Survey on Complex Event Processing and Predictive Analytics

Lajos Jenő Fülöp, Gabriella Tóth, Róbert Rácz, János Pánczél, Tamás Gergely, Árpád Beszédes
University of Szeged, Department of Software Engineering
Árpád tér 2., H-6720 Szeged, Hungary, +36 62 544145
{flajos,gtoth,gertom,beszedes}@inf.u-szeged.hu

Lóránt Farkas
Nokia Siemens Networks

July 13, 2010

Abstract

Observing failures and other – desired or undesired – behavior patterns in large scale software systems of specific domains (telecommunication systems, information systems, online web applications, etc.) is difficult. Very often, it is only possible by examining the runtime behavior of these systems through operational logs *or* traces*. However, these systems can generate data in order of gigabytes every day, which makes a challenge to process in the course of predicting upcoming critical problems or identifying relevant behavior patterns. We can say that there is a gap between the amount of information we have and the amount of information we need to make a decision. Low level data has to be processed, correlated and synthesized in order to create high level, decision helping data. The actual value of this high level data lays in its availability at the time of decision making (e.g., do we face a virus attack?). In other words high level data has to be available real-time or near real-time. The research area of* event processing *deals with processing such data that are viewed as events and with making alerts to the administrators (users) of the systems about relevant behavior patterns based on the rules that are determined in advance. The rules or patterns describe the typical circumstances of the events which have been experienced by the administrators. Normally, these experts improve their observation capabilities over time as they experience more and more critical events and the circumstances preceding them. However, there is a way to aid this manual process by applying the results from a related (and from many aspects, overlapping) research area,* predictive analytics*, and thus improving the effectiveness of event processing. Predictive analytics deals with the prediction of future events based on previously observed historical data by applying sophisticated methods like machine learning, the historical data is often collected and transformed by using techniques similar to the ones of event processing, e.g., filtering, correlating the data, and so on.*

In this paper, we are going to examine both research areas and offer a survey on terminology, research achievements, existing solutions, and open issues. We discuss the applicability of the research areas to the telecommunication domain. We primarily base our survey on articles published in international conferences and journals, but we consider other sources of information as well, like technical reports, tools or web-logs.

Keywords

Survey, complex event processing, predictive analytics.

Contents

1 Summary

The aim of this article is to give an overview on the terminology, existing solutions, methods and current problems of Complex Event Processing (CEP), and the related field of Predictive Analytics (PA). Both areas target the problems that are related to processing the large amount of runtime data of large information systems, in order to detect undesired behavior (like runtime failures or malicious activity) or other specific behavior patterns of interest. With these technologies, data is usually processed real-time or near real-time, and the decision is made based on current or past data, while sometimes the goal is the prediction of future events.

The survey introduces the current state-of-the-art and gives an overview on the problems and questions that have not been solved or answered in these fields yet. As a specific goal, the methods that can be applied in telecommunication, or other closely related application areas, are primarily collected, but the solutions that could be adopted to telecommunication from other domains are also considered. The two mentioned fields are presented side by side, due to the significant overlap in their goals and applied techniques. Yet, the literature normally treats these two fields fairly separately. However, after a careful observation of current achievements, we found that some results of one field can be readily applied to the other making it more efficient or more powerful. For example, the decision making process in CEP is often based on very simple principles which can be successfully extended by the more sophisticated prediction capabilities of PA, thus achieving a more intelligent and automatic solution to many of the addressed problems.

The paper is organized as follows. In Section 2, we examine the area of complex event processing. First, we give a background and overview about event processing in Section 2.1. This section is essential to understand the subsequent sections. Next, (in Section 2.2), we present the surveys that are similar to ours, which can be useful for the deeper understanding of the field. In the subsequent two sections (Section 2.3 and Section 2.4), we deal with the articles and tools that are related to event processing, and in the final subsections of Section 2, we classify and evaluate the articles, and present the open questions of the event processing research area. In Section 3, we offer a similar survey from the area of predictive analytics. In Section 4, we discuss the connection between complex event processing and predictive analytics more deeply, and we make conclusions in Section 5.

This work was prepared by the Department of Software Engineering, University of Szeged (SED) in cooperation with Nokia Siemens Networks, Budapest (NSN).

2 Detailed assessment of technologies of Complex Event Processing

2.1 Background and overview

In this section, we provide a background on complex event processing and an overview of related concepts (also see [72], [82], [43], [44], [66], and [96]). We summarize the basic concepts of event processing in Figure 1. Standard terminology has not yet been established for event processing, so most of the concepts have several synonyms which will be presented later in this section.

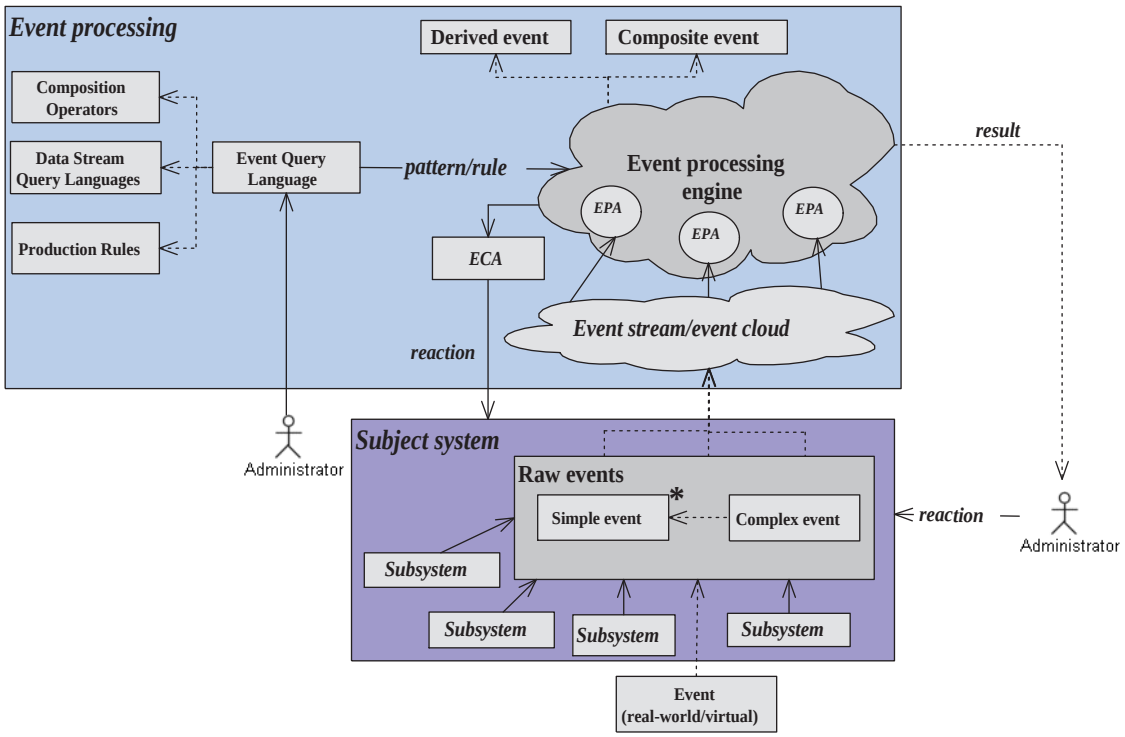

Figure 1. Event processing concept map

Event processing is usually applied on an already existing information system, which we call a *subject system*. The *subject system* is able to receive events. An event is "anything that happens, or is contemplated as happening" [63]. Events may come from the outside, which are either occurred in the real world (an airplane landed, a bus departed) or virtual (a storm prediction in weather forecast, an airplane landed in an airplane simulator). Physically, events can come from an external database, an RFID data sensor, a service, an enterprise information system (SCM, ERP, CRM), etc. [96]. The subsystems of the *subject system* also produce events internally. Events can be either simple or complex. A *complex event* is the abstraction of *simple events*, for example, the landing of an airplane is a *complex event* which is built up of several *simple events* (the pilot decreases altitude or speed, etc.). The events can be arranged into either an *event stream* or an *event cloud*; *Event Stream Processing* (ESP) deals with the former, while *Complex Event Processing* deals with the latter. In both cases, typically some kind of *event processing engine* plays the key role. The events are provided for the *engine* by *Event Processing Agents* (EPA), which are, for example, able to filter them. In the next step, the *engine* processes the events, and notifies the administrator (user of the *engine*) if the patterns/rules previously given by him are matched. A pattern or a rule can be defined in an *event query language* through which one can *subscribe* for certain event-patterns. The administrator executes the appropriate actions on the *subject system* when he is notified by the *engine* about which previously given pattern has been detected. These actions can be automated by using the *Event Condition Action* (ECA), where the *action* is automatically performed on the *subject system* if the *condition* proves to be true. During the processing of events, further events can be generated. The *derived event* is a completely new event which is generated from former events, while the *composite event* is a special variant of the *derived event* which aggregates several other events that could be queried later.

Event Query Languages can be grouped into three styles [43], Composition Operators, Data Stream Query Languages and Production Rules. Composition operators express complex events by composing single events using different composition operators (conjunction, sequence, negation) and nesting of expressions. IBM Active Middleware Technology and ruleCore are based on composition operators. Data stream query languages are based on SQL; the data streams containing events as tuples are converted into database relations. Afterwards, SQL queries can be evaluated, and finally, the relations are converted back to data stream. Nowadays, the most popular approach is data stream query languages, e.g., CQL, Coral8, StreamBase, Aleri, Esper, etc. Production Rules specify the actions that are to be executed when certain states are entered, so they do not constitute an event query language in the narrower sense, but they are suitable for CEP. Production Rules is an important part of TIBCO Business Events.

Another important factor, namely *time*, has to be mentioned. Two important parts of time appear in event processing, the time window and the time information about an event. The time window shows that the events between a start and an end time are examined. Time information about an event has three types [66], *occurrence time*, when the event physically happened, *detection time*, when the event has detected, and *transition time*, when the event was processed.

Since no widely accepted standard exists for the concepts of event processing, several synonyms appear in the literature of the terms presented in Figure 1. The Event Processing Technical Society [44] presented a glossary [63] in 2008, which was one of the most important steps of the standardization of event processing concepts. However, before its release and even in 2009, there are several synonyms used for CEP terms. In the followings, we show the definitions and synonyms – in parentheses – of some event processing concepts according to this glossary and the examined articles:

- Event: Anything that happens, or is contemplated as happening.
- Simple event: An event that is not an abstraction or composition of other events. (single event, base event, primitive event, atomic event, discreet event, constituent event, low-level event)
- Complex event: An event that is an abstraction of other events.(constructed event, high-level event; sometimes it means a composite and a derived event as well)
- Composite event: A derived, complex event that is created by combining base events using a specific set of event constructors such as disjunction, conjunction, sequence, etc. (compound event)
- Event type: An event type is a class of event objects. (event class, event definition, event schema)
- Event attribute: A component of the structure of an event. (event property)
- Event channel: A conduit in which events are transmitted from event sources (emitters) to event sinks (consumers). (event connection, event pathway, event topic)
- Situation: A specific sequence of events.
- Event: An object that represents encodes or records an event, generally for the purpose of computer processing. (representation of events, event object, event instance, event message, event tuple)
- Raw event: An event object that records a real-world event.
- Detection time: The timestamp when the event was detected. [66] (creation time)
- Transaction time: The timestamp when the event was processed. [66] (arrival time)
- Event Processing Engine: A set of event processing agents and a set of event channels connecting them. (Event Processing Network (EPN), Event Processing Platform (EPP)).

2.2 Related works

In this section, we present existing surveys in the area of complex event processing.

Owens states in his survey [72] that *"Event processing will change the information technology and information management communities as much as, if not more than, the introduction of the relational database"*. One key point of this report is that clarifies the terminology about event processing. The definitions (*real-world* and *representation*), the types (*simple* and *complex*), the existence (*event cloud* or *event stream*), and the operations (*querying* or *subscription*) of events are clarified. Owens introduces four application fields of event processing which are not a complete summary as he mentions. These are Data Analysis and Mining, Intrusion Detection Systems, Sensor Management, and Situational Awareness. Owens also summarizes the system requirements of event processing engines, namely *high availability*, *optimization*, and *scalability*. Furthermore, he presents several academic and commercial event processing systems as well, and summarizes the applied event processing languages. Finally, he provides benchmarks for event processing systems.

Schmidt et. al. also published a survey [82], however not directly about complex event processing, but 'event-driven reactivity'. In this paper, they investigate not only the events, but the automatically fired (re)actions as well if some conditions are met with the event (action processing). They call this technique event-triggered reactivity. Similarly to the work of Owens, they also emphasize the importance of CEP: *"Event-driven processing becomes ever important... The market value should increase tenfold by 2010 and should reach something like $4bn in total (source:IBM)"*. The authors differentiate *logic* and *non-logic* view of event-triggered reactivity. The non-logic view does not consider the formal (logical) representation of elements in reactive rules, while the logic view does. In the paper, they summarize the event-processing and the action processing approaches separately for the non-logic and logic views. Furthermore, they give the requirements for handling event-triggered reactivity for future systems. They present a vision about the future: the applications will be developed in an event-driven fashion – considering the context – which will reduce the complexity and the cost to react events. They cite that *"... context-based event processing is the next 'big thing' and should shape the future of the computing ... Context-Driven Architecture (CoDA) as the most promising paradigm that will extend SOA"*.

Eckert et. al. presented a paper about complex event processing [43] as well. First, they give the application areas of complex event processing: business activity monitoring, sensor networks, and market data. Afterwards, they divide complex event processing into two types; the first is where complex events are specified as a-priori known patterns over events; the second is where the previously unknown patterns should be detected as complex events. Next, they summarize the requirements for an event query language, namely data extraction, composition, temporal relationships, and accumulation. They also emphasize that two types of rules are important: deductive rules which define the new events based on event queries, and reactive rules which specify how to react to events. The key part of their paper is the summary about the categorization of the prevalent event query languages. The categories are *Composition Operators* (e.g., ruleCore), *Data Stream Query Languages* (e.g., Esper), and *Production Rules* (e.g., TIBCO Business Events). Similarly to other related surveys, they also mention the standardization as a future work, however not in general, but for the query languages. There are several efforts toward standardization. They mention the Event Processing Technical Society (EPTS) [44], and the Event Metamodel and Profile (EMP) proposal by OMG to support the modeling of events in UML.

2.3 Articles

In this section, we present papers from the complex event processing research area. In the first step, we used several keywords (related to telecommunication (15 labels), enterprise (17 labels), prediction (2 labels) and other labels (5 pieces)) in the search engine Google Scholar to find the baseline for further research. One of our selection criteria was whether the referenced papers had been published by IEEE, ACM, Springer, Elsevier, Wiley, or Addison. The latest articles were selected, and the older ones were eliminated. Later, we extended the searching fields for references and related surveys. We found 80 articles using this method. After a first review only 21 of them were examined in details and presented in this survey. These articles are organized into three groups: innovative papers, requirements and principles for complex event processing, methods and experiments.

We define evaluation aspects to raise important details for better visibility. First of all, the term **year** indicates the year when the paper was published. The term **number of references** indicates how many papers referred to the given paper. The term **realtime** tells us whether the method can be used in real time or not. The term **whether applied on live environment** shows the kind of source the method was tested on. If the paper defines a **tool** and/or a **language**, it is mentioned as an aspect as well. We examined in which area the given method can be used. The **applicability** aspect defines this area. If this area is not telecommunication, we further examine the paper **whether it is applicable in telecommunication**. As the aim of this survey is collecting methods for prediction, we have to look at the papers considering this aspect as well – **whether applicable for prediction**. Finally, we summarize the **goals of the different papers** and collect some typical **labels** from each paper. We do not include the aspects if they are not mentioned in the paper.

2.3.1 Innovative papers

2.3.1.1 Complex Event Processing Over Uncertain Data

Authors presented a solution [90] for generating events – they called it as materialization – with a probability or uncertainty. Uncertainty means that a (materialized or generated) event has a chance, for example, the chance of a flu outbreak is 90%. They provided two algorithms, an accurate but expensive which is based on the construction of a Bayesian network, and a Monte Carlo sampling algorithm which is heuristic. Finally, they made experiments in a simulation environment with

the presented algorithms and achieved good results. They mentioned, that the use of machine learning techniques for the automatic generation and maintenance of rules could be a promising research area.

