies achieved reasonable values in comparison to the values obtained for each web service data with its respective class labels. However, in general, better results are obtained by averaging all the weight scores.

Next, the TF-IDF weighting scores were used to classify normal and attack packets. Results for the TF-IDF weighting scores for different web service data with normal and attack massages are presented in Fig. 10, where the X-axis represents the number of features and the Y-axis represents TF-IDF weighting scores values. For all the web services a) WITHDRAW, b) UPGRADE, c) REGISTER, and d) TRANSFER, the results showed that the TF-IDF weighting scores values for normal messages are much smaller than the TF-IDF weighting scores values for attack messages. These weight score policies were obtained from the simulated dataset within the valid range of weight scores. Anomalous values that contain invalid payload values and out-of-range REST/SOAP payload can be then identified.

VI. EXPERIMENTAL RESULTS

This segment provides the experimental outcomes using the 2015 REST-IDS Dataset for the suggested Online Adaptive Deep-Packet Inspector (O-ADPI) model. The experiments were performed based on a standard stratified 10-fold cross-validation technique to estimate the Area Under Curve (AUC) for each target class. This is to prevent bias hiding in the suggested system's sequential information influencing the ordinary profile generation and detection efficiency. The results presented are three-fold; (1) analysis on optimal feature subspace selection, (2) analysis on detection performance, and (3) analysis on computational and speed scalability.

A. OPTIMAL FEATURE SUBSPACES

Analysis on the selected or filtered legitimate (normal) traffic was conducted using the Principal Component Analysis (PCA) algorithm that is designed to determine the optimal feature subspace for data representation of the entire training dataset.

Four feature subspaces were chosen with respect to the normal service payloads; $\mathbf{S}_W$, $\mathbf{S}_R$, $\mathbf{S}_U$ and $\mathbf{S}_T$, where they were used in both the training and testing phases so as to supply with accurate representation for all records. Next, the Cumulative Criterion and Guttman-Kaiser Criterion were employed during the election of the optimal feature subspaces. In order to determine the number of critical PCs to be retained for various types of web service payloads during the evaluations, Fig. 11 and Fig. 12 shows the plots for normal $\mathbf{S}_W$, $\mathbf{S}_R$, $\mathbf{S}_U$ and $\mathbf{S}_T$ service payloads extracted from REST-IDS Dataset 2015 against the Cumulative Criterion and Guttman-Kaiser Criterion, respectively. The figures of horizontal axes represent the number of PCs and the figures of vertical axes represent their own values (variances) with respect to the numbers of PCs displayed on the horizontal axes.

Note that the up-slopes on the plots for $\mathbf{S}_W$, $\mathbf{S}_R$, $\mathbf{S}_U$ and $\mathbf{S}_T$ service payloads are found lying at different numbers PCs over the feature selection engines. The same result can also be seen from the REST-IDS Dataset 2015 evaluation dataset. However, the numbers of PCs are not always practicable, and the best performance may be achieved around these numbers.

Next, Fig. 13 shows the potential optimal subspace for data representation so that the less important PCs can be eliminated and only the first several critical PCs to be retained as the new low dimensional feature space.

However, during the training phase, normal profiles were generated with respect to various types of web service payload. Since the plots of the Cumulative Criterion and Kaiser Criterion variances only suggest a preliminary result, further selection is necessary based on the suggestion from the preliminary outcomes. In this work, we test the potential

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

TABLE 5. Weighting scheme values for all data (Normal and attack) in different services

