Four Decades of Data Mining in Network and Systems Management

Khamisi Kalegele, *Member, IEEE,* Kazuto Sasai, Hideyuki Takahashi, Gen Kitagata, Tetsuo Kinoshita

Abstract—How has the interdisciplinary data mining field been practiced in Network and Systems Management (NSM)? In Science and Technology, there is a wide use of data mining in areas like bioinformatics, genetics, Web and more recently astroinformatics. However, the application in NSM has been limited and inconsiderable. In this article, we provide an account of how data mining has been applied in managing networks and systems for the past four decades, presumably since its birth. We look into the field's applications in the key NSM activities – discovery, monitoring, analysis, reporting and domain knowledge acquisition. In the end, we discuss our perspective on the issues that are considered critical for the effective application of data mining in the modern systems which are characterized by heterogeneity and high dynamism.

Index Terms—Data Mining, Network and Systems Management, Machine Learning.

1 Introduction

Data mining involves methods at the intersection of artificial intelligence, machine learning, pattern recognition and statistics, to mention a few. The field aims at extracting interesting information (patterns) from datasets and transform that information into understandable structures for further use [1]. The very nature of this aim makes data mining process a multifaceted problem with components like data preprocessing, retention, information and pattern modeling, interestingness metrics, algorithmic complexity, visualization etc. Clearly, outside a domain-specific context, there can only be a generic and inexplicable discussion about data mining. This explains the sprawl of domain-specific mining research platforms – bioinformatics, financial analysis, telecommunications [3], genetics [4], astroinformatics [5] etc. Astroinformatics [5] is one of the most recent introductions in data mining. As this paper title suggests, we focus on the context of Network and Systems Management (NSM). We will use the words "management" and "monitoring" interchangeably. The discussion presented by this paper covers selected papers which reflect specific areas where mining has been applied. The application areas are those within the scope of NSM activities [6] (discovery, monitoring, analysis, reporting and domain knowledge) as translated from the FCAPS (Fault, Configuration, Accounting, Performance, Security) ISO standard. The discussion also includes the authors' perspective on issues which either need improvements or they are bottlenecks to the effective use of data mining in NSM.

- *K. Kalegele is with the School of Computational and Communication Sciences and Engineering at the Nelson Mandela African Institution of Science and Technology.*
  E-mail: khamisi.kalegele@nm-aist.ac.tz, kalegs03@gmail.com
- *K. Sasai, H. Takahashi, G. Kitagata and T. Kinoshita are with the Research Institute of Electrical Communication at Tohoku University.*

Data mining application areas in discovering monitored objects, monitoring desired behaviors, analyzing target data, summarizing desired information for reporting purposes and acquiring new domain knowledge are vast. When such kind of application opportunities are seized, they can help in dealing with faults, configuring monitored objects, accounting for some concept of interest, improving performance and securing networks. In practice and literature, we have seen opportunities seized in traffic data analysis in order to secure networks [7], Web pages link data analysis in order to optimize the Web [8], software trace analysis in order to detect and isolate software bugs [9], [10], [11], etc. The literature has covered the way various techniques are used to address specific areas of NSM in specific domains, though not sufficiently. In [12], for example, approaches to fault management in Wireless Sensor Networks are explained. In all the numerous application areas, the core role and the concept of data mining remain unchanged, i.e., to inductively extract important information (patterns) from collected data and subsequently transform it into an understandable structure for further utilization. For students and researchers who wish to devise a mining-based approach in network management, an understanding of how the techniques have been used would be very helpful. In this article, we review the process of mining data, selectively provide brief but strong insights about possible data mining application areas, present the various published works in each area, and finally we provide a general discussion which includes the authors' perspective on bottleneck issues and various frontiers. The rest of this section provides introductions to data mining and NSM.

1.1 The data mining process

Selection, *preprocessing*, *transformation*, *mining* and *interpretation* are the typical steps in a mining process as in Fig. 1 (Han [13]). The process extracts important information

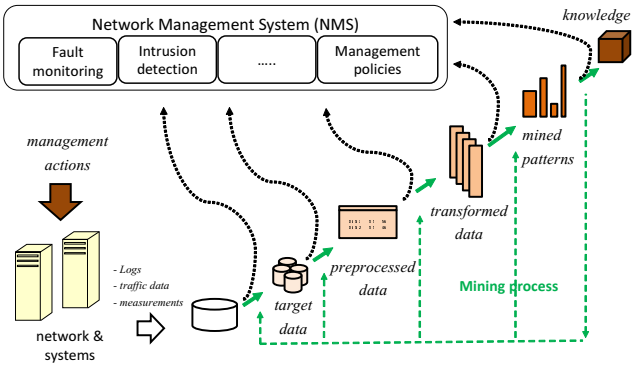

Fig. 1. The data mining process.

from data and transforms it into understandable and easy-to-use structures so that it can be presented as knowledge for further utilization. In *selection*, relevant target data is selected from retained data (typically very noisy) and subsequently *preprocessed*. This goes hand in hand with the integration from multiple sources, filtering irrelevant content and structuring of data according to a target tool (e.g. ARFF format of WEKA tool [14]). Although these steps can be considered trivial, preprocessing in particular, they have often spoiled many mining initiatives. By using ad-hoc tools, the preprocessed target data can be conveniently *transformed* as found fit in order to achieve the desired objectives, for instance, transformation of network flow intensities at selected measurement points into flow dynamics using time series techniques [15] (Section 3.3 provides details). Afterwards, machine learning algorithms, statistical and visualization techniques etc. are applied on the transformed data during analysis in attempts to identify interesting information patterns (e.g. association between items). Finally, the patterns are *interpreted* and *evaluated* into usable knowledge models (e.g. a neural network model for SPAM filtering). In many occasions, the process will require a repetition of the above stages in a recursive manner until the desired outcome is obtained.

In Section 3, mining applications are discussed in line with these steps. It will be seen that the process is heuristic, intriguing and very difficult to undertake without domain knowledge. For example, in the previous example of transforming flow intensities using time series modeling, one can not tell the impact of time granularity when making a time series out of the measurements without the knowledge about the flow intensities. The process is also very subjective. This is because; (1) it is too difficult to measure the outcome because it is limited by the properties of the used data. Proper data recording and warehousing can improve reliability of an outcome measure. However, many institutions are dilatory in implementing serious data retention policies. The various oversights during data recording and warehousing usually prove to be costly later; (2) An analyst's selection and preprocessing of data are substantially influenced by past experiences.

1.2 Network and systems management

Managing a network or a system means ensuring normal operations of its intended services, securing it and optimizing its performance in all aspects. The to-be-ensured objectives are dealt with from the perspective of a target specific domain, e.g. networking domain, and applications. Regardless of the domain, the following activities are usually undertaken when pursuing the objectives.

Discovery of the deployed assets (both hardware and software) and their inter-connectivity: Discovery can be done manually (e.g. using dictionaries and directories) or automatically (e.g. using self-advertisement techniques). More sophisticated approaches like traffic analysis and agent mining approaches can also be used. This activity is not trivial as mistakenly always considered.

The discovered assets and the offered services need to be *monitored* by constantly obtaining their operational status, configuration and topological information, and usage data from the IT infrastructure. The acquired information can then be consolidated into management data for further use and exploitation (e.g. analysis and reporting). The consolidated data constitutes data from systems, including application-related, and from network, including measurements. We refer to this data as network and systems generated data (NS-data).

By *analyzing* NS-data, solutions to ensure the above objectives are met can be devised. These include solutions to manage faults, configurations, assets accounting, performance, and security breaches (and vulnerabilities). Analysis is the most well-thought-of activity, perhaps because it is an activity that is central to key NSM operations, and probably the most difficult one, which requires critical and analytical thinking. Any method, based on any technology, can be used as long as the desired objectives are met. There is nothing like an "appropriate analysis" method. The commonly used approaches and technologies include agent technology, machine learning, statistics, fuzzy etc.

During an activity, the ability to *report* is equally vital. Usually, normal communication means (e.g. Email, SMS) are relied upon. Data mining approaches can be used in data description and summarization in efforts to improve the quality of the information being reported. If not well planned and organized, poor reporting can easily result into loss of corporate support for some NSM activities.

The outcome of management data analysis can lead to *reconfiguration* of the respective assets, for instance, a drop in performance or existence of a faulty component. Due to the large number of configurable assets, configuration operations can be facilitated by technologies like data mining and agent technology. This comes in three flavors – (1) Retention of existing configuration details – this can also be regarded as a part of the discovery activity, (2) Knowledge about configuration parameters – this knowledge must be acquired and maintained, and (3) Simulation and enforcement – any change in configuration parameter must not have negative effects on a system, and the changes are to be enforced in a harmless manner.

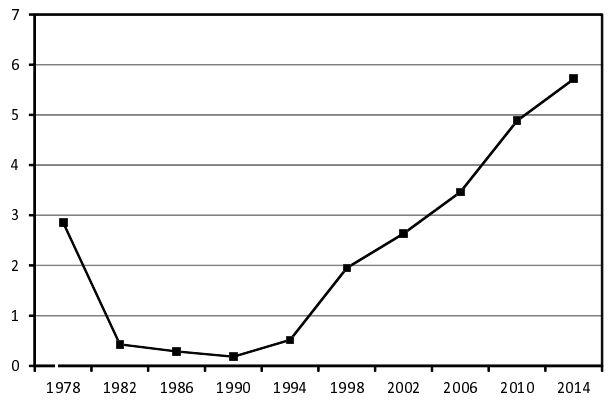

Fig. 2. The proportion of mining papers with application in NSM

2 Overview of the Four Decades

Over the years, the role of network and systems in our lives has been changing and so is the way data mining has been applied. In the 1970s, computers were used primarily to store data and to control production processes. Thus, unsurprisingly, some of the first applications of the concepts of data mining were in achieving Database Management System (DBMS) dependencies [16]. Dale Knudson [16] used system call trace data to find ways of achieving DBMS dependencies, though it is unclear whether he actually succeeded. Today, network and systems have taken hold of our lives. We literally can not live without them. The implication is that systems generate more NS-data, which reflects many aspects of our lives. Most failures occur because of human errors. Malicious users will cause the systems to generate data, which constitute an anomaly. NS-data can therefore be mined in order to identify the existence of the aspects, which reflect our practices. In this section, we narrate how the concepts of mining have been employed over the four decades. Our narration is based on publications (mostly IEEE, Springer and ACM), which have been categorized to contain "*data mining*" keyword. Some papers are not tagged by "*data mining* keyword," nonetheless they are categorized by publishers as to contain such keyword. Also, some of the surveyed papers come from renowned top-tier conferences like ICDM and ICML.

2.1 During 1980s

It was in the 1980s when data mining concepts became significantly important [17]. The paper [17] proposed a user-oriented performance index in an attempt to address the issue of having too many performance measures (e.g. grade of service, throughput) in packet switched networks. In the approach, three key measures (R– Rate, D– Delay and Q– Quality) were selected for the calculation of the performance index. Rate meant throughput, Delay meant data transmission time and Quality reflected human perceived grade of service. The three measures or dimensions were reduced in dimensionality into a single index, P, by learning a function such as a combinatorial computation ($P = D \times R \times Q$) or the one shown below,

$$P = w_d \times D + w_r \times R + w_q \times Q$$

where w is an induced weight.

In 1989, an unsupervised learning process was employed in the selection of electrical features which were most significant for accurate modeling of cable-motor system conditions [18]. In the process, they used a modeling software package PNETTR-4X to form polynomial networks called Adaptive Learning Networks (ALNs). ALNs were then used as feature selectors. This serves as an example of application of data mining on computer systems which were used to control production or industrial systems.

