Queryable Semantics to Detect Cyber-Attacks: A Flow-Based Detection Approach

Ahmed F. AlEroud and George Karabatis, *Member, IEEE*

***Abstract*—Cyber-attacks continue to increase worldwide, leading to significant loss or misuse of information assets. Most of the existing intrusion detection systems rely on per-packet inspection, a resource consuming task in today's high speed networks. A recent trend is to analyze netflows (or simply flows) instead of packets, a technique performed at a relative low level leading to high false alarm rates. Since analyzing raw data extracted from flows lacks the semantic information needed to discover attacks, a novel approach is introduced, which uses contextual information to automatically identify and query possible semantic links between different types of suspicious activities extracted from flows. Time, location, and other contextual information mined from flows is applied to generate semantic links among alerts raised in response to suspicious flows. These semantic links are identified through an inference process on probabilistic semantic link networks (SLNs), which receive an initial prediction from a classifier that analyzes incoming flows. The SLNs are then queried at run-time to retrieve other relevant predictions. We show that our approach can be extended to detect unknown attacks in flows as variations of known attacks. An extensive validation of our approach has been performed with a prototype system on several benchmark datasets yielding very promising results in detecting both known and unknown attacks.**

***Index Terms*—Context, information security, intrusion detection, netflow, network attacks, semantic link network (SLN).**

I. Introduction

MALICIOUS cyber-attacks may cause significant destruction whether they originate from a person, a group, or a country. Whereas it is theoretically possible to combat all types of cyber-attacks, most of the existing techniques provide reactive rather than proactive solutions to identify these attacks. The objective of proactive techniques is to eradicate the vulnerabilities in computer systems, a practically impossible task. Over the past decades, intrusion detection systems (IDSs) have been employed as one of the major reactive techniques against cyber-attacks. IDSs utilize logic operations, statistical, and data mining techniques to identify intrusions. Although existing IDSs have made a positive contribution to detection of attacks, they still have several significant limitations. 1) They analyze solely raw data, which is clearly insufficient; data needs to be analyzed at several layers. The data analyzed at the lowest layer overburdens cybersecurity systems, overwhelms human decision makers, and may not contain enough evidence about the intentions of an attacker. 2) Existing IDSs do not have the capability to analyze information in a relational manner. Primarily, information about the relationships of events is not available at prediction time [1]. Collectively integrating the relationships of events among themselves is very important in identifying relevant events that occur in similar contexts [2]. In general, events that target a system are not independent. Usually, a sequence of events initiated by an intruder has a specific objective; therefore, the discovery of relationships between these events can help in predicting cyber-attacks at an early stage. 3) The majority of existing IDSs perform deep-packet inspection—a fairly involved and resource-consuming task in today's high-speed Gigabit networks which carry vast volumes of network traffic [3].

A recent trend is to analyze netflows instead of packet payload. A (net) flow is a set of packets with common features such as source and destination IP addresses, ports, protocol interface, and service. One may think of a flow as metadata of a phone call: who called whom, when, etc., but without the conversation [4]. However, research in flow-based intrusion detection is criticized due to the limited amount of information a flow carries, which may not be adequate to predict attacks [5]. Therefore, a novel idea to overcome flow limitations is to consider the contextual factors that connect one suspicious flow to another such as their source and targets along with the time they occur. In essence, contextual information can be used to create attack prediction models. Intelligent techniques can take advantage of contextual information that is mined from past flows to assist in detecting attacks of incoming flows. Contextual semantic relations can be generated using extractable features of past suspicious flows such as: 1) the location targeted by suspicious flows; 2) the time and duration of suspicious flows; and 3) other features (e.g., TCP flags).

In this paper, we describe an intrusion detection approach that takes advantage of contextual information to identify relationships between suspicious activities discovered in flows. Such relationships are used to generate semantic link networks (SLNs) consisting of suspicious and benign activities

Manuscript received February 22, 2016; revised April 17, 2016; accepted July 27, 2016. This work was supported by the State of Maryland, TEDCO (MII) under Grant 01140-002. This paper was recommended by Associate Editor J. Lu.

A. F. AlEroud is with the Department of Computer Information Systems, Yarmouk University, Irbid 21163, Jordan (e-mail: ahmed.aleroud@yu.edu.jo).

G. Karabatis is with the Department of Information Systems, University of Maryland at Baltimore County, Baltimore, MD 21250 USA (e-mail: georgek@umbc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2016.2600405

(represented as nodes in SLNs). Run-time incoming flows are passed through a classifier to produce an initial prediction as a potential suspicious node in the SLN. Given this initial prediction, the preidentified semantic links on SLNs are queried to produce additional semantically relevant nodes that may be part of several related suspicious activities. After augmenting the initial prediction with semantically related ones based on the links in SLNs, feature-based profiles for benign activities are applied as prediction filters (PFs) to exclude "out-of-context" predictions, thus minimizing the side effects of the performed augmentation. We show that our approach can be extended to detect unknown attacks in flows as variations of known attacks.

The remainder of this paper is organized as follows. Section II outlines the motivations and contributions of utilizing contextual semantic relations in flow-based intrusion detection. Section III presents a background on flow-based intrusion detection. Section IV presents a theory about creating SLNs using contextual features. Section V discusses the process of constructing and using SLNs to detect cyber-attacks. Section VI discusses the evaluation of the proposed approach and describes the results. Section VII presents our conclusions and discusses future work.

II. Challenges, Motivations, and Contributions

Flows carry information only related to the features of network traffic, consequently, typical data mining techniques to identify cyber-attacks in flows may lead to a high false positive rate. One possible path to improve the effectiveness of detecting attacks from incoming flows is to examine the correlation between nominal (flags, protocol, service, etc.) and temporal features of past flows, and use it to predict attacks. Such correlation is still not sufficient to effectively detect attacks since there are several unknown relations among suspicious flows that attackers might exploit to execute attacks [6], [7]. Such relations are a key to identifying suspicious activities, however, they are usually identified by domain experts and/or described in ontologies; still, this process is manual, daunting, and inaccurate due to the vast number of semantic relations that need to be investigated. We are interested in an automatic creation of queryable semantic relations to identify attacks from flows. The following motivating example shows the importance of semantic relations for attack identification.

A. Motivating Example

One popular category of attacks is that of secure shell (SSH) daemons, where an attacker can gain access and potentially control a remote host. Once the host is compromised, the attacker uses it for scanning other systems. Whereas conventional intrusion detection techniques might be able to detect this attack, the context under which SSH attacks initiate cannot be easily bounded. For example, an attacker's intention may be to compromise Web servers to build SSH brute force botnet. This form of attack has been described by security experts as follows.

"There are strong indications that unidentified hackers are currently building a botnet, possibly by exploiting a vulnerability in outdated phpMyAdmin installations, and are using it to launch SSH brute force attacks" [8]. Fig. 1 shows sample log entries in alert logs corresponding to suspicious flows in a labeled dataset. The first log entry describes an attempt to compromise *phpMyAdmin* application on a specific server. The majority of such attacks come from a network of infected servers and target a wide range of open source PHP applications or modules. The second entry describes a successful brute force attempt on the same server. It is quite challenging to discover the relationships between these two attempts using the existing anomaly-based techniques that analyze the features of their corresponding flows. Alternatively, based on time, location, and other features of the initial attempt to compromise the *phpMyAdmin* and the brute force attempt, a security specialist may infer a relationship between these two activities through manual investigation. However, it may take quite some time to elicit, document, and use such knowledge. We propose an approach driven by database and graph mining techniques to automatically identify and query possible semantic links between different types of suspicious activities. Our approach makes the following contributions.

```
Sep 28 07:21:40 GET/administrator/phpMyAdmin-2.6.1-
pl1/main.php HTTP/1.0 XXX.XXX.XX.X
Sep 28 07:21:44 sshd[12556]: Accepted password for mySql
from XXX.XXX.XX.X port 52834 ssh2
```

Fig. 1. Alert log showing alerts raised in response to suspicious flows.

1) It improves the detection rate (DR) of cyber-attacks by analyzing flows. Contrary to other statistical intrusion detection models, SLN theory, and schema are utilized; reasoning on SLNs using semantic information of related attacks leads to quite satisfactory DRs. The objective is to prove that SLNs is the main reason of the enhancement in the values of precision (PR), DR, and F-score. Similar to some search engines, the proposed approach relies on query expansion followed by context-based filtering to handle an information security problem. To the best of our knowledge, this novel idea has not been utilized before.
2) It detects multistep attacks from sequences of flows by efficiently querying the relations produced via reasoning on SLNs and connecting one prediction to another based on several characteristics of flows such as source, target, and time of occurrence.
3) It alleviates the manual and daunting process of making decisions about possible semantic relationships between security incidents. Instead, it automates it by utilizing an inference process to generate these relationships based on time and location contextual features of the flows and the corresponding security alerts.
4) It achieves a good DR of unknown attacks using profile similarity as an indicator of the probability of an unknown attack. By creating feature-based profiles for known attacks our approach is able to predict the possible paths of unknown attacks. In addition, attack profiles created as linear functions lead to a satisfactory DR of unknown attacks in flows.

5) It reveals several theoretical properties using graph, probability, and information theories. It also validates the feasibility of contextual information fusion to create semantic links among security events.

III. Background and Related Work

In this section, we discuss related work on flow-based intrusion detection, SLNs, and context.

A. Flow-Based Intrusion Detection

Flow-based IDSs investigate the content of flows to detect attacks complementing the typical packet inspection intrusion detection approach [3], [9]. A flow has been defined as a set of unidirectional network packets sharing certain characteristics [10]. The definitions in [11] and [12] focus on time as an important aspect in flows.

Since the content of packets is absent in flows, attack identification approaches can only identify some categories of attacks, such as, denial of service (DOS) [12], [13], scanning [14], [15], worms [16]–[18], botnets [19], [20], and brute force attacks [21]. Several approaches utilize data mining techniques to identify attacks from flows such as hidden Markov models [6], and support vector machines (SVM) [10] to detect brute force attacks, entropy [22] to identify attacks as anomalies, gravitational search-based optimized neural networks [23], principal component analysis [24], and nonlinear regression [25]. The hidden Markov model and entropy approaches focus only on traffic distribution and temporal relations in order to identify anomaly windows. We argue that such approaches ignore other forms of relationships (e.g., the contextual features of alerts raised in response to flows). The one-class (SVM) approach [10] is a conventional flow-classification model that does not utilize contextual relations in the proposed detection technique. Recently, there have been few techniques that focus on designing multiagent systems to develop flow-based attack detection techniques. For instance, Hancock and Lamont [26] proposed an approach to dynamically discover and select the right nodes, among several others, to classify incoming flows. The technique utilized is still a multiclassifier system, focusing on raw features to discover attacks. Thus, it has the same limitations of other classification techniques.

Modern approaches focus on identifying sets of flow features which help in efficiently and accurately identifying attacks by combining the results of several feature selection techniques and then use feature support to identify a subset of features that cover optimality [27]. Fontugne *et al.* [28] introduced Hashdoop, a framework that splits traffic using hash functions to preserve traffic structures, using big data infrastructure to detect network anomalies.

There are previous works that utilize flow analysis to discover unknown attacks. François *et al.* [29] introduced an approach for the detection of large-scale anomalies or malicious events in flow records. This approach allows Internet operators to detect large-scale distributed attacks. The approach utilizes spatial-temporal flow record aggregation and applies entropic measures of traffic to discover known and unknown anomaly patterns. Choi *et al.* [30] presented a parallel coordinate attack visualization technique for detecting unknown large-scale Internet attacks including Internet worms, DOS attacks, and network scanning activities.

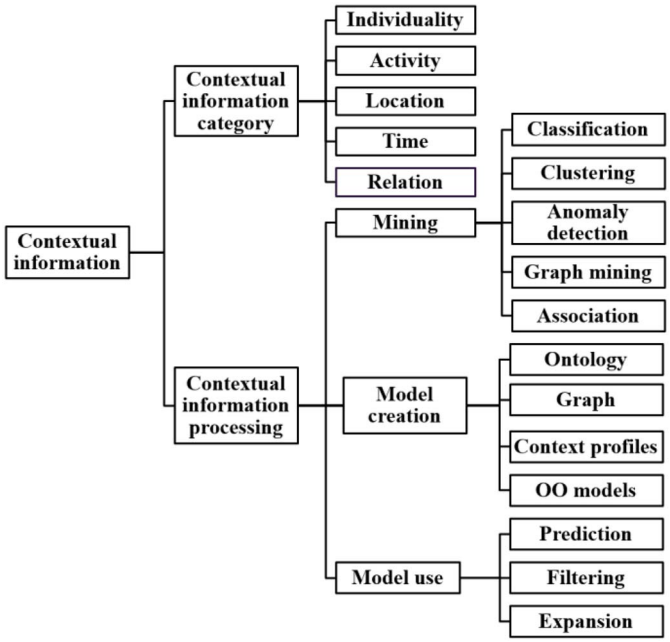

Fig. 2. Taxonomy for data mining-based intrusion detection techniques that utilize contextual information.