Feature	All services							
	(a) WITHDRAW		(b) UPGRADE		(c) REGISTTER		(D) TRANSFER	
	Normal	Attack	Normal	Attack	Normal	Attack	Normal	Attack
"	0.0000	0.0000	0.1531	0.2085	0.0000	0.0000	0.0000	0.0000
>	0.0000	0.0000	0.0000	0.0000	-0.0026	-0.0027	-0.0039	-0.0049
/	0.0000	0.0000	0.0000	0.0000	-0.0013	-0.0013	-0.0019	-0.0024
v	0.0000	0.0000	0.0026	0.0035	0.5088	0.4818	0.0000	0.0000
u	-0.0005	-0.0007	0.0585	0.0808	-0.0020	-0.0020	-0.0019	-0.0026
t	-0.0002	-0.0003	0.0418	0.0583	-0.0012	-0.0012	-0.0039	-0.0050
m	-0.0002	-0.0003	0.0524	0.0720	-0.0016	-0.0016	-0.0009	-0.0013
e	-0.0002	-0.0003	0.0674	0.0936	-0.0026	-0.0027	-0.0029	-0.0036
0	0.4000	0.5506	0.0069	0.0096	0.4024	0.3971	0.4348	0.5521
<	0.0000	0.0000	0.0000	0.0000	-0.0026	-0.0027	-0.0039	-0.0049
r	-0.0002	-0.0003	0.0570	0.0795	-0.0014	-0.0014	-0.0039	-0.0046
?	-0.0002	-0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
...	...	...	...	...	...	...	...	...
{	0.0000	0.0000	0.0191	0.0241	0.0000	0.0000	0.0000	0.0000
s	-0.0002	-0.0003	0.0241	0.0346	-0.0012	-0.0012	-0.0019	-0.0022
=	-0.0005	-0.0007	0.0000	0.0000	-0.0002	-0.0002	0.0000	0.0000
c	-0.0002	-0.0003	0.0536	0.0777	-0.0025	-0.0026	-0.0049	-0.0065
6	0.3923	0.6029	0.0033	0.0046	0.3906	0.3768	0.4019	0.5212
3	0.3503	0.5507	0.0037	0.0052	0.2958	0.4446	0.3702	0.5036
a	-0.0002	-0.0003	0.0639	0.0895	-0.0038	-0.0039	-0.0049	-0.0063
n	-0.0002	-0.0003	0.0546	0.0764	-0.0025	-0.0025	-0.0039	-0.0049
o	-0.0005	-0.0007	0.0589	0.0822	-0.0019	-0.0020	-0.0039	-0.0052
@	0.0000	0.0000	0.0114	0.0164	-0.0002	-0.0002	0.0000	0.0000
f	0.0000	0.0000	0.0026	0.0040	-0.0005	-0.0005	-0.0009	-0.0009
j	0.0000	0.0000	0.0046	0.0063	0.5404	0.5425	0.0000	0.0000
g	0.0000	0.0000	0.0219	0.0292	0.5291	0.5434	-0.0009	-0.0013

TABLE 6. Comparison of minimum, maximum and average obtained by the proposed weighting method based on REST-IDS 2015 evaluation dataset

Strategies scores	WITHDRAW		UPGRADE		REGISTER		TRANSFER	
	Normal	Attack	Normal	Attack	Normal	Attack	Normal	Attack
Minimum	-0.0005	-0.0007	0.0000	0.0000	-0.0038	-0.0039	-0.0049	-0.0065
Maximum	0.4023	0.6078	0.1531	0.2085	0.5829	3.2789	0.4348	0.6005
Average	0.1626	0.2425	0.0238	0.0327	0.1667	0.1667	0.1473	0.1957

TABLE 7. The number of principle components (PCs)

Name of Service		WITHDRAW	UPGRADE	REGISTER	TRANSFER
No. of PCs	Kaiser	11 PCs	23 PCs	39 PCs	12 PCs
	Cumulative	8 PCs	15 PCs	25 PCs	7 PCs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