- **year:** It was published in 2008.
- **number of references:** There are 2 references for this paper.
- **realtime:** The presented Monte Carlo sampling algorithm could be used in a real-time environment.
- **whether applied on live environment:** No, authors applied the algorithms in a simulation environment.
- **tools:** They presented the two algorithms in their framework, but the accurate name of the framework was not given.
- **language:** They used a rule based language, but it was not named.
- **applicability:** The presented method is general.
- **whether applicable in telecommunication:** Because it is general, yes, it could be applicable.
- **goal of the paper:** Generating or materializing events with a probability or uncertainty.
- **labels:** business, forecast, Business Activity Monitoring (BAM), Business Process Management (BPM), uncertainty

2.3.1.2 Complex Event Processing Beyond Active Databases: Streams and Uncertainties

In the first part of the article [81], Rizvi surveyed complex event processing work in multiple contexts, active databases among others. A metamodel for event algebras was defined after describing COMPOSE and SNOOP algebra. In the second part of the article, an overview and details of their system for complex event processing over streaming data were presented. TelegraphCQ data stream processor was extended to execute complex event continuous queries (CQs), along with traditional SQL-like continuous queries. It was shown how to use the defined system in a real-world scenario. After the details of the implementation of their tool, their experiment results confirmed that providing first-class support for sequence-aware operators in stream processing engines is important.

If events are probabilistic because of imprecise or fuzzy data streams, the state that results from these events is also probabilistic (Probabilistic Complex Event Processing (PCEP)). In the third part, an architecture for Probabilistic Complex Event Processing (PCEP) was presented. Based on this architecture, they built a small prototype for a smart home scenario.

- **year:** It was published in 2005.
- **number of references:** There are 20 references for this paper.
- **whether applied on live environment:** The resulting system was used as the core processing engine in a demonstration at SIGMOD 2005 (a real-world library).
- **tools:** TelegraphCQ was extended to execute complex event continuous queries.
- **language:** CQL.
- **applicability:** The system is general.
- **whether applicable in telecommunication:** The system is general, it can be applied in telecommunication as well.
- **goal of the paper:** They extended TelegraphCQ to help the debugging and analysis of complex systems. They showed a real-world scenario how to detect booklifting alerts.
- **labels:** Probabilistic Complex Event Processing (PCEP), TelegraphCQ

2.3.1.3 A Homogeneous Reaction Rule Language for Complex Event Processing

Paschke et. al. [74] presented an approach based on logic programming which combined and exploited the advantages of the different logic and rule-based approaches. They introduced some global reactions rules followed the ECA paradigm. They defined an interval-based event algebra with event definition, event selection and event consumption. They provided a homogenously-integrated event messaging reaction rule language, called Prova AA.

Then an efficient scalable middleware was implemented, which supports, e.g., complex event- and action processing, event communication/messaging, formalization of reaction rules in combination with other rule types such as derivation rules and transactional ID-based knowledge updates.

They proposed a reaction rule extension, called Reaction RuleML, a standard for rule interchange and rule serialization in XML. RuleML serves as a platform-independent rule interchange format, and it can be translated into platform-specific, executable rule languages (such as their homogenous reaction rule language).

- **year:** It was published in 2007.
- **number of references:** We didn't find any references for this paper.
- **language:** Interval-based Event Calculus Language, homogenously-integrated event messaging reaction rule language(Prova AA), Rule Markup Language (RuleML).
- **applicability:** The system is general.
- **whether applicable in telecommunication:** The system is general.
- **goal of the paper:** This paper defined a homogeneous integration approach that combined derivation rules, reaction rules and other rule types such as integrity constraints into the general framework of logic programming, the industrial-strength version of declarative programming.
- **labels:** reactive Event-Condition-Action (ECA) rules, Event Messaging Reaction Rules, interval based event calculus, Rule-based CEP Middleware

2.3.1.4 Complex Events in Business Processes

Barros et. al. [35] provided an assessment of Business Process Modeling Notation (BPMN) and Business Process Execution Language(BPEL). BPMN is a graphical representation for specifying business processes in a workflow, while BPEL is a standard executable language for specifying interactions with Web Services. BPMN is used as a graphical front-end to capture BPEL process descriptions – the BPMN specification includes an informal and partial mapping from BPMN to BPEL.

They introduced the following kind of common patterns to examine BPEL and BPMN support them or not: co-occurrence, time relation, data dependency, consumption patterns. More than half of these very basic eventing scenarios (8 from 13) are not supported neither by BPEL nor BPMN.

- **year:** It was published in 2007.
- **number of references:** There are 4 references for this paper.
- **language:** Business Process Execution Language(BPEL), Business Process Modeling Notation (BPMN)
- **applicability:** These languages can be used in Business Process Management.
- **goal of the paper:** The aim is to capture activities with their behavioral and data dependencies.
- **labels:** Business Process Execution Language(BPEL), Business Process Modeling Notation (BPMN)

2.3.1.5 Complex event processing in enterprise information systems based on RFID

Zang et. al. [96] implemented an event processing mechanism in enterprise information systems based on RFID, including the data structures, optimization strategies and algorithm that is considered as one of the contributions. Event meta model and rules of complex event processing were defined. An architecture of event processing in enterprise information systems was proposed. To improve the detection efficiency, classification and partitioning of event instances is utilized. Their prototype was compared to Esper, an open source software. The performance evaluations show that the method is effective in terms of scalability and the capability of event processing.

- **year:** It was published in 2007.
- **number of references:** There are 16 references for this paper.
- **realtime:** The detected event can be delivered to the subscribers in real time by RTE-CEP.
- **whether applied on live environment:** Yes, on a major refrigerator manufacturer in China.
- **tools:** A prototype was implemented, which was called RTE-CEP.
- **language:** They presented their own event processing language, including complex event pattern, operators and keys.
- **applicability:** Enterprise information systems based on RFID, sample application was a major refrigerator manufacturer in China.
- **whether applicable in telecommunication:** Partly, only in enterprise information systems based on RFID.
- **goal of the paper:** The aim is to give an architecture to enterprise information systems based on RFID to improve the detect efficiency.

- **labels:** Enterprise Architecture, Event driven architecture, Event meta model, RFID, Work Flow, Application Integration, Service-Oriented Architecture, Enterprise Software Systems

2.3.1.6 Domain Specific Reference Models for Event Patterns - for Faster Developing of Business Activity Monitoring Applications

In this paper [79], the authors connected the CEP to Business Activity Monitoring (BAM) and Business Process Management (BPM). Business Process Management and real-time Business Activity Monitoring are newly discussed as the preconditions for a so-called predictive business and the competitiveness of enterprises in the future. CEP is a technology that shall be the basis in order to achieve actionable, situational knowledge from distributed message-based system, databases and applications in real-time or near real-time.

- **year:** It was published in 2007.
- **number of references:** There are only 2 references for this paper.
- **realtime:** Yes, CEP is a real-time technology, therefore they use this to BAM.
- **whether applied on live environment:** This project has not been implemented yet.
- **tools:** DoReMoPat. In the project DoReMoPat, a catalogue of reference models for selected domains like automotive, finance, logistics, telco are developed and implemented as customizable prototypes.
- **language:** The authors showed an example on Coral8 language, and mentioned some other languages.
- **goal of the paper:** Event filtering and monitoring.
- **labels:** Internet, Enterprise, business, IT, SOA, SOAP, WSDL, web service, SLA, BPEL, Business Process Management (BPM), Business Activity Monitoring (BAM), multi-channeling

2.3.1.7 Identification of Suspicious, Unknown Event Patterns in an Event Cloud

While known event patterns can be derived from heuristics (e.g., from business activity monitoring view), unknown event patterns cannot. According to this article [92], unknown event patterns can be found with the help of event processing agents (EPAs) by analyzing the event cloud of an organization and using specific algorithm to detect them. If an EPA detects an unknown pattern – which seems to be a suspicious event combination – determined by discriminant analysis, EPA sends an alert and this pattern is saved in the database. Discriminant analysis analyzes multidimensional data to discover relationship between data. This method determines groups and the new objects are allocated into one of them. The first step of the the discriminant analysis is to define two groups of events: which are suspicious and which are not. The second steps is to compute the discriminant function and finally to determine the critical discriminant value, which determines which group the given event belongs to. The implementation of an experimental environment based on CEP to test their approach is in progress.

- **year:** It was published in 2007.
- **number of references:** There are 6 references for this paper.
- **realtime:** The discriminant analysis is executed by the CEP engine in real-time.
- **whether applied on live environment:** It has not been implemented yet, it will be tested later.
- **applicability:** In this paper, the method focused in banking domain, but there are further domains which are interested in finding event patterns.
- **whether applicable in telecommunication:** Detecting known and unknown event patterns can be useful in telecommunication area as well.
- **whether applicable for prediction:** Predictive Business predicts the business impact of occurring events or situations by processing real-time events together with historical data stored in DB's.
- **goal of the paper:** The goal of the paper is to identify unknown event patterns by classifying events into specific groups of events where a group can represent an unknown pattern.
- **labels:** internet, enterprise, business, ATM, predict, EDA, event patterns, event cloud, fraud detection

2.3.2 Requirements and principles for complex event processing

2.3.2.1 The 8 requirements of real-time stream processing

This paper [85] presents 8 premises to characterize the requirements for realtime stream processing. The essence of requirements shows some examples, such as military and business application. The presented requirements: keep the data moving, query using SQL on streams, handle stream imperfections, generate predictable outcomes, integrate stored and streaming data, guarantee data safety and availability, partition and scale applications automatically, process and respond instantaneously.

The article presents three different software system technologies. This three system technologies can potentially be applied for solving high-volume low-latency streaming problems. These technologies are DBMSs, rule engines, and stream processing engines. The authors observed that traditional system software failed when it met some of these requirements, justifying the need for and the relative benefits of SPEs.

- **year:** It was published in 2005.
- **number of references:** There are 49 references for this paper.
- **applicability:** Presented requirements are useful in event processing.
- **labels:** Phone, Cell, Wireless, business, DBMS, RFID

2.3.2.2 Wireless Integrated Network Sensors

This paper [78] presents WINS (Wireless integrated network sensors), which provide distributed network and Internet access to sensors, controls, and processors deeply embedded in equipment, facilities, and environment. Such applications require delivery of sensor information to the user at a low bit rate through low-power transceivers.

The authors analyzed power requirement and communication cost of WINS system entirely to level of architecture. They established that: “if the system is to detect objects reliably, it has to be distributed, whatever the networking cost” and “if the application and infrastructure permit, it pays to process the data locally to reduce traffic volume and make use of multihop routing and advanced communication techniques to reduce energy costs”.

- **year:** It was published in 2000.
- **number of references:** There are 1576 references for this paper.
- **realtime:** Yes, the event-detection process is continuous, the sensor, data converter, data buffer, and signal processing have to operate at micropower levels, using a real-time system.
- **whether applied on live environment:** The DARPA-sponsored low-power wireless integrated microsensors (LWIM) project demonstrated the feasibility of multihop, self-assembled, wireless networks.
- **tools:** WINS NG, a Windows CE-based device selected for the availability of low-cost developer tools.
- **applicability:** Applications in such industries as transportation, manufacturing, health care, environmental oversight, safety and security.
- **whether applicable in telecommunication:** Partly, but it is specialized for wireless integrated network sensors.
- **whether applicable for prediction:** WINS useful for data collection.
- **goal of the paper:** Complex data processing in wireless system with a low energy need.
- **labels:** Communication, Wireless, Radio, Media, Internet, data network, WINS, SNR

2.3.2.3 Seven Principles of Effective RFID Data Management

Radio-Frequency Identification (RFID) is a widely used technology nowadays. The main problem of RFID data management is the massive volumes of RFID data. If the most general conclusions are drawn, the value of that data will lost. Palmer [73] gave seven principles to help effectively managing RFID data:

1. **Digest the Data Close to the Source:** The digestion (which is more than basic filtering, it is data cleansing, consolidation, and summarization) ensures greater reliability and protect your IT infrastructure.
2. **Turns Simple Events into Meaningful Events:** CEP deals with the task of processing multiple streams of simple events with the goal of identifying the meaningful events within those streams. With the help of this principle, actionable knowledge can be derived from discreet events.

3. **Buffer Event Streams:** Employ data concentrators that buffer event stream flows by combining RFID middleware, event processing, and an in-memory data cache to achieve the reliable speed you need.
4. **Cache Context:** Determining complex events from simple RFID event data requires context. Context data needs caching for event data.
5. **Federate Data Distribution in Near Real Time:** Federate data distribution, your RFID system can scale globally and as close to real time as possible.
6. **Age RFID Data Gracefully:** Age RFID data to keep your working set of data manageable, enrich raw data with required context, and reduce the load on down-stream systems.
7. **Automate Exception Handling:** Exception handling is the primary job of any RFID system. Automate exception handling is very important to improve overall business efficiency.

- **year:** It was published in 2005.
- **number of references:** There are 16 references for this paper.
- **labels:** RFID, exception handling, data management, real-time

2.3.2.4 Design Patterns for Complex Event Processing

This paper [75] presents some CEP categories, such as categorization according to good and bad solutions, categorization according to the abstraction level, categorization according to the intended goal, categorization according to the management level. Furthermore, in this paper a distinction between the design/modelling perspective and the processing perspective is made by a set of fundamental definitions. Authors present two general pattern language templates for CEP patterns and antipatterns, introducing the necessary elements which should be commonly included into more specific pattern instantiations of these templates.

- **year:** It was published in 2008.
- **number of references:** There is only 1 reference for this paper.
- **labels:** Communication, Enterprise, business, BPM, BAM

2.3.3 Methods and experiments

2.3.3.1 Business Activity Monitoring of norisbank Taking the Example of the Application easyCredit and the Future Adoption of Complex Event Processing (CEP)

This paper [46] presents the kernel business process of easyCredit of the norisbank. Business Activity Monitoring (BAM) is a basic condition for the successful operation of the system. The norisbank faced with the future concept based on CEP/ESP. The norisbank has realized BAM by a pipeline model. This model assumes that each credit application runs through several processing steps within its entire life cycle. The single steps can be imagined as a production line or a pipeline.

A pipeline model is used for the technical monitoring of the business processes by the norisbank. This model assumes that every credit application within its total life cycle runs through several processing steps, like in a production line or in a pipeline.

- **year:** It was published in 2006.
- **number of references:** There are only 2 references for this paper.
- **realtime:** Yes, with CEP/ESP messages, information or data are correlated, aggregated, analyzed and evaluated in real time.
- **whether applied on live environment:** According to the results, they decision on the credit.
- **language:** Event-Processing Language (EPL).
- **applicability:** This is a case study in banking area.
- **whether applicable in telecommunication:** No.
- **goal of the paper:** BPEL-engine shall be used and which events of which actions in the credit process will have to be generated.
- **labels:** Internet, Enterprise, business, SOA, SOAP, WSDL, BPEL, BAM, ESP, Business Process Management System

2.3.3.2 Complex Event Processing in Distributed Systems

This paper [64] presents an overview of complex event processing (CEP). The authors applied CEP for a particular example of a distributed message-based system, a fabrication process management system. The concepts of causal event histories, event patterns, event filtering, and event aggregation were introduced and their application to the process management system was illustrated by simple examples. In this paper, the authors illustrated the concepts of RAPIDE complex event processing and how they applied it for a particular system. In RAPIDE complex event processing activities and operations can be placed into an abstraction hierarchy. In the case of message-based systems, authors can organize the activities and events into layers.

- **year:** It was published in 1998.
- **number of references:** There are 57 references for this paper.
- **realtime:** Yes, RAPIDE supports realtime data filtering and processing.
- **whether applied on live environment:** This paper illustrates how complex event processing. can be applied for using the RAPIDE toolset on one specific kind of system.
- **tools:** RAPIDE complex event processing.
- **language:** RAPIDE.
- **applicability:** Typical examples are systems supporting commercial applications such as distributed financial transaction processing systems, warehousing systems, and fabrication process control systems.
- **whether applicable in telecommunication:** No.
- **whether applicable for prediction:** Yes, data filtering.
- **goal of the paper:** The aim of the paper is data filtering.
- **labels:** Communication, Internet, Media, Enterprise, business, TIBCO

2.3.3.3 On classification and segmentation of massive audio data streams

The aim of the article [31] is quick online classification and recognition of massive voice streams. The authors constructed a number of techniques which were designed stream based online processing of massive number of audio data. Since popularly used models, such as Gaussian mixture modeling (GMM), are not very appropriate for real time speaker modeling (because of its high computational cost), they defined a micro-clustering technique for voice classification and recognition in order to create fast cluster based on signatures of the data. The experimental results showed that their technique was not only more efficient, but the accuracy of the micro-clustering process was significantly higher than the GMM approach. They determined outlier speech segments in order to identify anomalous patterns of the data stream.