B. Semantic Link Networks and Context

An SLN is a graph with nodes and edges, which is used to infer semantic links [31], [32]. A semantic link can be one of several types such as cause-effect link, implication link, subtype link, similar-to link, instance link, sequential link, and reference link [33]. Reasoning on an SLN is to derive the feasible relations between two nodes on a series of semantic links. SLNs have been utilized in several application domains such as software engineering [34], environmental research [35], citation networks, [36] and community detection [32]. Groups of nodes in SLNs have common characteristics including context. According to Brown *et al.* [37], context characterizes the environment of an object; it is a dynamic grouping mechanism that encloses all information related to a particular situation, including the time and order of events that target an entity, the location of an entity, the events that target it, and its relationship to other entities [38]. The main elements of contextual information fall into five categories: 1) activity; 2) location; 3) time; 4) relation; and 5) individuality [38]. Those categories are also presented in our taxonomy in Fig. 2 [39]. First, the location reveals the physical or logical information about location (e.g., IP addresses and port numbers). The IDS has to be aware of the location of victims and attackers.

Second, the IDS must be aware of time which refers to the time of events that target a particular entity. Third, activity describes events that are applicable to the system. The information in this category covers all events that occur during the system execution time (e.g., ping commands). As such, the set of activities that target the system can lead to one or

more attacks. Fourth, the relation category is important to identify dependencies between multiple events (e.g., relationships between two alerts raised by an IDS). The relation category is expressed over other categories such as time, location, and activity. Fifth, the environmental characteristics of computing entities are captured through the individuality category. For instance, the current characteristics of computer systems, applications, and patches applied are considered significant to assess the impact of the activities in progress on the targeted system. Therefore, some suspicious events are deemed as nonrelevant when the system is patched against them.

Several approaches use contextual information on security exploits to detect attack scenarios using contextual relationships [40]. Classification [41], clustering [42], [43], anomaly detection [44], association [45], and graph mining [46] are the main techniques that are utilized for mining contextual information. Ontologies [47], graphs [48], context profiles [49], and object-oriented models are the major techniques used for modeling contextual information. Our taxonomy reveals that attack graphs have been mainly utilized to overcome the limitations of several existing intrusion detection techniques [50]–[52]. Attack graphs model the vulnerabilities of the systems and their possible exploits. The dependency between nodes identifies relationships between attacks, hosts, exploits, network events, etc. These relationships describe potential attack scenarios. At run time, signature-based IDSs correlate these scenarios with run-time events.

Whereas attack graphs model the relationships between exploits, they are not efficient during system execution time. They require traversing large number of graph paths in order to perform correlation, which is computationally expensive. Moreover, domain and background knowledge incorporated in graphs needs to be collected or added by domain experts. For instance, in order to be aware of the latest attacks, machine exploits, and vulnerability information needs to be updated on a regular basis. Several approaches focus on correlating the IDS alarms with the vulnerabilities discovered using vulnerability detection systems. Despite the adoption of correlation, the problem of false positives generated by IDS is still on the rise due to their inability to profile correct and current vulnerability information [53]. Silveira *et al.* [54] proposed an anomaly detection technique that identifies strongly correlated flow changes to reveal anomalies (e.g., scanning and DDoS attacks, link outages, and routing shifts). Compared to this approach our SLN-based method contains semantic inference procedures to perform correlation between flows and the corresponding suspicious activities. While contextual information has been mainly utilized for attack prediction [55], [56], existing intrusion detection techniques do not apply semantic inference to automatically generate contextual relationships among security incidents.

IV. Theoretical Model for Creating Queryable Semantics Using Contextual Features

Table I shows the terminology used in this paper. Although there exists several techniques to automatically identify semantic links, we are not aware of any work that utilizes contextual features such as time and location and measures their effect to infer semantic links. The preliminary version of SLNs is modeled using a schema. An SLN schema is defined as follows [32].

TABLE I
Terminology Used in This Paper

Symbol	Description
m	An attack classification model created using flows in a raw dataset D.
NS, NB	Actual suspicious/benign activities in D
$N = \{n_1, \dots, n_p\}$	Types of nodes in D / types of known attacks and benign activities/ Consequences/alerts/ predictions at run time.
$fl \subseteq h_s \times h_d$	A flow established between a source (h_s) and destination host (h_d).
$F = \{fl_1, \dots, fl_k\}$	A set of incoming flows
$R = \{n_1, \dots, n_k\}$	Given a set of incoming flows $FL = \{fl_1, \dots, fl_k\}$, nodes in R are predicted by $m\|(fl_i, n_i) \in (S \text{ or } B) S \in$ suspicious flows, $B \in$ benign flows.
SLN	A graph with nodes and weighted edges
$n_i \overset{\alpha}{\rightarrow} n_j$	A semantic link that identifies a possible relationship between nodes n_i, n_j and α represents a numerical weight on that link.
$rs(n_i \rightarrow n_j)$	A metric used to measure the weight of a contextual relationship between any two nodes n_i, n_j connected via a path on an SLN.
C	Context.
$H(N'\|f)$	Conditional entropy for a subset of nodes N' conditioned on a feature f.
$H(n_i\|f)$	Conditional entropy for node n_i conditioned on a feature f.
$Pr(n_i\|f_j)$	Conditional probability of n_i given f_j which is one possible value for the feature f.
$\vec{V}_{n_i} = [f_1{:}\, d_i, \dots, f_m{:}\, d_j]$	A feature vector for node n_i, and d_j is the type of that feature (e.g. time, location, numerical, descriptive).

Definition 1 (SLN Schema): The SLN schema is a triple denoted by *<Nodes, SemanticLinks, Rules>*.

A *Node* is an object type denoted by n_i and its characteristics are represented using a vector $\vec{V}_{n_i} = [f_1 : d_j, \dots, f_m : d_y]$, where f_i is a feature of node n_i and d_j is the data type of that particular feature.

A *SemanticLink* is a *node* $\times$ *node* relation. Each sematic link l_i is represented as $l_i : n_i \overset{\alpha}{\rightarrow} n_j$. Each link identifies a possible semantic relation between nodes n_i, n_j, where α represents a numerical weight on that link.

A *Rule* is a reasoning mechanism on semantic links. A rule is denoted by $n_i \overset{\alpha}{\rightarrow} n_j,\ n_j \overset{\beta}{\rightarrow} n_r \Rightarrow n_i \overset{\gamma}{\rightarrow} n_r\ |\alpha, \beta, \gamma$ are weights on semantic links and $\alpha \cdot \beta \Rightarrow \gamma$. Using this rule two semantic links can indirectly derive a new link. Each implication generated via reasoning can be assigned a certainty degree called relevance score rs. A relevance score can be described in a specific metric space to represent the confidence of an implication generated by semantic reasoning [34]. SLN is initially represented as a similarity relationship matrix (SRM).

Definition 2 (Similarity Relationship Matrix): SRM N is an adjacency matrix where the element n_{ij} represents the weight on the semantic link from node n_i to n_j and n_{ji} is the weight

on the reverse link from n_j to n_i. If there are no semantic links between n_i and n_j, $n_{ij} = n_{ji} = 0$.

In the SRM N, $n_{ij} \times n_{jr}$ means that the node n_i can reach n_r via semantic links in one reasoning step using two links $n_i \rightarrow n_j$ and $n_j \rightarrow n_r$. Reasoning steps can be performed by raising N to the power k (i.e., $N^{k+1} = N^k \times N$), where $n_{ir}^{(k+1)}$ means that node n_i can reach node n_r in $k+1$ steps. The number of reasoning steps in an SLN is determined by $|N|-1$, where $|N|$ is the number of nodes [32]. Reasoning can derive implicit semantic links between nodes through addition and multiplication operations on a node to node relationship matrix using reasoning rules. Each node n_i in SLN is relevant to at least one specific contextual situation, identified by context C defined as follows.

Definition 3 (Context C): The context C is a combination of features $[f_1 : d_i, \ldots, f_m : d_j]$ that identify the settings or preconditions under which one or more consequences $N' = \{n_1, \ldots, n_k\}$ are likely to occur in a specific situation$|\ N' \subseteq N$, where N is the set of all possible consequences and $k < p|p$ is their number.

Based on the definition of context, there is a cause-effect relation among the features that characterize the context C and the corresponding consequences. In general, the features identifying context consequences are: *numerical*, *descriptive* and *time/location*-based features. The former two are used to create prediction models that distinguish among context consequences, such as predicting the type of a suspicious network based on the network traffic features (e.g., source bytes). Additionally, they can be used to identify relations among several consequences that are possible in a specific context. The latter (time- and location-based features) are dynamic in nature and cannot be used as prediction features (e.g., predicting suspicious activities based on the time of the day), but they are utilized to identify relationships among context consequences (e.g., the co-occurrence of two suspicious activities at several time windows). Since each node n_i in SLN is associated with one or more contexts, it represents one possible consequence of context. In general, nodes with common context share several characteristics including semantic closeness. Proposition 1 describes this relationship between the context associated with the nodes and their semantic closeness.

Proposition 1. The strength of semantic links between any two nodes n_i, n_j calculated via semantic reasoning on a set of paths $t_1, \ldots, t_m$ connecting n_i, n_j is affected by the context of the nodes along each of these paths.

Let c_1 and c_2 be two preidentified contexts where a specific feature f has weights H_1, H_2 calculated using conditional entropy—an information theory measure. The features of each context enable the preconditions that trigger one or more context consequences. H_1, H_2 identify the significance of feature f in distinguishing between consequences. Using conditional entropy as a measure for feature ranking, $H_1 \neq H_2$ implies that feature f occurs in both contexts with different probabilities, that is, the probability of occurrence of f in the consequences of c_1 differs from its probability in the consequences of c_2. According to the information theory measures introduced by Shannon and the simplest emerge principle (SEP) introduced in [32], *"the more stable entropy a path (that connects nodes) has, the less information it contains; therefore its semantics can be easily understood."*

Let us further assume that the feature f is used to discriminate among contexts and let each consequence be a node on a path. Let $N^{'} = \{n_1, n_2, n_3, n_4\}$ be a set of four nodes in a specific SLN; t_1, t_2 are two paths on that SLN, $t_1 : n_1 \xrightarrow{\alpha} n_2 \xrightarrow{\gamma} n_4$, $t_2 : n_1 \xrightarrow{\beta} n_3 \xrightarrow{\delta} n_4$, where $\alpha, \beta, \gamma, \delta$ represent the weights on links between $n_1 \rightarrow n_2, n_1 \rightarrow n_3, n_2 \rightarrow n_4$ and $n_3 \rightarrow n_4$, respectively. The weight of feature f in all nodes that belong to N' can be calculated using conditional entropy as

$$H(N'|f) = -\sum_{i=1}^{|N'|}\sum_{j=1}^{k} P(n_i, f_j)\log_2 P(n_i|f_j). \tag{1}$$

The conditional entropy H for a set of nodes $N'|N' \subseteq N$ given a feature f is the sum of entropy when all nodes that belong to N' are conditioned on f and it is calculated over all possible k values of f; where $P(n_i, f_j)$ is the joint probability of node n_i and f_j, (one possible value of f) and $Pr(n_i|f_j)$ is the conditional probability of n_i given f_j.