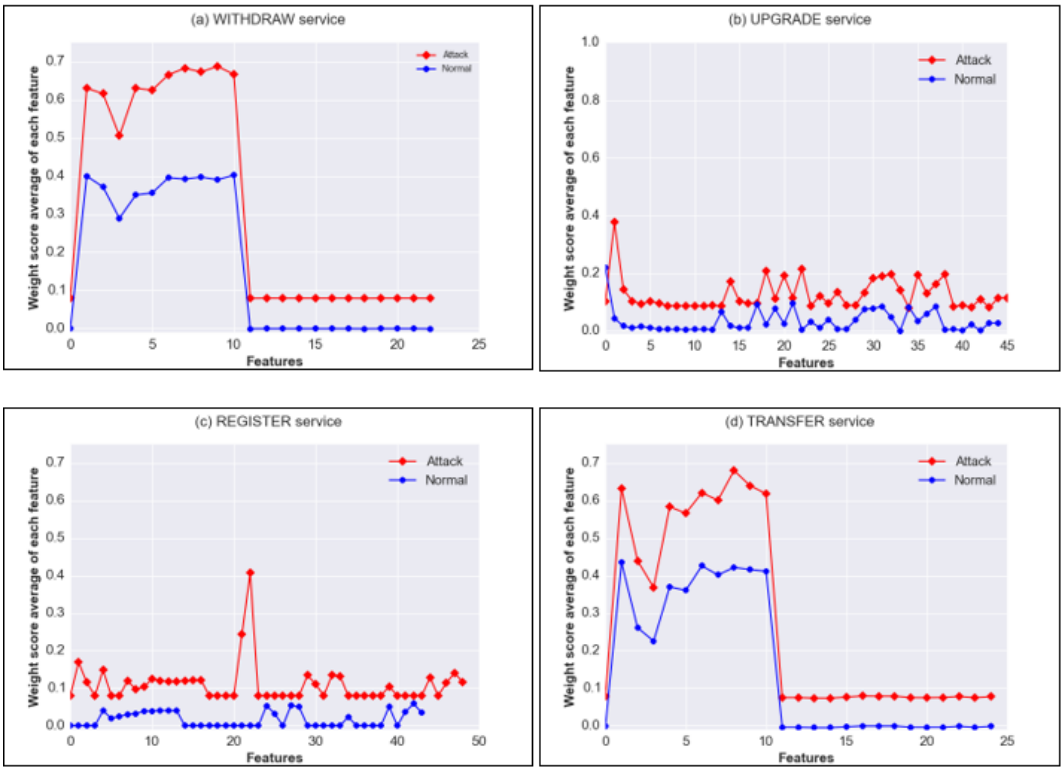


FIGURE 10. Weight scores of characters of (a) WITHDRAW service (b) UPGRADE service (c) REGISTER service (d) TRANSFER service

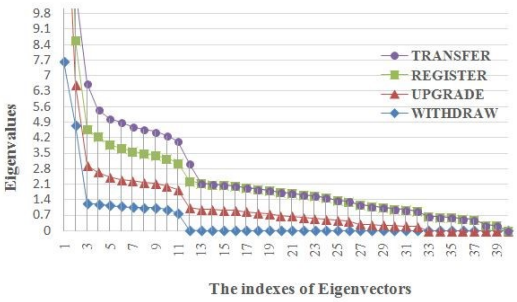


FIGURE 11. Guttman-Kaiser Criterion plot variance for REST-IDS Dataset 2015

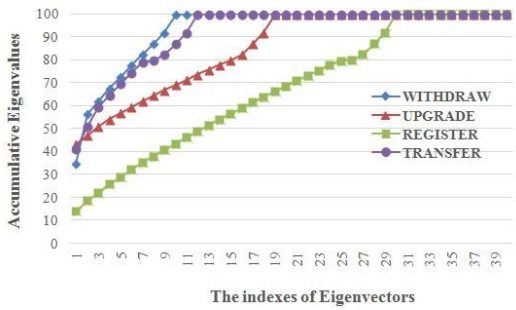


FIGURE 12. Cumulative Criterion plot variance for REST-IDS Dataset 2015

optimal subspace of for each type of web service payloads. Table 7 shows the numbers of PCs used in the potential optimal subspace for each type of web service payloads. Normal profiles were built with respect to the chosen feature subspaces (i.e., the aforementioned numbers of PCs). Then, the generated normal profiles were utilized during the testing phase.

B. DETECTION PERFORMANCE

Once the optimal feature subsets have been selected, the detection performance of the proposed O-ADPI model was evaluated against both the normal records and the attack records from the evaluation dataset. The threshold value was set to 2.5σ, with respect to different normal profiles using various sets of PCs as shown in Table 7. Note that the best classification result was classified and achieved with the first selected lower dimensional PCs for $\mathbf{S}_W$, $\mathbf{S}_R$, $\mathbf{S}_U$ and $\mathbf{S}_T$.