- **year:** It was published in 2008.
- **number of references:** We didn't find any references for this paper.
- **realtime:** The recognition process of massive voice streams is online.
- **whether applied on live environment:** No, the used data set was the HUB-64 data set, which contained the voice signal of 64 different public personalities such as Wolf Blitzer, Bill Clinton, Al Gore, Candy Crowley, etc.
- **tools:** An online voice recognition system.
- **applicability:** Telecommunication.
- **whether applicable in telecommunication:** This method is applicable in telecommunication, mainly in areas where voice stream is handled.
- **whether applicable for prediction:** With the help of outlier detection, anomalies can be detected and faults can be predicted based on these anomalies.
- **goal of the paper:** They concentrated on the problem of text-independent speaker classification and recognition in which the actual textual content of the speech is not available for modeling purposes.
- **labels:** VOIP, GMM

2.3.3.4 What is "Next" in Event Processing?

The aim of this paper [91] is to present a formal framework for the study of sequencing in event processing systems. They discussed the temporal model: how events are sequenced – what is the "next" event? – and how the time stamp of an event is

represented.

First, they emphasized the importance of event system's successor definition and examined how the different successor definitions were processed in three existing event systems: SnoopIB, Active Office, and Cayuga. All three use intervals as time stamps, and they have the same partial order on the intervals. However, the three systems differ in how they choose a successor.

Then they stated standard and desirable axioms, none of the definitions of successor of the mentioned tools satisfied all axioms. They tried to find the best model that satisfied all of their axioms. Finally they identified two canonical temporal models. The complete-history model had serious implementation issues because it required time stamps of unbounded size, while the interval-based time stamp model required time stamps of bounded size.

- **year:** It was published in 2007.
- **number of references:** There are 11 references for this paper.
- **realtime:** The events arrive real-time in a linear fashion.
- **whether applied on live environment:** It was not tested.
- **applicability:** This model is general.
- **whether applicable in telecommunication:** Yes, this model is general.
- **goal of the paper:** The aim is to define a model to determine how events are sequenced, and how the time stamp of an event is represented.
- **labels:** RFID, axiomatization, temporal model

2.3.3.5 Detecting Event Processing Patterns in Event Databases

Mihaeli et. al. [66] presented a generalized event database structure to design generalized algorithms, defined a particular event definition language, called AMIT. They realized an event query processor (EQP) that accepted as input the event rule definitions of the AMIT. EQP parses the event definition specification of the CEP rule engine and queries the database. AMIT uses XML specifications, which define entity-types – event, situation, lifespan, key – and situation operators – joining, counting, absence, temporal operators – of AMIT. Then they introduced layered information architecture for the query processor and considered the logic to implement the different AMIT constructs.

- **year:** It was published in 2007.
- **number of references:** We didn't find any references for this paper.
- **realtime:** This approach uses high-level event processing pattern language both for inline events and for retrospective processing.
- **language:** This approach uses high-level event processing pattern language, called AMIT.
- **applicability:** It is a general solution for detecting event processing patterns.
- **whether applicable in telecommunication:** Yes, it is a general solution for detecting event processing patterns.
- **goal of the paper:** The aim is to detect event processing patterns.
- **labels:** database, AMIT

2.3.3.6 Consistent Streaming Through Time: A Vision for Event Stream Processing

Barga et. al. [34] observed that CEP technologies (i.e. data stream management, complex event processing, asynchronous messaging) share a common processing model, but differ in query language features and consistency requirements. They predicated that these are complementary technologies. A new system, called CEDR (Complex Event Detection and Response), was developed, which integrated the mentioned technologies and supports a spectrum of consistency guarantees. They have only an initial implementation: a stream data model and a declarative query language with logical and run-time operators. In CEDR temporal stream model the notion of time divided into system time and application time, which was refined into two temporal dimensions, valid time and occurrence time. The CEDR query language is based on event pattern expression, instance selection and consumption, instance transformation. They handled negation, event selection and consumption, application driven modifications and out-of-order event delivery in flexible way.

In their another technical report, the authors compared CEDR to STREAM, Aurora, Niagra, Nile, Cayuga, HiFi.

- **year:** It was published in 2006.
- **number of references:** There are 20 references for this paper.
- **whether applied on live environment:** CEDR was not a complete system at that time, only the foundations of CEDR were overviewed. Testing this system is missing from this paper.
- **tools:** CEDR (Complex Event Detection and Response) tool was developed.
- **language:** CEDR query language was designed.
- **applicability:** The system is general, it can be applied in any kinds of areas.
- **whether applicable in telecommunication:** Since CEDR works with general events, the system can be applied in telecommunication as well.
- **goal of the paper:** The goal of the paper is developing CEDR in order to react immediately to business critical events.
- **labels:** enterprise, business, Microsoft, CEDR, Query Language

2.3.3.7 Bridging Physical and VirtualWorlds: Complex Event Processing for RFID Data Streams

First, this paper [89] presents RFID (radio frequency identification) technology. RFID provides fast data collection with precise identification of objects with unique IDs without line of sight, thus it can be used for identifying, locating, tracking and monitoring real world objects. In this paper, the authors take an event-oriented approach to process RFID data, by devising RFID application logic into complex events. An RFID system consists of a host computer, RFID reader, antenna (which is often integrated into readers), transponders or RF tags. An RFID tag is always uniquely identified by a tag ID stored in its memory, and can be attached to almost anything.

- **year:** It was published in 2006.
- **number of references:** Ther are 42 references for this paper.
- **realtime:** Yes, RFID rules can also provide effective support of real-time monitoring.
- **whether applied on live environment:** No, authors just tested the total event processing time versus the number of primitive events and versus the number of rules, and finally they achieved 1000 events per second.
- **tools:** The technology was developed in this paper is now integrated into Siemens RFID Middleware.
- **language:** The authors defined Pseudo Events language.
- **applicability:** With the significant advantages of RFID technology, RFID is being gradually adopted and deployed in a wide area of applications, such as access control, library checkin and checkout, document tracking, smart box, highway tolls, logistics and supply chain, security and healthcare.
- **whether applicable in telecommunication:** Partly, where sensors are used.
- **whether applicable for prediction:** RFID technology is useful for data collection.
- **goal of the paper:** The declarative event-based approach greatly simplifies the work of RFID data processing, and significantly reduces the cost of RFID data integration.
- **labels:** Enterprise, business, SAP, RFID, traditional ECA (Event-Condition-Action) rule systems, Real-Time Monitoring, RFID Complex Event Detection

2.3.3.8 The Event Tunnel: Interactive Visualization of Complex Event Streams for Business Process Pattern Analysis

Suntinger et. al. [86] presented an event-based business intelligence tool, the Event Tunnel framework. It is a framework for interactive visualization of event streams (display relationship between events) integrated with query and filtering tools for searching relevant events. According to the attributes of the event, the size and the color of the event can be different. It provides two main views: top view, which shows stream of events along the time-axis and side view, where the events are in temporal order. If the data set is larger than $40,000$, the events are clustered and aggregated groups of event into data points. The user can define clustering rules. They applied this framework for two business applications: automated fraud detection in online betting platforms and real-time monitoring of logistics and transportation processes.

- **year:** It was published in 2007.
- **number of references:** There are 5 references for this paper.

- **realtime:** Yes, this tool is suitable for automated fraud detection in online betting platforms and real-time logistics and resource planning.
- **whether applied on live environment:** No, the examples were generated using a simulation model.
- **tools:** They introduced Event Tunnel visualization tool.
- **language:** The analysis framework was written in C#.
- **applicability:** This framework can be used for automated fraud detection in online betting platforms and real-time logistics and transportation processes.
- **whether applicable in telecommunication:** Events from telecommunication area can be visualized by this tool as well.
- **goal of the paper:** The aim is a visualization tool for accomplishing back-tracking, drawing conclusions on the business performance and discovering undetected correlations and patterns.
- **labels:** Business process visualization, query-driven visualization

2.3.3.9 Real-time performance monitoring and optimization of cellular systems

This article [77] describes an *architecture* that employs event-based statistics for building effective and flexible performance monitoring and optimization applications. The prototype of monitoring and optimization system includes the drop rate measurement, failed handover rate measurement, traffic load measurement and hierarchical cell structure (HCS) control tasks, which implement the optimizing algorithm.

The *prototype system* could monitor 75 of the 105 cells. Capacity was improved, congestion was reduced from 5-10% to 1%. Latency measurements showed that it took between 10 and 35 seconds for an event in the network to be measured and displayed in the GUI. In most cases a latency of less than one minute is acceptable.

- **year:** It was published in 2002.
- **number of references:** There are 7 references for this paper.
- **realtime:** Yes, the R-PMO provides real-time presentation of several basic monitors related to the performance of the radio network.
- **whether applied on live environment:** Yes, the field trial took place in Hong Kong in cooperation with SmarTone with 105 test cells.
- **tools:** They introduce a real-time performance monitoring (R-PMO) application.
- **applicability:** The prototype can be used for monitoring and optimization of cellular networks.
- **whether applicable in telecommunication:** The prototype can be used in mobile communication (GSM systems).
- **goal of the paper:** The goal of the paper is to introduce R-PMO, whose aims are drop rate monitoring, traffic load monitoring and HCS control tasks.
- **labels:** Telecommunication, mobile phone, GSM, cellular, mobile station, WCDMA, Ericsson

2.3.3.10 High-Performance Complex Event Processing over Streams

This paper [94] presents the design, implementation, and evaluation of a system (called SASE) that executes complex event queries over real-time streams of RFID readings encoded as events. Authors proposed a complex event language that allows queries to filter and correlate events and transform the relevant ones into new composite events for output. The language provides features such as sequencing, negation, parameterization, and windowing necessary for emerging RFID-based monitoring applications. They demonstrated the effectiveness of SASE in a detailed performance study, and compare SASE to a relational stream processor, TelegraphCQ.

- **year:** It was published in 2006.
- **number of references:** There are 88 references for this paper.
- **realtime:** Yes, complex event queries over real-time streams of RFID readings are encoded as events.
- **whether applied on live environment:** No, but the authors demonstrated the effectiveness of SASE. Results of this study show that SASE can process $40,000$ events per second for a highly complex query in a Java-based implementation.

- **tools:** SASE, an event processing system that executes complex event queries over real-time streams of RFID readings.
- **language:** SASE Event Language. The SASE event language is a declarative language that combines filtering, correlation, and transformation of events.
- **applicability:** SASE is a monitoring system to streams of RFID readings.
- **whether applicable in telecommunication:** No.
- **goal of the paper:** The goal is to monitor streams of RFID readings.
- **labels:** wireless, radio, SASE, RFID, Stream-based, SASE Event Language, optimization

2.4 Tools

In this section, we present the tools of complex event processing of various companies, academics and some non commercial organizations. Similarly to the articles, we define evaluation aspects to raise important details for better visibility. First of all, the term **developer** shows the name of the producer company or the academy. The term **development state** means that the given tool is under development and/or still supervised by the owner, or by now the development of the tool has been finished, and there is no further support for it. The term **licence** tells us whether the tool is free or a licence fee has to be paid. The term **applicability** indicates the domains where the tools are usable, the kinds of problems which it is suitable for. It also gives some pieces of information about the capability of technology. In the cases where we find **references**, we can get some information about practicability, real applicability, and a picture about how this tool and its developer can handle real world blocks. We also describe **accessibility** about demos, presentations, or the free trials that are available for a given tool, while the **query language** aspect gives information about the event query language. At last, we introduce the potential **related tools**. A related tool can be, for example, a commercialized variant of the actual tool, a tool which employs some feature of the actual tool, etc. We do not include the aspects if they are not mentioned in the paper.

In Figure 2, we summarize the tools considering their development time, license type and whether it is released or not. Every tool is represented with a box, which has its left and right bounds are based on the time when the development is started and finished. However if it is currently under development, it is drawn as to be finished in 2010. (In cases of some tools, we could only give an estimation, especially for the start date, because the exact information is not available now). There is a time axis in the center which lasts from 2000 to 2010. The free tools are shown below the axis, while the commercial tools are shown above the axis. If a tool has not been released yet, then its box is drawn with a dash line. The connection between the tools is marked with arrows.

2.4.1 Commercial tools

2.4.1.1 Progress Software - Progress Apama

Apama Event Processing Platform [16] is a complete CEP based tool for Business Activity Monitoring. It provides many built-in applications for developing and supervising enterprise events. The CEP engine can handle inbound events within subseconds, find defined patterns, alert or respond to actions. With Apama Event Modeler, developers can create applications via graphical user interface, which are presentable with Apama Research Studio. Apama Dashboard Studio provides a set of tools to develop visually rich user interfaces. Via Apama dashboards, users can start/stop, parameterize and monitor event operations from both client and browser desktops. The Apama package includes many major adapters to handle communication with other components and applications. The company has bought the Apama software in 2004.

- **developer:** Progress Software
- **development state:** Live, and fully supported. (2004 - now)
- **licence:** Commercial licence.
- **applicability:** It can be used in wide variety of Business Activity Monitoring.
- **accessibility:** Demo request available.
- **technical data:** Sub-millisecond response time.
- **query language:** Apama Event Modeller graphical point and click application and Apama event processing language, Monitorscript via Java and native script version.

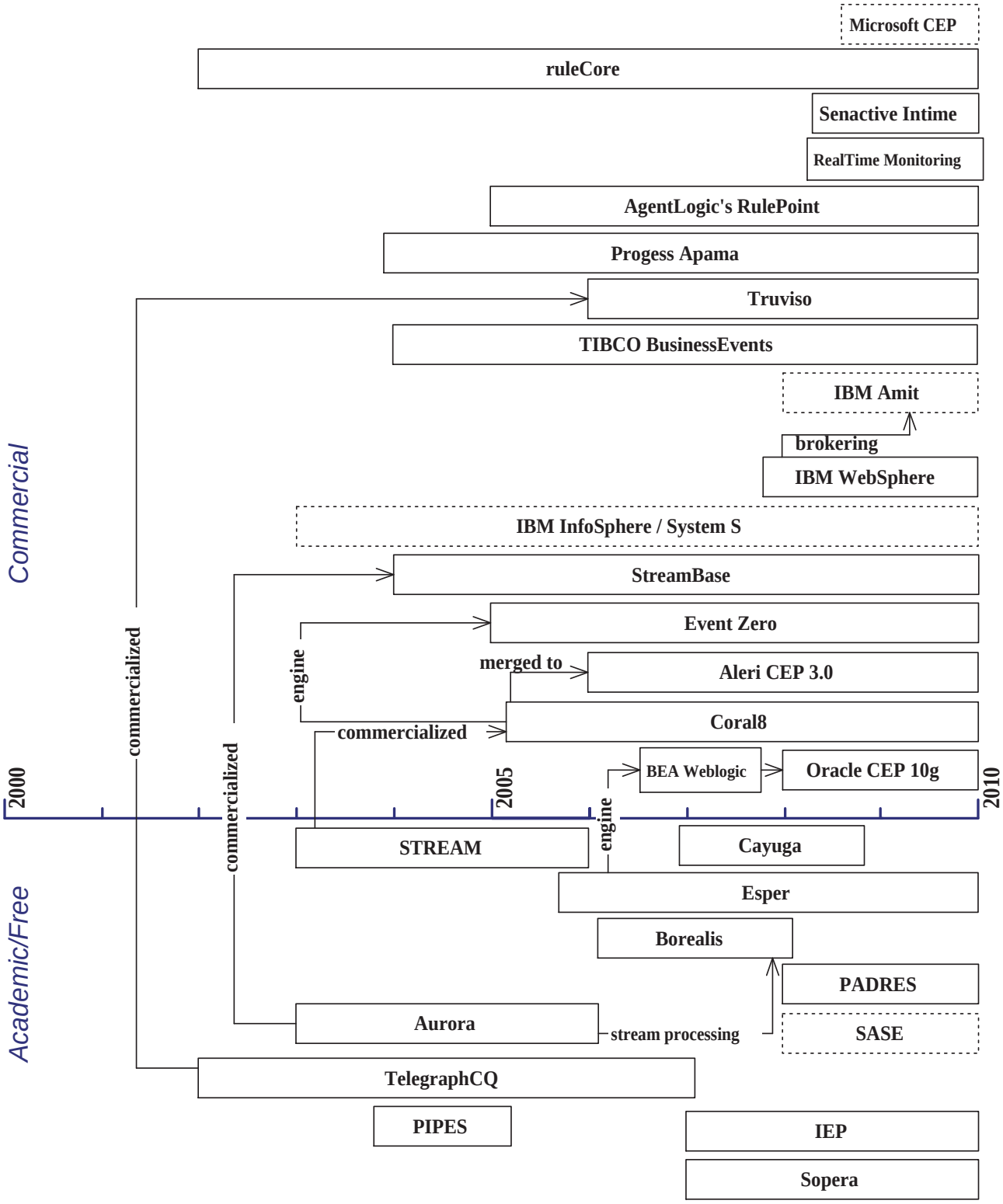

Figure 2. CEP tool timeline

2.4.1.2 Tibco - TIBCO BusinessEvents

It helps organizations to identify and address situations and trends by correlating massive volumes of data about discrete events and applying predefined rules to identify situations that require a response. After that, it can able to run processes, or notify human resource if it is necessary. With an UML-based modeling tools, users can define relation models, which describes causal relationships between event sources. For managing rules among events, there is a rule editor for IT members to determine how events correlate and aggregate, define filters and add thresholds. BusinessEvenst gives full support for other Tibco [26] software, and some other major developer's tools, like IBM. We mention that TIBCO developed a predictive analytic toolset as well, Spotfire Miner and TIBCO Spotfire S+ Serve.

