Multi-channel Network Flow Classification for Information Filtering (Postprint)

Authors: Wang Peng, Li Jun, Zhang Peng

Date: 2017-03-10T00:00:00+00:00

Abstract

With the rapid development of information technology, information security has attracted increasing attention from society at large. Among these concerns, network content security represents one of the most prominent issues, while network data stream filtering technology, as the core technology for processing network content security, is also facing new challenges. This paper investigates techniques for classifying network data streams using multi-channel information, starting from the problem of network data stream filtering. The research comprises three main contributions: (1) A study on multi-channel network flow classification models, proposing a flow classification model that can fuse network structure information and network content information; (2) A study on classification model indexing techniques, proposing an R-Tree-based classification model index structure that substantially improves the discrimination speed of network data streams; (3) The development of the F9 experimental platform for multi-channel network flow filtering systems, which supports multi-channel network flow discrimination and filtering and can serve as an experimental platform for novel models and algorithms. These three aspects of work systematically investigate multi-channel network flow classification systems for information filtering from the perspectives of model construction, model indexing, and model implementation.

Full Text

Preamble

Vol. 9 No. 3
Information Technology Letter
Research on Multi-Channel Network Flow Classification for Information Filtering
Peng Wang, Jun Li, Peng Zhang

Abstract

With the rapid development of information technology, information security has garnered increasing attention from society. Among the various concerns, network content security stands out as one of the most prominent issues, and network data flow filtering technology—the core technology for network content security processing—faces new challenges. This paper addresses the network data flow filtering problem by investigating techniques for classifying network data flows using multi-channel information. Our work encompasses three aspects: (1) research on multi-channel network flow classification models, where we propose a flow classification model that can fuse network structural information and network content information; (2) research on classification model indexing techniques, where we propose an R-Tree-based classification model index structure that significantly improves the discrimination speed of network data flows; and (3) development of the F9 experimental platform for multi-channel network flow filtering, which supports multi-channel network flow discrimination and filtering and serves as an experimental platform for new models and algorithms. These three aspects systematically investigate multi-channel network flow classification systems for information filtering from the perspectives of model construction, model indexing, and model implementation.

Keywords: multi-channel network flow, information filtering, data stream classification, model indexing, F9 filtering system

1 Background and Research Significance

In recent years, with the proliferation and development of information technology, the Internet has penetrated every aspect of social life. Consequently, the problem of using networks to disseminate malicious information such as reactionary or pornographic content has become increasingly serious, making network data flow filtering and classification critically important. Network data flows comprise various types of structured information, including IP information, URL (Uniform Resource Locator) information, and multimedia content such as text and images. Traditional network content security processing technologies often rely on only a single type of information for filtering and classification—for example, using webpage text or URLs to classify network flows. Such approaches suffer from poor accuracy and are easily circumvented (e.g., by embedding text within images), making them inadequate for practical application requirements. In contrast, classification and filtering that leverage multi-channel information from network data flows offer high precision and robustness against evasion, attracting widespread attention from both academia and industry and becoming a research hotspot in the network security domain.

Multi-channel network flow refers to the aggregation of network content information (such as text streams, image streams, video and audio streams) and network structural information (such as webpage address links, IP addresses, protocol types) corresponding to a single network request during network access.

Since user network access behavior is composed of multi-channel information, we can jointly utilize these channels to comprehensively determine whether a user access contains illegal information.

The core problem of information filtering is how to construct an accurate classification model on multi-channel network flows to correctly discriminate any unknown future traffic. One effective approach is the result fusion-based multi-channel network flow classification model, which first extracts feature information from each channel, constructs separate classifiers for each channel, and then fuses the results from these classifiers to make a comprehensive judgment.

[Figure 1: see original paper] Schematic diagram of multi-channel network flow

2 Related Work

Multi-channel network flow classification technology for filtering represents the core technology for network content security processing, closely related to Deep Packet Interception (DPI) technology [**?**], string matching technology [**?**], information extraction technology [**?**], multimedia feature extraction technology, data indexing technology, and database technology. Numerous efforts in both academia and industry have addressed this problem from various perspectives.

Network flow filtering can be viewed as a data classification problem. VFDT (Very Fast Decision Tree) [**?**] is a decision tree specifically designed for data stream classification that grows incrementally in correspondence with data stream patterns. However, VFDT cannot handle concept drift in data streams. CVFDT (Concept-adapting Very Fast Decision Tree) [**?**] solves this problem by continuously pruning the decision tree and generating new decision branches to accommodate new patterns, effectively addressing concept drift in data streams. VFDTc [**?**] extends VFDT to handle continuous attributes and concept drift situations. While such data stream classification methods exist, they do not allow manual customization of filtering rules and are therefore unsuitable for network flow filtering in the information security domain.