Table 8 shows the evaluation result for measuring the trade-off between TF, FP and F-value against different types of feature selection engines, each with its corresponding PCs. The proposed detection model enjoyed a promising performance on REST-IDS Dataset 2015 in TP, FP and F-value. The performance of the both feature selection engine achieved high efficiency. At this point, although the detection performance by O-ADPI dropped significantly across all services except one, but the both feature selection engine still have the best F-values rate. In order to illustrate a clearer

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2953791, IEEE Access

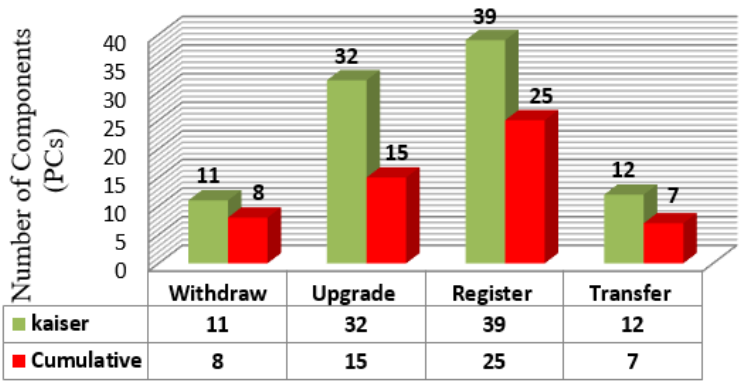


FIGURE 13. The numbers of principle components using in the training and testing for various services from REST-IDS Dataset 2015

picture on the relationship between F-values and PCs, Fig. 14 visualizes the trade-off between F-values and the PCs as well as different feature selection engines on REST-IDS Dataset 2015.

Overall, the proposed O-ADPI model performed very well across all services except one (REGISTER service). The results are contributed by two factors; (1) the relative frequencies from the Unigram-based Weighting Scheme (UWS) and (2) the weighting scheme method.

VII. COMPUTATIONAL AND SPEED SCALABILITY

Note that evaluation for computational and speed scalability are more challenging and more difficult than feature extraction. Once an anomaly is discovered in one or more subspaces, finding its other outer subspaces using current anomaly detection algorithms is a relatively simple challenge. The detection of anomalies from information streams is conducted online in this study while discovering the remote subspaces of the anomalies is operated offline. Therefore, the main concern would be how to ensure the speed for the O-ADPI model is scalable and able to operate in online mode.

This data set (REST-IDS Dataset 2015) contains XML/JSON messages to a real web service. In these services there are much more service payloads. Various subsets of the data are used to test the speed and especially the scalability of the O-ADPI model methodology to compare the efficiency on larger datasets. However, the size of the service payload is big enough (Up to 12,000 service payloads) to test the scalability aspects. However, we expect to be much less in the throughput than we used in real time for comparison.

Because real-time performance is of primary importance to any intrusion detection system, the O-ADPI was evaluated for speed efficiency by comparing the O-ADPI throughput with a similar environment used in a medium-sized web services corporation. The throughput comparison was based on the number of service payloads processed through such a web services against the service payload processing speed of our system considering the most ideal parameters.

Next, Table 9 shows the average processing time (memory usage) for all service payloads through O-ADPI model in different service payloads. All of these timings were obtained on a system with the following configuration.

- CPU: Intel (R) Core (TM) i-7 1.80GHz
- Memory (RAM): 16 GB
- OS: Microsoft Windows 8.1
- System type: 64-bit Operating System

VIII. CONCLUSIONS

This paper presented the Online Adaptive Deep-Packet Inspector (O-ADPI) for detecting web service message attacks. The O-ADPI model works on a high application layer and is heavily based on an underlying anomaly detector procedure where a deep-packet inspection is performed. The approach relies on the use of a message-specific service, Unigram-based Weighting Scheme (UWS) that combines text mining techniques with a PCA-based Feature Selection Engine (FSE) using the Guttman-Kaiser criterion and the Cumulative Criterion to effectively and efficiently explore subspaces in detecting anomalies embedded in feature spaces. We utilized a supervised intrusion detection algorithm based on mahalanobis distance map classifier in order to classify web service attacks into anomaly and normal.

The proposed approach was assessed using the REST-IDS Dataset 2015. The experimental outcome demonstrated that the proposed O-ADPI model achieved promising results in each service message. These tests are important for new REST-based services for intrusion detection.

The key to the O-ADPI model lies in the statistical approach for dimensionality reduction and feature selection in analyzing any web service message produced by various applications. The effectiveness of detecting anomalies in any text mining assignment is also evident. In addition, the analysis can be carried out on enormous amounts of information in adequate time. Larger volumes of information need to be analyzed for future studies to guarantee effective scaling. The system may be throttled by I / O instead of CPU, thus creating new difficulties.

