Hybrid Intelligence for Semantics-Enhanced Networking Operations

Bassem Mokhtar
The Bradley Department of Electrical and Computer
Engineering, Virginia Tech, Virginia, USA
bmokhtar@vt.edu

Mohamed Eltoweissy[1]
Department of Computer and Information Sciences
Virginia Military Institute, Virginia, USA
eltoweissymy@vmi.edu

Abstract

Endowing the semantically-oblivious Internet with Intelligence would advance the Internet capability to learn traffic behavior and to predict future events. In this paper, we propose a hybrid intelligence memory system, or NetMem, for network-semantics reasoning and targeting Internet intelligence. NetMem provides a memory structure, mimicking the human memory functionalities, via short-term memory (StM) and long-term memory (LtM). NetMem has the capability to build runtime accessible dynamic network-concept ontology (DNCO) at different levels of granularity. We integrate Latent Dirichlet Allocation (LDA) and Hidden Markov Models (HMM) to extract network-semantics based on learning patterns and recognizing features with syntax and semantic dependencies. Due to the large scale and high-dimensionality of Internet data, we utilize the Locality Sensitive Hashing (LSH) algorithm for data dimensionality reduction. Simulation results using real network traffic show that NetMem with hybrid intelligence learn traffic data semantics effectively and efficiently even with significant reduction in volume and dimensionality of data, thus enhancing Internet intelligence for self-/situation-awareness and event/behavior prediction.

Introduction

Due to semantically-oblivious protocol operations, the current Internet cannot effectively or efficiently cope with the explosion in services with different requirements, number of users, resource heterogeneity, and widely varied user, application and system dynamics (Feldmann, 2007). This is leading to increasing complexity in Internet operations, thus multiplying challenges to achieve better security and performance or even maintain satisfaction of applications with static or dynamic QoS requirements. The current Internet largely lacks capabilities to extract network-semantics to efficiently build runtime accessible dynamic behavior models of Internet elements (e.g., applications, services, protocols, etc.) at different levels of granularity to pervasively observe, inspect, analyze, predict and act upon network dynamics. We refer to the limited utilization of Internet traffic semantics in networking operations as the "Internet Semantics Gap".

The current and future internets (e.g., Internet of things) support a massive number of entities with extensive amounts of data. Fortunately these data generally exhibit multi-dimensional patterns (e.g., patterns with dimensions such as time, space, and various Internet elements) that can be learned to extract network-semantics (Srivastava et al., 2000). Network-semantics represent implicit information that can be extracted from analyzing raw Internet data using reasoning models for better understanding anomalous/emergent behavior of Internet elements. Recognizing and maintaining semantics as accessible behavior models related to various Internet elements will aid elements in possessing intelligence thus helping elements in predicting future events (e.g., attacks) and learning novel things. For instance, a router can classify a new running service in a network as a specific type of TCP-based file transfer service when it finds similarity between behavior of that new service and that of the already known service.

Hence, there is a need to endow Internet operations and running services and applications with intelligence to mitigate the Internet semantics gap. In the literature, Internet (or network) intelligence (referred here as InetIntel) is defined as the capability of Internet elements to understand network-semantics to be able to make effective decisions and use resources efficiently (Li *et al.*, 2007). InetIntel can be achieved via employing intelligence techniques to efficiently reason about semantics from tons of Internet traffic raw data and provide runtime accessible valuable information at different levels of granularity. InetIntel systems use either monolithic or hybrid intelligence techniques (HIT).

[1] Also affiliated with The Bradley Department of Electrical and Computer Engineering, Virginia Tech, USA.

Network-data characteristics include massive volume, high- and multi-dimensionality, dynamicity and complexity (variety in representation models and languages). In (Griffiths *et al.*, 2004), authors proposed a generative model based on Latent Dirichlet Allocation (LDA) (Blei *et al.*, 2003) and HMM (Rabiner and Juang, 1986) for learning words with short-range syntax and long-range semantics dependencies. Consequently, this aids in forming richer ontology with more associated semantic topics and classes. There is similarity between characteristics (e.g., huge volume, high dimensionality, and complexity) of datasets in networks and language modeling.