However, if

$$H_1 = H(n_1|f), H_2 = H(n_2|f), H_3 = H(n_3|f), H_4 = H(n_4|f)$$

$$\text{Then using SEP: } |H_1 + H_2| > |H_1 + H_3| \Rightarrow \beta < \alpha \tag{2}$$

$$|H_2 + H_4| > |H_3 + H_4| \Rightarrow \delta < \gamma. \tag{3}$$

Based on the entropy and probability relation [57], the expression on the left of the implication (2) is true *iff* the co-occurrence frequency of n_1 and n_2 when f is observed is greater than the co-occurrence frequency of n_1 and n_3 when f is observed, i.e., $Pr(n_1, n_2|f) > Pr(n_1, n_3|f)$. Implication (3) is true *iff* $(n_2, n_4|f) > r(n_3, n_4|f)$. Given these probabilities, the following implications are also true:

$$Pr(n_1, n_2|f) > Pr(n_1, n_3|f) \Rightarrow Pr(n_1 \rightarrow n_2|f) > Pr(n_1 \rightarrow n_3|f) \tag{4}$$

$$Pr(n_2, n_4|f) > Pr(n_3, n_4|f) \Rightarrow Pr(n_2 \rightarrow n_4|f) > Pr(n_3 \rightarrow n_4|f). \tag{5}$$

However, there are two paths between n_1 and n_4, $t_1 : n_1 \xrightarrow{\alpha} n_2 \xrightarrow{\gamma} n_4$ and $t_2 : n_1 \xrightarrow{\beta} n_3 \xrightarrow{\delta} n_4$. Based on inequalities (4) and (5), $\alpha \cdot \gamma > \beta \cdot \delta$, if the objective of a random walker is to identify the most feasible semantic link between n_1 and n_4, the path t_1 which contains the nodes that are closer in context is selected to identify such link, therefore, the above proposition holds.

V. Problem Formulation and Research Approach

Using the formal model above, we plan to: 1) improve the accuracy of flow-based attack prediction techniques using preidentified semantic links between suspicious nodes and 2) detect related suspicious activities given a sequence of flows. Throughout this paper the terms alert, suspicious activity, suspicious/benign node, label, consequence, and prediction are used interchangeably. Formally, let $\vec{V} =< f_1 : d_i, \ldots, f_n : d_j >$ be a feature vector extracted from a raw dataset D that consists of labeled flows. Let m be an attack

classification model created using the flows in D. Suppose that the role of m is to investigate the features of an incoming flow fl_i to produce an initial prediction n_i. Let $N = \{n_1, \ldots, n_p\}$ be the set of all possible types of attacks and benign activities in past flows. N represents all possible attacks and benign activities that occur in different contexts. Let $R = \{n_1, \ldots, n_k\}$ be the set of the predictions produced by the model m for a set of incoming flows FL $= \{fl_1, \ldots, fl_k\}|\ (fl_i, n_i) \in (SorB)$, where S and B are the sets of suspicious and benign flows, respectively. Let a multistep attack AS consist of a sequence of flows FL$' = \{fl_1, \ldots, fl_r\}|FL' \subseteq$ FL. Let $R' \subseteq R$ be the set of suspicious flows and the corresponding predictions produced by m for the flows in FL$'$. The predictions produced by m are therefore classified into the sets S, B, R'.

1) $(fl_i, n_i) \in S$ if fl_i is predicted as a suspicious activity.
2) $(fl_i, n_i) \in B$ if fl_i is predicted as a benign activity.
3) $(fl_{i'}, n_i) \in R'$ if $fl_{i'}$ is predicted as part of AS.

Let NS, NB, and NA be the sets that contain the flows in FL and their actual label.

1) NS is the set of actual suspicious flows in FL.
2) NB is the set of actual benign flows in FL.
3) NA is the set of actual suspicious flows in a multistep attack AS.
4) ¬ NA is the set of suspicious flows that are not part of the multistep attack AS.

Let rs$(n_i \rightarrow n_j)$ be a metric used to describe the strength of semantic relationship between any two nodes n_i, n_j based on their co-occurrence with the contextual features $<f_1 : d_i, \ldots, f_m : d_j>$. Given the values of rs, the objective is to expand the initial prediction n_i made to each fl_i using the prediction model m in order to find other relevant predictions so that the following objective is optimized:

$$\max_{fl_1,\ldots,fl_k} \{\text{obj}\}|\text{rs}(n_i \rightarrow n_j)\forall(fl_i, n_i) \in (S, B, R'), n_j \in N.$$

In particular, the objective (obj) is to optimize all of the following effectiveness measures.

1) Correct prediction of individual suspicious flows

$$\max_{fl_1,\ldots,fl_k} \{S \cap NS\}|\text{rs}(n_i \rightarrow n_j)\forall(fl_i, n_i) \in S, n_j \in N.$$

2) Correct prediction of benign flows

$$\max_{fl_1,\ldots,fl_k} \{B \cap NB\}|\text{rs}(n_i \rightarrow n_j)\forall(fl_i, n_i) \in B, n_j \in N.$$

3) Correct prediction of attack steps

$$\max_{fl_1,\ldots,fl_r} \{R' \cap NA\}|\text{rs}(n_i \rightarrow n_j)\forall(fl_i, n_i) \in R', n_j \in N.$$

The purpose of the first two measures is to increase the DR of suspicious flows and minimize the rate of false positives. The last one aims to maximize the DR of suspicious flows that actually belong to the multistep attack AS. Since R' can include large number of predictions, another objective function is to discard the predictions that are not part of the multistep attack AS as follows.

4) Incorrect prediction of attack steps

$$\min_{fl_1,\ldots,fl_r} \{R' \cap \neg NA\}|\text{rs}(n_i \rightarrow n_j)\forall(fl_{i'}, n_i) \in R', n_j \in N.$$

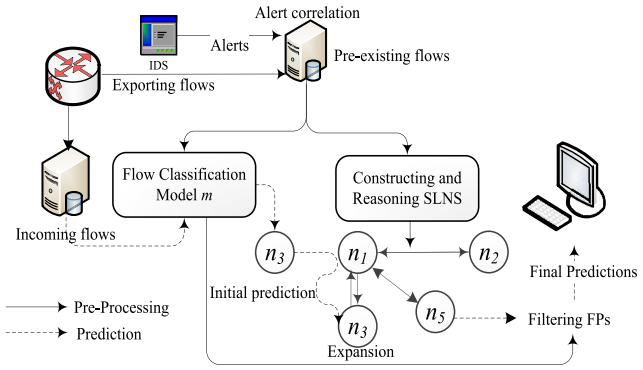

Fig. 3. Overview of the research approach.

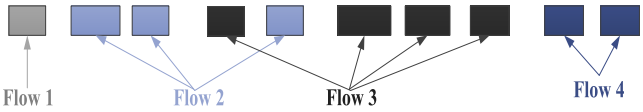

Fig. 4. Packets aggregation to create flows.

Our approach (shown in Fig. 3) is divided into two separate phases.

1) The *preprocessing phase* which utilizes pre-existing (past) flows to create a flow classification model, the SLNs, and other structures that will be used at run-time. One may think of this phase as calibration of the system before it becomes operational.
2) The *prediction phase* which occurs at run-time and it decides whether incoming flows are suspicious or benign based on the structures that were created in the preprocessing phase.

The rationale behind this separation into two phases is to isolate the resource and time consuming preprocessing phase (which occurs infrequently) from the prediction phase at run-time. This separation into two phases drastically improves the performance of the entire system.

The next sections discuss the following in detail.

1) The preprocessing phase which consists of: a) flow collection and alert correlation; b) constructing; and c) reasoning SLNs.
2) The prediction phase which consists of: a) flow classification; b) prediction expansion using SLNs; and c) discarding inaccurate predictions using PF.

A. Flow Collection and Alert Correlation

There are several mechanisms to generate flows. Each flow contains common features, e.g., same source and destination IPs, same protocol, etc. Fig. 4 shows an example of packets that are aggregated into four flows based on such features.

Packets that are close to each other in time and destined to same location are aggregated into a single flow. The flows are stored in a specific format for further monitoring and/or analysis. For the discussion in this paper a flow has the following structure:

$$fl = (I_{\text{src}}, I_{\text{dst}}, P_{\text{src}}, P_{\text{dst}}, \text{Prot}, Pckts, \text{Octs}, T_{\text{start}}, T_{\text{end}}, \text{Flags})$$

where the features I_{src} and I_{dst} represent the source and destination *IP* addresses. The features P_{src} and P_{dst} represent the source and destination ports. The features Pckts and Octs represent the count of packets and octets in the flow. The feature Flags represents the header flags. The start and end time of the flow is represented by T_{start} and T_{end}.

Data about alerts raised by IDSs in response to flows is extracted from IDS log files and then those alerts are correlated with the corresponding flows (more on this correlation approach is discussed in [4]). Data about those alerts includes, the timestamp of alert, the alert description (text) in natural language and its category which identifies the type of security incident (e.g., SSH suspicious connection). The flows correlated with alerts are used to create SLNs.

B. Constructing and Reasoning SLNs

The SLNs are constructed in two steps: 1) the creation of weighted links among nodes and 2) reasoning on such links to augment their semantics. Note that SLNs include both suspicious and benign activity nodes. Although benign and suspicious nodes have few common features, it is anticipated that semantic reasoning produces weak relationships between benign and suspicious nodes in SLNs. The two steps to construct SLNs are discussed below.

1) Creating Links Using Feature Similarity: The similarity among nodes is a measure of their co-occurrence. There are three categories of contextual features that are utilized to calculate similarity in our approach. *Time/location*, *numerical*, and *descriptive* features. *Time*-based features are the *timestamps* of alerts, the T_{start}, T_{end} of the flows, and the *duration* of those flows. *Location*-based features are the source and destination IPs and port numbers (I_{src}, I_{dst}, P_{src}, P_{dst}). Those features indicate relations among nodes in regards to source and target of activities. *Numerical* features identify traffic statistics such as the number of packets, octets (Pckts, Octs). *Descriptive* or *nominal* features describe other flow characteristics such as the flags, protocol type (Prot, Flags) and *alert description*. Some feature types are preprocessed before they are utilized for similarity calculation. For instance, binning is performed on numerical, time- and location-based features. After the stop-words are removed, the keywords of alert description are extracted and each word is treated as a feature. In addition to time and location features, the description of each alert is considered a significant factor to discover the actual semantic links between nodes based on their textual description. Mining semantic relationships based on the description of alerts reveals new knowledge that cannot be discovered by analyzing only the traffic features of the flows. After preprocessing of features, a global node-feature matrix F is created. It consists of all extracted features as rows, the node types as columns, and the normalized frequency of each feature f with each node n_i as a weight of f in that node. In addition, a feature vector V_{ni} is generated for each node type n_i.

The process of semantic reasoning starts with assigning initial weights on semantic links. These weights are calculated based on the similarity values between nodes on time, location, numerical, and/or descriptive features. *Pearson's* correlation and *Anderberg* similarity coefficients are the measures that calculate similarity between nodes. According to Weller-Fahy *et al*. [58] "the body of intrusion detection research has grown extensively but the knowledge of the utility of similarity measures within the field has not grown correspondingly." Most researchers did not evaluate more than one similarity measure at the detection phase. As such, one of the main recommendations in [58] is to "develop techniques to compare graph similarity matrices, and determine which thresholds are most useful under certain conditions." We utilize two different categories of similarity coefficients since each one generates slightly different SLNs. Pearson's correlation has been used in intrusion detection research [59]–[61]. It measures the linear association between two nodes n_i and n_j and is defined as $\text{cov}(\sigma_{V_{ni}}, \sigma_{V_{nj}})/\sigma_{V_{ni}} \times \sigma_{V_{nj}}$, where cov is the covariance of their corresponding feature vectors and $\sigma_{V_{ni}}$ is the standard deviation of entries in vector V_{ni}. The Pearson's correlation values are converted to values between [0-1]. The Anderberg coefficient works on binary feature vectors [62] and yields similarity values between 0 and 1. The weighted feature vector V_{n_i} for each node is converted into a binary vector by applying the relative cutoff data transformation technique [63]. Given two nodes, n_i and n_j with binary features, the Anderberg coefficient measures the overlap among the features of n_i and n_j. Each feature of n_i and n_j can be either 0 or 1 referring to the occurrence or absence of that feature.