TABLE 8. Evaluation of REST-IDS Dataset 2015 for four attack types

Services	10 Fold Cross Validation							
	Feature Selection Engines							
	Kaiser Rule			PCs	Cumulative			PCs
	Metrics				Metrics			
	TP	FP	F-Value		TP	FP	F-Value	
WITHDRAW	99.0%	0.76%	99.21%	11	99.40%	0.68%	98.80%	8
UPGRADE	98.75%	0.87%	98.83%	32	99.10%	0.72%	99.00%	15
REGISTER	81.63%	3.20%	98.00%	39	79.67%	8.18%	95.64%	25
TRANSFER	98.93%	0.79%	98.95%	12	98.30%	0.75%	99.38%	7

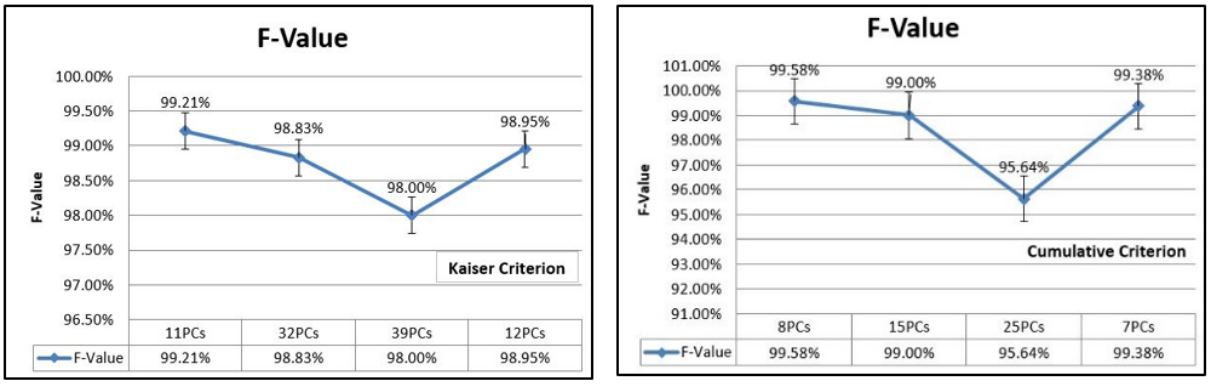


FIGURE 14. Trade-off between F-values and PCs

TABLE 9. Performance comparison (time in milliseconds)

Service Name	Selection Engine	Building Time	Testing Time
WITHDRAW	KC	1.043±0.042	0.010±0.038
	CC	1.145±0.012	0.013±0.005
UPGRADE	KC	0.426±0.012	0.004±0.005
	CC	0.596±0.021	0.006±0.008
REGISTER	KC	4.581±0.121	0.013±0.006
	CC	3.566±0.201	0.012±0.005
TRANSFER	KC	0.294±0.019	0.003±0.007
	CC	0.296±0.024	0.003±0.001

REFERENCES

[1] K. Goseva-Popstojanova, G. Anastasovski, A. Dimitrijevikj, R. Pantev, and B. Miller, "Characterization and classification of malicious Web traffic," Computers and Security, vol. 42, pp. 92–115, 2014.

[2] S. Suriadi, A. Clark, H. Liu, D. Schmidt, J. Smith, and D. Stebila, "Denial of Service Defence Appliance for Web Services," Springer, pp. 239–289, 2011.

[3] A. Vorobiev, "Security Attack Ontology for Web Services," in IEEE, Semantics, Knowledge and Grid, Second International Conference on, vol. 6, pp. 42–42, Ieee, 2006.

[4] R. Bunge, S. Chung, B. Endicott-Popovsky, and D. McLane, "An Operational Framework for Service Oriented Architecture Network Security," in Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), pp. 312–312, Ieee, 2008.

[5] S. Shah, Hacking Web Services. Charles River Media, 2007.

[6] M. Jensen, N. Gruschka, and R. Herkenhoner, "A Survey of Attacks on Web Services," Computer Science - Research and Development , Springer, vol. 24, no. 4, pp. pp 185–197, 2009.

[7] U. Thakar, N. Dagdee, and S. Varma, "Pattern Analysis and Signature Extraction For Intrusion Attacks On Web Services," International Journal of Network Security & Its Applications, vol. 2, no. 3, pp. 190–205, 2010.