In this paper, we present a system, called NetMem, for targeting InetIntel. NetMem provides a semantics reasoning model via a HIT integrating LDA and HMM algorithms for combining the advantages of both algorithms in efficiently: a) learning patterns of big data with high- and multi-dimensionality; and b) building dynamic network-concept ontology (DNCO) showing different and correlated concept classes (Rajpathak and Chougule, 2011). NetMem uses the proposed HIT-based reasoning model in order to have highly abstracted and associated application-agnostic semantics on different levels of granularity and related to various network concerns. We are motivated in our HIT design by the capability of LDA (Blei *et al.*, 2003) to discover latent and classified high-level features from multi-dimensional network-data patterns with long-range semantic dependencies. Those classified features will be sequenced to enable semantic reasoning via HMM with higher accuracy. HMM (Rabiner and Juang, 1986) are structured architectures that are able to predict sequences of semantic topics (related to different network concerns) based on input sequences of extracted network features by LDA. Depending on input sequences or pattern of highly-discriminative network-data features, HMM with forward and backward algorithms can learn semantics showing their functional, behavioral, and structural (FBS) aspects.

NetMem design is inspired by the functionalities of human memory (Hawkins and Blakeslee, 2005), which maintains conceptual models that describe associative concepts according to learned multi-dimensional patterns of data captured from the outside world through our sensory system. Those models are updated continually and used for future predictions achieving human intelligence. Analogy with human memory's functionalities, NetMem has a memory system structure comprising short-term memory (StM) and long-term memory (LtM). StM maintains for a short-time highly dynamic raw data while LtM keeps for long-time little varying information, which is used in matching and prediction processes. From a system's perspective, NetMem can be viewed as an overlay network of distributed "memory" agents targeting various data abstraction levels.

NetMem adopts Locality Sensitive Hashing (LSH) (Mimaroglu and Simovici, 2008) to reduce dimensions of large scale network-data. LSH can search for similarity in high-dimensional data and minimize required storage spaces. We focus in this paper on the design and evaluation of NetMem with the designed HIT by integrating LDA and HMM and with/without utilizing LSH. Our contributions are as follows:

- Memory system design with hybrid intelligence for efficient extraction of high-level features and reasoning about semantics in network traffic; and
- Runtime on-demand accessible application-agnostic customizable DNCO of concept classes related to various network concerns showing FBS aspects per each class; to have, for example, better enhanced decision making and anomaly detection.

Hybrid Intelligence Network Memory System

NetMem is a shared distributed semantics management system that can be built as an overlay network of "memory" agents. NetMem system can be built separately on multiple autonomous entities (e.g., intelligent agents) with capability of inter-communication and semantics integration.

NetMem Architecture Components

NetMem architecture, as shown in figure 1, comprises the following interacting components targeting InetIntel:

1- *Short-term memory (StM) and long-term memory (LtM)*: StM, or working memory, maintains raw data which possess higher levels of details and are related to various Internet elements. LtM maintains data semantics with higher levels of abstraction. StM and LtM consist of sets of big extensible relational data tables, which are a cloud-like data storage technique;

2- *Data virtualization and access (DVA)*: performs data collection and acquisition operations. DVA implements data virtualization techniques for data homogenization via using data models to unify