2.2 During 1990s

Many applications of data mining in NSM were proposed and the proportion of publications started to increase (see Fig. 2) as mining became more popular. Processes which were previously referred to as *data dredging* or *fishing* became known as data mining. One obvious reason for the increase in popularity is that information-rich communication networks became tougher to manage and alternative methods were being sought. Prominent application areas were in mailing systems [19], [20], networking [21], [22], [23], [24], [25], industrial process monitoring [26], electrical systems [27], telecommunication [28] and protocol analysis for accounting [22]. In most applications, techniques which included Neural Network (NN) and Principal Component Analysis (PCA) were adopted from their parent disciplines in order to simplify desired tasks. The mining process was still very simplistic; for instance, learning an NN to reduce dimensionality of the problems and extract attributes from NS-data, which best described the problem [26]. In a paper from AT&T [20], linking and interaction principles were used to build hierarchies, which facilitated navigation of large networks. The hierarchies helped in network exploration when answering questions like, for instance, "*Do similar people send similar amounts of mail*? *To similar people*?"

There were some sophisticated proposals in various NSM areas. In fault management, for example, fault scenarios were detected from selected Management Information Base (MIB) variables using an approach based on sequential Generalized Likelihood Ratio (GLR) [29]. In the approach, a GLR test was conducted to detect anomalies in MIB, and the information was temporally correlated with observed network faults and performance issues.

Later in the 1990s, there were proposals on using data mining to detect intrusions. As more people started to use emails for communication, intrusive events increased. One such proposal used "Repeated Incremental Pruning to Produce Error Reduction (RIPPER)" algorithm [30] to classify abnormality of system call sequences in a SENDMAIL program [19]. There were also uses of other technologies to either improve or facilitate the mining process. In the above RIPPER algorithm based approach [19], intelligent agents were used to facilitate the process.

2.3 During 2000s

In the years around 2000s, data mining became more vital for managing and securing operations. This is mostly attributed to the wide spread use of the Internet and the increasing online threats. Approaches to fuse data and exploit it were devised. For instance, Kangping et. al [31] fused reported service troubles and real time alarm messages by temporal and spatial associations to produce sequential events. The events were then mined for what they referred to as episode rules. An example of such rules is, "*If Type A alarm occurs, then Type B alarm will occur in 5 seconds with a probability of 80%*."

Beside faults and alarms management, mining concepts were also used to improve service delivery. A good example is in the then starting-to-increase *e*-commerce. Approaches like NN were used in Park et. al. [32] to mine customer data in order to deliver personalized services. Others include approaches to effectively manage network performance data for more effective analysis (e.g. fault, policies). In Gao et. al. [33], an approach to extract management information from MIBs of distributed systems was proposed. The extracted information formed the management knowledge base, which could be accessed seamlessly and dynamically to cater for management policy requirements. Developments in the various areas (especially data management) enabled researchers and analysts to conduct *mining* on heterogeneous data. Historical datasets were not singly mined. Mining was conducted on data which was correlated from historical data, configuration events and traffic data [34]. From the mid-2000s, applications on intrusion detection started to increase due to online threats (Fig. 2 reflects this). Detection of anomalous users, connections and malicious code, to mention a few, became the key to detect intrusions. System usage datasets and streams were mined in order to profile users using various techniques including clustering based methods [35]. Various network connections were mined in order to detect anomalous connections using techniques such as fuzzy-based [36], association rules [37], classification methods [38] etc. Time series concepts were also used to analyze network data. In Tianqi Yang [39], HTTP traffic data was treated as time series data and instead of applying the traditional approach of principal component analysis (PCA), AutoRegressive Moving Average (ARMA) and Hopfield models were employed to analyze the series. In efforts to supplement traditional signature-based malicious code detection approaches, there were efforts to mine various data in order to detect new malicious binaries [40].

Intrusion Detection Systems (IDS) became such a common place that proposals to optimize existing approaches started to increase [41]. In Yang Li et. al. [42], an approach for data instance selection and feature selection, which was based on Genetic Algorithms was proposed. The approach aimed at improving the quality of data, which is used by IDS in order to improve performance. In a similar category [67], a new method (instead of Information Gain) of determining the goodness of an attribute in order to minimize false positives of an IDS classifier is proposed.

In the late 2000s, the various tools which were developed by the data mining community motivated many analysts to evaluate existing algorithms, develop new ones and use other assisting technologies (e.g. agents, cloud computing) more easily. For instance, in the Machine Learning framework, WEKA [14], the number of plugins for achieving data manipulation tasks increased tremendously within five years. In Gorodetsky et. al [38], agent technology was used to assist mining processes in a ubiquitous environment.

In [43], *K*-Nearest Neighbor (*K*NN) method was used to identify poison message failures between components in IP networks. A testbed called OPNET was developed for the simulation of MultiProtocol Label Switching (MPLS) network in which the poison message could be a message in Border Gateway Protocol or Label Distribution Protocol.

It became much easier to conduct a mining process such that many application proposals were made in many areas using various techniques. Traffic data was being effectively analyzed [44], [45], program behaviors were being easily modeled [9], [46], [47] (to detect, for example, malicious activities), system component specific anomalies became a lot easier to detect [15], [48], Web page linking optimization became more efficient [8], and so on.

2.4 During 2010s

By 2010s, there has been a proliferation of data mining application proposals in order to cope with the new technological challenges. The increase in reliance on networked devices has led to a deluge of NS-data in our networks and thus new challenges in terms of retention and exploitation. New techniques and concepts like virtualization of resources have opened many avenues for the application of data mining. As a result, mining tools have become even more sophisticated. Approaches for acquiring and integrating the relevant datasets for mining have consequently become vital. The approaches cover many aspects including architectural issues [49], integration and correlation issues [49], [50], data retention issues [50], data interpretation issues [50] etc. More efficient algorithms with low False Positives and new evaluation methods are now needed. Also, due to NS-data abundance, the significance of supporting computational technologies (e.g. cloud computing) can now be easily demonstrated.

On the other hand, improvements in the way mining methods and algorithms are used have been tremendous. Some improvements have been necessities while dealing with emerging technologies. This has been demonstrated in many areas including in anomaly detection by using, for example, PCA [51], detection of bandwidth service violation [52], congestion control methods (for example, in SIP overload control [53]), SMS Spam filtering [54], intrusion detection [55], [67], feature extraction [56], post-processing malwares [57], and in parameter tuning for intrusion detection systems [58] etc.

Virtualization concepts have brought new difficulties because of the fact that isolation during analysis is becoming increasingly abstract. Example of such difficulties is in

isolation of faulty components by conducting a root cause analysis [59]. In He Yan et. al. [59], various datasets (router configuration data, layer-1 alarms, SNMP MIBs, routing data, end-to-end measurements) were used to build spatial models based on user-defined dependency rules. The spatial models were then used to isolate root causes.

Another computing challenge is on Next Generation Networks (NGNs). [60] gives a good technical narration of how mining concepts can be used to automate tasks.

The sophistication of tools and well-tested algorithms have created a completely new landscape. On one side, there is everything for the undertaking of effective mining processes. On another side, the efficient usage of the specialized tools and platforms is proving to be a huge challenge to miners who are still fighting to grasp an understanding of the many virtualized services and platforms.

2.5 Summary

The basis of data mining has not changed over the decades. Miners are still searching for useful patterns in NS-data. Depending on the target application, useful patterns can either be those which are frequent [61], [62] or those which are infrequent [63] or those which have been generalized. Associations based on frequent items are useful in, for instance, discovering communities and correlating events. Those based on infrequent items are useful in, for instance, identifying the rare intrusive and failure events [63].

In the 1980s, patterns were used in very simplistic ways to implement simple NSM operations. This was done at network levels which reflected some basic concepts like performance due to network delays. In the 1990s, patterns were used in learning specialized models (e.g. NN classifiers). During this period, intensive automation in many industrial, telecommunication and computer based communication systems was taking place. Mined patterns in terms of specialized models were mostly used to support engineers in managing the systems. In the 2000s, the discovered patterns were used in generating operational rules which were to be incorporated into monitoring systems to make them intelligent. In this period, the use of internet had significantly increased. Mining from datasets which were results of integration from multiple sources became the norm. A key trend during this period was the increase in analysis tools and the implementation of many algorithms. To date, we are still searching for patterns in NS-data. However, the 2010s marked a very special era of application of data mining in NSM. This is due to two key aspects – the tremendous amount of NS-data which is daily generated and the struggle to grasp understanding of the impact of the introduction of new concepts. Traditional approaches have failed to deliver in wake of serious cyber threats due to their data intensive processing nature. Currently, visualization techniques are aggressively being explored for possible alternatives to cyber security.

3 Application of Data mining in NSM

Today, data mining is an integral part of NSM operations. With the exception of reconfiguration activities, mining concepts can be applied in any of the NSM activities (refer Section 1.2). In *discovery* activities, mining approaches can be used in profiling deployed assets (i.e. deployed hardware and software, configuration, services, registered users, traffic data patterns, etc.) The profiles can then be used in modeling for the purpose of detecting new assets and accounting for the existing ones. *Monitoring* and *analysis* approaches are also common. The approaches are used to infer the various patterns, which are found in operational status data, usage data and configuration data of the deployed assets. These patterns are crucial in detecting and optimizing failures, performance and security breaches as it has been demonstrated in many cases [8], [64], [65], [66]. Other non-trivial mining tasks, which are commonly employed, are data description and summarization. These are useful in *reporting* activities by a management center. In this section, we explain selected cases for each activity (discovery, monitoring, analysis and reporting) in which data mining has been applied. Cases which represent an interesting mining style (based on mining steps described in Section 1.1) will be covered in detail.

3.1 Discovery

Discovery of assets in a network is not given much significance by most NSM researchers. We could not find articles, which either address a problem or propose a solution regarding discovery of assets. However, some system implementers and technologists have paid attention to the discovery of assets – discovery of deployed assets and discovery of inter-connectivity. Approaches to discovering the two asset categories (*deployed assets* and *interconnectivity* of assets) [6] are quite similar. The Intelligent Device Discovery (IDD), by the IBM Zurich Research Laboratory, discovers information technology assets. IDD uses active mapping, scanning and indirect discovery; it also can employ passive discovery based on traffic analysis. Active mapping and scanning involve active probing while indirect discovery involves asking third parties. Traffic analysis method is presented as a supplementary method to the above three. This method is the one which can employ data mining concepts and techniques. Traffic analysis (mostly TCP/IP) is employed by not only IDD but many other tools (e.g. ManageEngine, NetworkView, AlloyDiscovery, LogInventory, NetXMS, NetSurveyor etc). In the tools, discoveries are conducted by; (1) mining associations between IP addresses and MAC addresses, (2) analyzing IP headers to identify Operating Systems and software packages, and (3) mining system logs and traffic dump files for the discovery of fingerprints.

Discovery can also be conducted for better service management. Static and dynamic configurations and/or component interactions of a service are vital for the service optimization. In [68], a model-based approach and relevant metrics for performance and scalability are proposed for the discovery of static configurations. The approach was applied on e-commerce services, enterprise resource planning applications and Microsoft Exchange Services.