- **developer:** Tibco
- **development state:** Live and fully supported. ('90s - now)
- **licence:** Commercial license.
- **applicability:** Suitable for many kind of Event Driven business domains.
- **accessibility:** Presentation demo only.
- **query language:** UML-based modeling tool for relationships and rule editor for managing events.

2.4.1.3 Aleri - Aleri CEP 3.0

Aleri's CEP platform [1] is a complex, robust complex event processing application designed for analyzing large amount of various data in real time. It offers ability to filter, combine and normalize incoming data and can be used to detect important patterns, changed conditions, security problems and much more. It can be used to alert when events occurs, or able to react to events itself without supervising. Aleri provides wide range of integrated tools to improve productivity. With Studio 3, developers can create and manage their applications and the event processing course. Wide range of built-in adapters provide interfaces for JDBC, ODBC, JMS, etc. With the XML based AleriML language data models can be defined, while its SPLASH developer script language helps to develop much more complex applications what is unable to do with standard relational programming languages. Aleri had been merged with Coral8 and technology is integrated as well.

- **developer:** Aleri
- **development state:** Live and fully supported. (released 2006 - now)
- **licence:** Commercial licence.
- **applicability:** Wide range of enterprise solutions, like data analysis, supervising sensor data, detecting and reacting to network problems.
- **references:** Many commercial enterprises and global bank treasuries.
- **accessibility:** Commercial 180 day trial version available. (Full enterprise-class version of the Aleri Streaming Platform)
- **query language:** Graphical interface to users, AleriML for data models and SPLASH script language for control over CEP system.
- **related tools:** Aleri was merged with Coral8.

2.4.1.4 Coral8 Inc. - Coral8

Coral8 [6] engine provides CEP technology for live analytic tools and provides continuous up to moment business intelligence, trade data, online interaction and RFID support. With the CCL language, or by using related components derived for Coral8 Engine, developers and customers are able to create personalized applications quickly, event filtering patterns and graphical diagrams, visualize important data. With the SQL-based CEP language, CCL (Continuous Computation Language) programmers can easily create new applications, using the benefits of similarity to SQL. With Coral8 Studio, developers can manage modules, maintain streams and stream schemas, define adapters and compile, test and distribute CCL modules. The Coral8 Portal provides end-users fully graphically interface to query real time data, create diagrams, and supervise event flow. The system is tuned for handle many of data formats like SQL, XML, and much more. In 2009, Coral8 and Aleri were merged, and combined the two technology into one to create more powerful applications.

- **developer:** Coral8 Inc.

- **development state:** Live and fully supported. (released 2005 - now)
- **licence:** Commercial license.
- **applicability:** Provide Continuous Intelligence and real time data analysis in capital markets, telecommunications, e-commercials, RFID and sensor domains.
- **accessibility:** Developer version is available. Usable to create application for engine, test and evaluate it.
- **query language:** Continuous Computation Language and Coral8 Studio graphical environment.
- **related tools:** Coral8 was merged with Aleri.

2.4.1.5 Realtime Monitoring GMBH - RealTime Monitoring

RealTime Monitoring [17] is a pure Java CEP application, it is suitable for many IT event processing analysis. The engine is the CEP RTM Analyzer, which is able to filter, correlate and aggregate incoming events through SQL based queries. It supports multicore processors, and give solution to migrate working processes to other systems on-the-fly if necessary. Due to the pure Java engine, the tools are implementable in any operating systems, and it gives wide range of available adapters for accessing data sets, other applications or visualize results.

- **developer:** Realtime Monitoring GMBH.
- **development state:** Live and fully supported. (? - now)
- **licence:** Commercial licence.
- **applicability:** Suitable for many IT enterprise domain, like telecommunication solutions, e-commercial, etc.
- **references:** CITT, Cordys, Ubigrate, BITSz, Global Infinity, Bitkom, GFFS.
- **accessibility:** Demo presentations are available in Manufacturing, Capital Markets, and Telecommunication subjects.
- **technical data:** 100.000 events per second.
- **query language:** Java and .Net SDK to develop applications.

2.4.1.6 IBM - IBM WebSphere Business Events

IBM's WebSphere Business Events [11] is a Business Event Processing (BEP) tool. The tool is enable for business users to apply business concepts such as business vernacular, heuristics, business logic, and business event correlations and relationships. It has a graphical interface to manage system without any IT, or program coding knowledge. It can alert and trigger actions when determinated events occur. The main goal is to enhance existing Business Process Management (BPM) and service-oriented architecture (SOA) infrastructures. It supports both simple and complex business event pattern correlations.

- **developer:** IBM.
- **development state:** Live and fully supported. (? - now)
- **licence:** Commercial licence.
- **applicability:** Usable for Business Process Management and service-oriented architecture infrastructures.
- **references:** ActiveCare Network healthcare company.
- **accessibility:** Only illustration demo is available.
- **query language:** Fully graphical user interface for manage and supervise work. No need any IT or programming knowledge.

2.4.1.7 IBM - IBM InfoSphere Streams (supposed System S)

IBM's InfoSphere Streams [10] is a massive stream processing software platform for analyze huge continuous data stream on the fly. InfoSphere Streams supports high volume, structured and unstructured streaming data sources such as images, audio, voice, VoIP, video, TV, financial news, radio, police scanners, web traffic, email, chat, GPS data, financial transaction data, satellite data, sensors, badge swipes, etc. It provides large security. A prototype is now running on a Blue Gene in TD Bank Financial. The system handles 5 million messages per second with 150 microsecond latency range. Probable release date 2010.

- **developer:** IBM

- **development state:** Under development. No available version. (2003 - ?)
- **licence:** Commercial licence.
- **applicability:** Stream processing in huge and unstructured data domains.
- **references:** Financed and supported by US Government.
- **accessibility:** At this time, this tool is not available.
- **query language:** Spade, Semantic Solver, Eclipse-based Workflow Development Tool Environment (based on operators of Spade).

2.4.1.8 IBM - IBM Active Middleware Technology (Amit)

IBM Active Middleware Technology [9] is a lightweight and agile Java based complex event processing(CEP) engine. It aims to ease the cost of changing business logic, automating and monitoring business processes, and enabling an on demand business environment. The application uses WebSphere brokering technology, which is implemented through processing nodes. IBM Active Middleware Technology™ can be used to configure middleware that answers specific business needs.

- **developer:** IBM
- **applicability:** Wide range of commercial, financial and healthcare systems.
- **references:** University Hospital of Nice.
- **accessibility:** Presentation demos are available.
- **query language:** Java language.
- **related tools:** WebSphere brokering technology.

2.4.1.9 Oracle - Oracle Complex Event Processing 10g

Oracle Complex Event Processing [13] tool is a lightweight modular, fully java based CEP tool. It provides many built-in applications to manage event processing within enterprises. The system runs on JRockit JVM, which offers deterministic garbage collection. Oracle Complex Event Processing provides a declarative Event Processing Language (EPL) for creating sets of queries for processing incoming events and generates sets of developer configurated POJO (Plain Old Java Object) for further processing, what able with fully Java powered applications. Through the fully Java support, several built-in adapters are available, like JDBC, ODBC adapters. Oracle brought BEA in 2008, and integrated their solutions into Oracle Complex Event Processing 10g.

- **developer:** Oracle
- **development state:** Live and fully supported. (? - now)
- **licence:** Commercial licence.
- **applicability:** First of all for enterprise and commercial use. In that domain usable for various problems.
- **accessibility:** Demo is available for download.
- **technical data:** 1 million/sec++ throughput and microsecond latency.
- **query language:** Java
- **related tools:** In 2008 Oracle acquired BEA (WebLogic tool)

2.4.1.10 BEA - BEA WebLogic

BEA WebLogic [3] is a Java based event driven architecture based on Esper. Similarly to other tools, it has built-in adapters, various related tools, applications. Due the integration to Oracle, many BEA solutions had been built into Oracle Complex Event Processing 10g.

- **developer:** BEA
- **development state:** BEA solutions were integrated into Oracle applications, no further alone release. (2007 - 2008)
- **licence:** Commercial licence.
- **applicability:** Enterprise applications and SOA solutions

- **accessibility:** It is downloadable from Oracle's page, earlier releases are available freely.
- **query language:** Java based development language.
- **related tools:** Integrated to oracle. The base of its technology came from Esper.

2.4.1.11 Agent Logic - RulePoint

Rulepoint [19] is a server based Event Processing platform designed for enterprises. It supports detecting events and is able to responde, like running programs or alert users automatically. It can process many enterprise events. The system is configurable on the fly via explorer based point and click wizard, or using programming language.

- **developer:** Agent Logic
- **development state:** Live and fully supported. (1999 - 2009)
- **licence:** Commercial licence.
- **applicability:** Designed for enterprise use.
- **accessibility:** Only full version is available for license fee.
- **query language:** Point and click wizard, or programming language.

2.4.1.12 StreamBase Systems - StreamBase

StreamBase [24] is a high performance Event Stream Processing platform, which provides efficient solution to build powerful applications for almost any usage area. Usable for real time monitoring, analyzing, and handling problems. It supports fast development via graphical event-flow language and supports StreamSQL for providing ease of use, flexibility, and extensibility for developers. This widely used software gives solutions for Telecommunication, Capital Markets, Intelligence and Military, E-commercial, Multiplayer Online Gaming areas. In telecommunication, it gives services like Network Monitoring and Protection, Bandwidth and Quality-of-Service Monitoring, Fraud Detection and Location-based Services and even more. StreamBase technologies are widely used and known. It has won a lot of awards.

- **developer:** StreamBase Systems
- **development state:** Live and fully supported. (2001 - now)
- **licence:** Commercial licence.
- **applicability:** Event Stream Processing (telecommunication, e-commercial, markets)
- **references:** 6 of the top 10 global investment banks, various enterprises, government/intelligence agencies, multiplayer online gaming companies, large Web portals and virtual 3-D communities, and ISVs in anti-money laundering, fraud prevention, network monitoring using this application.
- **accessibility:** Free trial is available for download.
- **query language:** Graphical event-flow language and StreamSQL.

2.4.1.13 Truviso Inc. - Truviso

Truviso [27] is a lightweight tool based on TelegraphCQ academic approach, made by professors and PhD students of UC Berkeley in 2006. The main approach was to create online analytic tool for processing queries on the fly. The TruCQ Engine processes SQL query language (based on PostgreSQL) and supports publish/subscribe enterprise messaging environments, and can send or receive data streams from other instances of TruCQ. Many of major adapters are integrated, support connectivity with ODBC/JDBC and publish/subscribe interfaces.

- **developer:** Truviso Inc.
- **development state:** Live and fully supported. (2006 - now)
- **licence:** Commercial licence.
- **applicability:** Wide area of enterprise domains.
- **references:** Deliver solutions to logistics, media, social, content delivery, networking, and security industries.
- **accessibility:** Demo can be request via web.
- **query language:** SQL-Based Continuous Query language.

- **related tools:** Advance of the research of Berkeley's Telegraph project.

2.4.1.14 Rulecore - ruleCore CEP Server

RuleCore [18] is a Complex Event Processing engine fitted to work with third-party event processing applications through built in adapters and interfaces. It supports fragmented system domains. Queries are programable through declarative XML language, and it provides alerts and automatic actions. The program is available in OEM Kit, it provides access to source code to write better fitted applications to system states.

- **developer:** Rulecore
- **development state:** Live and fully supported. (2002 - 2009) (ruleCore released in 2008)
- **licence:** Commercial licence.
- **applicability:** It is suitable for any business solutions.
- **accessibility:** OEM Kit is available through support system.

2.4.1.15 Senactive - Senactive InTime

InTIme tool [20] is an event driven CEP application to analyze events, detect patterns and react those with automatic actions or alerts. The InTime box includes EventAnalyzer for analyze inbound events using matching patterns created in graphical Modellin Studio. Simulation Studio provides simulations to debug and check event rules and provide graphical results for users.

- **developer:** Senactive
- **development state:** Live and fully supported. (? - now)
- **licence:** Commercial licence
- **applicability:** For event driven solutions.
- **references:** AIM, AirPlus, One, Montana Capital, T-System Austria, T-Mobile Deutschland, Quelle and more.
- **accessibility:** Slideshow and free trial version are available.
- **query language:** Graphical event rule modeling tool.

2.4.1.16 Event Zero - Event Zero

Event Zero [8] is an Event Processing Network platform designed for many types of event processing solutions. It is able to capture incoming events for analyze those, and react to it, or alert supervisors. With Event Zero's Visualization and Response framework, users can get visual data about streams, the system state, and can visualize output data. It also supports external applications.

- **developer:** Event Zero
- **development state:** Live and fully supported. (2005 - now)
- **licence:** Commercial licence.
- **applicability:** Event driven networks.
- **references:** Coral8 is a solution partner.
- **accessibility:** Perpetual License with Maintenance, Software Subscription and a Hosted Service are available.
- **query language:** Graphical built-in interface.
- **related tools:** Coral8

2.4.2 Academic and free tools

2.4.2.1 EsperTech Inc. - Esper/NEsper

EsperTech [7] brings Event Stream Processing (ESP) and Complex Event Processing (CEP) to mainstream with an Open Source approach, ensuring rapid innovation with quality productization, support and services for mission critical environments, from SOA to eXtreme Transaction Processing deployments. EsperTech runs on any Java 5 or Java 6 JVM Fully embeddable.

- **developer:** EsperTech Inc.
- **development state:** Live and fully supported. (2008 - now)
- **licence:** Free licence.
- **applicability:** Esper Studio is an operational console for streamlined management and real-time view applications.
- **query language:** Event Processing Language (EPL).

2.4.2.2 Middleware Systems Research Group (MSRG)/University of Toronto - PADRES

PADRES (Publish/Subscribe Applied to Distributed Resource Scheduling) [14] is an enterprise-grade event management infrastructure that is designed for large-scale event management applications. Ongoing research seeks to add and improve enterprise-grade qualities of the middleware. A publish/subscribe middleware provides many benefits to enterprise applications. Content-based interaction simplifies the IT development and maintenance by decoupling enterprise components. As well, the expressive PADRES subscription language supports sophisticated interactions among components, and allows fine-grained queries and event management functions. Furthermore, scalability is achieved with in-network filtering and processing capabilities.

- **developer:** Middleware Systems Research Group (MSRG)/University of Toronto.
- **development state:** Live and fully supported. (2008 - now)
- **licence:** Free licence.
- **applicability:** Sensor integration, Business process execution.
- **references:** Cybermation Inc, NSERC, Sun Microsystems of Canada and OCE/CITO.
- **accessibility:** Full program can be downloaded. (Binary Release v1.5 [19-05-2009]).
- **query language:** PADRES subscription language.

2.4.2.3 CollabNet, Inc. - Intelligent Event Processor (IEP)

Intelligent Event Processor (IEP) [12] is an open source Complex Event Processing (CEP) and Event Stream Processing (ESP) engine. IEP is a JBI Service Engine and is a part of the Open ESB community. OpenESB is an open source project with the goal of building a world-class Enterprise Service Bus. An ESB provides a flexible and extensible platform on which to may build SOA and Application Integration solutions.

- **developer:** CollabNet, Inc.
- **development state:** Live and fully supported. (2007 - now)
- **licence:** Free licence.
- **references:** AllGateways Software, Pymma, AdvanTech, Imola Informatica Srl, Bostech Corporation, Eviware, Gestalt LLC
- **accessibility:** Full program can be downloaded.
- **query language:** Continuous Query Language (CQL).

2.4.2.4 SOPERA GmbH - Sopera

SOPERA [21] is a complete and proven SOA platform, which is rigorously oriented to practical requirements. Companies and organizations benefit from the SOA know-how integrated in SOPERA during implementation of sophisticated SOA strategies.

- **developer:** SOPERA GmbH
- **development state:** Live and fully supported. (2007 - now)
- **licence:** SOPERA ToolSuite download free. (Eclipse Public Licence)
- **references:** Ancud IT-Beratung GmbH, Engineering Group, EXXETA, Salmaco
- **accessibility:** Full program download.

2.4.2.5 UC Berkeley/University of Massachusetts Amherst - Stream-based And Shared Event Processing (SASE)

The goal of SASE [23] research project is to design and develop an efficient, robust RFID stream processing system that addresses the challenges in emerging RFID deployments, including the data-information mismatch, incomplete and noisy data, and high data volume, and it enables real-time tracking and monitoring.