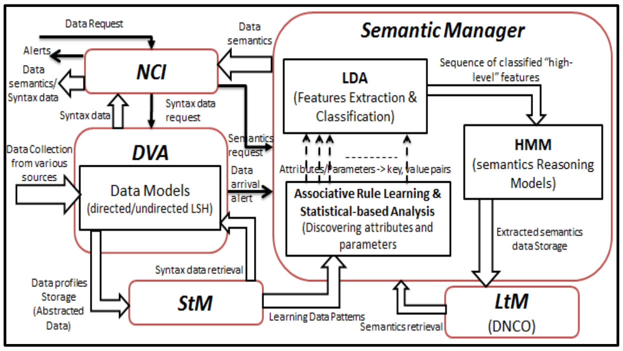

Data Request
Alerts
Data semantics/ Syntax data
NCI
Data semantics
Semantic Manager
Syntax data
Syntax data request
LDA
(Features Extraction & Classification)
Sequence of classified "high-level" features
HMM
(semantics Reasoning Models)
Data Collection from various sources
DVA
Data Models
(directed/undirected LSH)
Semantics request
Data arrival alert
Attributes/Parameters -> key, value pairs
Associative Rule Learning & Statistical-based Analysis
(Discovering attributes and parameters
Extracted semantics data Storage
Syntax data retrieval
Data profiles Storage (Abstracted Data)
StM
Learning Data Patterns
Semantics retrieval
LtM
(DNCO)

Figure 1: NetMem with HIT reasoner

network-data representation as profiles of attribute-value pairs in NetMem tables. DVA adopts LSH algorithm for data reduction, grouping, and similarity matching processes;

3- *Semantic manager (SM)*: executes semantics reasoning operations. SM is responsible for discovering/ generating/matching semantics in LtM using LDA and HMM algorithms (i.e., HIT) and based on: a) monitoring raw data or data profiles in StM and learning their patterns; and b) extracted and classified data attributes via adopting Associative Rule Learning (ARL) with Apriori algorithm and Fuzzy Membership Functions (FMF). The implemented ARL algorithm, as shown in figure 2, is used to relate groups of data attributes that are most found in data profiles. FMF can classify attributes based on their values; and

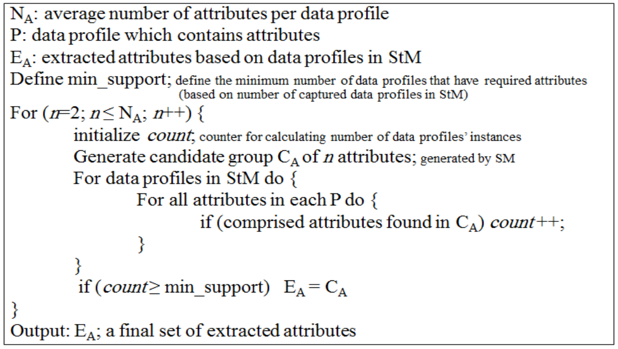
4- *NetMem controller and interface (NCI):* responsible for handling raw data and semantics (or concepts) requests from Internet elements in various networking domains. It has the capability to differentiate requests, and accordingly sends tasks, e.g., data discovery, to DVA or SM. It receives responses from DVA or SM including required data to be accessed by Internet elements.

Dimensionality Reduction

NetMem DVA applies LSH algorithm with bit sampling for hamming distance to reduce number of attributes represented by each profile and finding similarities among group of expressed profiles to be stored in one location in StM. Reduced dimensional profiles of length *k* attributes are constructed via selecting *k* attributes among *n* attributes where *n*>*k*. Attributes' selection depends on using hash functions which might be chosen randomly or according to specific directions such as focusing on attributes of certain Internet elements (e.g., services).

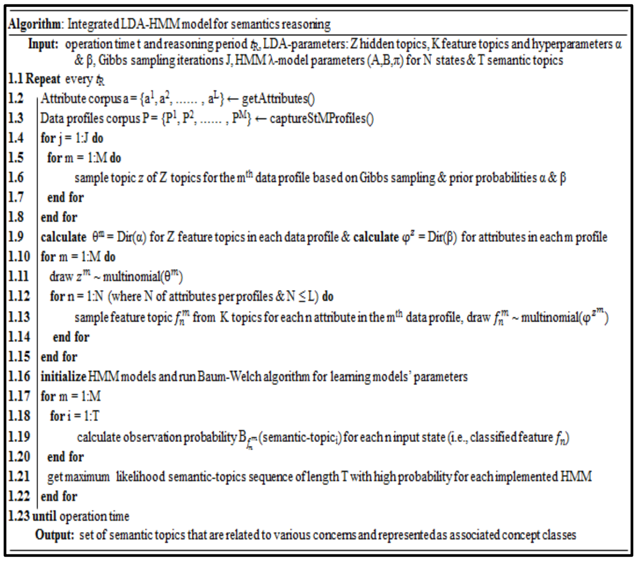
Hybrid Intelligence Technique for Semantics Reasoning