A rule which creates a 10s context SENDING_MAIL when a message enters queue manager:

```
#
type=Single
ptype=RegExp
pattern= ({my $var = get_pattern_qmgr_entry(); return $var})
context= SENDING_MAIL 10
```

A rule which fires when its pattern matches and the SENDING_MAIL context exists :

```
#
type=Single
ptype=RegExp
pattern== ({my $var = get_pattern_qmgr_entry(); return $var})
context= = SENDING_MAIL
desc=$0
action=write - $0
```

Fig. 3. Context pattern matching rule: When two messages enter queue in quick succession, generate an alarm and notify administrator.

3.2 Monitoring

Monitoring in NSM operations is in many aspects similar to data warehousing. They both involve collection, retaining, and reporting of data. In NSM, however, the activity is mostly an architectural problem – the most popular architecture being the *manager-agent* architecture. After the architecture has been decided, data mining can be applied in tasks like correlation of, for example, error messages from multiple sources. In order to monitor and report desired concepts (e.g. congestion or a failure), events from various sources within an IT infrastructure need to be correlated. Monitoring of static concepts like topological information is just an extension or repetition of the discovery process (Section 3.1). On the other hand, however, concepts like failures and errors require specialized monitoring solutions which involve domain based analysis. Over the decades, solutions have either been based on pattern matching, association rules or classification.

3.2.1 Pattern matching

Pattern matching is the most common solution due to its simplicity. When matched against recorded NS-data items, patterns (in terms of regular expressions) signify the occurrence of desired monitored concepts. Most NSM tools and approaches use this category as their primary monitoring solution. The tools allow that when a pattern (e.g. *@time\S+.*postfix/cleanup.*:\s+msg-id=.**) is matched, an action which ought to be taken can be specified. Examples of such tools include SEC [69] and NetXMS. Often, pattern matching approaches employ contexts in monitoring more complex scenarios. Fig. 3 shows a context used in a matching rule from the SEC tool [69].

3.2.2 Association and prediction

Association and prediction rules are usually derived from the above explained pattern matching contexts. They are sophisticated versions of pattern matching rules in that they are a result of certain generalization rather than expert's craft work. Prediction rules or models seek to achieve proactive monitoring. Traditional monitoring of waiting until an event is registered by logging systems might not yield desired results in some systems. The next two paragraphs cite examples of these monitoring approaches.

In the Data Fusion Supported Frequent Episodes Discovery (DFSFED) [31], service-oriented reported data and real alarm messages are fused by temporal and spatial associations to mine frequent error episodes. Key steps are 3 fusion phases and an incremental mining phase as follows.

- Noise filtering: The DFSFED uses service-oriented reported data, which might not have been correctly specified by a reporter. Such incorrect specified data (noises, redundancies) are removed in this phase.
- Temporal association: By keeping track of the spans between fault generation and restoration, temporal associations are easily established.
- Spatial association: Derivation of spatial associations is guided by Network Service Dependency Graphs (NSDGs), which represent correlations between services, devices, and faults.
- Incremental mining: Frequent episodes are mined using a support-confidence based association technique.

In the Managed Objects Behavior Patterns Learning (MOBPL) [34], similarly, various NS-data were fused and prediction rules mined. Regression trees were used to learn the patterns from which prediction rules were crafted. Table 1 shows examples of such rules.

Another common monitoring approach is based on signatures [70]. In this approach, failure events are correlated in time (using spherical covariance) and in space (using stochastic model) and then the correlated events are clustered into a tuple-based (e.g. fault id, time, fault type etc.) representation. The representation is called *failure signature* and is used in proactive management of systems.

In terms of categorization, there is a fine line between monitoring (Section 3.2) based on association, prediction, and classification on one side, and analysis (Section 3.3) of NS-data on another side. More cases of association based, prediction based and classification based monitoring are covered in Section 3.3.

3.2.3 Classification

In Freire et. al. [45], an approach and metrics on how to distinguish Skype and HTTP traffic are presented. Skype uses port 80 just like HTTP. Traffic classification of the two protocols can not be achieved based on, for example, difference in port number (a conventional approach). In their work, Chi-squared (χ^2) and the Kolmogorov-Smirnov tests were used to distinguish the distributions of the traffic models for the two protocols. The distributions are useful in, for example, tracking internet usage at a workplace.

STAD [71] is a framework for detecting alerts from logs. In this approach, log files are decomposed in space and time using nodehour (one hour of log information from

TABLE 1
Rules for predicting events of the W3svc service

Rule A	IF	Oldness("WntEventLog=ErrorCodeX")=[50;65]seconds, AND Repitition("NetLogon=ErrorCodeY")=[1;1] in 720 seconds
	THEN	"W3svc=ErrorCodeZ" in [90:129] seconds. Support: 2.7%
Rule B	IF	Oldness("WntEventLog=ErrorCodeX")=[50;65]seconds, AND Repitition("NetLogon=ErrorCodeY")=Never in last 720 seconds
	THEN	"W3svc=ErrorCodeZ" in [451:491] seconds. Support: 1.5%

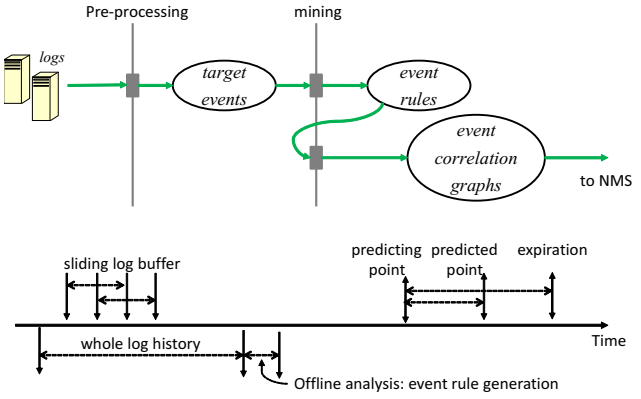

Fig. 4. An event prediction approach: schematic

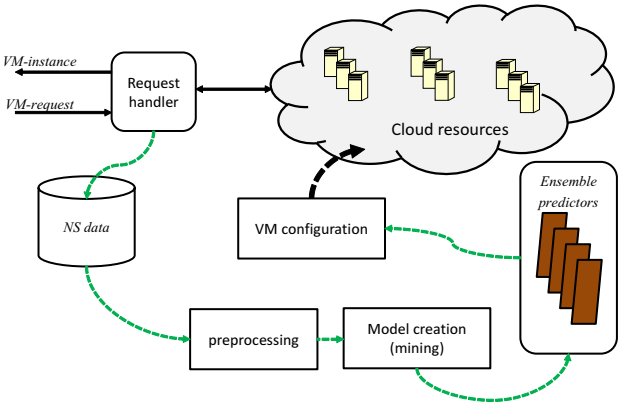

Fig. 5. Virtual machine capacity prediction approach: schematic

a single node) into spatio-temporal units. The units are then grouped using improved entropy-based scores based on the *Nodeinfo-Uniq* equation. Alerts are detected from the anomalous units using a rule based testing approach.

3.3 Analysis

This activity plays central roles in all the others. Mostly, analysis is needed to achieve specialized tasks – that is, predicting a desired event, detecting a desired phenomenon, optimizing a system, classifying items and identifying the root cause of a failure. We established these five to be the key roles, which analysis plays in NSM operations.

3.3.1 Prediction

Prediction is very useful in many aspects. Mostly, prediction approaches have been employed in predicting two key desired events. The first one is *logged events of interest* like errors, and the second one is *desired capacity* for system dimensioning and provisioning.

Fig. 4 shows a typical schematic of a mining process (by the LogMaster [65]) employed to predict failures in cluster systems. The LogMaster mines *correlations of events from the logs* of a cluster systems (Cloud and HPC) using Apriori-like algorithms and Event Correlation Graphs (ECG) in order to predict failures. The traditional Apriori requires a property that, "*any subset of a frequent itemset must be frequent*" to hold. This property does not hold in systems log data. In the LogMaster, a simplified algorithm that generates an n-ary event rule candidate iff its two $(n$-$1)$-ary adjacent subsets are frequent was proposed. This simplified algorithm produces event rules, which are then made into an Event Correlation Graph (ECG). From the ECG, an event prediction algorithm is used to predict failure events. The key to the algorithm are the three timing points, which are shown in the lower part of Fig. 4. As shown in Fig. 4, the algorithm analyses the whole log history by taking one sliding window of events at a time. The confidence-support based algorithm then predicts the occurrence of the desired event sometime before the actual occurrence. For evaluation and testing, the LogMaster was used to analyze logs from a 260-nodes Hadoop system, which ran MapReduce-like applications. HPC cluster logs and BlueGene/L System logs were also used. The logs included /dev/error, /var/log, IBM Tivoli, HP openview, and NetLogger. Evaluation was based on how fast the analysis was done and how accurate were the predictions (non-failure, failure and fatal events).

With paradigms like Infrastructure As A Service (IaaS) on the rise, cloud computing has increasingly became popular. Through cloud systems analytics, various challenges which still hamper cloud computing can be addressed effectively. One of such challenges is the provisioning of accurate resources for effective cloud resource utilization without affecting Service Level Agreements (SLAs). By analyzing data traces of Virtual Machines (VMs) in a cloud, computing capacities can be effectively provisioned. Some published research works [64] have addressed this issue. In Jiang et. al. [64], the capacity provisioning problem is treated as a prediction problem where by an asymmetric and heterogeneous measure called Cloud Prediction Cost (CPC) was proposed to quantify the prediction errors. Fig. 5 shows the schematic of the approach. In the approach, an ensemble

of predictors based on moving average, auto regression, artificial neural network and support vector machines techniques was made. The ensemble becomes responsible for determining appropriate capacities of requested resources and the configurator configures them accordingly.

In another case [72], in an attempt to achieve automated management, a periodicity mining based approach to resource demand prediction (for proactive scheduling) was proposed. In the approach, periodicities (periodical behaviors) were mined using a modified version of a statistical algorithm by Ma and Hellerstein [73], and then used to predict demand. The modifications were on adding change-adaptability. Prediction is done using probability mass functions (pmf) of the historical demands.

3.3.2 Detection – Link analysis

The Web constitutes an enormous amount of operations starting from contents management, Web server administration to link analysis. Most data mining applications, specific to the Web, have been in link analysis.

There is a scramble among website developers to boost ranking of their pages so that they come first in Web search results. There are many ongoing activities to mislead search engines to boost rankings of certain pages. These activities are known as *link spam* activities and their beneficiary pages are referred to as *link spam target pages*. Data mining has been applied in detecting these target pages.

Definition 3.1. *For a page p, the page farm of p is the set of pages on which the PageRank score of p depends. Page p is called the target page.*

In Zhou et. al [8], the concept of page farms is used. The concept involves modeling of Web pages and hyperlinks as directed *Web graph* $G = (V, E)$ where V is the set of Web pages and E is the set of hyperlinks. Using the graph G, a two step approach is employed. In step one, methods based on contributions to PageRank (page by page, or path by path) are used to extract page farms (Definition 3.1). In step two, spamicity score of a Web page is calculated from its page farm to establish the likelihood that the page is a *link spam target*. The used spamicity score is called utility-based spamicity, which is described as follows. If p is the target of a link spam, then the page farm of p should try to achieve the PageRank of p as high as possible. The maximum PageRank can be calculated using the same number of pages and the same number of hyperlinks as p's page farm. The utility-based spamicity of the page farm of p is the ratio of the PageRank of p against the maximum PageRank that can be achieved. The utility provides a measure of the likelihood that p benefits from the link spam.

3.3.3 Detection – Intrusion detection

The ongoing global fight against cyber security threats is all about detecting intrusions and limiting their impacts. By analyzing relevant NS-data, anomalous intrusive activities can be detected, prevented and quarantined. Many published works have demonstrated how data mining can be applied for this purpose. We, herein, present selected cases.