The total number of matches and mismatches between each pair of nodes n_i and n_j is specified as follows [39]. Suppose y_{11} represents the total number of features where V_{ni} and V_{nj} both have a value of 1, y_{01} represents the total number of features where V_{ni} is 0 and V_{nj} is 1, y_{10}represents the total number of features where V_{ni} is 1 and V_{n_j} is 0, y_{00} represents the total number of features where V_{ni} and V_{nj} both have a value of 0. Thus, the value of Anderberg similarity is calculated as

$$AD_{(n_i,n_j)} = \frac{y_{11}}{y_{11} + 2(y_{10} + y_{01})}. \tag{6}$$

After the similarity values are calculated, an SRM N is generated. The matrix N expresses links among, for example five nodes $n_1, \ldots, n_5$. The numbers represent the weights of direct links (i.e., similarity values) between nodes. According to the definition of the transition matrices [64], it is necessary to normalize the transition probabilities represented by link weights to convert the matrix into a right stochastic matrix. The rows of matrix N are normalized as $n_{i,j} = (n_{i,j}/\sum_{m=1}^{p} n_{i,m})$. An initial SLN is then created with weights on edges as shown in Fig. 5

$$N = \begin{array}{c} \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{array} \begin{array}{c} \begin{array}{ccccc} n_1 & n_2 & n_3 & n_4 & n_5 \end{array} \\ \begin{bmatrix} 0 & 0.6 & 0.5 & 0 & 0.1 \\ 0.6 & 0 & 0 & 0.6 & 0.7 \\ 0.5 & 0 & 0 & 0.3 & 0 \\ 0 & 0.6 & 0.3 & 0 & 0 \\ 0.1 & 0.7 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

Algorithm 1 summarizes the steps required to create the initial SLN after normalizing the similarity values. Lines 1–8

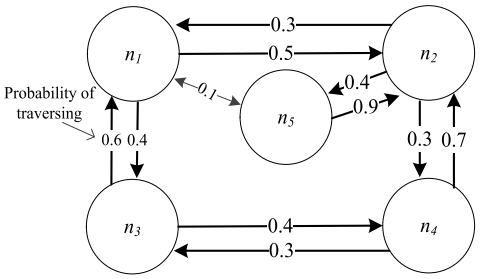

Fig. 5. Initial semantic link network (SLN) with nodes and edges.

Algorithm 1 Generation of the Initial SLN

Input: Node set $N = \{n_1, \ldots, n_p\}$
Feature vector $V_{ni} = [f_1, \ldots, f_a]$ $\forall n_i \in N$
Output: Initial SLN N

1: **Begin**
2: **For** $i = 1$ to $p - 1$ **do**
3: $deg_{n_i} = 0$
4: **For** $j = i + 1$ to p **do**
5: $SIM(n_i, n_j) = SIM(n_j, n_i) = SIM\ (V_{ni}, V_{nj})$
6: $deg_{n_i} = deg_{n_i} + SIM(n_i, n_j)$
7: **End For**
8: **End For**
9: **For** $i = 1$ to p **do**
10: **For** $j = 1$ to p **do**
11: $SIM(n_i, n_j) = \frac{SIM(n_i, n_j)}{deg_{n_i}}$
12: **If** $SIM(n_i, n_j) > \vartheta$ **then**
13: $N_{(n_i \to n_j)} = SIM(n_i, n_j)$
14: **Else**
15: $N_{(n_i \to n_j)} = 0$
16: **End If**
17: **End For**
18: **End For**
19: **Return** N
20: **End**

show how the similarity $\text{SIM}(n_i, n_j)$ between the node n_i and all other nodes is used to find the degree of n_i. Lines 9–19 summarize the process of normalizing similarity, assigning weights on edges and retrieving the initial SLN N. The user-defined threshold ϑ enforces minimum connectivity values between any two nodes.

2) Reasoning on Initial Semantic Links: A reasoning process is performed on the initial SLNs to discover the implicit relationships between pairs of nodes. The outcome of this reasoning process is the degree of relevance or the relevance score between nodes n_i and n_j, a metric that measures one or more types of semantic relations between these nodes (e.g., cause-effect, implication, and sequential), defined as [34].

Definition 4 (Relevance Score rs): If n_i and n_j are two nodes of an SLN N and there are m paths $t_1, \ldots, t_m$ between n_i and n_j where the path $t_l (1 \leq l \leq m)$ consists of nodes $n_{l_1}, \ldots, n_{l_{|t_l|+1}}$ ($|t_l|$ is the length of path t_l), the $\text{rs}_{(n_i \to n_j)}$ is defined as $\sum_{t_l} \prod_{1 \leq i \leq |t_l|} \text{SIM}(n_{l_i}, n_{l_{i+1}})$

The relevance score rs between n_i and n_j is calculated as the sum of the products of the similarity values on all paths of length $|t_l|$ connecting n_i and n_j. Suppose that we want to calculate rs between n_3 and other nodes in k reasoning steps

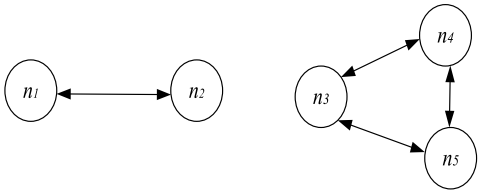

Fig. 6. An example of two disconnected initial semantic linked networks (SLNs).

where $k \leq N - 1$ using the matrix multiplication rules, and for any given pair (n_i, n_j) with $i \neq j$, the rs for all paths between n_i and n_j with length $|t_l|$ is equivalent to $N^{|t_l|}_{n_i \to n_j}$ (e.g., $\text{rs}_{(n_3 \to n_j)} \Leftrightarrow N^{|t_l|}_{n_3 \to n_j}$), where $N^{|t_l|}$ is the product of self-multiplying N, $|t_l|$ times. For example, the relevance scores between n_3 and other nodes $\text{rs}_{(n_3 \to n_j)}$ over all paths with lengths 2–5 (calculated using the weights shown in Fig. 5) are as follows:

$$
\begin{array}{lccccc}
 & n_1 & n_2 & n_3 & n_4 & n_5 \\
N^2_{n_3 \to n_j} = [& 0.00 & 0.58 & 0.36 & 0.00 & 0.06] \\
N^3_{n_3 \to n_j} = [& 0.39 & 0.05 & 0.00 & 0.31 & 0.23] \\
N^4_{n_3 \to n_j} = [& 0.04 & 0.63 & 0.25 & 0.02 & 0.06] \\
N^5_{n_3 \to n_j} = [& 0.35 & 0.09 & 0.02 & 0.30 & 0.26].
\end{array}
$$

Since several relevance scores are calculated based on the length of the path, we need the score that identifies the most feasible link between the corresponding nodes. According to [34], the maximum rs is the best indicator of such a link, hence it is selected to measure the strength of semantic links between them. For instance, the $\text{rs}_{(n_3 \to n_2)}$ is 0.63 (see the third vector $N^4_{n_3 \to n_j}$ above) and it is obtained after three reasoning steps ($N^4_{n_3 \to n_2}$) with path length = 4.

One problem that needs to be handled is the dangling nodes (nodes with no outgoing edges), in other words, SLNs with multiple connected components. For instance, in the two SLNs in Fig. 6, an attacker who starts at the connected component on the left-side cannot reach node 5 of the right-side since the nodes 1 and 2 have no links to reach node 5. In order to overcome this problem, we need a positive constant p between 0 and 1, which is called the damping factor (a typical value is 0.85). An attacker traverses from the current node and arbitrarily chooses a different node from the set of the remaining nodes to go to. The procedure of handling dangling nodes is part of the matrix creation step. As such, it is applied after the similarity matrix N is created and before the reasoning process starts, as follows.

1) Create a $n \times n$ matrix B with all elements equal to 1.
2) Use $\alpha = 0.85$ as a damping factor.
3) Construct a matrix $N = \alpha \times N + (1 - a) \times B/|N|$.

Finally, since the similarity-based network is restricted to the subjective aspects hidden in the dataset, the SLN is adjusted to generate another version using domain knowledge. Specifically, the relevance scores between nodes of the first SLN are updated using relationships between those nodes in existing taxonomies using our approach in [65].

C. Flow Classification and Prediction Expansion Using SLNs

During the prediction phase at run-time, incoming flows are analyzed and marked either as benign or suspicious. It starts by investigating the set of incoming flows FL $= \{fl_1, \ldots, fl_k\}$ to produce an initial prediction n_i for each flow. The objective of this step is to classify individual flows and identify suspicious activities by applying a rule-based classification model that is created using the ID3 decision tree algorithm [66]. The produced rule-based model is used at the beginning of the prediction phase during which the features of incoming flows are examined using the classification rules. If one of the rules is triggered, an initial suspicious node is selected; if no rule is triggered, a benign activity node is selected as an initial prediction. The initial prediction is passed to SLNs that expand it to include several additional related nodes $R = \{n_1, \ldots, n_m\}$. A flow can be predicted as a suspicious (that represents a step in a multistep attack) or a benign activity. During multistep attacks, several alerts are raised indicating a suspicious activity in the attack. SLNs identify the possible links between these nodes based on their relevance score rs to the initial prediction. A user defined threshold tr controls the scope of the expansion. For instance, if n_3 (in Fig. 5) is selected as an initial prediction for a specific flow fl, a threshold tr $= 0.6$ indicates the inclusion of n_2 as another prediction to flow fl since the rs($n_3 \rightarrow n_2$) equals 0.63 and it is greater than the threshold tr.

D. Discarding Inaccurate Predictions

Based on an initial prediction, the expanded set of predictions R that is generated for a specific flow fl_i may include both suspicious and benign activities. It is then necessary to discard possible inaccurate predictions (i.e., false positives or false negatives). Therefore, another rule-based classification model is created and used to examine flow features. Its main objective is to identify benign activities. Based on the distinct types of protocols found in the pre-existing flows, the data is divided into several disjoint splits that are trained separately. Each split consists of benign and suspicious flows that share the protocol features. The outcome is a set of rule-based profiles called PFs that describe different types of benign and suspicious activities. We only utilize profiles that describe benign activities for filtering. PFs are only applied to incoming flows for which the predictions of SLN include both suspicious and benign nodes. For any such flow under investigation, if a profile PF_i is triggered and the corresponding benign activity node is included in the prediction set R, all suspicious predictions made for that flow are discarded, and only the corresponding benign activity node is kept as a final prediction; the purpose of this step is to remove possible false positives. In contrast, if no profile is triggered, all benign predictions included for that flow are discarded and only the suspicious predictions are kept; the purpose of this step is to remove false negatives. The remaining predictions are expected to be part of semantically related activities or a possible multistep attack.

Algorithm 2 Analyzing Incoming Flows Using SLNs and PFs

Input: A set of incoming flows $FL = \{fl_1, \ldots fl_k\}$, A classification model m, $rs(n_i, n_j) \forall n_i, n_j \in N$, $PF = \{PF_{n1}, \ldots PF_{nu}\}$
Output: a set of one or more prediction(s) $R = \{n_{1,\ldots,}n_m\}$ for each flow fl_i

```
1: Begin
2:   For i = 1 to k do
3:     R = {∅}, SC=0, NC = 0
4:     Find an initial prediction n_i to fl_i using m
5:     R = {n_i}
6:     If n_i is suspicious then
7:       SC = AS + 1
8:     Else
9:       NC = NC + 1
10:    End If
11:    For j = 1 to pp=|N| do
12:      Retrieve rs(n_i, n_j)
13:      If rs(n_i → n_j) > tr then
14:        R= {R} ∪ n_j
15:        If n_j is suspicious then
16:          SC = SC + 1
17:        Else
18:          NC = NC + 1
19:        End If
20:      End If
21:    End For
22:    If NC ≥ 1 AND SC ≥ 1 then
23:      For s = 1 to u do
24:        If PF_ns match fl_i then
25:          setNC = 0
26:          Exit loop
27:        End If
28:      End For
29:      If NC = 0 then
30:        ∀n_j ∈ R|n_j is benign
31:        R= {R} − n_j
32:      Else
33:        ∀n_j ∈ R|n_j is suspicious
34:        R= {R} − n_j
35:      End If
36:    End If
37:    Return R
38:  End For
39: End
```

The attack detection steps are summarized in Algorithm 2. The input is a set of incoming flows FL. The output is one or more predictions in R for each flow based on the value of threshold tr. Lines 1–21 produce an initial prediction and expand it based on the value of tr. Lines 22–36 discard the inaccurate predictions of SLNs when the set of predictions R contains both suspicious and benign activity nodes [if the suspicious and benign activity counts in R are greater than 1 (NC ≥ 1 and SC ≥ 1)].

E. Complexity Analysis

As mentioned earlier, our approach consists of two phases, a preprocessing phase, which creates SLNs, and a prediction phase at run-time. Since there are new attacks that are discovered on a regular basis, SLNs need to be recreated. The complexity of the algorithm used to create the feature vectors for all nodes is $O(P|T|)$, where P is the number of nodes and $|T|$ is the size of the feature vector. The complexity of

the algorithm that calculates the similarity between nodes is $O(P^2|T|)$, where P is the number of nodes and $|T|$ is the size of the feature vector used for similarity calculation. We utilize the matrix power method on the similarity matrix to perform the reasoning step which retrieves the relevance scores with time complexity of $O(P^3)$.