We provide for NetMem the hybrid LDA-HMM or HIT model, as illustrated in figure 3, through integrating two monolithic intelligence techniques (i.e., LDA and HMM) to make use of their abilities. We provide the hybrid model to strengthen the capability of NetMem for reasoning efficiently about network-semantics based on learning patterns of full or reduced-dimensional data. Semantic reasoning process via the implemented HIT in SM is executed for learning high-level data features with long-range syntax and semantic dependencies. The process is performed every defined reasoning window or through other criteria as triggering signal sent from NetMem DVA to SM. The HIT operation depends on learning patterns of kept profiles and comprised attributes in StM. Groups of profiles' attributes are discovered using the applied ARL algorithm and a defined set of FMFs in SM. For instance and through a defined reasoning window, SM aggregates information, initially, via learning patterns of TCP data profiles in StM through recognizing and classifying attributes of each profile. Analyzing TCP data profiles in StM by ARL and FMF might give for LDA the following classified attributes: 10000 profile's instances, large_TCP_packet_size,TCP-SYN_packet_type, file_ transfer_service_type.



```
$N_A$: average number of attributes per data profile
P: data profile which contains attributes
$E_A$: extracted attributes based on data profiles in StM
Define min_support; define the minimum number of data profiles that have required attributes
                    (based on number of captured data profiles in StM)
For ($n$=2; $n \leq N_A$; $n$++) {
        initialize count; counter for calculating number of data profiles' instances
        Generate candidate group $C_A$ of $n$ attributes; generated by SM
        For data profiles in StM do {
                For all attributes in each P do {
                        if (comprised attributes found in $C_A$) count++;
                }
        }
        if (count≥ min_support)  $E_A = C_A$
}
Output: $E_A$; a final set of extracted attributes
```

Figure 2: associative rule learning algorithm in SM



```
Algorithm: Integrated LDA-HMM model for semantics reasoning
   Input: operation time t and reasoning period $t_R$, LDA-parameters: Z hidden topics, K feature topics and hyperparameters $\alpha$
          & $\beta$, Gibbs sampling iterations J, HMM $\lambda$-model parameters (A,B,$\pi$) for N states & T semantic topics
Repeat every $t_R$
   Attribute corpus a = {$a^1, a^2, \ldots\ldots , a^L$} ← getAttributes()
   Data profiles corpus P = {$P^1, P^2, \ldots\ldots , P^M$} ← captureStMProfiles()
   for j = 1:J do
     for m = 1:M do
          sample topic $z$ of Z topics for the $m^{th}$ data profile based on Gibbs sampling & prior probabilities $\alpha$ & $\beta$
     end for
   end for
   calculate $\theta^m = Dir(\alpha)$ for Z feature topics in each data profile & calculate $\varphi^z = Dir(\beta)$ for attributes in each m profile
   for m = 1:M do
     draw $z^m \sim multinomial(\theta^m)$
     for n = 1:N (where N of attributes per profiles & $N \leq L$) do
          sample feature topic $f_n^m$ from K topics for each n attribute in the $m^{th}$ data profile, draw $f_n^m \sim multinomial(\varphi^{z^m})$
      end for
   end for
   initialize HMM models and run Baum-Welch algorithm for learning models' parameters
   for m = 1:M
     for i = 1:T
          calculate observation probability $B_{f_n^m}$(semantic-topic$_i$) for each n input state (i.e., classified feature $f_n$)
     end for
     get maximum likelihood semantic-topics sequence of length T with high probability for each implemented HMM
   end for
until operation time
   Output: set of semantic topics that are related to various concerns and represented as associated concept classes
```