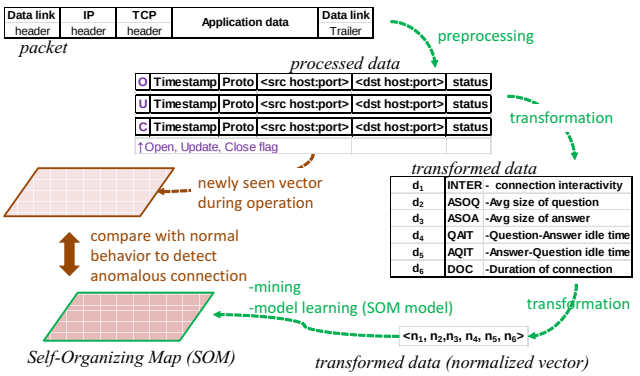

Fig. 6. An anomaly detection approach: data

In the Integrated Network-Based Ohio University Network Detection Services (INBOUNDS) [7], shown in Fig. 6, network connections (traffic) data is characterized into six parameters, which are then used to train a Self-Organizing Map (SOM) for the detection of intrusive connections. In the process, shown in Fig. 6, traffic data (TCP) is summarized into parameters – interactivity per connection, i.e. questions per second (INTER), average size of questions (ASOQ), average size of answers (ASOA), question-answer idle time (QAIT), answer to question idle time (AQIT) and duration of connection (DOC). This processing, indicated as preprocessing in Fig. 6, is followed by a transformation stage in which the parameters are transformed into six dimensional vectors. Normalized vectors are then used to train Self Organized Map (SOM) models. At run-time, a distance-based threshold approach is used to detect the connections as intrusive or not. Connection data to be tested is run through the same transformation as above and transformed into a six-dimension vector. When the vector is fed into the trained SOM, its distance from the training winner neuron is identified. By applying distance-based threshold, the connection type can be detected.

One of the most common types of intrusions or attacks is Distributed Denial of Service (DDoS). There have been too many efforts on this to match the threats. In [74], a methodology for utilizing monitoring systems for early detection of distributed denial of service attacks (DDoS) was proposed. The approach is based on the concept of time series quantization and in the application of the Granger Causality Test (GCT) for the selection of variables that, likely, contain precursors. In terms of NSM activities, integrating monitoring systems with intrusion detection systems is increasingly becoming a daunting task due to volume and heterogeneity of data. Sufficient insight on the concepts behind intrusion detection systems and how they work is very vital. The reviews, [105], [106], present an overview of the systems and associated methods and tools. In [106], anomaly-based intrusion detection was presented in six (6) aspects – input data types, appropriateness of proximity measure of deviation from normality, labeling of data, relevant feature identification, and anomaly reporting. Since intrusion detection can be treated as a pattern

recognition problem, proximity measures are very useful, and sufficiently covered in [106]. In the paper, methods and systems are classified into six (6) distinct classes and each is comprehensively covered. The six (6) classes are statistical [107], classification based [108], clustering and outlier based [109], soft computing, knowledge based and combination learners. Each class is analyzed in detail, examples provided, and sample datasets, which was used to evaluate its methods and techniques, are provided.

3.3.4 Detection – Anomaly/failure detection

Over the decades, a variety of approaches have been employed to detect failures and anomalies. A review of network anomalies, feature selection for anomaly detection and detection methods is provided in [75]. Few selected cases are covered here. One case in the late 1990s involved the use of GLR test to detect anomalies in MIB variables [29]. Detected anomalies were then temporal correlated with observed faults and reported issues in order to infer failures. The process had three key steps: (1) data processing – division of the time series of MIB variables into piecewise stationary segments modeling using an Auto Regressive (AR) process; (2) change detection – determination of GLR and the use of a sequential hypothesis test to determine whether a change has occurred. If there is a change, an alarm is raised based on change (in MIB variable levels) threshold; and (3) combiner or correlator – application of a duration filter on variable levels in order to temporal correlate the changes. This was done on the premise that, a change observed in a particular variable would propagate into another variable that was higher up in the protocol stack. For instance, in the case of $ifIO$ MIB variable, the traffic flow is towards $ifIR$.

In Kiciman et. al. [76], classification techniques are used to localize anomalies for the detection and monitoring of application-level failures in Internet services. In the approach, components interactions are used to model the behavior of a system. The interactions are captured as ordered sets of logical software components, which are used to service a client request. The ordered sets are referred to as *paths* and they have varying shapes. By using decision trees, they are able to correlate anomalies and features of these paths. A decision tree not only tells whether a path possibly correlates with an anomaly but also tells which components were interconnected to lead to the anomaly. From the possible anomalous path-based model, anomalies are verified by measuring the deviation between a single component's current behavior and the reference model using the Chi-square (χ^2) test of goodness-of-fit.

Another case of interesting approaches was presented in the 26th International Conference on Machine Learning (ICML 2010) [77]. In the approach, programs of two systems (Darkstar and Hadoop) were analyzed and the desired information features extracted from their console logs. The analysis and the extracted features were employed in a Principal Component Analysis (PCA) based method to detect system failures.

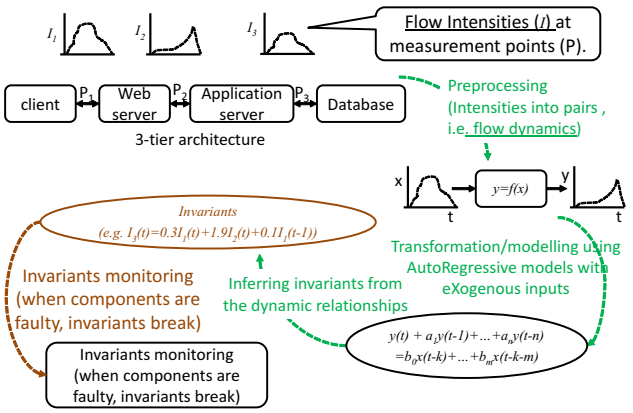

Fig. 7. A failure detection approach using invariants

Of recent, researchers at the NEC Corporation [15], [48] used invariant analysis technique to detect failures and characterize complex systems. The basic concepts behind the approach are summarized in Fig. 7. In this example, the approach is used to characterize a 3-tier system in order to detect failures (including silent failures). Flow intensities are established from the various measurements. The flow intensities are then modeled as flow dynamics and transformed into AutoRegressive models with eXogenous inputs (ARX models). Among the modeled dynamic relationships between flow intensities, invariants are inferred. A failure of an anomaly is detected when an invariant breaks. An example of an invariant, which was discovered in their experimentation involved intensities of created enterprise java beans I_{ejb} and processing time of java virtual machine I_{jvm}. This is shown by Equation 1.

$$I_{ejb}(t) = 0.07I_{ejb}(t-1) - 0.57I_{jvm}(t) \qquad (1)$$

There is no clear distinction between anomaly detection and intrusion detection (see Section 3.3.3). They are technically similar in many aspects. The one aspect, which poses the biggest challenge is how to efficiently process the voluminous data the systems generate. The approach which was presented in Section 3.3.3 extracts 6-dimensional vectors in dealing with the challenge. Similarly, in [78], in order to ensure that they do not end up with the curse of dimensionality, *log keys* were extracted from the unstructured log messages. In many cases, the steps which follow the extraction are conceptually the same in the sense that they strive to model normal behavior and attempt to detect whatever that deviates from that behavior. For instance, in [78], Finite State Automation (FSA) based method was used while in [7], Self Organizing Map (SOM) based method was used for the same.

3.3.5 Optimization

Optimization is one of the most interesting applications of mining in NSM. By analyzing relevant data, we could, for example, correctly estimate the exact amount of computational resources needed, as was presented in Section 3.3.1. Also hidden faults, which do not manifest themselves

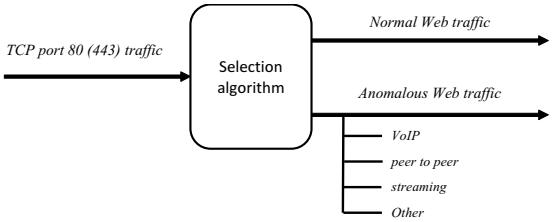

Fig. 8. Structure traffic classification

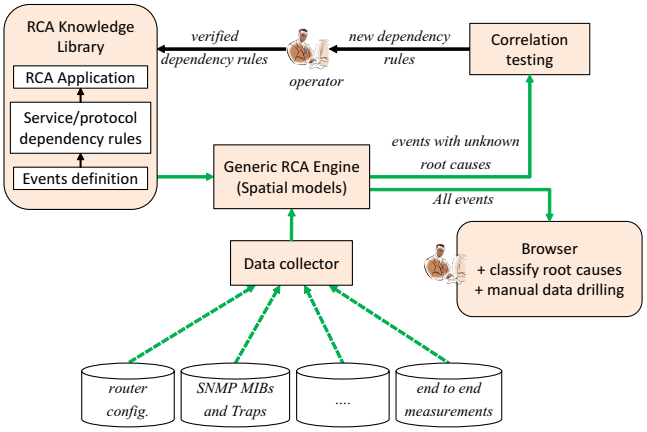

Fig. 9. An example approach for root cause analysis

as alerts in conventional tools, could be detected. As an example, in [79], an approach called StackMine was proposed for the analysis of execution traces in order to detect hidden bugs with negative performance effect.

In Section 3.3.1, we presented a capacity prediction case [64]. The same paper also addressed an optimization problem in commissioning Virtual Machines (VMs). Even though being able to predict resources in order to effectively utilize resources is a mile stone, in terms of SLA, the time it takes for a cloud system to commission a VM is equally crucial. For example, a state of the art technology would take minutes to commission requested resources in terms of VM units. When on-the-fly computational scaling is sought, this can really cause customer dissatisfaction. By employing mining techniques, preprovisioning of resources can be achieved [64] and enable the commissioning of requested resources in seconds rather than minutes.

An invariant analysis based approach (Section 3.3.3) [15] can also be regarded as an optimization solution. The solution is good in detecting silent failures, which are not always covered by many other approaches.

Probably, one of the best and popular optimization cases is in controlling TCP congestion in order to improve services [80]. Of recent, with the evolution of broadband mobile networks, operational cost (OPEX) reduction efforts by automating various tasks (e.g. radio parameter tuning [81]) have been increasing.

3.3.6 Accounting

Often in NSM operations, we want to account for the assets and usage of services and resources. In one publication [45], a metric was proposed and statistical tests conducted to clearly distinguish Skype flows, which are hidden in HTTP traffic. As shown in Fig. 8, HTTP traffic is the normal traffic while attempts to identify anomalous Skype traffic are made. They applied a two-step approach. In one step, normal HTTP traffic is modeled using well-known HTTP modeling methods – Arlitt and Williamson workload model [82], Mah workload model [83], and Choi and Limb Workload model [84]. Based on these three models, a new model was defined to constitute five parameters – request size, response times, request inter-arrival times, number of requests per page, and page retrieval time. In step two, when new data is seen, Chi-squared (χ^2) and the Kolmogorov-Smirnov are used to evaluate how close the new data flow in monitored traffic is to the empirical distributions derived from the training data. This approach achieved a 90% performance in terms of rate of detection of VoIP calls hidden in Web traffic with a false positive rate of only 2%. A system based on this approach is claimed to be in use at Motorola Inc. On daily basis, the system is used to analyze about 76 million data records with about 600 attributes from phone usage logs. The objective is to detect possible call failures and improve the performance of Motorola phones. The tool, which employs the approach, offers visuals about distributions of the attributes and identifies those which largely contribute to a particular class of failures.