Currently, there are also application systems internationally that specifically manage and discriminate data streams. A representative example is Stanford University' s STREAM (STanford stREam datA Manager) system [**?**]. It supports CQL (Continuous Query Language) [**?**], a data stream query language similar to SQL[1]. Through CQL, users can register continuous queries to perform query operations on data streams. CQL supports most SQL syntax, but due to the special nature of data streams, such queries always target a specific time window and return approximate results. If registered continuous queries are regarded as classification rules, the STREAM system can be considered a network flow classification and filtering system that supports complex rules, but it lacks support for complex filters.

[1]SQL: Structured Query Language, a standard language for relational database management systems

Despite the numerous existing solutions, several shortcomings remain: (1) these solutions can only handle single-form data streams and cannot process multi-channel network data flows; (2) most research on data stream classification focuses solely on how to build classification models with emphasis on classification accuracy, without considering classification speed—yet in high-speed network flow filtering, speed is as important as accuracy; and (3) currently developed systems are tested in simulated environments, lacking large-traffic and high-intensity testing in real network environments.

3 Introduction to Our Research

Focusing on three aspects of multi-channel network flow classification—model construction, model indexing, and system development—we have conducted research on ensemble classification models for multi-channel network flows, R-Tree[2]-based classification model indexing techniques, and the construction of the F9 experimental platform. The classification model research addresses fundamental theoretical problems in network flow classification; the model indexing research primarily solves real-time discrimination problems; and the F9 system development work mainly addresses model testing and application transformation issues. These efforts mutually reinforce each other, forming an integrated whole.

3.1 Multi-Channel Network Flow Classification Model for Filtering

When fusing information from various channels, decisions from each channel may contradict each other. As shown in Figure 2 [Figure 2: see original paper], for a webpage containing multimedia content, the discrimination results from the text channel and image channel may differ at the content level. From the network structure perspective, the IP address channel or webpage address channel may also produce different decision results. In summary, when fusing information from various channels, an inevitable problem is that decisions from different channels may conflict.

Furthermore, when fusing discriminators from multiple different channels, the decision model must consider: (1) the domains of the discriminators differ—they may target different objects such as text, images, or webpage addresses; (2) the discrimination capabilities of the discriminators vary—since training costs on data streams are high, we can often construct only a small number of classifiers, while constructing clusterers is relatively easier; and (3) the discrimination capabilities of each discriminator change over time—since data streams evolve continuously, the discrimination ability of each discriminator generally decays over time.

We propose a decision model combining clusterers and classifiers that addresses

[2]R-Tree (real-tree) is a multidimensional extension of B-tree that partitions space objects by range, with each node corresponding to a region and a disk page. It is currently a popular spatial indexing method. See §3.2 for details.

these three issues through a three-step approach from the perspective of decision fusion:

First Step: Decision Encoding This essentially solves the similarity measurement problem between different models. For example, suppose we need to classify a webpage into one of three categories (high risk, medium risk, low risk), and we have trained four different types of discriminators 1, 2, 3, and 4, where the first two are classifiers and the latter two are clusterers. For seven incoming webpages x1, ···, x7, assume the discrimination results are as shown in Table 1 , where gi represents the basis for the corresponding encoding. Since this is a three-class classification problem, each model uses three bases, and each sample corresponds to a coordinate—for instance, webpage x1 is classified as class 1 by model 1, with coordinates [1, 0, 0]. This allows us to measure similarity between two bases using Jaccard distance[3]. For example, the similarity between bases g4 and g8 is 2/3, while the similarity between g8 and g9 is 1/5, indicating that g4 and g8 are more similar.

Second Step: Decision Propagation After calculating the similarity between various bases, the next problem is how to propagate similarity among these bases. The goal is to map all bases corresponding to clusterers (Cluster ID) to actual class labels (Class Label). For a clusterer with unknown labels, our basic idea is to first combine all classifiers to derive the clusterer' s label, then use other clusterers to refine the result. This is because when there are few classifiers, relying solely on them cannot derive precise labels, and we need to utilize structural similarity among clusterers for correction.

Third Step: Decision Negotiation When underlying patterns in the data stream change continuously, the role of each model in decision-making will change over time. Therefore, according to the "most recent, most similar" principle for patterns in data streams, each model is weighted based on its similarity to the most recent model. The final weighted average is used for the ultimate decision.