Figure 3: Pseudo code of the LDA-HMM-based reasoning model

Based on learned attributes, LDA extracts and classifies high-level features associated with each analyzed *m* data profile of total *M* profiles in StM. LDA samples a hidden semantic topic *z* for each *m* data profile through calculating sampled posterior probability vector θ of topic-data profile association which depends on prior association weight α, number of the m^{th} profile's attributes related to a certain topic *z*, and total number of attributes in the m profile. Also, LDA calculates sampled posterior probability φ of attribute-topic association based on prior attribute-topic association weight β, number of attribute instances assigned to topic *z*, and total number of attributes in all *M* profiles assigned to topic *z*.

In our proposed HIT, through a certain number of iterations and using Gibbs sampling (Casella and George, 1992) LDA draws hidden topics for each *m* data profile and then draws feature topics for comprised attributes in each analyzed data profile. For example, three feature or semantic topics are defined in LDA: (“normal TCP packet”, “normal comm-flow”,“TCP comm-protocol”). Ten data profiles (*M*=10) in StM yield the same three classified attributes. Each attribute and profile has a prior topic association weight vector. Based on the overall prior weight vectors *(α and β)* and number of semantic topics, a sampled topic association probability vector p_{assoc} of length equals the number of available semantic topics is calculated like p_{assoc}=p(semantic_topic_1)=0.75, p(s_2)= 0.2, p(s_3)=0.05. In each LDA iteration, the current assigned topics for a data profile and comprised attributes are removed. Then, a random number ***u*** is sampled based on p_{assoc} and the summation of its contents. The higher *p* topic association value will be chosen and the related topic is assigned. For example, if ***u*** equals 0.6, number of attributes and related profiles assigned to the first semantic topic (i.e., the new topic) increases since (p(s_1) =0.75) is greater than 0.6. Thereafter, updates will be happened to posterior association weights θ and φ according to changes in number of attributes and profiles that relate to the first semantic topic. Hence, the posterior association weight of the first topic with data profiles and comprised topic-related attributes increases.

Extracted and classified features, output from LDA, form a sequence and convey to parameters of HMM (A, B, π) to generate semantics. The HMM parameters, discussed shortly, are trained and assigned using the unsupervised Baum-welch learning algorithm. That algorithm depends on an initial developed HMM for finding the maximum likelihood HMM parameters through iteratively training the parameters of the initial model relied on the observed output sequence. The ability of input to HMM sequence of data features related to diverse network concerns enables getting output sequence with associated semantic topics or concept classes on different level of abstraction. For example, an input sequence to HMM might be (*“normal TCP packet size”, “normal comm-flow”, “TCP comm-protocol”*) with equal initial state probability π (i.e., π =1/3) and state transition probabilities A (i.e., A_{ij} = 1/2 for i≠j and A_{ij} = 0 for i=j where A_{ij} is the transition probability form state *i* to state *j*). The first feature can be classified as an application concern and the other two features as communication concerns. Accordingly, the expected HMM output observation based on the previous sequence with any feature order might be “normal TCP-based service”. To get the previous output, the observation probability B matrix, which relates each input state with an output, regarding that concept class (i.e., output) will be high. For instance, B matrix might consist of three rows ***r*** and three columns ***c***; and it might equal ((0.3,**0.5**,0.2),(0.2,**0.8**,0.0), (0.4,**0.45**,0.05)) where the number of ***r*** equals the number of input features and the number of ***c*** equals the number of output concept classes. According to the previous example, all input features have high observation probability with the “normal TCP-based service” concept.