3.3.7 Causality analysis

When a failure or an intrusion alert has manifested itself, the next step is finding out its root cause. Data mining based techniques can be applied on known failure cases and other network measurements data in the course of establishing the root causes. On the other hand, efforts ([15]) have also been made to establish root causes for hidden errors. In this subsection, we present few cases of causality analysis.

In science and engineering, we strive to find causes of problems in order to improve systems availability. Electronic devices (e.g. cell phones) fail, all or specific brand models. In Zhao et. al. [85], a deployed mining system for *analyzing failure causes* of cell phones is presented. The system was deployed and has been in regular use at Motorola since around year 2004. The system uses cell phone usage data from operators (in a typical classification dataset format) and attempts to predict when a cell phone will fail. The employed approach is based on three key concepts – *Class Association Rules* (CAR), *general impressions*, and *visualization*. An improved CAR mining is used as a solution to the problems, which render traditional mining (decision trees, Naive Bayesian, SVM etc) less useful for the production of practical rules. The problems are mostly completeness related. For instance, decision tree algorithms focus on producing rules which are only sufficient enough to classify items. The improvements were such that minimum support and confidence of the conventional association rule mining are set to zero so that

all CAR rules are mined. Then a meta-mining process is taken to discover knowledge from discovered rules. The challenge then was how to present the rules to the user for knowledge discovery. In Zhao et. al. [85], three *general impressions* (discrimitive attributes, trend attributes and similar values) were used as a way-around to this challenge.

Another root cause analysis case is in Service Quality Management (SQM) [59]. Generic-Root Cause Analysis (G-RCA) [59] is a customizable platform for service quality management in large IP networks, which provides different root cause analysis tools for new incoming problems. The key feature to G-RCA is the comprehensive service dependency model, which includes topological and cross layer relationships, protocol interactions, and routing dependencies. These are automatically determined by examining collected management data. For instance, network paths can be computed from BGP and OSPF route-monitoring data. The architecture (Fig. 9) of the G-RCA approach, which was implemented in a tier-1 ISP network had five main components: (1) data collection and management, (2) the service dependency model, (3) spatial-temporal correlation, (4) reasoning logic, and (5) domain knowledge building. In data collection component, the key is the use of a spatial event model, which maps events and locations and also the technique of associating an event with a script, which would enable G-RCA to acquire diagnostic information. The spatial model also serves as the service dependency model and it spans multiple OSI layers. In this approach, the timing issue is handled by the spatial-temporal correlator, which uses sets of rules linking spatial and temporal aspects of events. This is achieved by expanding an event time window using temporal rules specified by six parameters – the left expansion margin X, right margin Y, and an expanding option (Start/End, Start/Start, or End/End) for each of the symptom event and diagnostic event. The G-RCA's reasoning logic is largely rule-based but also uses Bayesian-based inference technique. The diagnostic domain knowledge is specified by operators using diagnostic graphs when defining a new SQM application. Issues related to inaccuracy of human specified knowledge are addressed by the inclusion of the Correlation Tester. The effectiveness of G-RCA was demonstrated using three case studies – BGP flaps, end-to-end throughput management in a content delivery network service, and network PIM flaps in multicast VPN.

3.4 Reporting

Usually, NSM center must report critical events and statuses of a system or network to the management. Using data mining techniques, complex details can easily be put in a form which is much easier to understand and interpret. The techniques can also be used to link with historical data in order to establish trends for better decision making process. A wide range of techniques, such as data description techniques (i.e. statistical measures like measures of central tendency), visualization and pattern recognition techniques, which can be used for this purpose, are available.

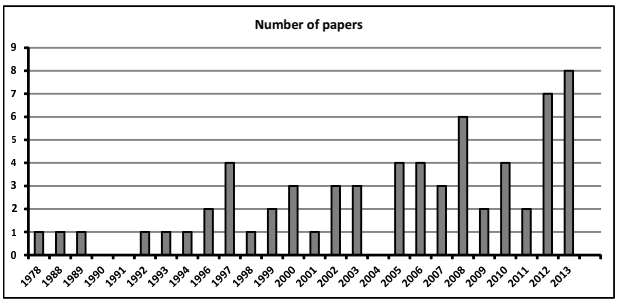

Fig. 10. Number of surveyed papers per year

3.5 Domain knowledge

Google and others provide platforms (e.g. blogs, wikis, forums) in which a deluge amount of knowledge (troubleshooting, optimization, etc) is shared. Through text mining and Information Extraction (IE), important information can be acquired from the Web. Aspects like Web crawling, text categorization, concept extraction and data integration, to mention a few, are central to many Web-centered businesses. Natural Language Processing (NLP) is the key to most aspects of text mining and IE. There are a number of publications in which the various techniques, including NLP, have been used to acquire domain knowledge for NSM operations. Following the "2011 earthquake off the Pacific coast of Tohoku," a knowledge-based approach for autonomous failure isolation and recovery support was proposed. The approach aimed at enabling novice administrators to restore and run operations in the aftermath of a disaster. Central to the approach are the pre-crafted knowledge resources for diagnosis and fault countermeasures (e.g. system commands). Due to difficulties in crafting every possible knowledge resource, an approach [86] based on the concept of Named High Cardinality Entity (NHCE) [87] was proposed to acquire system commands for supplementing the crafted countermeasure knowledge resources. The IE approach [86] uses NLP techniques and the GATE framework [88]. In Bozdogan et.al. [66], a combination of a clustering algorithm and genetic algorithm were used to extract information from Web crawled data. The extracted information was used to support IT management operations in an organization. Unfortunately, they did not explicitly explain the kind of information which was extracted.

NSM involves management of services that are supported by a system. Due to heterogeneity of the services, it is sometimes difficult to utilize them. Wang et. al. [89] proposed a framework for the acquisition of service support knowledge. In the framework, Bayesian networks are built as knowledge from the Configuration Management DataBase (CMDB) and system logs. Items from CMDB and event entities from logs make the nodes of the network while the edges represent their relationship. The networks were used to localize problems.

TABLE 2
Application areas of data mining in network and systems management

NSM Actitity	Application area	Description
Discovery	Assets [24], [68]	Discovery of existing assets (software, hardware, configurations, topologies) in a network.
Monitoring	Desired events related to faults, anomalies and usage [4], [7], [15], [17], [18], [19], [21], [23], [26], [31], [34], [38], [40], [46], [48], [52], [56], [63], [65], [90], [91], [92], [93], [94], [95]	Various data mining based approaches for monitoring network and systems events which are desired and related to errors, faults, anomalies and usage. The approaches are mostly based on pattern matching, association, prediction and classification.
Analysis	Prediction [4], [17], [27], [28], [29], [34], [96]	Analysis of NS-data in order to predict the occurrence of a desired event (e.g. performance drop, failure) or dimension the network or system.
	Detection [7], [23], [34], [35], [36], [39], [40], [42], [70], [74], [76], [77], [92]	Analysis of NS-data for modeling (including correlation) the normal behaviors of a system so that some other desired behaviors (e.g. fault) can be detected.
	Optimization [16], [22], [32], [79], [92]	Analysis of NS-data (e.g, system call trace, logs) in order to optimize the network or system which generated that data.
	Classification [19]	Classification of data (e.g. Communication protocol data) for various purposes like accounting, forensics etc.
	Causality [59], [85], [94]	Analysis to establish the root cause of a desired event.
Reporting	Data description [63]	Describing raw data is a daunting task. The task can be simplified if techniques like measures of central tendency are used.
	Visualization [20]	Complex concepts (e.g. outliers, large scale networking) can be simplified and made easily understandable to human using visual objects (e.g. graphs).
Domain knowledge	Management Information Base [22], [33], [42], [50], [90], [97]	Mining of correlated information which is useful in NSM operations (e.g. Policy making, Usage and transactions tracking).
	Operational rules [31], [63]	Mining of association between items for, for example, anomaly or fault detection.
	Web information [86], [87]	Extraction of useful information and knowledge (e.g. System operation commands) from the Web (Technical forums, Wikis, SNSs) for, for example, use in knowledge based systems or NSM helpdesks.

4 DISCUSSION

In Section 3, we presented selected cases on how data mining has been applied in various activities of NSM. In practice, solutions are used in combinations to achieve desired objectives. Experts present discussions about these solutions differently. Our presentation style followed a known categorization [6] whereby NSM functions are deemed to involve five activities – *discovery*, *monitoring*, *analysis*, *reporting* and *domain knowledge*. Table 2 summarizes the key application areas in each of the activities for the papers which we have surveyed. Fig. 10 shows the distribution of the papers. Across the areas, differing techniques, algorithms and evaluation measures have been used for each of the data mining process steps in Fig. 1. In this Section, we discuss the techniques and methods used in the mining steps and our perspective on issues which we consider critical for the improved and effective use of data mining concepts in the exploitation of NS-data.

There are three sources of performance data in any computer based network – system logged data, user-defined measurements, and traffic data. These sources plus online content are the main sources of information and knowledge for network administration. The presented and other mining cases in NSM are applied on data from all these sources – *log data* [64], [65], [71], [85], [89], [91], [94], *measurements* [15], [64], [76], *traffic data* [7], [8], [45] and *Online data* [66], [86]. When using log and traffic data, miners have leveraged various well known tools (tcpdump, tcppurify, wireshark, syslog, netflow etc) to extract NS-data. Measurements were conducted using system tools (e.g. SNMP) and various tool-specific libraries. Online sources were exploited via search engines. In most cases, the properties (e.g. dimensionality, numerosity) of the target dataset were defined by experts. Preprocessing in order to achieve the desired properties was manually done and, in most cases, it has not been explained how. This is because it is considered trivial. Examples are (1) in ANDSOM [7], it was not explicitly narrated how datasets for the six parameters were obtained from the traffic dump, and how nuisance was that processing to the whole mining process, and (2) in NEC [15], [48] where there was no guideline on how to decide about the amount of data items,

which should be seen before declaring an invariant out of flow dynamics relationships.

In most cases, target sets were transformed into forms which were desired by the miners. Some of the transformations are summarized in Table 3. In some cases, dedicated modules or mechanism were employed to achieve the transformation. Examples include Common Information Adapter (CIA) [70] for processing *synched common info* and ETL [50] for processing *NHCE instances*.

4.1 Measures of Interestingness

Data mining is a recursive and iterative process. Multiple algorithms and methods (including human interventions) are run on transformed datasets in succession. During such mining iterative process, value is added and a step towards the discovery of patterns is made in each pass on data. In this subsection, we look into how interestingness of the added values has been measured.