Overall, three prediction models are utilized to investigate incoming flows: the classification model m for initial prediction, SLNs to expand such a prediction, and PFs to filter-out non relevant predictions. Let W be the number of distinct types of activities (labels) to be predicted by a classification model m. Let K be the number of flows to be analyzed. At the beginning it is required to scan the created classification rules to identify suspicious activity types; the complexity of such a scan is $O(KW)$, since each flow has initially one prediction, K also refers to the number of predictions (single prediction per each flow). The K predictions are expanded using the precalculated relevance score rs. The length of the expansion depends mainly on the value of rs and the number of semantic links each node has with other nodes. In the worst case, the number of semantic links is P (i.e., the node has relationships to all other nodes), therefore, the overall complexity of expansion is $O(KP)$.

The size of SLNs poses a minor effect on the performance of the entire system. First, SLNs are represented using the relational database model where each node is connected to at most $P-1$ nodes. In addition, the complexity of the algorithm used for scanning those relationships at run-time is $O(P)$. The modeling with database technology gives more flexibility to create the prediction models that work in a complementary manner. Without such a robust modeling technique, more complex correlation techniques would be needed to integrate evidence from different prediction models. Second, it has been shown that the growth rate in the primary signatures of cyber-attacks increases at slow [67] and almost linear rates with respect to time [68]. An experimental study on snort rules was conducted to observe the addition of new rules over a period of several years. From April 2005 to October 2008 versions V2.1–V2.4 and V2.6–V2.8 were released. The experiments have shown that "whereas new rules are added based on newly discovered attack patterns, the major part of snort databases follows an accumulating update policy." Similarly, the growth rate in the number of nodes in SLNs follows a comparable trend to that of snort signatures. The last component in the run-time complexity is filtering inaccurate predictions using PFs. If the number of predictions after expansion is K_R, then the complexity of filtering out inaccurate predictions is $O(K_R U)$, where U is the number of the profiles used for filtering. Thus, the overall complexity in the prediction process is $O(K(P+W)+(K_R U))$. Since our approach operates on the outputs of existing IDSs (e.g., snort), W can be discarded, resulting in an actual complexity of $O((KP)+(K_R U))$. For the expansion process, it is expected that the growth in the number of nodes is linear, since the approach used is a signature-based one. In addition, the higher the value of the relevance score threshold the less the number of relationships to be scanned which reduces further the complexity of the expansion step.

VI. Experiments and Evaluation

Evaluating intrusion detection techniques is a challenging task due to the scarcity of labeled intrusion detection datasets. Sperotto *et al.* [3], [69] were one of the first who contributed a labeled flow-based dataset which has been used for evaluating flow-based attack detection technique. This dataset contains suspicious flows only. Our approach requires evaluation on both benign and suspicious traffic, therefore, there is a need to augment the dataset by including benign flows that are relatively similar to suspicious ones. By listening to a network interface in a testbed environment that consists of four Linux and Windows-based hosts and using public labeled PCAP files [70], we simulate the generation of benign traffic [4] to create benign flows and the snort IDS to label the collected traffic. Whereas such a flow collection method does not capture all intended characteristics of benign activities [71], related published research describing experiments on the same dataset utilized this approach [10], [72]. The resulting dataset consists of both benign and suspicious flows.

A. Data Preparation and Preprocessing

The data set provided by Sperotto *et al.* [3], [69] contains malicious flows. Each suspicious flow is correlated with one alert that describes the type of security incident (i.e., the label) of that flow. The security incidents in the dataset fall into two categories: 1) basic alerts and 2) clustered alerts. Basic alerts represent single security incidents and are directly correlated with one or more flows. Most alerts in this category are HTTP and SSH suspicious connection attempts. As a side effect of these attempts, ICMP and AUTH/IDENT traffic is generated. Although the side effect flows have not been described as suspicious activities, they were treated as consequences of SSH and HTTP suspicious connection attempts. The description features of the basic alerts in this dataset are analyzed using SQL queries. We found that the majority of basic alerts are SSH and HTTP connection attempts. Nevertheless, based on the header flags we discovered 12 distinct SSH scan types. Based on the targeted application we also discovered 11 types of alerts that are raised in response to HTTP connection attempts. These types represent the classes of attacks to be identified by the classifier used for initial prediction, and they also form the suspicious nodes in SLNs. These alerts are considered as the ground truth to validate the effectiveness of our approach in identifying the exact type of each suspicious flow. Clustered alerts represent logical groups of alerts. They describe attack scenarios during which several suspicious connection attempts are observed. Using the output of several SQL queries on the dataset, we found that the duration of each attack is between 5 s and 1 h. The dataset contains three types of clustered alerts; the first type represents the SSH scan attempts which consist of several SSH connection attempts. The second type of attacks is HTTP scans. As part of scan attacks, side-effect traffic is generated. The third type is a two-step attack representing attacker's HTTP connection attempts, and subsequently the attacker used the honeypot to launch SSH scans and dictionary attacks. Clustered alerts

TABLE II
CHARACTERISTICS OF THE DATASET USED IN THE EXPERIMENTS

Activity category	# of distinct node types in SLNs	# of selected flows	# of attacks
SSH suspicious connection attempts	12	350000	45 scans
SSH benign traffic	1	6428	-
HTTP suspicious connection attempts	11	9228	4 scans
HTTP benign traffic	1	95873	-
ICMP side effect traffic	1	16403	21 scans
IRC side effect traffic	1	7383	
AUTH/IDENT side effect	1	191325	
ICMP benign traffic	1	1573	-
Successful login attempts	-	21	21
Total	26 suspicious nodes, 3 benign nodes	574360 suspicious flows, 103874 benign flows	91

that aggregate attack steps in this dataset are considered as the ground truth to measure the effectiveness of the proposed approach in detecting multistep attacks. We selected 574 360 suspicious flows from this dataset with the corresponding alerts. The selected flows are used in creating SLNs, training the decision tree classifiers and testing the success rate of our approach in detecting both suspicious flows and multistep attacks. The data is divided in several consecutive time windows with various lengths in order to include the majority of suspicious activity types in the data.

During data preprocessing time bins are manually created. The width of each bin is 25 min. The bin width is selected based on the average duration of multistep attacks in this data. A total of 30 time bins are created as part of data preprocessing. Equal width binning approach is used to discretize other feature types including the location features such as source and destination IP. A total of 103 874 benign flows of HTTP, SSH, and ICMP traffic are used in the experiments. Each type is represented as a benign node in the SLNs. Table II shows the characteristics of the selected suspicious and benign flows.

B. Experimental Setting

The dataset is further partitioned into training and evaluation parts: 1) 70% of data is selected to train the decision tree classifiers and create SLNs and 2) the rest 30% is used for evaluation. The training and evaluation datasets contain benign flows and suspicious flows representing different types of basic and clustered alerts. The features, Pckts, Octs, Duration ($T_{\text{end}} - T_{\text{start}}$), P_{src}, P_{dst}, Flags, and Prot are used during the training phase of the decision tree classifiers and for the creation of PFs. Out of the 91 multistep attacks in the dataset, 45 attacks are used during training and 46 during evaluation. We created two types of SLNs: the first type does not include time and location contextual features in similarity calculations, the second is created with time and location features. The approach is evaluated to measure its accuracy in: 1) initially predicting the actual alert type, if any, using the classification model at the beginning of the prediction process; 2) identification of other relevant nodes that belong to a possible multistep attack using SLNs; and 3) filtering-out inaccurate predictions using the PFs. PR, DR, and F-score are the metrics used to evaluate our approach as defined

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$DR = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$F\text{score} = \frac{\left(1 + \beta^2\right) \times \text{PR} \times \text{DR}}{\beta^2 \times (\text{PR} + \text{DR})} \beta^2 = 1. \tag{9}$$

TP, FP, and FN represent the true positives, false positives, and false negatives, respectively. A TP represents a suspicious flow correctly recognized as suspicious. A TP for such a flow reveals the correct basic alert type n_i and other alerts that are semantically related to such alert. This includes other alerts which belong to multistep attacks in which n_i is observed, and/or alerts which cause/are caused by n_i. A FP arises when a specific benign flow under evaluation is incorrectly recognized as an alert. A FN arises when an incoming flow is an alert but it is incorrectly recognized as benign activity. The flows which are part of multistep attacks are scanned as sequences of observations and tested using our prototype. Sequences of benign flows are also tested to measure the effectiveness of the approach in differentiating between sequences of benign and attack activities. The predictions made for each flow in the sequence are collected and the union of these predictions is taken and used as a final prediction for that sequence. The final predictions are compared with the clustered alerts that correspond to flows under analysis to determine if our approach can actually detect attack steps.

The complexity of selecting distinct predictions for all flows in the sequence is $O(P)$, where P is the number of predictions made to each individual flow in the flow sequence. We performed our experiments on a testbed environment that includes a prototype implementation of our approach in an Oracle database running on a 64-bit Windows with Intel Pentium D Dual Core 3.4 GHz CPU and 32 GB RAM.

C. Effect of Context Fusion in Semantic Links on Attack Detection Rate

The first part in the evaluation process compares the effectiveness of SLNs created without time and location features (no suffix in the experiment name) versus the ones created with time and location features (with suffix TL).

We conducted experiments on the SLNs created using Anderberg (AD_SLN) and Pearson correlation (P_SLN) similarity measures, where threshold tr is used as a tuning parameter to observe the changes in PR, DR, and F-score values. The values of PR, DR, and F-score in our experiments are shown in Fig. 7. The observations in this figure can be summarized as follows.

First, the best PR value ($\approx$0.97) is noticed when threshold tr = 0.6 [Fig. 7(a)] and this result is achieved using AD_SLN_TL. Initially, the PR values observed in this experiment at small values of threshold tr are low, indicating that some benign activities were initially predicted as suspicious and due to the expansion phase, several other suspicious nodes

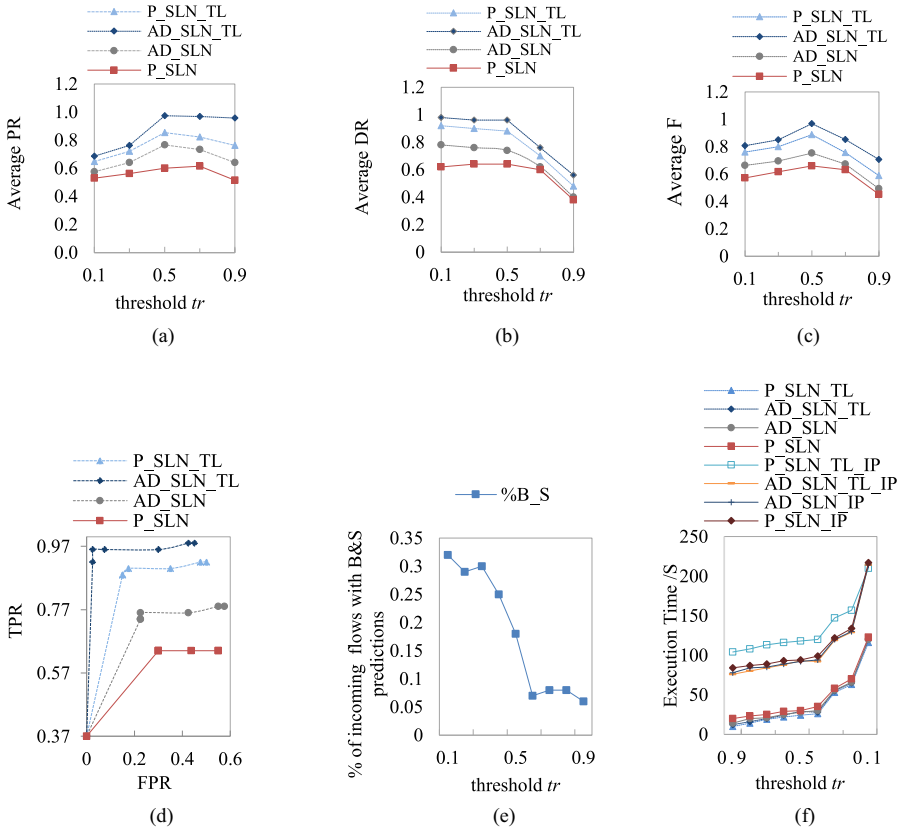

Fig. 7. Results of the first experiment. Average (a) precision for different SLNs, (b) DR for different SLNs, and (c) F-score for different SLNs. (d) ROC for SLNs with/without TL. (e) Percentage of incoming flows for which the predictions of SLN include both suspicious (S) and benign (B) nodes. (f) Execution time at different values of threshold without/with time for initial prediction.