Evaluation

We study the effectiveness of NetMem’s HIT via simulation. We target a case study where NetMem is used to achieve predictive networking operations via better anomaly detection. A practical and simple network of five entities was implemented. Those entities connect to the Internet. There were two hosts and two servers. FTP and Web servers were implemented over two static laptops running Windows 7. Two hosts were built over another two static laptops to handle data from servers. NetMem was implemented over a Windows 7 laptop, an entity with routing functionalities that can capture data/control packets going from/to hosts to/from servers and Internet. NetMem was implemented as an application written in Java for the operations of NetMem entities: NCI, DVA, StM, LtM, and SM. Real traffic was collected via snort (Caswell and Beale, 2004). We compared NetMem system effectiveness with snort in learning classes of normal/abnormal data flows and detecting classes of attacks. Table 1 shows metrics used to evaluate performance of NetMem. In our scenario, hosts run file

Table 1: Prediction analysis metrics and equations

Metric	Function	Equation	Unit
Processing Overhead (T)	Measures the time complexity of the implemented semantics reasoning model for detecting and learning all behavior classes of running service flows	$T = T_R$ where T_R is the time-overhead caused by the implemented algorithms in the semantic Reasoning model	min
Space Complexity(S)	Identifies the storage space required for maintained data that will be used by the implemented semantics reasoning model	S = StM_size where StM_size is the storage space of the working memory for learning data patterns	Kbytes
Prediction Accuracy (A_c)	Calculates ratio of true positive (Tp) classes and true negative (Tn) classes that are learned with respect to all behavior classes (N_c) that should be learned	A_c =(Tp+Tn) / N_c	---
False Positive (Fp) Ratio	Calculates ratio of normal behavior classes that misclassified to abnormal classes according to N_c	Fp / N_c	---
False Negative (Fn) Ratio	Calculates ratio of abnormal behavior classes that misclassified to normal classes according to N_c	Fn / N_c	---
Recall (R)	Measures the effectiveness of system to learn abnormal behavior classes (including classes of attacks) with respect to Fn	R= Tp/(Fn+Tp)	---

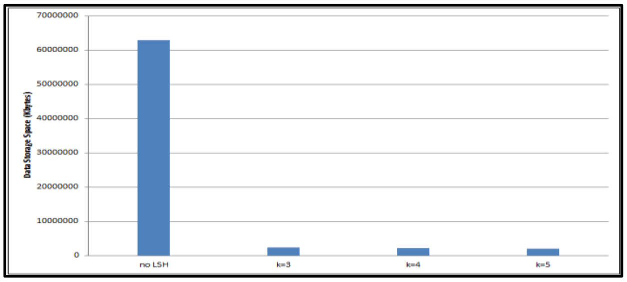


Figure 4: Storage space saving in StM due to using LSH

transfer services on top of TCP to get files and access pages from FTP and Web servers, respectively. Also, hosts utilize UDP to transfer data packets through the Internet. One of the two hosts was malicious and it sent the web server succession of TCP SYN requests to form TCP-SYN flood attack. The other host is a legitimate user which requests file from FTP server.

In our evaluation, we tested NetMem performance in correctly learning semantics of running services' flows and forecasting behavior of unfamiliar traffic flows using LDA-, HMM- and HIT-based reasoning models. We compared results with snort based on its set of defined traffic-content-based rules. We evaluated NetMem performance without LSH and with LSH at various *k* values (3, 4 and 5). The *k* value refers to the reduced number of attributes in reduced-dimensional data profiles.

Figure 4 shows the impact of adopting LSH in minimizing StM storage space required for reasoning about semantics compared with different operation cases; NetMem system without LSH and with LSH at different values of *k*. Figure 5 shows the percentage of storage space saving according to the usage of LSH. As in the figure, the usage of LSH saved over 96% of storage space than in the case of operations without LSH. Figure 6 illustrates the processing time overhead of NetMem's semantic reasoning model using different algorithms to recognize semantics of normal/abnormal flows and to detect running malicious flows and attacks accordingly.

Figure 7 shows the prediction accuracy and false negative ratio of NetMem versus snort for learning normal/abnormal flows' behavior classes and detecting related attacks. The NetMem's semantic reasoning model using HIT with/without LSH was able to learn most of the behavior classes with higher accuracy. Simulation results showed that snort was not able to learn correctly all normal behavior classes of running flows because snort is considered a stateless firewall in general (Caswell and Beale, 2004). As an example, a stateless firewall will prevent legitimate packets concerning a TCP-based traffic from passing since it has no knowledge that those packets destined to a protected network adopt a certain host's destination port such as 51282. From the previously obtained results, we conclude the following: a) NetMem can provide an effective and efficient capability through the designed HIT for learning behavior classes of normal/abnormal network-data traffic relying on learning patterns of full- or reduced-dimensional profiles of traffic data; and b) NetMem can support existing networking tools (e.g., intrusion detection systems such as snort) to execute their functions more effectively and efficiently.