Through these three steps, we can construct a decision (result) fusion model on multi-channel network flows at a relatively low cost. As shown in Figure 3 [Figure 3: see original paper], test results on the UCI Malicious Website Detection dataset (left) and the KDDCUP' 99 Intrusion Detection dataset (right) demonstrate that the classification model fusing information from multiple channels (ECU) achieves higher accuracy than any previous single-channel classification models (EC1 and EC2). Note that accuracy here refers to the proportion of correctly classified data among all data, ranging from 0 to 1.

3.2 R-Tree Based Classification Model Indexing

Decision trees are a type of classification model that is simple to construct and fast to execute, making them well-suited for network flow classification. When

[3]Jaccard distance measures the dissimilarity between two sets by the proportion of elements that differ between them.

implementing multi-channel data stream filtering, using a single decision tree often fails to meet practical application requirements in terms of classification accuracy. Ensemble classifiers can effectively solve this problem. An ensemble classifier uses multiple classifiers to classify data and then synthesizes the results from each classifier to obtain the final classification result. Experiments show that using ensemble classifiers for multi-channel data stream filtering can achieve satisfactory accuracy, but the time overhead increases linearly with the number of classifiers (as shown in the experimental results in Figure 5 [Figure 5: see original paper]). In high-speed network flow environments, this is unacceptable. By indexing the decision trees of ensemble classifiers using an R-Tree-based index structure, classification time overhead can be significantly reduced, making ensemble classifiers feasible in high-speed network data stream environments.

We now introduce the R-Tree [**?**, **?**, **?**, **?**]. Proposed by Antonin Guttman in 1984, the R-Tree is a multidimensional extension of the classic B-Tree index structure. Like B-Trees, R-Trees are height-balanced multi-way search trees and external memory index structures, though their concepts easily extend to in-memory indexing. An R-Tree node is an array of index records. For leaf nodes, index records have the form (I, tuple-ID), where I is an n-dimensional rectangle representing a spatial object and tuple-ID is the identifier of that spatial object. For non-leaf nodes, index records have the form (I, child-pointer), where I is an n-dimensional rectangle and child-pointer is a pointer to a child node at the next level, with I being the minimal rectangle covering all rectangles in the child node.

The R-Tree retrieval process addresses a simple problem: given an n-dimensional rectangle, determine which rectangles in the index cover it. The retrieval process traverses from the root downward. However, unlike B-Trees, since rectangles in each node of an R-Tree may overlap, every index record in a node must be examined sequentially, potentially requiring traversal of multiple subtrees of the current node. In the worst case, R-Tree retrieval may need to access all nodes in the tree—meaning it might need to compare data samples with classification rules one by one. On average, however, retrieval can be completed by accessing only a few nodes.

Since R-Trees are spatial indexes, they can only index continuous attributes. Additionally, due to the sparsity of high-dimensional data, R-Trees face difficulties when handling ultra-high-dimensional data. Our main contribution is adding a layer of hash structure on top of the R-Tree to index discrete attributes in the data, using hashing to find the appropriate entry node into the R-Tree. The data structure is shown in Figure 4 [Figure 4: see original paper].

We conducted tests on artificial datasets (Figure 5-I), the UCI Malicious URL Detection dataset (Figure 5-II), the UCI Spam Email Detection dataset (Figure 5-III), and the KDDCUP' 99 Intrusion Detection dataset (Figure 5-IV). The results demonstrate that using the R-Tree-based index structure significantly reduces time overhead compared to not using an index structure. Moreover,

after indexing, the classification time overhead becomes insensitive to increases in the number of classification models in the ensemble classifier.

3.3 Multi-Channel Network Flow Classification and Filtering Experimental Platform: F9

Real network environments are complex and unpredictable, causing many theoretically sound algorithms and models to perform below expectations in practice. Furthermore, due to the complexity of network flows, artificially generated experimental data struggles to simulate real data. The design goal of the F9 system is to capture network data flows from real network environments, evaluate the performance of models and algorithms, and serve as a prototype system for real data stream filtering.

The F9 system implements the following functions:

1. **Data Stream Analysis**: This function uses an analysis engine to effectively perform protocol restoration and flow reassembly on data streams from high-speed data flow gateways, then uses the engine' s matching algorithm to efficiently match the restored data streams against rules to detect flows that satisfy the matching conditions.
2. **Rule Base Management**: System administrators dynamically add or delete filtering rules through a configuration management interface. The feature extraction and analysis module passes dynamically generated rule data to the analysis engine, which dynamically adjusts its in-memory data structures to activate the new rules.
3. **Network Connection Blocking**: Through dynamic configuration of blacklists and whitelists and using the system' s series connection access method, real-time blocking of specific data flows is achieved.
4. **Protocol Restoration and Analysis**: Through effective protocol restoration of data flows in the gateway, content extraction is performed according to protocol categories.