1) *Precision, Recall, F-measure and AUC*: This group of measures comprises of the most used measures. Simple examples include the use of precision and recall in terms of true positive and true negative following a hit or a miss database write operation. This was used when detecting malicious activities in a database system [47].
2) *Classification error*: When classifiers are learned (e.g. in ensembles [46], [99]), classification errors and accuracy are used to test how useful the classifiers are. The same applies to other ensemble models.
3) *Confidence score, θ*: This is used to determine whether a model has the potential to be an invariant in an invariant analysis based failure detection approach [15], [48]. The following equation is used.
$$p_k(\theta) = \frac{p_{k-1}(\theta).(k-1) + f(F_k(\theta))}{k}$$
where for a set of models $M_k = \{\theta \| p_k(\theta) > P\}$, P is a confidence threshold for determining whether a model has the potential.
4) *χ^2 test of goodness-of-fit*: It is used to verify a null hypothesis that observed frequencies of some independent events follow a specified distribution. For instance, it was used to test how close the results obtained from traffic flows are to the empirical distributions [45].
5) *Minimum support*: Commonly used in association mining. It specifies the minimum support value for an association to be considered interesting. It has been used in many cases [34], [63], [71] to verify association rules during mining processes.
6) *Entropy based measures* [89]: This is one of the well known measures to evaluate uncertainties or randomness of a variable X characterized by a probability distribution, $P(x)$ (see the equation below).
$$H(X) = -\sum_{x \notin X} P(x) log P(x)$$
7) *Granger Causality Test (GCT)*: Uses statistical functions to test if lagged information on a time series provides any statistically significant information about another time series variable. In one error correlation approach [94], a variable y was modeled by two auto-regression models – AutoRegressive Model (AR) and AutoRegressive Moving Average Model (ARMA). GCT was used to compare the residuals of the AR model with that of the ARMA model.
8) *Utility-based spamicity*: A utility used to measure the likelihood that a page is benefiting from a link spam. It was used in a page-farm based approach to detecting the beneficiaries of link spam. It was defined that, for a page farm of a page p which is represented as a graph $Farm(p) = (V, E)$, utility-based spamicity is $USpam(p) = \frac{PR(p)}{PR_{max}(V,E)}$, $PR =$ Page Rank.
9) *RMS Error*: This measure represents statistical and regression based measures. It is commonly used when learning from data using regression algorithms. For instance, it is used when learning from behavior summaries [34] during event prediction.
10) *Kolmogorov-Smirnov test*: Used to decide whether samples follow a particular distribution. It resembles χ^2 test and is also used in traffic modeling [45].
11) *Minimum probability*: Like minimum support, this is a probability threshold for mutually dependent items to be considered interesting, for instance, in verifying custom-defined *m-patterns* in an approach to isolating failures based of items infrequency.
12) *Slack variable, ξ*: In an intrusion detection approach based on ellipsoid [98], a slack variable was introduced to penalize large distances from hyper-ellipses.
13) *Similarity*: When jobs are similar but have occurred in different nodes at the same time, they are considered to be spatially correlated [70]. Besides using similarity to compare jobs, the metric can be used to compare Web services (e.g. IO Similarity [100]).
14) *Time scale parameter, θ* [70]: This is an adjustable timescale parameter used to determine temporal relevancy of two failure events. It was used in an approach based on spherical covariance, C, given by the below equation where α and β are positive constants, and d time distance between the failures.
$$C_T(d) = \begin{cases} 1 - \alpha\frac{d}{\theta} + \beta(\frac{d}{\theta})^3 & \text{if } 0 \leq d \leq \theta \\ 0 & \text{if } d > \theta \end{cases}$$
15) *Discriminative Significance*: A pattern (e.g. of itemsets which makes up a bug signature [11]) is considered discriminative if it isolates one class from the other. In [11], based on Information Gain (IG), Discriminative Significance (DS) of a pattern P out of a database of transactions D is given as
$$DS(p,n) = \begin{cases} IG(p,n) & \text{if } \frac{n}{\|D^-\|} > \frac{p}{\|D^+\|} \\ 0 & \text{otherwise} \end{cases}$$
16) *Proximity measures*: An overview in [106] presented the concept of proximity as a measure, which allows one to compute the degree of deviation from normality when analyzing anomalies in intrusion detection

TABLE 3
Examples of transformations and approaches before actual mining processes

Transformation	Target mining results	Description
n-measurements workload model [45]	traffic accounting	A five parameter workload model for describing HTTP traffic (request size, response size, inter-arrival time, requests per page and retrieval time).
n-attributes tuple [65]	Log event prediction	n=9 event summarizing tuples – time, severity, type, id, node, app name, process, log id and user.
Behavior summaries [91]	Chaotic event sequence detection	A behavior summary (behSum) is a temporal description of a system situation determined by the past and present observed characteristics as well as the predicted future.
Hyper alert time series [94]	Causality: attacks relationships	Time ordered sequences of alerts that belong to same alert clusters after alerts have been aggregated and clustered.
Normalized vectors [7]	connection class intrusive, non intrusive	n=6 connection summarizing parameters – INTER (interactivity per connection, i.e. questions per second), ASOQ (average size of questions), ASOA (average size of answers), QAIT (question-answer idle time), AQIT (answer to question idle time) and DOC (duration of connection).
Runtime paths [76]	behavior anomaly detection in applications	runtime paths are ordered sets of coarse-grained components, resources and control flow used to service user requests.
n-measurements items [63]	infrequent (but highly correlated) event discovery	any set of desired measurement attributes.
Association rules [85]	Root cause correlations	Any class association rule (CAR) with both minimum support and minimum confidence set to zero.
Intensity flow series [15], [48]	Invariants	Time series of measurements which can be paired into flow dynamics (e.g. y=f(x) where x and y are measurements from two different points.).
Failure signatures [70]	failure prediction	A quantified set of performance variables that makes an indexable representation which distills the essential characteristics of a system.
High info gain *n*-grams [46]	intrusion detection	feature set with high information gain.
Page farms [8]	spam target page detection	For a page *p*, the page farm of *p* is the set of pages on which the PageRank score of *p* depends.
Normalized samples [98]	intrusion detection	a set of samples as desired by the miner.
Entity sequences [87]	entity extraction	Sequences of sub-entities as defined by the validation template in a sequence validation based approach.
Rrequests series [64]	VM capacity prediction	Time series of VM resource demands (requests) submitted by users.
MSG type transformation [71]	Anomalous log event detection	A more concise message representation. (we could not find sufficient information)
Synched common info [89]	Domain knowledge acquisition	Data from IT infrastructure which has been synchronized in time and is from diverse sources.
NHCE instances [50]	training dataset	Miner defined, model specific set of attributes and their respective domains.

systems. The measures included Euclidean, Squared χ^2, Minkowski, canberra, Jaccard, to mention a few. Testing interestingness of the outputs remains a huge challenge. In most cases, the above measures were used out of a miner's intuition (experience and domain knowledge) than out of an analytic evaluation. Experience and domain knowledge become more critical with advances in technology because of the increase in levels of recursiveness during mining. For instance, the proliferation of TCP-based protocols and applications (e.g. HTTP, HTTPS, SMTP, Skype etc) has complicated traffic characterization. This might necessitate additional iterations of the mining algorithms through NS-data. This is also true for security measures like layering of HTTP with SSL/TLS protocol (i.e. HTTPS).

4.2 Issues

Despite the advances in mining techniques, *information integration* and *interpretation*, and *domain knowledge* remains the biggest challenges. These two are such a mountain to climb that one can hardly find a thorough text which addresses them sufficiently. *Information integration* refers to the extraction of information features from NS-data and *integrating* (and/or correlating) them from across multiple sources. The commonly used approach is pattern matching rules using pre-defined correlation features (e.g. timestamp). However, the required amount of preprocessing

can easily deter the mining process. Imagine cultivating a field to grow unfamiliar seeds which could probably turn to be weeds. This is how preprocessing feels to data miners. Thus, efforts are needed for effective, sophisticated and easy-to-use NS-data preprocessing tools. The tools will facilitate NS-data retention. The very first challenge for such efforts is the formulation of correlation features for which *domain knowledge* is vital. When NS-data is retained, there is another challenge of *interpreting* what has been retained. Similar challenge was addressed in databases by the introduction of Data Manipulation Language(DML). Useful database techniques, which could be used to streamline data flow in networks, exist. Nevertheless, there is a need to devise methods which go beyond DML-like approaches. For example, DML offers limited features reduction (e.g. "group by"). Few efforts on addressing the deficiencies exist (refer Section 4.3). The issue of *domain knowledge* has been emphasized enough in the past.

4.3 The ongoing and new frontiers

4.3.1 Algorithms, methods and techniques

Data mining application in NSM is not as mature as that in bioinformatics and other disciplines. Algorithms which have been specifically designed for NS-data are very few. The challenges in NSM are characterized by multitude of behaviors that a network can exhibit. Capturing the many behaviors into an algorithm leads to algorithms, which are sometimes too difficult to understand and implement. The heuristics and the calculations, which an algorithm constitutes, have varying effects when run on NS-data that comes from different systems. Thus, one of the biggest challenges is in generalization of heuristics, which were devised based on a particular system behavior.

4.3.2 Bigness of NS-data

Big data analytics is currently a common place for researchers and practitioners. Big data analytics techniques can be very useful in areas like precise reporting, quick detection of concepts and detailed troubleshooting. Companies have identified this potential and a number of big data analytics tools in network monitoring are on the rise.

4.3.3 NS-data processing and retention

Increasingly, awareness about the significance of the customary "in-advance" preprocessing of NS-data for retention has been growing. Previous practices included the retaining of raw data and traditional rotation by logging servers. ETL [50] is a multiagent based approach to processing of NS-data. The approach extracts information features, integrates the features, and retains them according to a specific target-use format, which is defined using a retention model known as Named High Cardinality Entity. Likewise, Common Information Adapter (CIA) [89] is designed to allow for the extraction of synchronized information from specific NS-data items. There are also devised methods for automated processing in specific setups, for instance, in [101], a method for the automatic generation of training sets for the retraining of P2P traffic classifiers is proposed. In another effort, the *C5 Sigma* tool, by Command Five Pty Ltd, preprocesses packet captures and map them into a relational database.

4.3.4 Knowledge and experience sharing

The advances in technology have widened knowledge gaps within the computing society. As a result, knowledge and experience sharing is crucial than ever. But yet, due to skepticism about competition, losses and security aspects, institutions are not entirely engaging. Disclosing NS-data as a way of knowledge and experience sharing is unwelcome. This explains the not very successful missions for online knowledge sharing databases (Common Event Expression [102] and EventTracker Knowledgebase [103]). Some of the recent efforts include one which uses the concept of Active Information Resources (AIR) to share knowledge models produced during mining processes [104].

5 Conclusion

In this paper, we have explained how data mining has been applied in various network and systems management operations for the last four decades. For each decade, an overview of what happened was presented and then few selected sample cases were explained in detail. Narration of the selected cases was according to the NSM activities (*discovery*, *monitoring*, *analysis*, *reporting*, and *configuration*). The last section presented a thorough discussion covering accounts of the mining steps, apparent issues which still hamper the process and the ongoing frontiers.

The survey revealed that most applications are in intrusion/anomaly detection. Other areas include failure (hardware and software) detection and prediction, SPAM classification, causality and resource prediction and optimization. Over the decades and across the application areas, the central role of data mining process has remained the same – that is to detect interesting pattern from recorded data. This is achieved by using various learning methods and algorithms, which are commonly categorized into data summarization, classification, clustering, association, regression and visualization. Other methods and techniques mainly facilitate these categories. In a mining process, the methods, techniques and algorithms are used recursively and iteratively for a specific objective.

New frontiers include the growing interest in proper management of recorded NS-data in order to facilitate exploitation through mining. Some of the efforts include multiagent based approaches to extracting, integrating, retaining and interpreting NS-data for the effective integration with exploitation tools. We established that in order to better deal with the challenges caused by the introduction of new technologies, proper NS-data management is crucial. Other aspects are: (1) sustained adaptation of methods and algorithms, and (2) acquisition of domain knowledge.

REFERENCES

[1] S. Chakrabarti, M. Ester, U. Fayyad, J. Gehrke, J. Han, S. Morishita, G. Piatetsky-Shapiro, and W. Wang, "Data mining curriculum," ACM SIGKDD, Tech. Rep., 2006.

[2] S. Dua, and P. Chowriappa, *Data Mining for Bioinformatics*, CRC Press, 2013.