- **developer:** UC Berkeley/University of Massachusetts Amherst
- **development state:** It is developed since 2008, no release version.
- **applicability:** RFID-based retail and inventory management, healthcare, financial services, and transport networks.
- **accessibility:** Full program can be downloaded.
- **query language:** SQL-like.

2.4.2.6 Cornell University - Cayuga

The Cayuga [5] system architecture is designed to efficiently support a large number of concurrent subscriptions. Its core components include a query processing engine, an index component, a meta data manager, and a memory manager. The query processing engine extends the functionality of traditional publish/subscribe to support stateful subscriptions. Cayuga requires users to further specify their interests in the structured Cayuga Event Language (CEL).

- **developer:** Cornell University.
- **development state:** released (2006 - 2008)
- **licence:** Free licence (BSD license).
- **applicability:** Monitoring The Blogosphere and Social Networks, Technical Analysis for Stock Investors, etc.
- **accessibility:** Full source code downloaded.
- **query language:** Cayuga Event Language (CEL).

2.4.2.7 Brandeis University, Brown University, and MIT. - Aurora

The primary goal of the Aurora project [2] is to build a single infrastructure that can efficiently and seamlessly meet the requirements of demanding real-time streaming applications. This project has been superseded by the Borealis project.

- **developer:** Brandeis University, Brown University, and MIT.
- **development state:** The Aurora project has been superseded by the Borealis project. (released 2003 - 2006)
- **licence:** Free licence.
- **applicability:** Real-time monitoring applications, environmental monitoring, surveillance, tracking, plant maintenance, and telecommunications data management.
- **accessibility:** Full program can be downloaded.
- **query language:** Aurora Stream QUery ALgebra (SQUAL)
- **related tools:** Borealis

2.4.2.8 UC Berkeley - TelegraphCQ

TelegraphCQ [25] supports continuous queries over a combination of tables and data streams.

- **developer:** UC Berkeley
- **development state:** 2002-2007
- **licence:** Free licence (BSD license).
- **accessibility:** Full program can be downloaded.
- **query language:** Continuous query language (CQL).

2.4.2.9 Brandeis University, Brown University, and MIT. - Borealis

Borealis [4] is a distributed stream processing engine (SPE). It takes a set of streams as input, and continuously correlates, aggregates, and filters them to produce outputs of interest to applications. In Borealis, the operators that process streams can be spread across multiple physical machines, called processing nodes (or simply nodes).

- **developer:** Brandeis University, Brown University, and MIT.
- **development state:** released (2006 - Summer 2008)
- **licence:** Free licence.
- **applicability:** Real-time monitoring applications, environmental monitoring, surveillance, tracking, plant maintenance, and telecommunications data management, etc.
- **accessibility:** Full program can be downloaded.
- **related tools:** Aurora

2.4.2.10 Stenford University - STREAM

Stream [22] is useful in applications such as network monitoring, telecommunications data management, clickstream monitoring, manufacturing, and sensor networks.

- **developer:** Stenford University.
- **development state:** The STREAM project has officially wound down. (released 2003 - 2006)
- **licence:** Free licence (BSD license).
- **accessibility:** Full program can be downloaded.
- **query language:** Continuous query language (CQL).

2.4.2.11 University of Marburg - PIPES

PIPES [15] is a flexible and extensible infrastructure providing fundamental building blocks to implement a data stream management system (DSMS). PIPES covers the functionality of the Continuous Query Language (CQL).

- **developer:** University of Marburg
- **development state:** First and last public release from 2004.
- **licence:** GNU Lesser General Public License.
- **query language:** Continuous Query Language (CQL).

2.4.3 Current research projects of industrial CEP vendors

This section differs from the previous ones. In this section, we summarize the ongoing projects of large vendors, which have not been released yet.

MicroSoft has announced the use of its own CEP engine, which will be developed approximately by the first half of 2010. This engine will be built on MicroSoft SQL Server 10, and codenamed Kilimanjaro. It will process the event flow and give CEP capabilities for their applications. Their aim is to fill the gap in their technology of business processing. There will be several tools available for enterprise applications, which will become more accurate with this approach. The engine is made by a SQL server team. At the moment, there is not so much information about it. No data or demos have been publicated yet. More information is expected in July 2009, but nothing new have been released until the closing of this paper.

IBM has made its new tool, named System S, which is similar to Complex Event Processing systems, but it is built to support higher data rates and a broader spectrum of input data modalities. It has a parallel pipeline architecture for simultaneous access, and it analyzes thousands of data streams from both inside and outside the corporate firewall including stock prices, traffic sensor readings, and weather reports. It also delivers the results directly to the business applications. Moreover, it can trigger applications to make automated responses to certain events or combinations of events, or it can notify the workers of potential events through associated visualization technology and dashboards. There are three available methods for end-users to operate on streaming data: SPADE (Stream Processing Application Declarative Engine), MARIO (Mashup Automation with Runtime Invocation and Orchestration), and Workflow Development Tool Environment.

2.5 Evaluation

In this section, we give the classification and evaluation of the event processing area. Table 1 categorizes the articles considering five aspects. Each column in Table 1 represents an aspect:

- *Requirements defined (Req.)*: papers are signed with an *X*, if they collect requirements for CEP algorithms, tools, or engines.
- *Method*: if the paper applies or defines a model, an algorithm, a method, or a new theoretical result, then it will be shortly described in this column.
- *Tool/Project*: if the paper uses or describes certain tools or contributes to a certain project, then it will be shortly described in this column.
- *Query language*: if the paper applies or defines an event query language, then it will be shortly described in this column.
- *Real experiments (Real exp.)*: papers presenting real world experiments (case studies) are marked with an *X* in this column.
- *Applicable in telecommunication (App. in telcom)*: papers are marked with *X* if their algorithm can be used or adopted to telecommunications.

In the following paragraphs, we discuss the findings summarized in Table 1.

Req.: Three of the articles contain the principles of effective RFID data management [73], real-time stream processing [85], and design patterns [75]. The physical principles are mentioned in the fourth paper [78], where the main aim is not to collect these principles but to describe a tool called WINS NG.

Method: The main aim of eight of these articles is to use or define a model or an algorithm. Two of the papers [81, 90] present the results about the probabilistic event model. The event engine must consider if the data stream is dirty or unreliable. Although this research area is in an early stage, it is very important, because the events are probabilistic in the real world. Two of the papers [89, 96] deal with complex event detection, especially RFID complex event detection. Both of them give an event detection algorithm, which have been implemented.

Tool/Project: Ten of the articles deal with a tool or a project. While one of them is about a project [79], the others are about a tool. They usually define an event engine to process events and four of them are completed to an own query language. In the article [81], the authors extend an existing tool, called TelegraphCQ. In another paper [35], a comparison is detailed between Business Process Execution Language (BPEL) and Business Process Modeling Notation (BPMN). Finally, there is a technical report [86] which presents an event-based business intelligence visualization tool, the Event Tunnel framework.

Query language: Six articles introduce query languages. In three of these articles the languages have the same names as the engines. These languages are not always named in the articles, the authors only referred to them as a reaction rule language or a language based on COMPOSITE and SNOOP algebra [81]. These 6 languages are a specialized languages for certain problems. In each article, a problem is introduced and the authors give a language, which can handle this special problem.

Real exp.: Finally, we determine the group of articles where the introduced systems are applied in real environments. One of the papers [46] only describes how easyCredit of the norisbank is faced with two related areas of CEP: BAM and BPEL-engine.

App. in telcom: Since in almost all of the (5 from 21) articles general event processing methods are presented, these methods can be used in telecommunication area as well. There are some articles, which are about a certain case study or a comparison of two languages, where the applicability in telecommunication area is meaningless.

The presented papers and tools are generic; they can be used in most industrial areas, including telecommunication as well. Therefore, we cannot distinguish them from the point of view of application areas, they should rather be evaluated on the basis of other indicators. These indicators basically aim at performance counters and successive rates. Evaluating the tools regarding these points of view should require their practical investigation and comparison, which are not the goals of this survey.

After all, there have been several attempts to evaluate the solutions and tools of complex event processing. One benchmark is proposed by the STAC Benchmark Council [84], however concrete information is missing, even if one of the biggest CEP providers, Aleri, is certified by STAC [32].

The Linear Road [49, 33] project provides a potential benchmark for complex event processing solutions. It has been applied by the Aurora and the STREAM CEP projects. This benchmark compares the performance characteristics of Stream Data Management Systems by simulating a variable tolling system.

Article (section,[cite])	*Req.*	*Method*	*Tool/Project*	*Query language*	*Real exp.*	App. in telco
StonebrakerThe8 (2.3.2.1,[85])	X					X
RizviComplex (2.3.1.2,[81])		PCEP	extended TelegraphCQ	COMPOSITE, SNOOP algebra	X	X
GustasReal-time (2.3.3.9,[77])			R-PMO server		X	X
AggarwalOn_Classification (2.3.3.3,[31])		micro-clustering				X
PottieWireless (2.3.2.2,[78])	X		WINS NG			X
BargaConsistent (2.3.3.6,[34])			CEDR	CEDR		X
LuckhamComplex (2.3.3.2,[64])			RAPIDE	RAPIDE	X	
WidderIdentification (2.3.1.7,[92])		discriminant analysis				X
PalmerSeven (2.3.2.3,[73])	X					X
PaschkeAHomogeneous (2.3.1.3,[74])				reaction rule language		X
WangBridging (2.3.3.7,[89])		RCEDA				X
SuntingerTheEvent (2.3.3.8,[86])			Event Tunnel			X
GreinerBusiness (2.3.3.1,[46])					X	
AmmonDomain (2.3.1.6,[79])		domain specific reference model	DoReMoPat			
WuHigh-Performance (2.3.3.10,[94])			SASE	SASE		
MihaeliDetecting (2.3.3.5,[66])				AMIT		X
WhiteWhatIsNext (2.3.3.4,[91])		temporal model				X
ZangComplex (2.3.1.5,[96])		Complex event detection	RTE-CEP		X	X
BarrosComplex (2.3.1.4,[35])			BPEL, BPMN			
WasserkrugComplex (2.3.1.1)		probabilistic event model				X
PaschkeDesign (2.3.2.4,[75])	X					X

Table 1. Issue of the articles.

Another benchmark proposal is developed in the BiCEP [48, 37] project at the University of Coimbra. This benchmark is now in the design stage, but it should be dealt with huge attention. The first outcome of the project is FINCoS [65], a framework that can be used to benchmark CEP systems. The members of the BiCEP project will present a tutorial [87] about the use cases of event processing, which could be really interesting. After all, the major measurement aspects of event processing based on the BiCEP proposal are sustainable throughput, response time, scalability, adaptivity, computation sharing, similarity search, and precision and recall. The BiCEP project will be finished in June, 2010.

NEXMark [50] is also a proposal about a benchmark for event processing. It extends the XMark benchmark to an online auction system, and gives hundreds of open auctions. The benchmark evaluates how fast and accurate the data is processed by an event processing engine.

2.6 Open research questions

In this section, we identify some main open research questions related to complex event processing. During examination of the articles, blogs, and tools, we met some research problems, which should be solved in the future.

How to standardize CEP? Different vendors use different types of events, terminology, customized event hierarchies, and different kinds of event processing components in their operation. Consequently, standardization is one of the most important aims of the CEP community. It could make the solutions more portable and allow easier development of third-party applications. Several attemtps exist for standardization. First, the paper of Mishra et. al. [53] was the result of the nearly two-year long research done by StreamBase Systems Inc., a leader in high-performance CEP, and Oracle Corporation, the world's largest enterprise software company. They conducted research on CEP techniques and advance convergence on standard language implementation issues. In their paper, StreamBase and Oracle proposed the integration of the Oracle CQL and StreamBase StreamSQL into a common language standard. The execution model of StreamBase is event-based, which is useful for deterministic event-by-event data processing where the outcome of business logic is determined by the temporal order of data. The time-based model of Oracle processes data in sets, which is an approach optimized for processing stored event data. The novel standards-based approach that combined both models resulted in an event processing approach that addressed both real-time and historical event processing in one common language. However, according to the chief architect

of StreamBase, Richard Tibbetts [29], standardization is a very slow process. It will take several years to have a complete and common CEP language standard. Many open parallel standards will be created and lots of work between groups of vendors will be needed to resolve their differences. An event processing technical society [44] is also making attempts for standardization, for example, they have presented a glossary for the terms used in the area of CEP.

Why SQL is the base of CEP languages in most cases? SQL is the most widely used language for relational databases. Although SQL is both an ANSI and an ISO standard, many CEP vendors support SQL with proprietary extensions for event processing. The implementations of SQL are inconsistent and usually incompatible between vendors. According to StreamBase, SQL is the only viable candidate for the language of standard event processing. Even StreamBase's own language, StreamSQL, is based on the standard database language, SQL. Similarly to StreamBase, the languages of most CEP products' are similar to SQL. These vendors added a lot of features to their own SQL-derived language in order to work with streams of data. Most of these languages are powerful and can be used to create complex systems. These SQL dialects can be found in products of Coral8, Esper, BEA WebLogic Event Server, StreamBase, and Aleri. Due to its request-response nature, SQL is unsuitable for low-latency event processing. Since SQL-based systems do not include the notion of time, it is difficult to code complex event patterns with SQL-based systems. Despite these facts, most of the CEP languages are SQL-based.

How the performance (e.g., latency) of searching for event patterns can be improved? Optimization is very important in event processing systems. The system resources are limited and clients need information in near-real-time or even in real-time. It is important to consider rule set optimizations to speed up rule execution without changing the externally observable behavior of the implemented application. We examined some articles [96, 85, 94, 77] and tools [2, 4, 5, 6], which show highly optimized implementation or operator. To work with CEP in real-time, further real-time optimization techniques are required.

How to deal with uncertain data? We usually assume that data streams are precise. However, the data reported by real-world sources are often unreliable. What happens if our data streams or generated virtual events are imprecise or fuzzy? In these cases, the actual happening of an event is not certain, e.g., 90% chance for a sunny day. We should not forget about these cases either. Two of our collected papers [81, 90] give a preliminary solution for these problems. One of them introduces Probabilistic Complex Event Processing to handle these data streams, the other gives two algorithms to approximate event probabilities. However, this problem is not globally handled by CEP solutions, now they assume precise and certain data.

How to deal with incomplete data? Rizvi et. al. [81] deal with incomplete data as well. If the incoming events have missing attributes or properties, the CEP engine should handle them as well as the other normal cases. However, similarly to uncertain data, most CEP solutions do not handle this problem either.

How to automate the manual phases of CEP? There are several steps in the process of event processing which require human actions. For example, the definition of the rules and patterns with some kind of event query language, performing reactions based on the alerts, and so on. For these problems, certain solutions are published but not widely used. For example, ECA (Event Condition Action) [74] for automatic reaction in the case of a given event which is accepted by a given condition. An other solution is to automatically determine the rules or patterns for the CEP engine [92]. This could be done effectively with the techniques of Predictive Analytics as it is mentioned by Wasser et. al. [90].

3 Detailed assessment of the technologies of Predictive Analytics

In this section, we assess the achievements and open questions in the field of Predictive Analytics along similar guidelines as used for the CEP field. Our main focus area is telecommunication. We survey the relevant articles using the following method. In the first step, we used several keywords (prediction, predictive analysis, forecasting, learning, automatic detection, data mining, telecommunication, network, log processing, trace processing) in the search engine Google Scholar to find the baseline for further research. One of our selection criteria was whether the referenced papers had been published by IEEE, ACM, Springer, Elsevier, Wiley, or Addison. In the next step, we extend the searching fields for references and related surveys. The main aspects, used in the evaluation of results, are usability in telecommunication, the probability of implementation, and the performance and accuracy rate.

3.1 Background and overview

In the case of Predictive Analytics Methods based on Complex Event Data, one can make predictions about some attributes of the monitored system based on the previously monitored events. Such a prediction process can be divided into four steps: (1) collect and preprocess raw data; (2) transform preprocessed data into a form that can be easily handled by the (selected) machine learning method; (3) create the learning model (training) using the transformed data; (4) report predictions to the user using the previously created learning model. Future events will be predictable by using recent data based on the learning model trained for the previously monitored events.

While collecting the papers, we considered the publications from the areas of distributed or multi-agent networks, web applications, and the financial sector besides telecommunication. The papers were evaluated on the basis of well-defined viewpoints, e.g., whether the presented method was applied in a live environment, what kind of learning method(s) were used, or whether the method was applicable in telecommunication.

As mentioned previously, we examined more application areas. Many promising, usable solutions were found to discover the errors of different layers in telecommunication, e.g., the Pinpoint in [60]. This tool materializes a fault detection model which works on the components of complex web applications. This online fault detection tool detects 89% - 100% of the major faults. Furthermore, we found an article [67] which achieved results on the transmission element of a telecommunication system. This article also introduces the use of the neural network for video traffic prediction. Another article [61] deals with the receiver element of a telecommunication system. This paper describes a theoretical method which can predict file change in mobile computers and is capable to predict the downloaded online contents.