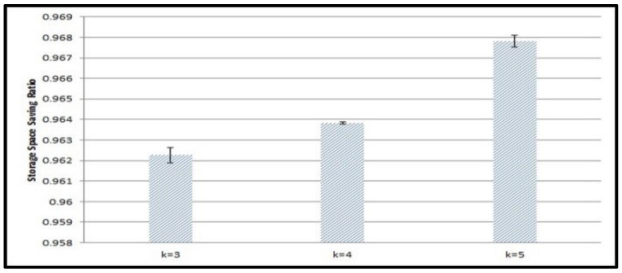


Figure 5: Storage space reduction

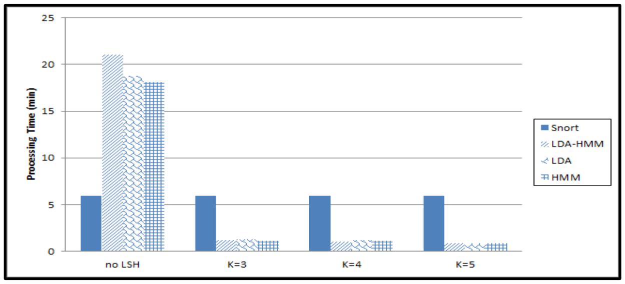
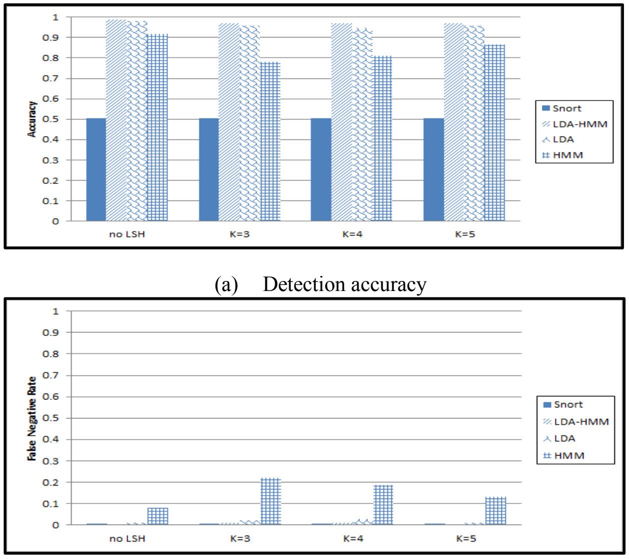


Figure 6: Processing time overhead by reasoning processes

Related Work

Recent works targeted intelligence-based solutions to enhance operation performance in networking. In (Idris and Shanmugam, 2005), authors provided a HIT-based system for intrusion detection that combines: a) misuse detection (e.g., detect attacks based on learning patterns and matching with already known attack patterns); and b)



(a) Detection accuracy

(b) False negative (Fn) ratio

Figure 7: Detection accuracy and Fn ratio of NetMem system and Snort in detecting classes of abnormal flows and attacks

anomaly detection (e.g., learn unfamiliar attacks or threats by applying statistical analysis methods over data and compare results with historical knowledge). The proposed HIT comprised fuzzy logic and a data mining mechanism with the usage of NN. They depended on extracting features from large sets of real network-data and applying those features over simple if-then fuzzy rules. Khan et al. (Khan *et al.*, 2012) provided HMM for discovering abnormal temporal events in sensor networks. The assumption was that unusual events are rare and not enough data are found for training. Different detection models were implemented where abnormal activities were detected if their likelihood were below defined thresholds.

How NetMem Differs?