[3] R. Mattison, *Data warehousing and data mining for telecommunications*, Artech House, 1997.

[4] W. H. Au, K. C. C. Chan, and X. Yao, "A novel evolutionary data mining algorithm with applications to churn prediction," *IEEE Transactions on evolutionary computation*, vol. 7, pp. 532–545, 2003.

[5] N. Kirov, "Astroinformatics and digitization of astronomical heritage," *Review of the National Center for Digitization*, vol. 19, pp. 7–10, 2011.

[6] D. C. Verma, *Principles of Computer Systems and Network Management*, 1st ed, Springer Publishing Company, Incorporated, 2009.

[7] M. Ramadas, S. Ostermann, and B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science, G. Vigna, C. Kruegel, and E. Jonsson, Eds. Springer Berlin Heidelberg, 2003, vol. 2820, pp. 36–54.

[8] B. Zhou and J. Pei, "Link spam target detection using page farms," *ACM Trans. Knowl. Discov. Data*, vol. 3, no. 3, pp. 1–38, Jul. 2009.

[9] D. Lo, H. Cheng, J. Han, S. C. Khoo, and C. Sun, "Classification of software behaviors for failure detection: A discriminative pattern mining approach," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09., New York, NY, USA: ACM, 2009, pp. 557–566.

[10] M. Brodie, S. Ma, L. Rachevsky, and J. Champlin, "Automated problem determination using call-stack matching," *Journal of Network and Systems Management*, vol. 13, no. 2, pp. 219–237, 2005.

[11] C. Sun and S. C. Khoo, "Mining succinct predicated bug signatures," in *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2013. New York, NY, USA: ACM, 2013, pp. 576–586.

[12] L. Paradis and Q. Han, "A survey of fault management in wireless sensor networks," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171–190, 2007.

[13] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, ser. The Morgan Kaufmann Series In Data Management Systems. Diane Cerra, 2006.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[15] G. Jiang, H. Chen, and K. Yoshihira, "Discovering likely invariants of distributed transaction systems for autonomic system management," *Cluster Computing*, vol. 9, no. 4, pp. 385–399, Oct. 2006.

[16] D. Knudson, "Obtaining data base management system independence," in *Computer Software and Applications Conference, 1978. COMPSAC '78. The IEEE Computer Society's Second International*, pp. 376–381, 1978.

[17] J. W. LaPatra and M. Speck, "A dynamic performance index for packet switched networks," in *Signals, Systems and Computers, 1988. Twenty-Second Asilomar Conference on*, vol. 1, pp. 92–96, 1988.

[18] G. Homce, "Application of adaptive learning networks for the detection of failing power system components," *Industry Applications, IEEE Transactions on*, vol. 25, no. 6, pp. 986–991, 1989.

[19] G. G. Helmer, J. S. K. Wong, V. Honavar, L. Miller, and L. Miller, "Intelligent agents for intrusion detection," in *In Proceedings, IEEE Information Technology Conference*, pp. 121–124, 1998.

[20] S. Eick and G. Wills, "Navigating large networks with hierarchies," in *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on*, pp. 204–210, 1993.

[21] R. Gardner and D. Harle, "Fault resolution and alarm correlation in high-speed networks using database mining techniques," in *Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on*, vol. 3, pp. 1423–1427, 1997.

[22] L. Gaspery and L. Tarouco, "Managing users, applications and resources with rmon2," in *Global Telecommunications Conference, 1999. GLOBECOM '99*, vol. 3, 1999, pp. 1997–2001 vol.3.

[23] C. Hood and C. Ji, "Proactive network-fault detection [telecommunications]," *Reliability, IEEE Transactions on*, vol. 46, no. 3, pp. 333–341, 1997.

[24] Y. Murayama, "Configuration detection as a problem of knowledge discovery in computer networks," in *Cooperative Information Systems, 1997. COOPIS '97., Proceedings of the Second IFCIS International Conference on*, 1997, pp. 47–55.

[25] L. Conradie and M. A. Mountzia, "A relational model for distributed systems monitoring using flexible agents," in *Services in Distributed and Networked Environments, 1996., Proceedings of Third International Workshop on*, 1996, pp. 10–17.

[26] C. Peel, A. C. G. Saunders, A. J. Morris, and C. Kiparissides, "Neural network feature detection and process monitoring," in *Intelligent Control, 1992., Proceedings of the 1992 IEEE International Symposium on*, 1992, pp. 560–565.

[27] J. G. Roos, I. E. Lane, E. Botha, and G. Hancke, "Using neural networks for non-intrusive monitoring of industrial electrical loads," in *Instrumentation and Measurement Technology Conference, 1994. IMTC/94. Conference Proceedings. 10th Anniversary. Advanced Technologies in I amp; M., 1994 IEEE*, 1994, pp. 1115–1118 vol.3.

[28] R. Sasisekharan, V. Seshadri, and S. Weiss, "Data mining and forecasting in large-scale telecommunication networks," *IEEE Expert*, vol. 11, no. 1, pp. 37–43, 1996.

[29] M. Thottan and C. Ji, "Statistical detection of enterprise network problems," *Journal of Network and Systems Management*, vol. 7, no. 1, pp. 27–45, 1999.

[30] W. W. Cohen, "Fast effective rule induction," in *In Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 115–123.

[31] L. Kangping, L. Zengzhi, W. Zhiwen, Z. Jing, and T. Yazhe, "Towards more intelligent network management: service-oriented proactive fault management using kdd techniques," in *Intelligent Control and Automation, 2000. Proceedings of the 3rd World Congress on*, vol. 1, 2000, pp. 715–718 vol.1.

[32] S. Park, "Neural networks and customer grouping in e-commerce: a framework using fuzzy art," in *Research Challenges, 2000. Proceedings. Academia/Industry Working Conference on*, 2000, pp. 331–336.

[33] F. Gao and X. Ye, "Managing the management: using active metamib for policy based management," in *Communication Technology Proceedings, 2000. WCC - ICCT 2000. International Conference on*, vol. 1, 2000, pp. 91–94 vol.1.

[34] M. Nunez, R. Morales, and F. Triguero, "Automatic discovery of rules for predicting network management events," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 4, pp. 736–745, 2002.

[35] S. H. Oh, J. S. Kang, Y. C. Byun, G. L. Park, and S. Y. Byun, "Intrusion detection based on clustering a data stream," in *Software Engineering Research, Management and Applications, 2005. Third ACIS International Conference on*, 2005, pp. 220–227.

[36] W. Li, K. Zhang, B. Li, and B. Yang, "An efficient framework for intrusion detection based on data mining," in *Computational Intelligence Methods and Applications, 2005 ICSC Congress on*, 2005, pp. 37 – 50.

[37] Z. Youdong, "Multi relational mining in network intrusion detection," in *Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on*, vol. 4, 2008, pp. 445–448.

[38] V. Gorodetsky, O. Karsaev, V. Samoylov, and S. Serebryakov, "Interaction of agents and data mining in ubiquitous environment," in *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on*, vol. 3, 2008, pp. 562–566.

[39] T. Yang, "A time series data mining based on arma and hopfield model for intrusion detection," in *Neural Networks and Brain, 2005. International Conference on*, vol. 2, 2005, pp. 1045–1049.

[40] M. Schultz, E. Eskin, E. Zadok, and S. Stolfo, "Data mining methods for detection of new malicious executables," in *Security and Privacy, 2001. S P 2001. Proceedings. 2001 IEEE Symposium on*, 2001, pp. 38–49.

[41] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems." *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 38, no. 5, pp. 649–659, 2008.

[42] Y. Li and L. Guo, "An efficient network anomaly detection scheme based on tcm-knn algorithm and data reduction mechanism," in *Information Assurance and Security Workshop, 2007. IAW '07. IEEE SMC*, 2007, pp. 221–227.

[43] X. Du, "Identifying control and management plane poison message failure by k-nearest neighbor method," *Journal of Network and Systems Management*, vol. 14, no. 2, pp. 243–259, 2006.

[44] A. N. Mahmood, C. Leckie, and P. Udaya, "An efficient clustering scheme to exploit hierarchical data in network traffic analysis." *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 6, pp. 752–767, 2008.

[45] E. Freire, A. Ziviani, and R. Salles, "On metrics to distinguish skype flows from http traffic," in *Network Operations and Management Symposium, 2007. LANOMS 2007. Latin American*, 2007, pp. 57–66.

[46] M. M. Masud, T. M. Al-Khateeb, K. W. Hamlen, J. Gao, L. Khan, J. Han, and B. Thuraisingham, "Cloud-based malware detection for evolving data streams," *ACM Trans. Manage. Inf. Syst.*, vol. 2, no. 3, pp. 16:1–16:27, Oct. 2008.

[47] Y. Hu and B. Panda, "Design and analysis of techniques for detection of malicious activities in database systems," *Journal of Network and Systems Management*, vol. 13, no. 3, pp. 269–291, 2005.

[48] G. Jiang, H. Chen, and K. Yoshihira, "Modeling and tracking of transaction flow dynamics for fault detection in complex systems," *Dependable and Secure Computing, IEEE Transactions on*, vol. 3, no. 4, pp. 312–326, 2006.

[49] L. Li and D. bao Xiao, "Research on the network security management based on data mining," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 5, 2010, pp. 184–187.

[50] K. Kalegele, J. Sveholm, H. Takahashi, K. Sasai, G. Kitagata, and T. Kinoshita, "Multiagent-based processing and integration of system data," *Int. J. Intell. Syst. Technol. Appl.*, vol. 12, no. 2, pp. 128–155, 2013.

[51] Y. J. Lee, Y. R. Yeh, and Y. C. F. Wang, "Anomaly detection via online oversampling principal component analysis," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 7, pp. 1460–1470, 2013.

[52] A. Ahmed, A. Jantan, and M. Rasmi, "Service violation monitoring model for detecting and tracing bandwidth abuse," *Journal of Network and Systems Management*, vol. 21, no. 2, pp. 218–237, 2013.

[53] R. G. Garroppo, S. Giordano, S. Niccolini, and S. Spagna, "A prediction-based overload control algorithm for sip servers." *IEEE Transactions on Network and Service Management*, vol. 8, no. 1, pp. 39–51, 2011.

[54] Q. Xu, E. Xiang, Q. Yang, J. Du, and J. Zhong, "Sms spam detection using noncontent features," *Intelligent Systems, IEEE*, vol. 27, no. 6, pp. 44–51, 2012.

[55] S. Mabu, C. Chen, N. Lu, K. Shimada, and K. Hirasawa, "An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 1, pp. 130–139, 2011.

[56] A. Makanju, A. Zincir-Heywood, and E. Milios, "A lightweight algorithm for message type extraction in system application logs," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 11, pp. 1921–1936, 2012.

[57] Y. Ye, T. Li, Q. Jiang, and Y. Wang, "Cimds: Adapting postprocessing techniques of associative classification for malware detection," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, no. 3, pp. 298–307, 2010.

[58] A. Sperotto, M. Mandjes, R. Sadre, P. T. de Boer, and A. Pras, "Autonomic parameter tuning of anomaly-based idss: an ssh case study," *Network and Service Management, IEEE Transactions on*, vol. 9, no. 2, pp. 128–141, 2012.

[59] H. Yan, L. Breslau, Z. Ge, D. Massey, D. Pei, and J. Yates, "G-rca: A generic root cause analysis platform for service quality management in large ip networks," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 6, pp. 1734–1747, 2012.