The survey gives detailed descriptions about the information collected from non-telecommunication articles. We examined whether their methods in these articles can be used in the telecommunication area or not. Based on our evaluation models our opinion is that all of the methods used in web application or distributed or multi-agent networks can be used in telecommunication as well. The software error detection and/or prediction methods used on web applications can predict/detect the faults of the transmitter system. The quality of the communication medium (bandwidth, throughput, latency, etc.) can be predicted by the methods used in the network.

During our work, we recorded the unresolved problems and open questions described in the observed papers. One of the problem group deals with the effect of integrating the forecasting method into a live system, where the following questions may arise:

- How will the original system be affected after the integration of the forecasting system?
- When should the learning model be optimally refreshed?
- Which frequency is the best for pattern sampling?

Further questions can be asked about the implementation of the forecasting methods.

- How scalable is a forecasting method?
- How can the methods be compared in practice?
- Can the predictors be effectively determined?
- What are the names of the certain steps of the forecasting process?

3.2 Articles

In this section, we present papers from the predictive analytics research area. We define evaluation aspects to raise important details for better visibility. First of all, the term **basic information** indicates the author(s), the year when the paper was published, and the number of references. The term **realtime** tells us whether the method can be used in real time or not. The term **model refreshment** indicates whether the model can be refreshed or not. The term **whether applied on live environment** shows the kinds of source the method was tested on. The term **preprocessing method** shows the second step of predictive analytics, where the raw data is filtered or transformed. The term **learning method** indicates the third step of predictive analytics, where the learning method learns the earlier saved data and gives a model. If the paper defines a **tool**, it is mentioned as an aspect. We highlight the kinds of **metrics and predictors** as well. We examined in which area the given method can be used. The **applicability** aspect defines this area. If this area is not telecommunication, we further examine the paper **whether it is applicable in telecommunication**. We try to evaluate the **difficulty to implement**. We summarize the **goals of the different papers**. Finally, a short **evaluation** is given about the paper. We do not include the aspects if they are not mentioned in the paper.

3.2.1 Fault prediction and detection

3.2.1.1 Forecasting Field Defect Rates Using a Combined Time-based and Metric–based Approach a Case Study of OpenBSD

This paper presents an empirical case study of ten releases of OpenBSD. The findings in this paper are steps towards managing the risks associated with field defects. In this paper it is identify the SRGM that produces the most accurate forecasts and determine subjectively the preferred metrics-based prediction method and set of predictors [62].

- **Basic information:** Written by Paul Luo Li, Jim Herbsleb and Mary Shaw. Published in 2005 by IEEE, there are 7 references for this paper.
- **realtime:** No.
- **model refreshment:** Yes (at every release).
- **whether applied on live environment:** OpenBSD operation system.
- **preprocessing method:** Statistic.
- **learning method:** Decision Trees (Exponential model).
- **tools:** Theil forecasting statistic.
- **metrics, predictors:** 145 predictors (product metrics, development metrics, deployment and usage metrics, and software and hardware configurations metrics).
- **applicability:** Software defect prediction.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** Difficult.
- **goal of the method:** Software defect prediction.
- **evaluation:** Usable to detection of failures before software release.

3.2.1.2 Failure detection and localization in component based systems by online tracking

The authors presented their new approach to detect and localize service failures in component based systems. They tracked high dimensional data about component interactions to detect failure dynamically and online. They divided the observed data into two subspaces: signal and noise subspaces, and extracted two statistics that reflected the data distribution in subspaces for tracking. The subspaces are updated for every new observation, the data distribution is learnt by SDEM. This decomposition method is not a novel approach but they employed online distribution learning algorithms instead of normal distribution [38].

- **Basic information:** Written by Haifeng Chen, Guofei Jiang, Cristian Ungureanu and Kenji Yoshihira. Published in 2006 by ACM, there are 8 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Yes.

- **whether applied on live environment:** No, real system (PetStore application demo) with a list of injected faults.
- **preprocessing method:** They divided the observed data into two subspaces: signal and noise subspaces with Hotelling T^2 score and SPE(squared prediction error).
- **learning method:** SDEM (sequentially discounting expectation maximization) with EWMA filter (exponentially weighted moving average).
- **tools:** PinPoint (collect observations), SmartSifter (SDEM).
- **metrics, predictors:** Activity of component interactions.
- **applicability:** Component based systems.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** The mathematical base of thess algorithms are given, they can be implementable.
- **goal of the method:** They give algorithms to reduce the dimensionality of monitoring data.
- **evaluation:** It is important to reduce the recovery duration.

3.2.1.3 Path-based Failure and Evolution Management

This is a new approach to managing failures and evolution using paths. The authors provide automated statistical analysis of single and multiple paths. They provide complex detection and diagnosis operations without any knowledge of the system components. One of the testing environments of the method is from telecommunication. Its main feature is not prediction, but the last step of failure management is feedback (predictive impact analysis – predict potential outcomes in the system) [39].

- **Basic information:** Written by Mike Y. Chen, Anthony Accardi, Emre Kiciman, Jim Lloyd, Dave Patterson, Armando Fox and Eric Brewer. Published in 2004 by USENIX, there are 99 references for this paper.
- **realtime:** No.
- **model refreshment:** No.
- **whether applied on live environment:** Yes. *Not live:* Pinpoint (research prototype – an analysis framework for an open-source, 3-tier Java 2 Enterprise Edition (J2EE) application platform, JBoss) *Live:* two large, commercial systems that service millions of requests per day: ObsLogs (part of a path-based infrastructure at Tellme Networks, an enterprise voice application network), SuperCal (the logging infrastructure at eBay, an online auction site).
- **preprocessing method:** Aggregation (SQL operations), clustering, classification.
- **learning method:** Trained PCFG, decision tree, association rules.
- **tools:** Tracers, Aggregator, Repository, Analysis Engines, Query Engine.
- **metrics, predictors:** Path (collection of observations: timestamp, component name, host name, version number).
- **applicability:** Large, complex distributed system (e.g., telecommunication).
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** Difficult.
- **goal of the method:** Detect and diagnose failures (correctness and performance problems), and understand the evolution of a system.
- **evaluation:** It has maintainable, extensible, reusable architecture.

3.2.1.4 Capturing, indexing, clustering, and retrieving system history

The authors presented a method for automatically extracting an indexable signature from a running system to identify when an observed system state is similar to a previously-observed one. They created an index from the values of raw system metrics. After clustering, the problems they identified were similar to the ones that was occurred earlier. [42].

- **Basic information:** Written by Ira Cohen, Steve Zhang, Moises Goldszmidt, Julie Symons, Terence Kelly and Armando Fox. Published in 2005 by ACM, there are 65 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Online.
- **whether applied on live environment:** Yes. *Not live:* realistic testbed with injected performance faults. *Live:* real globally-distributed system.

- **preprocessing method:** Pattern classification: signature clustering (k-means, k-medians).
- **learning method:** TAN (Tree-Augmented Bayesian Network).
- **tools:** HP OpenView Operations Agent, HP OpenView Performance Agent, httperf.
- **metrics, predictors:** Application-level performance data, system-level resource utilization metrics (e.g., CPU load, disk I/O rates).
- **applicability:** Distributed application.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** It seems to be not so difficult to implement.
- **goal of the method:** Identify when an observed system state is similar to a previously-observed state and evolve to the failure.
- **evaluation:** Not prediction.

3.2.1.5 Tracking Probabilistic Correlation of Monitoring Data for Fault Detection in Complex Systems

The authors modified their earlier paper [56] with a new variant of the EM algorithm to learn the parameters of the Gaussian mixtures, further estimate a boundary of normal data distribution, and detect faults in real time. In fact, they used this boundary to distinguish the outliers from normal data points, to detect the outliers with extremely low probability density. They proposed a procedure to automatically search and validate probabilistic relationships between each pair of the monitoring data [47].

- **Basic information:** Written by Zhen Guo, Guofei Jiang, Haifeng Chen and Kenji Yoshihira. Published in 2006 by IEEE, there are 10 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Yes.
- **whether applied on live environment:** No, real system (PetStore e commerce application demo) with a list of injected faults.
- **preprocessing method:** Used Gauss mixture models to characterize probabilistic correlation between flow-intensities measured at multiple points.
- **learning method:** A novel algorithm derived from Expectation-Maximization (EM) algorithm that is able to tune the number of clusters dynamically.
- **tools:** New framework with other tools (e.g., JMX).
- **metrics, predictors:** Access logs of web request, numbers of threads and EJBs, network traffic statistics, OS audit data (CPU, memory usage), SQL queries.
- **applicability:** Three-tier systems.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** The mathematical base of this algorithm is given, it is implementable.
- **goal of the method:** Fault detection.
- **evaluation:** It does not predict, but it is usable in complex systems.

3.2.1.6 Discovering likely invariants of distributed transaction systems for autonomic system management

The authors completed their earlier paper [56] with an extracting algorithm which automatically searches and extracts the invariants from a large number of flow intensity measurements from dynamic systems. Invariants are widely used in transaction systems: fault detection and diagnosis,performance debugging and capacity planning, system evolution, intrusion detection in security, etc. [55].

- **Basic information:** Written by Guofei Jiang, Haifeng Chen and Kenji Yoshihira. Published in 2006 by Kluwer Academic Publisher, there is only 1 reference for this paper.
- **realtime:** Yes.
- **model refreshment:** No.

- **whether applied on live environment:** No, real system (Pet Store e-commerce application demo) with a list of injected faults.
- **preprocessing method:** Finding invariants where fitness score greater than 50 (it is strong linear correlation).
- **learning method:** ARX (AutoRegressive models with eXogenous inputs).
- **tools:** New framework with other tools (e.g., JMX).
- **metrics, predictors:** Monitoring and calculating the flow intensity at multiple checkpoints inside the system (CPU, Disk, OS, Network, JVM, JBoss, Apache, MySQL metrics).
- **applicability:** Distributed transaction system.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** This method is easily implementable, the used algorithm is given in pseudo code form.
- **goal of the method:** Extract invariants.
- **evaluation:** This method can not predict, but it has usable steps in complex systems for prediction.

3.2.1.7 Modeling and Tracking of Transaction Flow Dynamics for Fault Detection in Complex Systems

A new concept named flow intensity is introduced to measure the intensity with which the internal monitoring data reacts to the volume of user requests in distributed transaction systems. The authors give a new approach to automatically model and search relationships between the flow intensities which are measured at various points of the system [56].

- **Basic information:** Written by Guofei Jiang, Haifeng Chen and Kenji Yoshihira. Published in 2006 by IEEE, there are 8 references for this paper.
- **realtime:** Yes.
- **model refreshment:** No.
- **whether applied on live environment:** No, real system (Pet Store e-commerce application demo) with a list of injected faults.
- **preprocessing method:** Finding invariants where fitness score greater than 50 (it is strong linear correlation).
- **learning method:** ARX (AutoRegressive models with eXogenous inputs).
- **tools:** New framework with other tools (e.g., JMX).
- **metrics, predictors:** Monitoring and calculating the flow intensity at multiple checkpoints inside the system (CPU, Disk, OS, Network, JVM, JBoss, Apache, MySQL metrics).
- **applicability:** Distributed transaction system.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** This method is easily implementable.
- **goal of the method:** Fault detection.
- **evaluation:** This method can not predict, but it has usable steps for prediction, invariant searching is improved in some other articles [57, 55, 47].

3.2.1.8 Efficient and Scalable Algorithms for Inferring Likely Invariants in Distributed Systems

The computational complexity of the previous invariant (constant relationship between flow intensities) search algorithm [56] is high, so it cannot scale well with a lots of measurements in large systems. The authors give two efficient but approximate algorithms (SmartMesh, SimpleTree) for inferring invariants in large-scale systems. The computational complexity of the new randomized algorithms is significantly decreased, and the experimental results from a real system show the accuracy and efficiency of their new algorithms [57].

- **Basic information:** Written by Guofei Jiang, Haifeng Chen and Kenji Yoshihira. Published in 2007 by IEEE, there are 2 references for this paper.
- **realtime:** Yes.
- **model refreshment:** No.
- **whether applied on live environment:** No, real system (Pet Store e-commerce application demo) with a list of injected faults.

- **preprocessing method:** Finding invariants where fitness score greater than 50 (it is a strong linear correlation), Smart-Mesh algorithm, SimpleTree algorithm.
- **learning method:** ARX (AutoRegressive models with eXogenous inputs).
- **tools:** New framework with other tools (e.g., JMX).
- **metrics, predictors:** Monitoring and calculating the flow intensity at multiple checkpoints inside the system (CPU, Disk, OS, Network, JVM, JBoss, Apache, MySQL metrics).
- **applicability:** Distributed transaction system.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** This method is easily implementable, the used algorithms are given in pseudo code form.
- **goal of the method:** Fault detection.
- **evaluation:** This method can not predict, but it has usable steps in large-scale systems for prediction.

3.2.1.9 Diagnosis of Recurrent Faults by Invariant Analysis in Enterprise Software

In real-world systems, similar faults tend to reoccur. It is therefore possible to reach better results. The authors presented methods to determine the most important metrics to track for efficient failure detection, and the correlations that add the most to diagnosis accuracy. They applied neural networks to identify the relevant correlations to reduce the level of interaction. Their approach requires minimal monitoring and no manual configuration of thresholds [58].

- **Basic information:** Written by Michael Jiang, Mohammad A. Munawar, Thomas Reidemeister and Paul A.S. Ward. This article has not published yet.
- **realtime:** No.
- **model refreshment:** No.
- **whether applied on live environment:** No, real system with a list of injected faults.
- **preprocessing method:** $R^2 > 0.9$.
- **learning method:** Neural network, Cook's distance (0.8).
- **tools:** WebSphere 6 Application Server, Weka 3, DB2 UDB 8.2 database server.
- **metrics, predictors:** Data reflects performance, state, and errors of components of the application and the application server.
- **applicability:** Enterprise-software system.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** The pseudo code of this algorithm is given, it is implementable.
- **goal of the method:** Specific fault detection, generic fault detection, generic fault diagnosis.
- **evaluation:** Effective (low false positive rate) but not for real programs.

3.2.1.10 Detecting Application-Level Failures in Component-based Internet Services

This paper presents Pinpoint, a methodology for automating fault detection in Internet services by observing the low-level, internal structural behaviors of the service. It models the majority behavior of the system correctly and detects the anomalies in these behaviors as the possible symptoms of the failures [60].

- **Basic information:** Written by Emre Kiciman and Armando Fox. Published in 2005 by IEEE, there are 61 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Yes.
- **whether applied on live environment:** Application tested on RUBiS system.
- **preprocessing method:** No.
- **learning method:** Decision Trees, ID3 algorithm.
- **tools:** Java middleware, L7.
- **metrics, predictors:** 2 (recall, precision).

- **applicability:** Telecommunication.
- **whether applicable in telecommunication?:** Yes.
- **goal of the method:** Detect and diagnose failures.
- **evaluation:** Usable to detection of failures.

3.2.1.11 Adaptive Monitoring in Enterprise Software Systems

The authors have a new approach to monitoring at a minimal level in order to determine system health and automatically adaptively increase the monitoring level if a problem is suspected. The relationships between the monitored data are used to determine normal operation and – in the event of anomalies – identify areas that need more monitoring. In this paper the authors urge the need for adaptive monitoring and describe an approach to drive this adaptation [68].

- **Basic information:** Written by Mohammad Ahmad Munawar and Paul A. S. Ward. Published in 2006, there are 6 references for this paper.
- **realtime:** No.
- **model refreshment:** No.
- **whether applied on live environment:** No, real system with a list of injected faults.
- **preprocessing method:** Correlation coefficient.
- **learning method:** Linear regression.
- **tools:** WebSphere 5 Application Server, DB2 UDB 8.1 database server.
- **metrics, predictors:** Interaction between components.
- **applicability:** Enterprise software system.
- **whether applicable in telecommunication?:** Yes.
- **goal of the method:** Find activity, performance, state and errors of components and services in the server.
- **evaluation:** Self-adaptation of monitoring can reduce time spending with detection of problems.

3.2.1.12 A comparative study of pairwise regression techniques for problem determination

The authors compared the use of simple linear regression (SLR) to some of its more complex variants including autoregressive regression and locally weighted regression. They considered component coverage, model robustness, accuracy of diagnosis, and computation cost. ARX was found to be the most accurate one, but unfortunately this regression type is the second most expensive and the least robust one [69].