Compared with related work such as (Idris and Shanmugam, 2005; Khan *et al.*, 2012), NetMem provides a storage memory structure which comprise StM and LtM to facilitate data patterns learning and semantics reasoning/retrieval for matching and prediction processes. Additionally, NetMem's HIT can provide behavior models related to various Internet elements through the formed application-agnostic runtime accessible ontology. NetMem's HIT achieves ontology of richer network concept classes, extracted with high prediction accuracy and good level of computation time compared with NetMem at using monolithic reasoning algorithms. Furthermore, the formed ontology can be customized to maintain concept classes related to specific Internet elements showing various FBS aspects per each class. Derived concept classes are accessed and learned by Internet elements at runtime and on-demand enabling, for example, learning services' QoS requirements and better anomalies detections.

Conclusion

In this paper, we presented hybrid intelligence memory system, or NetMem, targeting Internet intelligence. NetMem integrated LDA and HMM for semantic reasoning, and utilized LSH to reduce data dimensionality and to have low time-overhead for learning and matching Internet behavior classes. NetMem provided a memory structure mimicking the human memory functionalities via StM and LtM to enable learning data patterns and semantics reasoning. Evaluation of networking operations using real-time Internet traffic data showed the efficacy of NetMem for learning behavior classes of normal/anomalous flows and attacks. Also, NetMem with hybrid intelligence provided better effectiveness and efficiency compared with monolithic intelligence techniques. Future work includes (i) leveraging NetMem's hybrid intelligence capability of learning dynamic and abnormal behavior to enhance behavior prediction and self- and situation-awareness by the various Internet elements, and (ii) expanding our evaluation to study a fuller large-scale NetMem system with distributed intelligent heterogeneous multi-agent system with dynamic adaptation to various reasoning models.

References

Blei, D.M., A.Y. Ng and M.I. Jordan, 2003. Latent dirichlet allocation. the Journal of machine Learning research, 3: 993-1022.

Casella, G. and E.I. George, 1992. Explaining the gibbs sampler. The American Statistician, 46(3): 167-174.

Caswell, B. and J. Beale, 2004. Snort 2.1 intrusion detection. Syngress.

Feldmann, A., 2007. Internet clean-slate design: What and why? ACM SIGCOMM Computer Communication Review, 37(3): 59-64.

Griffiths, T.L., M. Steyvers, D.M. Blei and J.B. Tenenbaum, 2004. Integrating topics and syntax. In: Advances in Neural Information Processing Systems. pp: 537-544.

Hawkins, J. and S. Blakeslee, 2005. On intelligence. St. Martin's Press.

Idris, N.B. and B. Shanmugam, 2005. Artificial intelligence techniques applied to intrusion detection. In: Annual IEEE INDICON. IEEE: pp: 52-55.

Khan, S.S., M.E. Karg, J. Hoey and D. Kulic, 2012. Towards the detection of unusual temporal events during activities using hmms. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. ACM: pp: 1075-1084.

Li, D., L. Xiao, Y. Han, G. Chen and K. Liu, 2007. Network thinking and network intelligence. In: Web intelligence meets brain informatics, N. ZhongJ. LiuY. YaoJ. WuS. Lu and K. Li, (Eds.). Springer Berlin Heidelberg: pp: 36-58.

Mimaroglu, S. and D.A. Simovici, 2008. Approximate computation of object distances by locality-sensitive hashing. In: Proceedings of the 2008 International Conference on Data Mining, Washington, DC.

Rabiner, L. and B. Juang, 1986. An introduction to hidden markov models. ASSP Magazine, IEEE, 3(1): 4-16.

Rajpathak, D. and R. Chougule, 2011. A generic ontology development framework for data integration and decision support in a distributed environment. International Journal of Computer Integrated Manufacturing, 24(2): 154-170.

Srivastava, J., R. Cooley, M. Deshpande and P.-N. Tan, 2000. Web usage mining: Discovery and applications of usage patterns from web data. ACM SIGKDD Explorations Newsletter, 1(2): 12-23.