[8] G.-Y. Chan, C.-S. Lee, and S.-H. Heng, "Discovering fuzzy association rule patterns and increasing sensitivity analysis of XML-related attacks," Elsevier Science, Journal of Network and Computer Applications, vol. 36, pp. 829–842, mar 2013.

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," ACM Computing Surveys, vol. 41, pp. 1–58, jul 2009.

[10] Y. Yu, "A survey of anomaly intrusion detection techniques," Journal of Computing Sciences in Colleges, vol. 28, no. 1, pp. 9–17, 2012.

[11] C. Kruegel, G. Vigna, and W. Robertson, "A multi-model approach to the detection of web-based attacks," Elsevier Science, Computer Networks, vol. 48, pp. 717–738, aug 2005.

[12] A. Jamdagni, Z. Tan, P. Nanda, X. He, and R. P. Liu, "Intrusion detection using GSAD model for HTTP traffic on web services," in Proceedings of the 6th International Wireless Communications and Mobile Computing Conference on ZZZ - IWCMC '10, (New York, New York, USA), p. 1193, ACM Press, 2010.

[13] M. Kakavand, N. Mustapha, A. Mustapha, and M. T. Abdullah, "A Text Mining-Based Anomaly Detection Model in Network Security," Global Journal of Computer Science and Technology, vol. GJCST 201, no. 5, pp. 23–31, 2015.

[14] A. Oza, K. Ross, R. M. Low, and M. Stamp, "HTTP attack detection using n-gram analysis," Computers and Security, vol. 45, no. 2011, pp. 242–254, 2014.

[15] P. Garcia-Teodoro, J. E. Diaz-Verdejo, J. E. Tapiador, and R. Salazar-Hernandez, "Automatic generation of HTTP intrusion signatures by selective identification of anomalies," Computers and Security, vol. 55, pp. 159–174, 2015.

[16] F. Angiulli, L. Argento, and A. Furfaro, "PCkAD: an unsupervised intrusion detection technique exploiting within payload n-gram location distribution," Cryptography and Security, pp. 1–6, 2014.

[17] A. Juvonen, T. Sipola, and T. Hamalainen, "Online anomaly detection using dimensionality reduction techniques for HTTP log analysis," Computer Networks, vol. 91, pp. 46–56, 2015.

[18] M. Yousefi-azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based Feature Learning for Cyber Security Applications,"

in 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017.

[19] Z. Zhang, R. George, and K. Shujaee, "An Approach to Malicious Payload Detection," in 2018 World Automation Congress (WAC), pp. 1–5, IEEE, 2018.

[20] M. Kakavand, N. Mustapha, A. Mustapha, M. T. Abdullah, and H. Riahi, "Issues and Challenges in Anomaly Intrusion Detection for HTTP Web Services," Journal of Computer Sciences, vol. 11, no. 11, pp. 1041–1053, 2015.

[21] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Computer Networks, vol. 51, no. 12, pp. 3448–3470, 2007.

[22] C. G. Yee, W. H. Shin, and G. Rao, "An Adaptive Intrusion Detection and Prevention (ID/IP) Framework for Web Services," in International Conference on Convergence Information Technology (ICCIT), pp. 528–534, Ieee, nov 2007.

[23] J. Wang and L. L. Iacono, "Intrusion Detection and tolerance in Grid-based applications," in Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks, SecureComm, pp. 177–185, IEEE, 2007.

[24] H.-m. Lee and M. R. Mehta, "Defense Against REST-based Web Service Attacks for Enterprise Systems," Communications of the IIMA, vol. 13, no. 1, pp. 57–68, 2013.

[25] H. Haas and A. Brown, "Web Services Glossary," 2004.

[26] B. Upadhyaya, Y. Zou, H. Xiao, J. Ng, and A. Lau, "Migration of SOAP-based services to RESTful services," Proceedings - 13th IEEE International Symposium on Web Systems Evolution, WSE 2011, pp. 105–114, 2011.

[27] M. S. Najjar and M. Abdollahi Azgomi, "A distributed multi-approach intrusion detection system for web services," in ACM, Proceedings of the 3rd international conference on Security of information and networks - SIN '10, vol. 7, (New York, New York, USA), p. 238, ACM Press, 2010.

[28] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, 2000.