were included as relevant to the initial prediction resulting in some false positives. With threshold tr values between 0.4–0.6 we observed better PR values. The PR decreases again at very high values of threshold tr since some true positives that should be included as part of the predicted multistep attacks are missed due to their weak relation to the initial prediction. Second, the fusion of time and location contextual features in SLNs yields better PR than SLNs created using only numerical and descriptive features. This shows in both SLNs types created using Pearson and Anderberg similarity measures. Third, although the difference is not very significant, the SLNs created using Anderberg similarity measure (AD_SLN and AD_SLN_TL) achieve better DRs. The Anderberg similarity coefficient focuses on positive matches (1-1) in similarity calculation, thus, it can aggregate activities that occur in similar contexts. Some of these observations can be noticed in Fig. 7(b) where the DR values are higher when threshold tr ranges between 0.1 and 0.6 and lower at more refined values of threshold tr. This decline in the values of DR is due to missing some relevant alerts which are either part of a multistep attack or semantically relevant to the initial prediction but have been missed during the expansion performed by SLNs. A similar trend can be seen in Fig. 7(c) which shows the F-score values at different values of threshold tr. The best F-score value is 0.97, achieved at 0.5 value of the threshold tr. These results indicate the positive contribution of time and location contextual features on the DR of attacks when those features are utilized to create SLNs. To measure this positive impact in terms of intrusion detection parameters, the receiver operating characteristic (ROC) curve is utilized. The ROC is a popular measure that has been used to compare intrusion detection techniques and to plot TP and FP rates associated with various operating points when different intrusion detection techniques are used. The values of TP and FP rates (TPR and FPR) are calculated as

$$TPR = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$FPR = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{11}$$

Fig. 7(d) shows the ROC of both approaches using six operating points (threshold tr $= 0.1 - 0.6$). The results clearly indicate the role of time and location contextual features in increasing the TPR and decreasing the FPR. The main observation is the reduction in FPR when time and location-based features are utilized in creating SLNs. Time and location features discover the relations between suspicious nodes that are observed in similar contexts. In particular, they improve the quality of semantic relationships between suspicious nodes that are observed together in several time bins. Therefore, the overlap between contexts of benign and suspicious activities is decreased, leading to lower false positive rates.

PFs are only applied to incoming flows for which the predictions of SLN include both suspicious and benign nodes. The

latter represent a small percentage of incoming flows especially when the value of threshold tr is high (>=0.5) as shown in Fig. 7(e). Therefore, even if such a filter does not work as intended it has a very minor effect on the values of TP and FP rate. To summarize, this set of filtering rules are only applied to small subset of incoming flows. The expansion process discussed earlier can affect the run time performance of the system. However, the number of semantic links that need to be scanned to expand the initial prediction decreases as the value of threshold tr increases. The execution time (the time to process incoming flows) of the experiment is affected by variations in the value of threshold tr. To observe such an effect, we measure the execution time using different forms of SLNs created with/without the time-location features as shown in Fig. 7(f), which also shows the execution time when the classification time at the beginning of the prediction process is included. The increase in the execution time is almost linear when three instances of the prediction procedures are executed concurrently on the testing dataset. Another observation is the slightly less execution time when SLNs with time and location features are utilized. The incorporation of time and location features adds more constraints on the resulting semantic links, limiting the number of weak and unnecessary semantic links at high values of threshold tr, leading to a narrower results scope, thus, decreasing the total time needed to find relevant predictions.

D. Effect of Semantic Links on Result Significance

We show that SLNs with PFs can identify attacks by examining flows. One question to be answered is: *can the same results be obtained using the typical association and similarity techniques to expand the initial prediction*? Two research hypotheses are explored to answer this question.

- H_0: The DR of SLNs can be achieved using measures such as similarity and association relationships.
- H_1: Semantic links discovered through reasoning rules are significantly more effective to detect attacks than typical similarity and association relationships.

To test the hypotheses above, we utilize association confidence (AC) measure and the typical similarity values calculated using AD and Pearson correlation (PC) to expand the initial prediction and observe the results. We use the AC between the nodes defined on SLNs as a measure of their co-occurrence. AC among two nodes is defined as $AC(n_i \rightarrow n_j) = (Pr(n_i \cup n_j)/Pr(n_i))$, where $Pr(n_i \cup n_j)$ is the probability that two nodes co-occur and it is calculated in terms of feature co-occurrence. The value of AC is equivalent to rs and it is used to expand the initial prediction. Therefore, we reran our experiments using the AC score to expand the initial prediction. The initial prediction is also expanded using the typical similarity values (SIM) calculated using the AD and PC measures. The results of this experiment are shown in Fig. 8. The symbols AC, and SIM refer to the AC and the typical similarity values.

The values of rs, AC, and SIM are calculated with and without time and location features and the averages indicate the effectiveness of the measure under analysis. As shown

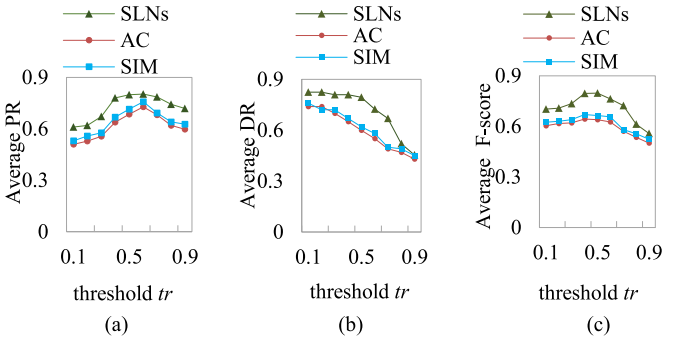

Fig. 8. Results of the second experiment. Average (a) precision for different expansion techniques, (b) DR for different expansion techniques, and (c) F-score for different expansion techniques.

in Fig. 8(a)–(c), SLNs show better performance compared to other expansion techniques using PR, DR, and F-score. To determine if the results achieved are statistically significant, the paired-samples T Test is used to measure the statistical difference of SLNs versus the AC and the typical similarity values. We measured the statistical significance using the F-score values since it summarizes the trend of PR and DR values. The sig (2-tailed) P-value at the 95% confidence interval and $\alpha = 0.05$ is ≈ 0.000 for the pairs (SLNs versus AC) and (SLNs versus SIM). The null hypothesis is therefore rejected. This implies that semantic links are the main factor that leads to such an improvement in the attack prediction process. The inference process during the creation of SLNs strengthens the relationships between nodes that co-occur with each other. Whereas both similarity and AC measures have the capability to create some association between nodes, they produce unnecessary relationships between benign and suspicious nodes, therefore, they lead to an increase in FPs and FNs rates.

E. Validation on Other Datasets

The results presented so far empirically prove the accuracy of our approach in identifying suspicious activities that are related in terms of time, location and other contextual features. One concern is if the same level of accuracy can be achieved using datasets with different characteristics. Since the dataset used in the previous experiments contains aggregated events according to their time and location features one may argue that they may be the reason of the achieved accuracy. So a question to answer is: *can we generalize the quality of the produced semantic links to other datasets that do not include time and location features*? To answer this question, we examine the validity of our approach when features do not clearly identify correlations in the dataset. In particular, we examine the validity of our approach using the DARPA dataset which contains TCP connections instead of flows. Each connection is described using 41 features and it can be a benign activity or an attack. Attacks that belong to each category G are contextually similar as shown in Table III. Thus, if one attack $n_i \in G$ is initially predicted by the rule-based classifier, an SLN is supposed to retrieve other attacks that belong to the same category as relevant to n_i. Events in the DARPA dataset are used to recreate: 1) the classifier that produces the initial prediction; 2) thSLNs; and 3) the PFs.

TABLE III
TAXONOMY OF ATTACKS IN THE DARPA DATASET

Attack Category	Attack Name
Denial of Service (DoS)	smurf, neptune, back, teardrop, pod, land
Remote to Local (R2L)	warezclient, Guess_Password, Warezmaster, imap, ftp_write, multihop, phf, spy
User to Root(U2R)	buffer_overflow, rootkit, loadmodule, perl
Probe	satan, ipsweep, portsweep, nmap

TABLE IV
CONNECTIONS SELECTED FROM DARPA INTRUSION DETECTION DATASET

Category	# of connections selected for training	# of connections selected for evaluation
Normal	147250	110620
Probe	4112	4171
Denial of Service (DoS)	391458	229853
Remote to Local (R2L)	1130	16354
User to Root(U2R)	50	69

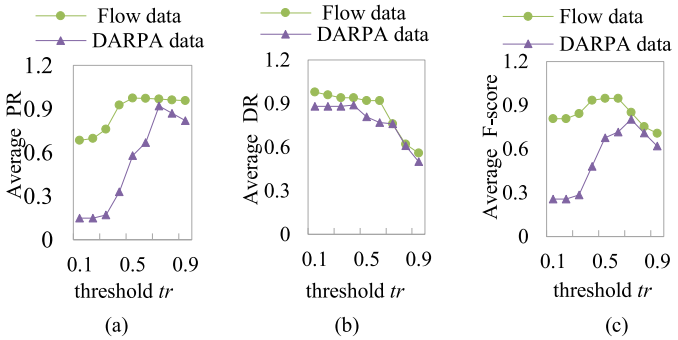

Fig. 9. Results of the third experiment. Average (a) precision using each dataset, (b) DR using each dataset, and (c) *F*-score using each dataset.

TABLE V
UNKNOWN TYPES OF ATTACKS IN THE FLOWS DATASET

Category	Unknown types of attacks	Number of flows
SSH-based attacks	2	10000
HTTP-based attacks	3	7230
Benign activities	-	40000
Total	5	57230

We utilized 544 000 connections to create: 1) the classifier that produces the initial prediction; 2) the SLNs; and 3) the PFs. Such prediction models are created based on the features of the selected connections and their label. Each label is represented as a single node in SLNs. We then used 361 067 connections during the evaluation phase to run this experiment. The number of connections used in this experiment per category is shown in Table IV. Each connection is initially classified as an attack or a benign activity using the rule-based classifier. The relevant predictions are then retrieved using SLNs. Finally, PFs are applied to minimize the percentage of inaccurate predictions.

We ran this experiment by comparing the performance of AD-based SLN created using the flows dataset, versus the one created using the DARPA dataset. For SLNs created using the flows dataset, we utilize the version of SLN with time and location features (AD_SLN_TL). For DARPA dataset we created the SLNs using the same algorithm described in Section V. Since the DARPA dataset does not contain time and location features, the SLNs created using this dataset do not utilize those features for attack discovery. The values of PR, DR, and *F*-measure are shown on Fig. 9.

First, the values of PR [Fig. 9(a)] when the flows dataset is used are better than their counterparts using DARPA dataset, however, the difference is not very significant at high values of threshold tr. Although time and location features are absent from connections in the DARPA dataset, the PR values of the SLNs equal 0.92 at 0.7 level of threshold tr. Notice that the DARPA dataset contains more types of attacks than the flows dataset, so one should expect to observe some unnecessary links between dissimilar attacks at low values of threshold tr. However, this fades away when raising the values of threshold tr. Regarding DR [Fig. 9(b)] we do not see significant difference in our approach when applied on either dataset. This also validates that SLNs can satisfactorily be used on datasets which lack time and location features. The F-measure values [Fig. 9(c)] achieved on SLN based on the DARPA dataset are lower compared to SLN based on flows dataset. Once again this is less frequent at high values of threshold tr. To conclude, we believe that when time and location contextual features exist in the dataset, the quality of semantic links are better than when such features are absent, however, we can still achieve a satisfactory DR by adjusting the value of the threshold tr.

F. Experiments on Discovering Unknown Attacks

In this section, we perform experiments to evaluate our approach in terms of detecting unknown attacks by analyzing flows. We utilize the PFs created using flow-features to investigate whether unknown traffic patterns can be identified using contextual similarity. The data set by Spretto *et al.* [69] is used in our experiments. Since the testing part of this dataset is from the same distribution of the training part, the testing data does not contain traffic patterns with unknown characteristics (i.e., unknown attacks). Alternatively, we can still examine if our approach will identify unknown traffic patterns by changing some characteristics of the training data. Therefore, we removed five types of SSH and HTTP attacks from the training part of this data, thus, they become partially unknown to our prediction models. Based on the experiments conducted earlier, most of the flows that correspond to these attacks—after removing them from the training part—are predicted as benign activities although they are actually attacks.