- **Basic information:** Written by Munawar, Mohammad A. and Ward, Paul A. S. Published in 2007 by ACM, there are 4 references for this paper.
- **realtime:** No.
- **model refreshment:** No.
- **whether applied on live environment:** No, real system with a list of injected faults.
- **preprocessing method:** $R^2 > 0.6$.
- **learning method:** SLR (simple linear regression), SLRT (SLR using Transformed Data), SLRS (SLR using Smoothed Data), ARX (AutoRegressive models with eXogenous inputs), LWR (Local Weighted Regression).
- **tools:** WebSphere 6 Application Server, Weka 3, DB2 UDB 8.2 database server.
- **metrics, predictors:** Components' metrics.
- **applicability:** Component based system.
- **whether applicable in telecommunication?** Yes.
- **difficulty to implement:** This regression methods are easily implementable.
- **goal of the method:** Find performance, state and errors of components of the application and the application server.
- **evaluation:** Significant, comparison of different regression methods, parameterizable.

3.2.1.13 Predicting the Location and Number of Faults in Large Software Systems

In this paper the statistical model assigns a predicted fault count to each file of a new software release based on the structure of the file and its fault and change history in previous releases. The higher the predicted fault counts, the more important it is to test the file early and carefully [71].

- **Basic information:** Written by Thomas J. Ostrand, Elaine J. Weyuker and Robert M. Bell. Published in 2005 by IEEE, there are 9 references for this paper.
- **realtime:** No.
- **model refreshment:** Yes (at every release).
- **whether applied on live environment:** Followed the inventory system for five additional releases, and the overall prediction results are presented for 17 successive releases covering four years of field usage.
- **preprocessing method:** Square root, logarithm.
- **learning method:** Negative Binomial Regression.
- **tools:** SAS/STAT.
- **metrics, predictors:** The log of the number of lines of code, file age, and the square root of the number of prior faults.
- **applicability:** Software defect detection.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** Difficult.
- **goal of the method:** Software defect detection.
- **evaluation:** Usable to detection of failures before software release.

3.2.1.14 Defect Prediction using Combined Product and Project Metrics – A Case Study from the Open Source "Apache" MyFaces Project Family

In a closed source software development, the prediction of defects between releases can provide benefits such as guiding the testing of the next release. This paper presents a prediction model that uses combination of project and product metrics [88].

- **Basic information:** Written by Dindin Wahyudin, Alexander Schatten, Dietmar Winkler, A Min Tjoa, Stefan Biffl. This article has not published yet.
- **realtime:** No.
- **model refreshment:** Yes (at every release).
- **whether applied on live environment:** MyFaces (Core and Tobago, 6-6 release).
- **preprocessing method:** Pearson bivariate correlation model.
- **learning method:** Linear regressions.
- **tools:** SPSS, Metrics plug-in for Eclipse, Check style plug-in for Eclipse.
- **metrics, predictors:** Project and Product Metrics, normalized number of defects.
- **applicability:** Software defect prediction.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** Difficult.
- **goal of the method:** Software defect prediction.
- **evaluation:** They reached good results by correlation between metrics.

3.2.2 Performance prediction

3.2.2.1 Correlating instrumentation data to system states: a building block for automated diagnosis and control

Tree-Augmented Bayesian Networks (TANs) are used to find correlation between certain combinations of system-level metrics and threshold values and high-level performance states (SLOs – Service Level Objectives) in a three-tier web service. This paper shows that TANs are powerful enough to recognize the performance behavior of a representative three-tier web service. It can forecast SLO violations with stable workloads as well [41].

- **Basic information:** Written by Ira Cohen, Moises Goldszmidt, Terence Kelly, Julie Symons and Jeffrey S. Chase. Published in 2004 by USENIX Association, there are 164 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Not online, but it can be.
- **whether applied on live environment:** No, only real system (modified Petstore e-commerce application system).
- **preprocessing method:** Pattern classification.
- **learning method:** TAN (Tree-Augmented Bayesian Network).
- **tools:** HP OpenView (collect a set of system metrics), httperf (load generator).
- **metrics, predictors:** 124 system level metrics: CPU, Memory, Network data, application-level performance measurements (e.g., response time, throughput).
- **applicability:** Three-tier web service.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** It is easily implementable.
- **goal of the method:** Performance diagnosis and forecasting.
- **evaluation:** TANs are practical: they are efficient to represent, evaluate, interpretable, modifiable, simple, compact.

3.2.2.2 Load Forecasting

This paper discusses various approaches to load forecasting. It describes different methods like statistical techniques and artificial intelligence algorithms for different time intervals. The importance of measures, which highly affects forecast accuracy is also mentioned [45].

- **Basic information:** Written by Eugene A. Feinberg and Dora Genethliou. Published in 2005 by Springer, there are 10 references for this paper.
- **model refreshment:** Various.
- **whether applied on live environment:** Show frequently used methods, but no exact data.
- **preprocessing method:** Depends on used methods.
- **learning method:** Statistical techniques and artificial intelligence algorithms.
- **metrics, predictors:** Raw data(date, temperature, geographic and social data) and composited weather variables (THI (temperature-humidity index), WCI (wind chill index)).
- **applicability:** Electric load forecasting.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** These techniques are easily implementable.
- **goal of the method:** Load forecasting.
- **evaluation:** These techniques has various usage area.

3.2.2.3 An Analysis of Trace Data for Predictive File Caching in Mobile Computing

The data gathered and the analysis performed in this study strongly indicate that predictive file caching for mobile computing is a feasible approach. It looked at two measures that had special applications to mobility: write conflicts and attention shifts. A prototype system of this sort was developed under CMU's Coda system and proved successful. However, it was inconvenient for the user, and it was tested only in one application environment [61].

- **Basic information:** Written by Geoffrey H. Kuenning, Gerald J. Popek and Peter L. Reiher. Published in 1994, there are 64 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Yes.
- **whether applied on live environment:** 47 users running business-oriented applications, recording $4,637,924$ accesses during 1563 hours.
- **preprocessing method:** No.

- **learning method:** No.
- **tools:** No.
- **metrics, predictors:** 6 data management metrics (read, write, . . .).
- **applicability:** Mobile computers.
- **whether applicable in telecommunication?:** Yes.
- **goal of the method:** Predicting data cache.
- **evaluation:** This concept is feasible.

3.2.2.4 Data Mining in Social Networks

This paper presents a new datamining approach which works with the relational links of huge relational data bases. Given a set of data, like the IMDB movie database, various values can be forecasted with this tool. The links between the objects are presented with graphs in relational probability trees. This method is currently in an early phase, but seems it to be useful [54].

- **Basic information:** Written by David Jensen and Jennifer Neville. Published in 2003 by National Academies Press, there are 29 references for this paper.
- **model refreshment:** Once per run.
- **whether applied on live environment:** Yes, used on IMDB database.
- **preprocessing method:** Creating graphs of object links.
- **learning method:** RPT (relational probability trees).
- **tools:** PROXIMITY framework, QGraph.
- **metrics, predictors:** Relational links between objects.
- **applicability:** Social networks, relational databases.
- **whether applicable in telecommunication?:** Probably yes.
- **difficulty to implement:** At this time, the technology is in test phase.
- **goal of the method:** Predict various data using relational data.
- **evaluation:** This method probably usable for marketing, and other problems. But at this time hard to tell, how hard to implement to a real problem with good accuracy.

3.2.2.5 Video Traffic Prediction Using Neural Networks

The goal of this paper is to predict the video time series for an efficient bandwidth allocation of the video signal using neural networks. The application of the presented methods is in traffic and congestion procedures of communication networks. The authors tried more neural network models, backpropagation through time (BPTT), radial basis functions (RBF) and multilayer perceptron (MLP). The best results of prediction for test set were achieved by using BPTT network with configuration 3-20-1 [67].

- **Basic information:** Written by Milos Oravec, Miroslav Petrás and Filip Pilka. This article has not published yet.
- **realtime:** Yes.
- **model refreshment:** Yes.
- **preprocessing method:** No.
- **learning method:** Neural network.
- **tools:** ANFIS.
- **metrics, predictors:** Normalized Length Of Frame.
- **applicability:** Video traffic prediction.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** The implementation of neural network is moderately difficult.
- **goal of the method:** Normalized Length Of Frame.
- **evaluation:** This method is usable to video traffic prediction.

3.2.2.6 Active Sampling Approaches in Systems Management Applications

A novel approach is presented: active sampling in distributed computer systems, that includes an active probing approach for problem diagnosis and active learning for end-to-end performance prediction and resource allocation in distributed systems and networks. The prediction of the best server for the given client is based on the history of interactions between servers and clients. With the help of this method, the number of samples can be reduced by 60-75% [80].

- **Basic information:** Written by Irina Rish. White paper about performance prediction and resource allocation in distributed systems and networks.
- **realtime:** Yes.
- **model refreshment:** Yes.
- **whether applied on live environment:** Yes: IBM's Download Grid (100-100 Server/Client), PlanetLab.
- **preprocessing method:** MMMF (max-margin matrix factorization), median threshold to convert the bandwidth data to binary value.
- **learning method:** Active learning heuristic (minimum margin sample).
- **tools:** Active Probing algorithm.
- **metrics, predictors:** Probe (test transactions: ping, traceroute, webpage-access, database query, an e-commerce transaction, etc.).
- **applicability:** Systems Management Applications.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** It is easily implementable.
- **goal of the method:** Predicting the quality of service which the server could provide: accurate prediction of end-to-end performance and resource allocation(bandwidth, latency), reduce the number of samples.
- **evaluation:** Significant method, active sampling is more accurate than random sampling.

3.2.2.7 A Scalable Multi-Agent Architecture for Remote Failure Detection in Web-Sites

The authors presented GOA-Net which detects application-level remote failures. Forecast Aggregation Nodes (FANs) are responsible for forecasting performance degradations in a given number of web-sites in the near future. FAN issues a forecasting report, GOA (Global Observation Agents) issues an error report to MA (Master Agent), which finally forwards it to the system administrator. The authors give a comparative table about the main features of four widely used, commercial remote monitoring solutions and Goa-Net. Their tool seems to have almost every feature which have the others [83].

- **Basic information:** Written by Décio Sousa, Nuno Rodrigues, Luis Silva and Artur Andrzejak. This technical report was published in 2007.
- **realtime:** Supported.
- **model refreshment:** No.
- **whether applied on live environment:** It has not implemented yet.
- **preprocessing method:** Time series analysis, floating average.
- **tools:** GOA-Net.
- **metrics, predictors:** Performance metrics (DNS res Time, TCP connect, Whole Page, Vieo QoS), Content Checking (static and dynamic).
- **applicability:** Web-sites.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** Difficult.
- **goal of the method:** Performance degradation forecasting, application level failures and operator errors, service outages.
- **evaluation:** Its advantages: high scalability, proactive forecasting, internet video QoS monitoring.

This tool has three level of accuracy: low, medium, high.

3.2.2.8 Forecasting network performance to support dynamic scheduling using the network weather service

The author focuses on the dynamic prediction of network performance with the support of parallel application scheduling. A prototype of Network Weather Service (NWS) was developed to forecast future latency, bandwidth and available CPU percentage for each monitored machine. Three kinds of forecasting methods are implemented: mean-based, median-based and autoregressive method. NWS automatically identifies the best prediction method for the resource [93].

- **Basic information:** Written by Rich Wolski. Published in 1997 by IEEE, there are 213 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Choose between 3 different models.
- **whether applied on live environment:** Yes, 5 hosts located far from each other in USA.
- **learning method:** Mean-based method (running average, sliding window average, uses only the last measurement, adaptive average, stochastic gradient), median-based methods (median, adaptive median, trimmed mean), autoregressive method (autoregressive model).
- **tools:** Network Weather Service.
- **metrics, predictors:** Latency, CPU percentage, throughput.
- **applicability:** Network.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** This method is not so difficult to implement.
- **goal of the method:** Predict network performance (TCP/IP end-to-end throughput, latency), system resource.
- **evaluation:** Significant: extensible, modifiable.

3.2.2.9 Ensembles of Models for Automated Diagnosis of System Performance Problems

The authors extended their earlier approach [41] with working under changing (real) workloads instead of stable workloads. This new method is not only inexpensive but efficient enough to be used in real time. They worked with an ensemble of models to identify the likely causes of performance problems. They continuously induced new models and enhanced the ensemble if a new model was captured [97].

- **Basic information:** Written by Steve Zhang, Ira Cohen, Julie Symons and Armando Fox. Published in 2005 by IEEE, there are 37 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Yes.
- **whether applied on live environment:** No, real system (PetStore e-commerce application demo).
- **preprocessing method:** Pattern classification.
- **learning method:** TAN (Tree-Augmented Bayesian Network).
- **tools:** HP OpenView (collect a set of system metrics), httperf (load generator).
- **metrics, predictors:** Low level system metrics from each tier (e.g., CPU load, memory usage) and application level metrics (e.g., response time and request throughput).
- **applicability:** Three-tier web service.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** Difficult.
- **goal of the method:** "Our goal is to understand the connection between high-level actionable properties such as SLO's, which drive these adaptation efforts, and the low-level properties that correspond to allocatable resources, decreasing the granularity of resource allocation decisions." [97]
- **evaluation:** Inexpensive enough to work with it in real time.

3.2.3 Methods and experiments

3.2.3.1 Simon Fraser University Database and Data Mining lab

This page is the Database and Data Mining Lab of Simon Fraser University, Canada. There are some related tools, downloadable demos and descriptions about those techniques. Here are the most interesting tools [28]:

FIHC Frequent Itemset-based Hierarchical Clustering (FIHC) is a program that constructs a document cluster hierarchy from a set of unlabeled documents based on frequent item sets. The output of FIHC is an XML file which allows the user to visualize the hierarchy, or serves as input data for further processing.

ADT ADT is the abbreviation for "Association-based Decision Tree". ADT is a program that constructs classifiers based on association rules. Given a data set, ADT will generate association rules first, if it is necessary, then build a decision tree in which each node is a rule. The over fitting rules will be pruned based on the measurement of estimated errors. Finally, ADT will test the accuracy of the decision tree on testing cases if required so. To classify a case, the most confident matching rule in the ADT is used. ADT can handle both table and transactional data. If the data set contains numeric values, ADT will automatically discretize them first.

3.2.3.2 QMON: QoS- and Utility-Aware Monitoring in Enterprise Systems

QMON is an online monitoring system. This program provides the abstraction of "Monitoring Channels" as a means to implement differential quality of service in monitoring [30].

- **Basic information:** Written by Sandip Agarwala, Yuan Chen, Dejan Milojicic and Karsten Schwan. Published in 2006 by IEEE, there are 11 references for this paper.
- **realtime:** Yes.
- **model refreshment:** No.
- **whether applied on live environment:** No.
- **tools:** Dproc, Application Response Measurement (ARM).
- **metrics, predictors:** CPU, memory, network, block I/O.
- **applicability:** Monitoring network systems.
- **whether applicable in telecommunication?:** Yes, online monitoring.
- **evaluation:** Does not predict and can not intervene into communication of monitored system.

3.2.3.3 Regrets Only! Online Stochastic Optimization under Time Constraints

This paper considers online stochastic optimization problems where time constraints severely limit the number of offline optimizations, which can be performed at decision time and/or in between decisions. It proposes a novel approach called Regret algorithm, which combines the salient features of the expectation and consensus algorithms. It shows experimental results in Packet Scheduling and Vehicle Routing problems [36].

- **Basic information:** Written by Russell Bent and Pascal Van Hentenryck. Published in 2004, there are 29 references for this paper.
- **realtime:** Yes.
- **whether applied on live environment:** No, only test results are available.
- **learning method:** Regret algorithm.
- **tools:** Online stochastic optimization framework.
- **metrics, predictors:** Requests.
- **applicability:** Packet scheduling, vehicle routing, elevator dispatching.
- **goal of the method:** Choosing optimal decisions under limited time interval.
- **evaluation:** Effective.

3.2.3.4 Regression Cubes with Lossless Compression and Aggregation

Data cubes in OLAP (online analytical processing) technologies often become high and unmanageable, if the raw data set is complex, infinite or continually changing behavior. A new technology is introduced which reduces the dimensionality of raw data using lossless compression. After this, the new high level measures are usable for data analysis without accessing raw data [40].

- **Basic information:** Written by Yixin Chen, Guozhu Dong, Jiawei Han, Jian Pei, Benjamin W. Wah and Jianyong Wang. Published in 2006 by IEEE, there are 4 references for this paper.
- **model refreshment:** Yes.
- **whether applied on live environment:** No, only test data are available.
- **preprocessing method:** NCR (nonlinear compression representation).
- **tools:** OLAP engine.
- **metrics, predictors:** Various measures, like usage, user, etc.
- **applicability:** Online analytical methods.
- **whether applicable in telecommunication?:** Probably yes.
- **goal of the method:** Lossless compression for saving time and resource.
- **evaluation:** It seems to be a usable and acceptable technique.