[29] C. Luca and P. Stefano, "HTTP Parameter Pollution, a new category of web attacks," 2009.

[30] C. Severance, "Discovering JavaScript Object Notation," IEEE Computer Society, pp. 6–8, 2012.

[31] J. Aviram, "Testing JSON Applications for Security Holes," 2009.

[32] R. G. Esfahani and M. A. Azgomi, "Towards an Anomaly Detection Technique for Web Services Based on Kernel Methods," in Innovations in Information Technology, pp. 345–349, IEEE, 2009.

[33] R. Banchs, Text Mining with MATLAB. New York, NY: Springer New York, 2013.

[34] I. T. J. Jolliffe, Principle Component Analysis. Wiley Online Library, 2005.

[35] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," Journal of Machine Learning Research, vol. 3, no. 2, pp. 1157–1182, 2003.

[36] C. E. Lance and R. J. Vandenberg, Statistical and Methodological Myths and Urban Legends: Doctrine, Verity and Fable in Organizational and Social Sciences. Taylor & Francis Group, 2009.

[37] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," Computer Networks, vol. 34, no. 4, pp. 579–595, 2000.

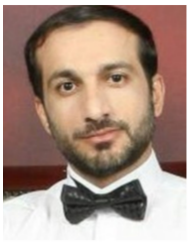
[38] J. Mchugh, "Testing Intrusion Detection Systems : A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2001.

[39] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," 2010 IEEE Symposium on Security and Privacy, vol. 0, no. May, pp. 305–316, 2010.

[40] SoapUI, "Getting Started with REST Testing," 2015.

[41] A. Shiravi, H. Shiravi, M. Tavallaee, and A. a. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Computers & Security, Elsevier, vol. 31, pp. 357–374, may 2012.

[42] The Open Web Application Security Project (OWASP), "The Ten Most Critical Web Application Security Risks," 2013.



MOHSEN KAKAVAND received the Ph.D. degree in intelligent computing from the University Putra Malaysia (UPM), Malaysia in 2017. He is currently a Lecturer with the Department of Computing and Information Systems, Faculty of Science and Technology, Sunway University in Malaysia. His research interests include aspects of data mining, intelligent computing, machine learning, intrusion detection systems (IDSs), and cybersecurity.

AIDA MUSTAPHA received the B.Sc. degree in Computer Science from Michigan Technological University and the M.IT degree in Computer Science from UKM, Malaysia in 1998 and 2004, respectively. She received her Ph.D. in Artificial Intelligence focusing on dialogue systems. She is currently an active researcher in the area of Computational Linguistics, Soft Computing, Data Mining, and Agent-based Systems.

ZHIYUAN TAN received the Ph.D. degree in computer systems from the University of Technology Sydney, Ultimo, NSW, Australia in 2014. He was a Post-Doctoral Researcher of cybersecurity with the University of Twente, The Netherlands from 2014 to 2016; and a Research Associate with the University of Technology Sydney in 2014. He is currently a Lecturer of cybersecurity with the School of Computing, Edinburgh Napier University, U.K. His research interests include cybersecurity, machine learning, pattern recognition, data analytics, virtualization, and cyber-physical system. He is an EAI and BCS Member. He was the Chair of international workshops and conferences, such as SECSOC, SITN, EAI Future 5V, and EAI BD:TA 2018. He serves on the Editorial Board for the International Journal of Computer Sciences and its Applications. He is an Associate Editor of the IEEE Access and a Guest Editor of Special Issues for the Computers and Electrical Engineering, and the IEEE Access.

SEPIDEH FOROOZAN YAZDANI with the background of pure mathematics, received her M.Sc. degree in Information Technology from Multimedia University (MMU), Malaysia. She is currently working towards her Ph.D. in Information Systems at Computer Science and Information Technology faculty, University Putra Malaysia (UPM). Her current research interests include natural language processing, data mining, text mining, sentiment mining and information systems and their application as interdisciplinary studies.

LINGGES ARULSAMY is currently pursuing dual degree in BSc (Hons) Information Technology (Computer Networking and Security) from Sunway university and Lancaster university. He is an active researcher in the area of artificial Intelligence in Agriculture, Crypto-Ransomware Anomaly Classification, Encryption Through Pangram, Smart Light Using Electromagnetic Wave, and Cybersecurity.