The distribution of unknown attacks and benign activities in the testing part of the flow data which was used during this experiment and the number of the corresponding flows are shown in Table V. There are two sets of rules that are generated when PFs are created, the first set is used to classify benign activities. Those rules are applied to filter-out

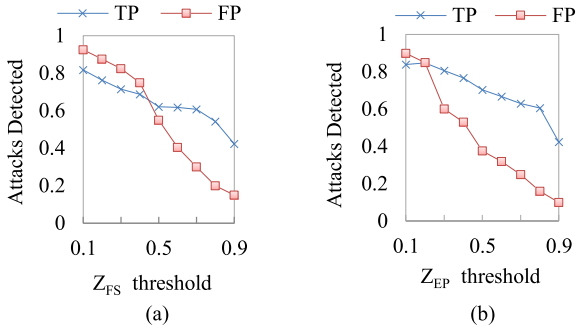

Fig. 10. Detecting unknown attacks in flows. TP and FP rates for detecting unknown attacks in flows using (a) AFs and (b) DPs.

some predictions of SLNs (as described in Algorithm 2). The second set identifies suspicious activities. Therefore, this set of rules, known as attack profiles AFs, is used to identify unknown attacks. Specifically, the process of identifying unknown attacks is performed as follows.

1) The ID3 classification rules are utilized to examine the features of incoming flows.
2) The flows which trigger any rule based on their features are initially classified as suspicious activities and an initial prediction is generated. Such flows are then processed by SLNs.
3) Other flows that do not trigger any rule are processed by AFs. We utilize partial similarity with AFs as a metric to identify unknown attacks. High similarity S between the features of a flow fl_i and the features of a profile AF_{ni} is an indicator of a possible unknown attack in that flow. A threshold Z_{FS} is then utilized to classify that flow as an unknown attack (i.e., IFS $\geq Z_{FS}$) or otherwise as a benign activity.

This part of the experiments is carried out by varying the values of profile similarity threshold Z_{FS} from 0.1 to 0.9. Fig. 10(a) shows that AFs achieve a TP rate of 0.54 and a FP rate of 0.20 when $Z_{FS} = 0.7$. Unknown attacks have several unique characteristics, therefore, matching the features of incoming flows with the discretized numerical features of AFs is not very effective approach to discover similarity. This leads to a high number of mismatches between the features of incoming flows and those of AFs, resulting in a low similarity between incoming flows and AFs, which leads to high FP rate. Therefore, PFs are recreated as discriminant functions (DPs) using linear discriminant analysis Technique [40]. The DPs are utilized to process the features of incoming flows. Each DP gives a probability value called estimated probability (EP) to each incoming flow. EP indicates the probability of an unknown attack given the features of the flow under analysis. Once the estimated probability EP is calculated for each flow, it is compared to a user tunable threshold (Z_{EP}). Alerts about unknown attacks are raised if the estimated probability for a specific flow is greater than Z_{EP}. Fig. 10(b) shows the TP and FP rates for DPs with values ranging from 0.90 to 0.42 for TP rate and 0.84 to 0.10 for FP. These rates are better than those achieved by AFs. These results indicate that DPs are better at handling the numerical features of incoming flows than rule-based classifiers.

TABLE VI
COMPARISON WITH OTHER FLOW-BASED INTRUSION DETECTION TECHNIQUES

Approach	t	t value	FPR	PR	DR	F
AD SLNs With time and location features	*tr*	0.5	**0.018**	**0.98**	**0.96**	**0.97**
		0.6	**0.017**	**0.98**	0.92	**0.95**
		0.7	**0.016**	**0.98**	0.80	**0.88**
		0.8	**0.015**	**0.98**	0.71	**0.82**
		0.9	**0.015**	0.98	0.68	0.80
ANN	-	-	0.03	0.57	0.93	0.70
KNN			0.05	0.56	0.94	0.70
SVM			0.03	0.50	**0.96**	0.66
LibLinear			0.05	0.67	0.91	0.77
ID3	-	-	0.09	0.55	0.9	0.68
RC	-	-	0.06	0.50	0.85	0.62
BC	-	-	0.03	0.52	0.82	0.63
AVG_ID3+SLNs	*tr*	0.5	0.04	0.80	0.81	0.80
AVG_RC+SLNs	*tr*	0.5	0.06	0.80	0.78	0.79
AVG_BC+ SLNs	*tr*	0.5	0.05	0.81	0.77	0.79

G. Comparison With Other Flow-Based Intrusion Detection Approaches

We conducted an experiment to compare our technique with different classification approaches. A representative instance selection technique was proposed by Guo *et al.* [72] to select representative samples of flows and use them as input to several techniques that classify them as attacks or benign activities.

In order to make the settings of our experiment consistent with those of each approach, we reduced the number of benign flows for this comparative experiment. In the experiments conducted in [72], the size of benign traffic is small (about 1000 flows) compared to the suspicious traffic. Additionally, the comparison is conducted based on recognizing suspicious activity as suspicious and benign activity as benign without focusing on the exact type of the suspicious activity. FPR, PR, DR, and F-score are reported in Table VI. A dash "-" means that the measure is either not used or cannot be calculated. The results reported in the table are the averages under different experiment settings using (ANN, KNN, SVM, and LibLinear) classification techniques. The overall values of PR and F-score are lower than ours.

The last part of this experiment is related to the classifier used for initial prediction and if such a classifier has a significant effect on the results generated using SLNs, that is, if such results depend on the classification technique used to produce the initial prediction. To better understand the effects of this classification technique let us explore two more hypotheses.

H_0: The classifier used to produce the initial prediction has no significant effect on the results yielded by SLNs
H_1: The classifier used to produce the initial prediction has a significant effect on the results yielded by SLNs.

To test the hypothesis above, we utilize two classifiers to produce the initial prediction, the Bayesian and RIPPER Classifiers. The Bayesian classifier (BC) is utilized as follows: the probability of each feature f_i with each node type $pr(f_i|n_i)$ is precalculated and profiled. We used the precalculated probability values at prediction time to find the MAP decision rule, that is, the probability of the most potential node based on the

feature vector $f_1, \ldots, f_n$ of the corresponding flow. Once the MAP decision rule is triggered, the initial prediction is passed to the SLNs. The SLNs expand such an initial prediction and pass it to PFs. The latter is recreated as a rule-based filter using the probability values generated using BC. The second classifier of this experiment is RIPPER (RC), which is a rule-based classifier. We ran this experiment using all classifiers without SLNs (ID3, RC, and BC) and with SLNs (ID3 + SLNs, BC + SLNs, and RC + SLNs). The results are averaged over all types of SLNs and are reported in Table VI. Note that using SLNs on top of other classification techniques achieves better PR, DR, and F-score values than using ID3, RC, and BC without any semantics. The paired T-sample test is used to measure the statistical difference between the use of SLNs on top of ID3 classifier (AVG_SLN+ID3) versus the use of SLNs on top of Bayesian and RIPPRER classifiers (AVG_SLN+RC, AVG_SLN+BC). We measure the statistical significance based on the F-measure values reported in Table VI. The sig 2-tailed P value at the 95% confidence interval for using SLNs on top of ID3 versus on top of BC classifier $\approx 0.276 > 0.05$. Similarly, the sig (2-tailed) P value at the 95% confidence interval for using SLNs on top of ID3 versus RIPPER classifier $\approx 0.66 > 0.05$. Since both p values are not significant, the null hypothesis is accepted, that is, the classifier used is not the major reason of the significant results generated using SLNs. Although classification and anomaly detection techniques can still work in case of flow-based intrusion detection, the major disadvantage of these techniques is the lack of semantics, which is needed to detect multistep attacks. Whereas such techniques can be used in signature or anomaly-based IDSs with a satisfactory rate of success, they are not capable enough to handle the sophistication and complexity of today's attacks.

VII. Conclusion

Analyzing flows is a major challenge for IDSs, due to a lack of information available for analysis. To overcome this problem, we propose an approach to detect cyber-attacks from flows using preidentified and automatically constructed semantic links among alerts raised in response to these flows. We focus on fusion of contextual information represented by time, location and other features to identify links among alerts using SLNs. The created links expand the predictions produced by other flow classification models. The purpose of such an expansion is improving the effectiveness of these models and detecting multistep attacks. Our experiments exhibit the importance of contextual information in semantic links to detect security alerts and multistep attacks from flows. Our approach also achieves a good DR of unknown attacks in network flows using profile similarity as an indicator of the probability of unknown attacks. We introduce an automatic technique that creates static semantic links. In the future, we plan to dynamically update the links to include new suspicious nodes.

References

[1] S. Jajodia, P. Liu, V. Swarup, and C. Wang, *Cyber Situational Awareness: Issues and Research*, vol. 14. Boston, MA, USA, Springer, 2010.

[2] T. Liu *et al.*, "Abnormal traffic-indexed state estimation," *Future Gener. Comput. Syst.*, vol. 49, pp. 94–103, Aug. 2015.

[3] A. Sperotto *et al.*, "An overview of IP flow-based intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 3, pp. 343–356, 3rd Quart., 2010.

[4] T. Ding, A. AlEroud, and G. Karabatis, "Multi-granular aggregation of network flows for security analysis," in *Proc. IEEE Int. Conf. Intell. Security Informat. (ISI)*, Baltimore, MD, USA, 2015, pp. 173–175.

[5] A. AlEroud and G. Karabatis, "Context infusion in semantic link networks to detect cyber-attacks: A flow-based detection approach," in *Proc. IEEE Int. Conf. Semant. Comput. (ICSC)*, Newport Beach, CA, USA, 2014, pp. 175–182.

[6] A. Sperotto, R. Sadre, P.-T. De Boer, and A. Pras, "Hidden Markov model modeling of SSH brute-force attacks," in *Proc. 20th IFIP/IEEE Int. Workshop Distrib. Syst. Oper. Manag. (DSOM)*, Venice, Italy, 2009, pp. 164–176.

[7] A. Valdes and K. Skinner, "Probabilistic alert correlation," in *Proc. 4th Int. Symp. Recent Adv. Intrusion Detection (RAID)*, Davis, CA, USA, 2001, pp. 54–68.

[8] L. Constantin. (2010). *Compromised Web Servers to Build SSH Brute Force Botnet*. Accessed on Nov. 15, 2013. [Online]. Available: http://news.softpedia.com/news/Compromised-Web-Servers-Used-to-Build-SSH-Brute-Force-Botnet-151779.shtml

[9] R. Hofstede and A. Pras, "Real-time and resilient intrusion detection: A flow-based approach," in *Dependable Networks and Services*, vol. 7279. Heidelberg, Germany: Springer, 2012, pp. 109–112.

[10] P. Winter, E. Hermann, and M. Zeilinger, "Inductive intrusion detection in flow-based network data using one-class support vector machines," in *Proc. 4th IFIP Int. Conf. New Technol. Mobility Security (NTMS)*, Paris, France, 2011, pp. 1–5.

[11] J. Quittek, T. Zseby, B. Claise, and S. Zander. (2004). *Requirements for IP Flow Information Export (IPFIX)*. Accessed on Oct. 20, 2013. [Online]. Available: http://tools.ietf.org/html/rfc3917

[12] B. Claise. (2008). *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information*. Accessed on Nov. 24, 2013. [Online]. Available: http://www.ietf.org/rfc/rfc5101.txt

[13] Y. Gao, Z. Li, and Y. Chen, "A DoS resilient flow-level intrusion detection approach for high-speed networks," in *Proc. 26th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Lisbon, Portugal, 2006, p. 39.

[14] A. Wagner and B. Plattner, "Entropy based worm and anomaly detection in fast IP networks," in *Proc. 14th IEEE Int. Workshops Enabling Technol. Infrastruct. Collaborative Enterprise*, Linköping, Sweden, 2005, pp. 172–177.

[15] C. Gates, J. J. McNutt, J. B. Kadane, and M. I. Kellner, "Scan detection on very large networks using logistic regression modeling," in *Proc. 11th IEEE Symp. Comput. Commun. (ISCC)*, Cagliari, Italy, 2006, pp. 402–408.

[16] F. Dressler, W. Jaegers, and R. German, "Flow-based worm detection using correlated honeypot logs," in *Proc. ITG-GI Conf. Commun. Distrib. Syst. (KiVS)*, Bern, Switzerland, 2007, pp. 1–6.

[17] M. P. Collins and M. K. Reiter, "Hit-list worm detection and bot identification in large networks using protocol graphs," in *Proc. 10th Int. Conf. Recent Adv. Intrusion Detection (RAID)*, 2007, pp. 276–295.