[60] A. Bashar, G. Parr, S. McClean, B. Scotney, and D. Nauck, "Application of bayesian networks for autonomic network management," *Journal of Network and Systems Management*, pp. 1–34, 2013.

[61] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, Jun. 1993.

[62] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. of 20th Intl. Conf. on VLDB*, 1994, pp. 487–499.

[63] S. Ma and J. Hellerstein, "Mining mutually dependent patterns for system management," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 4, pp. 726–735, 2002.

[64] Y. Jiang, C. S. Perng, T. Li, and R. N. Chang, "Cloud analytics for capacity planning and instant vm provisioning," *Network and Service Management, IEEE Transactions on*, vol. 10, no. 3, pp. 312–325, 2013.

[65] X. Fu, R. Ren, J. Zhan, W. Zhou, Z. Jia, and G. Lu, "Logmaster: Mining event correlations in logs of large-scale cluster systems," in *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on*, 2012, pp. 71–80.

[66] C. Bozdogan, A. Zincir-Heywood, and Y. Gokcen, "Automatic optimization for a clustering based approach to support it management," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, 2013, pp. 1233–1236.

[67] S. Ohta, R. Kurebayashi, and K. Kobayashi, "Minimizing false positives of a decision tree classifier for intrusion detection on the internet," *Journal of Network and Systems Management*, vol. 16, no. 4, pp. 399–419, 2008.

[68] P. Garg, M. Griss, and V. Machiraju, "Auto-discovering configurations for service management," *Journal of Network and Systems Management*, vol. 11, no. 2, pp. 217–239, 2003.

[69] J. P. Rouillard, "Refereed papers: Real-time log file analysis using the simple event correlator (sec)," in *Proceedings of the 18th USENIX Conference on System Administration*, ser. LISA '04. Berkeley, CA, USA: USENIX Association, 2004, pp. 133–150.

[70] S. Fu and C. Z. Xu, "Quantifying temporal and spatial correlation of failure events for proactive management," in *Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on*, 2007, pp. 175–184.

[71] A. Makanju, A. Zincir-Heywood, and E. Milios, "Investigating event log analysis with minimum apriori information," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, 2013, pp. 962–968.

[72] A. Andrzejak and M. Ceyran, "Characterizing and predicting resource demand by periodicity mining," *Journal of Network and Systems Management*, vol. 13, no. 2, pp. 175–196, 2005.

[73] D. Dentcheva and W. Rmisch, "Optimal power generation under uncertainty via stochastic programming," in *Stochastic Programming Methods and Technical Applications*, ser. Lecture Notes in Economics and Mathematical Systems, K. Marti and P. Kall, Eds. Springer Berlin Heidelberg, 1998, vol. 458, pp. 22–56.

[74] J. Cabrera, L. Lewis, X. Qin, W. Lee, and R. Mehra, "Proactive intrusion detection and distributed denial of service attacksa case study in security management," *Journal of Network and Systems Management*, vol. 10, no. 2, pp. 225–254, 2002.

[75] J. K. Bhattacharyya, Dhruba Kumar; Kalita, *Network Anomaly Detection: A Machine Learning Perspective*. Chapman and Hall/CRC, Jun. 2013.

[76] E. Kiciman and A. Fox, "Detecting application-level failures in component-based internet services," *Trans. Neur. Netw.*, vol. 16, no. 5, pp. 1027–1041, Sep. 2005.

[77] W. Xu, L. Huang, A. Fox, D. A. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs." in *ICML*, J. Frnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 37–46.

[78] Q. Fu, J. G. Lou, Y. Wang, and J. Li, "Execution anomaly detection in distributed systems through unstructured log analysis." in *ICDM*, W. W. 0010, H. Kargupta, S. Ranka, P. S. Yu, and X. Wu, Eds. IEEE Computer Society, 2009, pp. 149–158.

[79] S. Han, Y. Dang, S. Ge, D. Zhang, and T. Xie, "Performance debugging in the large via mining millions of stack traces." in *ICSE*, M. Glinz, G. C. Murphy, and M. Pezz, Eds. IEEE, 2012, pp. 145–155.

[80] A. Al-Naamany and H. Bourdoucen, "Tcp congestion control approach for improving network services," *Journal of Network and Systems Management*, vol. 13, no. 1, pp. 1–6, 2005.

[81] M. Tiwana and M. Tiwana, "A novel framework of automated rrm for lte son using data mining: Application to lte mobility," *Journal of Network and Systems Management*, pp. 1–24, 2013.

[82] M. Arlitt and C. L. Williamson, "A synthetic workload model for internet mosaic traffic," Saskatoon, Sask., Canada, Canada, Tech. Rep., 1995.

[83] B. Mah, "An empirical model of http network traffic," in *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, vol. 2, 1997, pp. 592–600 vol.2.

[84] H. K. Choi and J. O. Limb, "A behavioral model of web traffic," in *Network Protocols, 1999. (ICNP '99) Proceedings. Seventh International Conference on*, 1999, pp. 327–334.

[85] K. Zhao, B. Liu, J. Benkler, and W. Xiao, "Opportunity map: identifying causes of failure - a deployed data mining system," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 892–901.

[86] K. Kalegele, Y. Tanimura, J. Sveholm, K. Sasai, and T. K. Gen Kitagata, "A knowledge-based method for autonomous failure isolation and recovery support," in *Proceedings of the IEICE Technical Committee on Mobile Network and Applications*. JAPAN: IEICE, 2013, pp. 1–5.

[87] K. Kalegele, H. Takahashi, K. Sasai, G. Kitagata, and T. Kinoshita, "Sequence validation based extraction of named high cardinality entities," *International Journal of Intelligence Science*, vol. 2, no. 4A, pp. 190–202, 2012.
[88] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: A framework and graphical development environment for robust NLP tools and applications." in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA*, 2002.
[89] W. Wang, H. Wang, B. Yang, L. Liu, P. Liu, and G. Zeng, "A bayesian network-based knowledge engineering framework for it service management," *Services Computing, IEEE Transactions on*, vol. 6, no. 1, pp. 76–88, 2013.
[90] A. Murray and J. Penman, "Extracting useful higher order features for condition monitoring using artificial neural networks," *Signal Processing, IEEE Transactions on*, vol. 45, no. 11, pp. 2821–2828, 1997.
[91] M. Nunez, R. Morales, and F. Triguero, "Automatic discovery of rules for predicting network management events," *IEEE J.Sel. A. Commun.*, vol. 20, no. 4, pp. 736–745, Sep. 2006.
[92] Y. Tang, S. Krasser, P. Judge, and Y.-Q. Zhang, "Fast and effective spam sender detection with granular svm on highly imbalanced mail server behavior data," in *Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006. International Conference on*, 2006, pp. 1–6.
[93] T. Li, W. Peng, C. Perng, S. Ma, and H. Wang, "An integrated data-driven framework for computing system management," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 1, pp. 90–99, 2010.
[94] X. Qin and W. Lee, "Statistical causality analysis of infosec alert data," in *In Proceedings of The 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003*, 2003, pp. 73–93.
[95] W. Zhuang, Y. Ye, Y. Chen, and T. Li, "Ensemble clustering for internet security applications," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 6, pp. 1784–1796, 2012.
[96] C. Chao, D. Yang, and A. Liu, "An automated fault diagnosis system using hierarchical reasoning and alarm correlation," *Journal of Network and Systems Management*, vol. 9, no. 2, pp. 183–202, 2001.
[97] A. Schoorl and N. Dimopoulos, "Client mobility and fault tolerance in a distributed network data system," in *Communications, Computers and Signal Processing, 1999 IEEE Pacific Rim Conference on*, 1999, pp. 593–596.
[98] M. Ghasemigol, R. Monsefi, and H. Sadoghi-Yazdi, "Intrusion detection by ellipsoid boundary," *J. Netw. Syst. Manage.*, vol. 18, no. 3, pp. 265–282, Sep. 2010.
[99] R. Zarei, A. Monemi, and M. Marsono, "Automated dataset generation for training peer-to-peer machine learning classifiers," *Journal of Network and Systems Management*, pp. 1–22, 2013.
[100] Y. Xu, W. Niu, H. Tang, G. Li, Z. Zhao, and C. Song, "A policy-based web service redundancy detection in wireless sensor networks," *Journal of Network and Systems Management*, vol. 21, pp. 384–407, 2013.
[101] R. Zarei, A. Monemi, and M. Marsono, "Automated dataset generation for training peer-to-peer machine learning classifiers," *Journal of Network and Systems Management*, pp. 1–22, 2013.
[102] CEE. (2008) Common event expression.
[103] Prismmicrosys. (2007) Eventtracker-knowledgebase, http://www.prismmicrosys.com/newsletters august2007.php
[104] K. Kalegele, H. Takahashi, K. Sasai, G. Kitagata, and T. Kinoshita, "System monitoring models as active information resources," in *Awareness Science and Technology, 2013. 5th IEEE International Conference on*, 2013, pp. 226–230.
[105] M.H. Bhuyan, H.J. Kashyap, D.K. Bhattacharyya, and J.K. Kalita, "Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions," *The Computer Journal*, Oxford University Press, 2013.
[106] M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," Communications Surveys and Tutorials, IEEE , vol.16, no.1, pp. 303–336, First Quarter 2014.
[107] Z. Zhang, J. Li, C. N. Manikopoulos, J. Jorgenson, J. Ucles, "HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification," Proc. IEEE Workshop on Information Assurance and Security, pp. 85–90, 2001
[108] Daniel Barboard, Julia Couto, Sushil Jajodia and Ningning Wu, "ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection, " SIGMOD Record, Vol. 30, no. 4, pp. 15-24, 2001.
[109] L. Ertoz, E. Eilertson, A. Lazarevic, P. Tan, J. Srivastava, V. Kumar, P. Dokas, "The MINDS - Minnesota Intrusion Detection System," Next Generation Data Mining, MIT Press, 2004.

Khamisi Kalegele is a lecturer at the Nelson Mandela African Institution of Science and Technology, Tanzania. He received his PhD in Information Sciences from Tohoku University, Japan in 2013 and his M.Eng. in Computer Sciences from Ehime University, Japan in 2010. He is currently researching on ICT for development, application of Data Mining, and Network and systems management.

Hideyuki Takahashi is an assistant professor of Research Institute of Electrical Communication of Tohoku University, Japan. He received his doctoral degree in Information Sciences from Tohoku University in 2008. His research interests include ubiquitous computing, green computing and agent-based computing. He is a member of IEICE and IPSJ.

Kazuto Sasai is an assistant professor of Research Institute of Electrical Communication, Tohoku University, Japan. He received his Dr. Sci. in Earth and Planetary Science from Kobe University. His research interests include network management, network analysis, complex systems and agent-based systems. Dr. Sasai is a member of IEICE, IPSJ, JSAI and BSJ.

Gen Kitagata is an associate professor of Research Institute of Electrical Communication of Tohoku University, Japan. He received a doctoral degree from the Graduate School of Information Sciences, Tohoku University in 2002. His research interests include agent-based computing, network middleware design, and symbiotic computing. He is a member of IEICE and IPSJ.

Tetsuo Kinoshita is a professor of Research Institute of Electrical Communication of Tohoku University. He received B.E. degree in electronic engineering from Ibaraki University, Japan, in 1977, and M.E. and Dr. Eng. degrees in information engineering from Tohoku University, Japan, in 1979 and 1993, respectively. His research interests include agent engineering, knowledge engineering, knowledge-based systems and agent based systems. He received the IPSJ Research Award in 1989, the IPSJ Best Paper Award in 1979 and the IEICE Achievement Award in 2001.