F9 adopts a design combining series filtering control and bypass monitoring to achieve real-time and efficient supervision and control of multimedia streams. The system mainly includes: a connection blocking module, protocol restoration and analysis module, data analysis module, rule discovery module, and rule base management module. The connection blocking module performs effective address analysis and necessary interception and blocking of data flows based on managed blacklists. The protocol restoration and analysis module uses bypass listening technology to import media stream data from network cards into data analysis servers, then performs effective protocol analysis on the imported raw data streams to identify the protocols carried by the data flows and extracts content according to the rules of the identified protocols, such as images, text, webpage addresses, etc. The data analysis module calls relatively independent analysis processes within the analysis engine based on the category

of data information imported from the protocol restoration and analysis module to determine whether the imported data information is sensitive; if it is sensitive, its address information is recorded and imported into the blacklist used by the connection blocking module. The rule discovery module uses clustering technology to quickly extract and aggregate data streams (including text and images) obtained from the content analysis server, displays the clustering results in page form, and writes them into the rule base dataset based on manual selective labeling. The rule base management module mainly completes add, delete, modify, and query operations on the rule base through the interface, along with some auxiliary permission control operations. The workflow of the F9 system is shown in Figure 6 [Figure 6: see original paper].

4 Conclusion

Multi-channel network flow classification technology for filtering is the core technology for network content security processing. Utilizing multi-channel information for data flow filtering offers tremendous advantages in accuracy and anti-circumvention compared to traditional methods. In recent years, our research on multi-channel network flow classification has made significant progress, including the development of accurate flow classification models, the construction of efficient model index structures, and the development of the application-oriented F9 multi-channel network flow filtering platform. These complementary efforts form an organic whole, laying a foundation for future in-depth research in multi-channel network flow filtering.

References

[?] Domingos, P., Hulten, G. (2000) Mining high-speed data streams. Proceedings KDD 2000, ACM Press, New York, NY, USA, pp. 71–80.
[?] Hulten, G., Spencer, L., Domingos, P. (2001) Mining time-changing data streams. Proceedings KDD 2001, ACM Press, New York, NY, pp. 97–106.
[?] Gama, J., Fernandes, R., & Rocha, R. (2006) Decision trees for mining data streams. *Intelligent Data Analysis* 10 23-45.
[?] A. and Babcock, B. and Babu, S. and Cieslewicz, J. and Datar, M. and Ito, K. and Motwani, R and Srivastava, U. and Widom, J. (2004) STREAM: The Stanford Data Stream Management System. Technical Report. Stanford InfoLab.
[?] A. Arasu, S. Babu, and J. Widom. The cql continuous query language: Semantic foundations and query execution. Technical report, Stanford University Database Group, Oct. 2003.
[?] UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data, http://archive.ics.uci.edu/ml/datasets/URL+Reputation
[?] Antonin Guttman: R-Trees: A Dynamic Index Structure for Spatial Searching, Proc. 1984 ACM SIGMOD International Conference on Management of Data, pp. 47–57. ISBN 0-89791-128-8
[?] Yannis Manolopoulos, Alexandros Nanopoulos, Apostolos N. Papadopoulos,

Yannis Theodoridis: R-Trees: Theory and Applications, Springer, 2005. ISBN 1-85233-977-2
[?] N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger: The R-*Tree: An Efficient and Robust Access Method for Points and Rectangles. SIGMOD Conference 1990: 322-331*
[?] Scott T. Leutenegger, Jeffrey M. Edgington and Mario A. Lopez: STR: A Simple and Efficient Algorithm for R-Tree Packing
[?] Dr. Thomas Porter (2005-01-11). "The Perils of Deep Packet Inspection" . Security Focus. Retrieved
[?] S. Wu and U. Manber, "A fast algorithm for multi-pattern searching" , Dept. of Computer Science, University of Arizona, Tucson, AZ, TR-94-17, 1994
[?] R. K. Srihari, W. Li, C. Niu and T. Cornell, "InfoXtract: A Customizable Intermediate Level Information Extraction Engine" , Journal of Natural Language Engineering*, Cambridge U. Press, 14(1), 2008, pp.33-69

Author Biographies

Peng Wang: Ph.D. candidate, Information Security Research Center, Institute of Computing Technology, Chinese Academy of Sciences, wangpeng@software.ict.ac.cn
Jun Li: Ph.D. candidate, Information Security Research Center, Institute of Computing Technology, Chinese Academy of Sciences
Peng Zhang: Ph.D. candidate and Assistant Researcher, Information Security Research Center, Institute of Computing Technology, Chinese Academy of Sciences

Note: Figure translations are in progress. See original paper for figures.