[18] M. Grill, I. Nikolaev, V. Valeros, and M. Rehak, "Detecting DGA malware using NetFlow," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, Ottawa, ON, Canada, 2015, pp. 1304–1309.

[19] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in *Proc. 1st Conf. Hot Topics Understand. Botnets (HotBots)*, Cambridge, MA, USA, 2007, p. 7.

[20] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol-and structure-independent botnet detection," in *Proc. 17th Conf. Security Symp. (USENIX)*, San Jose, CA, USA, 2008, pp. 139–154.

[21] O. van der Toorn, R. Hofstede, M. Jonker, and A. Sperotto, "A first look at HTTP (S) intrusion detection using NetFlow/IPFIX," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, Ottawa, ON, Canada, 2015, pp. 862–865.

[22] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 217–228, 2005.

[23] M. Sheikhan and Z. Jadidi, "Flow-based anomaly detection in high-speed links using modified GSA-optimized neural network," *Neural Comput. Appl.*, vol. 24, nos. 3–4, pp. 599–611, 2014.

[24] G. Fernandes, Jr., J. J. P. C. Rodrigues, and M. L. Proença, Jr., "Autonomous profile-based anomaly detection system using principal component analysis and flow analysis," *Appl. Soft Comput.*, vol. 34, pp. 513–525, Sep. 2015.

[25] J. Frecon *et al.*, "Non-linear regression for bivariate self-similarity identification—Application to anomaly detection in Internet traffic based on a joint scaling analysis of packet and byte counts," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Shanghai, China, 2016, pp. 4184–4188.

[26] D. L. Hancock and G. B. Lamont, "Multi agent system for network attack classification using flow-based intrusion detection," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, 2011, pp. 1535–1542.

[27] A. Fahad, Z. Tari, I. Khalil, I. Habib, and H. Alnuweiri, "Toward an efficient and scalable feature selection approach for Internet traffic classification," *Comput. Netw.*, vol. 57, no. 9, pp. 2040–2057, 2013.
[28] R. Fontugne, J. Mazel, and K. Fukuda, "Hashdoop: A mapreduce framework for network anomaly detection," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, 2014, pp. 494–499.
[29] J. François, C. Wagner, R. State, and T. Engel, "SAFEM: Scalable analysis of flows with entropic measures and SVM," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, Maui, HI, USA, 2012, pp. 510–513.
[30] H. Choi, H. Lee, and H. Kim, "Fast detection and visualization of network attacks on parallel coordinates," *Comput. Secur.*, vol. 28, no. 5, pp. 276–288, 2009.
[31] H. Zhuge, Y. Sun, and J. Zhang, "Schema theory for semantic link network," in *Proc. 4th Int. Conf. Semant. Knowl. Grid (SKG)*, Beijing, China, 2008, pp. 189–196.
[32] H. Zhuge, "Communities and emerging semantics in semantic link network: Discovery and learning," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 6, pp. 785–799, Jun. 2009.
[33] H. Zhuge, "Active e-document framework ADF: Model and tool," *Inf. Manag.*, vol. 41, no. 1, pp. 87–97, 2003.
[34] G. Karabatis *et al.*, "Using semantic networks and context in search for relevant software engineering artifacts," in *Journal on Data Semantics XIV* (LNCS 5880). Heidelberg, Germany: Springer, 2009, pp. 74–104.
[35] Z. Chen, A. Gangopadhyay, G. Karabatis, M. McGuire, and C. Welty, "Semantic integration and knowledge discovery for environmental research," *J. Database Manag.*, vol. 18, no. 1, pp. 43–67, 2007.
[36] Z. X. Huang and Y. Qiu, "Construction and aggregation of citation semantic link network," in *Proc. 4th Int. Conf. Semant. Knowl. Grid (SKG)*, Beijing, China, 2008, pp. 247–254.
[37] P. J. Brown, J. D. Bovey, and X. Chen, "Context-aware applications: From the laboratory to the marketplace," *IEEE Pers. Commun.*, vol. 4, no. 5, pp. 58–64, Oct. 1997.
[38] A. Zimmermann, A. Lorenz, and R. Oppermann, "An operational definition of context," in *Proc. 6th Int. Interdiscipl. Conf. Model. Using Context (CONTEXT)*, Roskilde, Denmark, 2007, pp. 558–571.
[39] A. AlEroud, "Contextual information fusion for the detection of cyber-attack," Ph.D. dissertation, Dept. Inf. Syst., Univ. Maryland Baltimore County, Baltimore, MD, USA, 2014.
[40] A. AlEroud and G. Karabatis, "Toward zero-day attack identification using linear data transformation techniques," in *Proc. IEEE 7th Int. Conf. Softw. Security Rel. (SERE)*, Gaithersburg, MD, USA, 2013, pp. 159–168.
[41] S. Peddabachigari, A. Abraham, and J. Thomas, "Intrusion detection systems using decision trees and support vector machines," *Int. J. Appl. Sci. Comput.*, vol. 2, no. 1, pp. 18–134, 2004.
[42] M. Jianliang, S. Haikun, and B. Ling, "The application on intrusion detection based on k-means cluster algorithm," in *Proc. Int. Forum Inf. Technol. Appl. (IFITA)*, Chengdu, China, 2009, pp. 150–152.
[43] J. Mazel, P. Casas, R. Fontugne, K. Fukuda, and P. Owezarski, "Hunting attacks in the dark: Clustering and correlation analysis for unsupervised anomaly detection," *Int. J. Netw. Manag.*, vol. 25, no. 5, pp. 283–305, 2015.
[44] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Inf. Sci.*, vol. 177, no. 18, pp. 3799–3821, 2007.
[45] W. Xuren and H. Famei, "Improving intrusion detection performance using rough set theory and association rule mining," in *Proc. Int. Conf. Hybrid Inf. Technol. (ICHIT)*, 2006, pp. 114–119.
[46] X. Qin, "A probabilistic-based framework for INFOSEC alert correlation," Ph.D. dissertation, College Comput., Georgia Inst. Technol., Atlanta, GA, USA, 2005.
[47] G. Jakobson, "The technology and practice of integrated multiagent event correlation systems," in *Proc. Int. Conf. Integr. Knowl. Intensive Multi Agent Syst.*, Cambridge, MA, USA, 2003, pp. 568–573.
[48] S. Mathew, C. Shah, and S. Upadhyaya, "An alert fusion framework for situation awareness of coordinated multistage attacks," in *Proc. 3rd IEEE Int. Workshop Inf. Assurance*, College Park, MD, USA, 2005, pp. 95–104.
[49] Y. Bouzida, F. Cuppens, N. Cuppens-Boulahia, and S. Gombault, "Efficient intrusion detection using principal component analysis," in *Proc. 3rd Conf. Security Netw. Architect.*, La Londe, France, 2004, pp. 381–395.
[50] S. Noel, E. Robertson, and S. Jajodia, "Correlating intrusion events and building attack scenarios through attack graph distances," in *Proc. 20th Annu. Comput. Security Appl. Conf. (CSAC)*, Tucson, AZ, USA, 2004, pp. 350–359.
[51] S. Noel, M. Jacobs, P. Kalapa, and S. Jajodia, "Multiple coordinated views for network attack graphs," in *Proc. IEEE Workshop Visualization Comput. Security (VizSEC)*, Minneapolis, MN, USA, 2005, pp. 99–106.
[52] L. Wang, A. Singhal, and S. Jajodia, "Toward measuring network security using attack graphs," in *Proc. ACM Workshop Qual. Protect.*, Alexandria, VA, USA, 2007, pp. 49–54.
[53] F. Massicotte, L. C. Briand, M. Couture, and Y. Labiche, "Context-based intrusion detection using snort, nessus and bugtraq databases," in *Proc. Int. Conf. Privacy Security Trust (PST)*, 2005.
[54] F. Silveira, C. Diot, N. Taft, and R. Govindan, "ASTUTE: Detecting a different class of traffic anomalies," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 267–278, 2010.
[55] S. S. S. Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Exp. Syst. Appl.*, vol. 39, no. 1, pp. 129–141, 2012.
[56] X.-B. Li, "A scalable decision tree system and its application in pattern recognition and intrusion detection," *Decis. Support Syst.*, vol. 41, no. 1, pp. 112–130, 2005.
[57] T. M. Cover and J. A. Thomas, *Elements of Information Theory: Entropy, Relative Entropy and Mutual Information*. New York, NY, USA: Wiley, 2012, ch. 2.
[58] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 70–91, 1st Quart., 2015.
[59] Q. Wu, D. Ferebee, Y. Lin, and D. Dasgupta, "An integrated cyber security monitoring system using correlation-based techniques," in *Proc. IEEE Int. Conf. Syst. Syst. Eng. (SoSE)*, Albuquerque, NM, USA, 2009, pp. 1–6.
[60] J. Beauquier and Y. J. Hu, "Intrusion detection based on distance combination," in *Proc. World Acad. Sci. Eng. Technol. (WASET)*, 2007, pp. 172–180.
[61] A. Hassanzadeh and B. Sadeghian, "Intrusion detection with data correlation relation graph," in *Proc. 3rd Int. Conf. Availability Rel. Security (ARES)*, Barcelona, Spain, 2008, pp. 982–989.
[62] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *Proc. 8th SIAM Int. Conf. Data Min. (SDM)*, Philadelphia, PA, USA, 2008, pp. 243–254.
[63] R. G. Pensa, C. Leschi, J. Besson, and J.-F. Boulicaut, "Assessment of discretization techniques for relevant pattern discovery from gene expression data," in *Proc. 4th ACM SIGKDD Workshop Data Min. Bioinformat. (SIGKDD)*, Seattle, WA, USA, 2004, pp. 24–30.
[64] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[65] P. He and G. Karabatis, "Using semantic networks to counter cyber threats," in *Proc. IEEE Int. Conf. Intell. Security Informat. (ISI)*, Arlington County, VA, USA, 2012, p. 184.
[66] J. W. Grzymala-Busse, "Selected algorithms of machine learning from examples," *Fundamenta Informaticae*, vol. 18, no. 1, pp. 193–207, 1993.
[67] H. Chen, D. H. Summerville, and Y. Chen, "Two-stage decomposition of SNORT rules towards efficient hardware implementation," in *Proc. 7th Int. Workshop Design Reliable Commun. Netw.*, Washington, DC, USA, 2009, pp. 359–366.
[68] S. Sen. (2006). *Performance Characterization & Improvement of Snort as an IDS*. [Online]. Available: http://www.tc.umn.edu/∼ssen/papers/bell_labs_report_snort.pdf
[69] A. Sperotto, R. Sadre, F. Vliet, and A. Pras, "A labeled data set for flow-based intrusion detection," in *Proc. 9th IEEE Int. Workshop IP Oper. Manag. (IPOM)*, Venice, Italy, 2009, pp. 39–50.
[70] NETRESEC. (2013). *Network Forensics and Network Security Monitoring*. [Online]. Available: http://www.netresec.com/?page=PcapFiles
[71] D. Miller. (2013). *Softflowd: A Flow-based Network Traffic Analyser*. [Online]. Available: http://www.mindrot.org/projects/softflowd/
[72] C. Guo *et al.*, "Efficient intrusion detection using representative instances," *Comput. Security*, vol. 39, pp. 255–267, Nov. 2013.

Ahmed F. AlEroud received the B.S. degree in software engineering from Hashemite University, Zarqa, Jordan, and the M.S. and Ph.D. degrees in information systems from the University of Maryland at Baltimore County (UMBC), Baltimore, MD, USA.

He is an Assistant Professor of Computer Information Systems with Yarmouk University, Irbid, Jordan. He was a Visiting Associate Research Scientist with the UMBC, researching on cybersecurity research projects. His current research interests include cybersecurity, data mining for privacy preserving network data analytics, and detection of social engineering attacks.

George Karabatis (M'94) received the B.S. degree in mathematics from Aristotelio University, Greece, and the M.S. and Ph.D. degrees in computer science from the University of Houston, Houston, USA.

He is an Associate Professor of Information Systems with the University of Maryland at Baltimore County, Baltimore, MD, USA. He was a Research Scientist with Telcordia Technologies (formerly Bellcore), Piscataway, NJ, USA. His current research interests include cybersecurity, specifically on intrusion detection utilizing intelligent manipulation of information through semantics and context, semantic information integration, workflow systems, big data, multidatabase systems, and concurrency control.