3.2.3.5 Artificial intelligence in short term electric load forecasting

This paper provides an overview of AI technologies, expert systems, artificial neural networks, generic algorithms, and their current use in the field of short term electric load forecasting. It also gives some historical descriptions about the changes of technologies with time [59].

- **Basic information:** Written by K. Metaxiotis, A. Kagiannas, D. Askounis and J. Psarras. Published in 2002 by Elsevier, there are 29 references for this paper.
- **realtime:** Yes.
- **whether applied on live environment:** Yes.
- **learning method:** Expert system, artificial neural network, generic algorithm.
- **applicability:** Electric load forecasting.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** These techniques are easily implementable.
- **goal of the method:** Electric load forecasting.
- **evaluation:** This is a state-of-the-art survey of commonly used AI technologies for load forecasting.

3.2.3.6 Logistic Model Trees

In this paper, Landwehr et. al. presented an algorithm called Logistic Model Tree (LMT), that combines logistic regression and tree induction methods for the prediction of nominal classes and of continuous numeric values. LMTs are trees that contains logistic regression functions in their leaves. The model was evaluated against 'model trees' (described in a previous work) that use linear regression functions [70].

- **Basic information:** Written by Niels Landwehr, Mark Hall, and Eibe Frank. Published in 2003 by Springer, there are 113 references for this paper.
- **model refreshment:** Yes.
- **whether applied on live environment:** No, only benchmarked.
- **preprocessing method:** Simple linear regression with (five fold) cross validation and transform data to numeric ones (if raw data not numeric).
- **learning method:** LMT (Logistic Model Tree).
- **tools:** CART pruning algorithm, LogitBoost.

- **metrics, predictors:** 32 UCI datasets.
- **applicability:** Accurate learning algorithm for forecasting techniques and classification tasks.
- **whether applicable in telecommunication?:** Probably yes.
- **goal of the method:** Accurate classification and prediction.
- **evaluation:** LMT has very good accuracy but high computational complexity. On future work speed may improvable by new regression methods or optimize LogitBoost method.

3.2.3.7 H-mine: hyper-structure mining of frequent patterns in largedatabases

This paper describes a memory based mining structure, H-struct, and an algorithm for it called H-mine. This technology is designed for low and medium density data sets in large data bases. The data collections are projected in a special link based data model, and the data partitions switch between memory and HDD for better performance. If the density of the data set grows, the algorithm changes into the FP-grow model [76].

- **Basic information:** Written by Jian Pei, Jiawei Han, Hongjun Lu, Shojiro Nishio, Shiwei Tang and Dongqing Yang. Published in 2001 by IEEE, there are 172 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Yes.
- **whether applied on live environment:** No, test cases only.
- **preprocessing method:** Create link projection.
- **learning method:** H-mine.
- **metrics, predictors:** Raw data.
- **applicability:** Usable in various cases.
- **whether applicable in telecommunication?:** Probably yes.
- **difficulty to implement:** This technique seems easily implementable.
- **goal of the method:** Fast and scalable method for data mining.
- **evaluation:** These techniques have various usage areas and good performance.

3.2.3.8 On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms

In this paper the SmartSifter tool is presented. It detects outliers in an online process through the online unsupervised learning of a probabilistic model (using a finite mixture model) of the information source. SmartSifter employs an online discounting learning algorithm to study the probabilistic model. It calculates a score for every data, the higher this score is, the more possible it is that the data reflects a statistical outlier. This tool has some novel features: adaptive to non-stationary sources of data, clear statistical meaning, low computational cost, handles not only categorical but continual variables [95].

- **Basic information:** Written by Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams and Peter Milne. Published in 2000 by ACM, there are 134 references for this paper.
- **realtime:** Yes.
- **model refreshment:** Online.
- **whether applied on live environment:** Yes: network intrusion detection for KDD Cup 1999, rare event detection for Australia's Health Insurance Commission.
- **preprocessing method:** Aggregation.
- **learning method:** SDLE (Sequentially Discounting Laplace Estimation), SDEM (Sequentially Discounting Expectation and Maximizing).
- **tools:** SmartSifter.
- **metrics, predictors:** Categorical and continuous attributes (entities: doctor, patient, laboratory).
- **applicability:** Network.
- **whether applicable in telecommunication?:** Yes.
- **difficulty to implement:** This method is easily implementable.

- **goal of the method:** Outlier detection.
- **evaluation:** New efficient approach for outlier detection with their novel tool.

SmartSifter has two different version: a parametric and a non-parametric version. The former one seems to be more precise to detect failure.

3.3 Classification and comparison of methods

In this section, the examined articles are presented from different aspects.

First, Table 2 shows the use area of each article. There are 29 articles which use the methods in different applicability areas. Half of them (exactly 14) have been evaluated with the help of a typical three-tier architecture (web server, application level, database level) of enterprise systems. Most of these technologies have been tested with the PetStore sample application demo [51]. Its functionality consists of store front, shopping cart, purchase tracking, etc. Usually a client emulator is built to generate workload similar to that created by typical user behavior. With the help of this demo application, faults can be simulated.

Although there are only five papers used in telecommunication, most of the mentioned techniques from the examined papers can be used in this area. This information is given in the detailed list of articles above.

Four of the articles deal with network topology problems, and there are some other papers which have special applicability areas.

The SPSS Statistics tool is the one we have found to be used in marketing, finance, CRM, etc.

Secondly, Table 3 presents the most important goals of the methods mentioned in the articles. The goals of the methods are divided into four subareas: fault (1 article) or performance prediction (9 articles), failure detection (13 articles) and other methods (6 articles). Not only prediction methods but failure detection methods are examined as well, because these themes have similar steps. There are two articles which present only one step of the full method. Although the goal of these articles is not finding a predicting or a detecting failure, based on the presented step, both can be done.

According to this classification, we show the main steps of the examined methods. We determine 4 main general steps:

- raw data generation: What kind of data is retrieved? (monitoring metrics)
- filtering, transformations: How is the monitored information reduced or aggregated?
- learning: How is the relationship between data learned? (learning algorithm)
- testing, forecasting: What was their purpose? (detect or predict failure or performance degradation)

The methods in Table 4 predict performance degradation. These methods observe system-level resource metrics like bandwidth, latency, throughput, CPU use percentage, accessibility of components, etc. These metric values are continuously monitored, and if an anomaly appears, these techniques forecast the change of performance. Three of our examined articles [41, 42, 97] employs TANs to capture the pattern of performance behavior. They determine the subset of metrics that are most highly correlated with the SLO (service level objectives) performance violation. SLOs are usually expressed in terms of high-level behaviors (e.g. average response time, request throughput). Only two of them is listed in Table 4, the third one is mentioned only for detection, not for prediction (see Table 5).

Table 5 shows the methods for failure detection. The preprocessing method is usually in a linear correlation with a threshold. In most cases, the employed learning algorithm is a kind of linear regression. One of these articles [69] compare this kind of linear regression techniques. ARX was been found to be the most accurate one, but it is very expensive and not so robust. Simple linear regression is the least expensive one. Four of our examined articles [47, 55, 56, 57] present a novel monitoring method, flow intensity. These articles have the same authors and they improve their base algorithm to retrieve invariants more and more efficiently. Two of these methods [42, 58] try to identify the failures that are similar to the ones that occurred earlier. This helps to address the cause of the failure.

Table 6 shows the fault prediction processes and other approaches. Several partial technologies, or interesting articles are not close-knit with our issue, but we regard them as interesting or probably usable approaches to improve future works. In our research, we have found the SmartShifter tool [95] being used in other context, and we studied the high precision rate of LMT [70] earlier. There are some new approaches in data mining technologies [40, 76] which try to minimize the resources needed for a huge set of databases.

Article [cite]	*enterprise application*	*telecom-munication*	*network topology*		*finance*	*other*
			hardware	*software*		
AgrawalaChen2006QMON [30]	H	H	Z	Z	Z	L
BentHentenryck2004OnlineStochasticOptimization [36]	L	L	Z	Z	Z	L
ChenJiang2005FailureDetection [38]	H	L	Z	Z	Z	L
ChenAccardi2004Path-BasedFailure [39]	H	H	Z	Z	Z	L
2006RegressionCube [40]	Z	Z	Z	Z	Z	H
CohenGoldszmidt2004CorrelatingInstrumentation [41]	H	L	Z	Z	Z	L
CohenZhang2005CapturingIndexing [42]	H	L	Z	Z	Z	L
FeinbergGenethliou2005LoadForecasting [45]	Z	Z	Z	Z	Z	H
KuenningPopek1994anAnalysisofTrace [61]	L	H	Z	Z	Z	L
GuoJiang2006TrackingProbabilistic [47]	H	L	Z	Z	Z	L
JensenNeville2002DataMininginSocialNetworks [54]	L	L	Z	Z	H	H
JiangChen2006DiscoveringLikely [55]	H	L	Z	Z	Z	L
JiangChen2006ModelingandTracking [56]	H	L	Z	Z	Z	L
JiangChen2007EfficientandScalable [57]	H	L	Z	Z	Z	L
JiangMunawar0000DiagnosisofRecurrent [58]	H	L	Z	Z	Z	L
KicimanFox2005DetectingApplication-Level [60]	H	H	Z	Z	Z	L
LiHerbsleb2005ForecastingField [62]	L	Z	Z	Z	Z	H
OravecPetras2008VideoTrafficPrediction [67]	H	H	Z	Z	Z	L
MunawarWard2006AdaptiveMonitoring [68]	H	L	Z	Z	Z	L
MunawarWard2007aComparativeStudy [69]	H	L	Z	Z	Z	L
LandwehrHall2004LogisticModel [70]	Z	Z	Z	Z	Z	H
OstrandWeyuker2005PredictingtheLocation [71]	L	L	Z	Z	Z	H
2001Hmine [76]	Z	Z	Z	Z	Z	H
Rish0000ActiveSampling [80]	L	L	H	H	Z	L
SousaRodrigues2007aScalableMulti-Agent [83]	Z	Z	Z	H	Z	L
WahyudinSchatten2008DefectPrediction [88]	L	L	Z	Z	Z	H
Wolski1997ForecastingNetwork [93]	Z	L	H	Z	Z	L
YamanishiTakeuchi2000OnlineUnsupervised [95]	Z	Z	Z	H	Z	L
ZhangCohen2005EnsemblesofModels [97]	H	L	Z	Z	Z	L
SPSS Statistics	Z	Z	Z	Z	H	L

Table 2. Relevance considering telecommunications or enteprise systems (H-High, L-Low, Z-Zero).

The testing of methods is very important for estimating their performance. Precision and recall are common metrics used to evaluate accuracy.

Precision for a class is the number of true positives (the number of items correctly labeled as belonging to the class) divided by the total number of elements labeled as belonging to the class (the sum of true positives and false positives, which are the items incorrectly labeled as belonging to the class).

Recall is defined as the number of true positives divided by the total number of elements that actually belong to the class (the sum of true positives and false negatives, which are the items that are not labeled as belonging to the class but should have been).

In Table 7, we collect testing information about testbed and accuracy. Unfortunately, the testing phase is missing in some articles, because the authors have only a prototype, or have not implemented the program yet.

In 16 articles, the program is tested only in a test environment. Actually, the authors could easily check their algorithms in a testprogram with injected faults. In 7 articles, the methods are applied in test and live environments as well, while in four articles the authors write about their testing experiments only in live environments. Finally, there are two articles in which the testing phase is missing due to the lack of implementation.

Precision is mentioned in less than half of the articles, and recall in only four cases. The complexity of the fault prediction and detection analysis is high, but its accuracy is not as high as it is expected. In the articles where accuracy is mentioned, the lowest precision value is 50% and the highest is 100%. Most of them have about 90 - 95% precision values, which are accurate enough. The recall values are between 93 - 97% except the one which has a recall value of 34.2% in [42]. In this article, the failure detection method has a precision value of 92%, which means that when the top 100 signatures are

Article (cite, section)	*fault prediction*	*performance prediction*	*detection*	*other*
AgrawalaChen2006QMON [30] 3.2.3.2				X
BentHentenryck2004OnlineStochasticOptimization [36] 3.2.3.3				X
ChenJiang2005FailureDetection [38] 3.2.1.2			X	
ChenAccardi2004Path-BasedFailure [39] 3.2.1.3			X	
2006RegressionCube [40] 3.2.3.4				X
CohenGoldszmidt2004CorrelatingInstrumentation [41] 3.2.2.1		X		
CohenZhang2005CapturingIndexing [42] 3.2.1.4			X	
FeinbergGenethliou2005LoadForecasting [45] 3.2.2.2		X		
KuenningPopek1994anAnalysisofTrace [61] 3.2.2.3		X		
GuoJiang2006TrackingProbabilistic [47] 3.2.1.5			X	
JensenNeville2002DataMininginSocialNetworks [54] 3.2.2.4		X		
JiangChen2006DiscoveringLikely [55] 3.2.1.6			X	
JiangChen2006ModelingandTracking [56] 3.2.1.7			X	
JiangChen2007EfficientandScalable [57] 3.2.1.8			X	
JiangMunawar0000DiagnosisofRecurrent [58] 3.2.1.9			X	
KicimanFox2005DetectingApplication-Level [60] 3.2.1.10			X	
LiHerbsleb2005ForecastingField [62] 3.2.1.1	X			
OravecPetras2008VideoTrafficPrediction [67] 3.2.2.5		X		
MunawarWard2006AdaptiveMonitoring [68] 3.2.1.11			X	
MunawarWard2007aComparativeStudy [69] 3.2.1.12			X	
LandwehrHall2004LogisticModel [70] 3.2.3.6				X
OstrandWeyuker2005PredictingtheLocation [71] 3.2.1.13			X	
2001Hmine [76] 3.2.3.7				X
Rish0000ActiveSampling [80] 3.2.2.6		X		
SousaRodrigues2007aScalableMulti-Agent [83] 3.2.2.7		X		
WahyudinSchatten2008DefectPrediction [88] 3.2.1.14			X	
Wolski1997ForecastingNetwork [93] 3.2.2.8		X		
YamanishiTakeuchi2000OnlineUnsupervised [95] 3.2.3.8				X
ZhangCohen2005EnsemblesofModels [97] 3.2.2.9		X		

Table 3. Goal of the method.

retrieved, 92 of them are correctly retrieved. A recall value of 34.2% means that 34.2% of all the retrieved signatures are correctly retrieved.

The articles in the next list include performance information:

- **ChenAccardi2004Path-BasedFailure [39]:** The tool analyzes 200K paths in less than 3 seconds.
- **2006RegressionCube [40]:** The PC used for the tests is an AMD PC 1.0GHz with 1G memory, running Windows 2000 server. It is clearly readable from the table, that working with the compressed data is faster and more efficient than working with the exhaustive data.
- **CohenZhang2005CapturingIndexing [42]:** "Each tier (Web, J2EE, database) runs on a separate HP NetServer LPr server (500 MHz Pentium II, 512 MB RAM, 9 GB disk, 100 Mbps network cards, Windows 2000 Server SP4) connected by a switched 100 Mbps full-duplex network. In our prototype implementation in Matlab, given an ensemble with approximately 41-67 models (generated by using one month of system data), it takes about 200ms to compute a signature for one epoch. Using the k-medians algorithm (with k=10) to cluster 7507 signatures (about one month when using 5-minute epochs) takes less than 10 seconds. Finally, retrieving the top 100 matching signatures from a database of 7700 signatures takes less than one second."
- **FeinbergGenethliou2005LoadForecasting [45]:** These techniques are implemented in the Electric industries. This paper does not specify exact data, but describes good precision by using forecasting methods.

Article (cite, section)	*raw data generation*	*filtering, transformations*	*learning*	*testing, forecasting*
CohenGoldszmidt 2004 CorrelatingInstrumentation [41] 3.2.2.1	application-level performance data, system-level resource utilization metrics	pattern classification	TAN (Tree-Augmented Bayesian Network)	performance diagnosis and forecasting
FeinbergGenethliou 2005 LoadForecasting [45] 3.2.2.2	Weather and social datas	Composition of raw data	Regression, neural network, fuzzy logic and statistical methods	Forcasting Load usage data
KuenningPopek 1994 anAnalysisofTrace [61] 3.2.2.3	file system	statistics	N/A	N/A
JensenNeville 2002 DataMininginSocialNetworks [54] 3.2.2.4	Relational data of Social networks	QGraph (create graph of relations)	RPT (Relational probability trees)	Forecasting method by connections between relation of objects
OravecPetras 2008 VideoTrafficPrediction [67] 3.2.2.5	video stream	N/A	neural network	throughput of video stream
Rish 0000 ActiveSampling [80] 3.2.2.6	probe (test transactions: ping, traceroute, webpage-access, database query, an e-commerce transaction, etc.)	MMMF (max-margin matrix factorization)	active learning heuristic (minimum margin sample)	predicting the quality of service which the server could provide: accurate prediction of end-to-end performance and resource allocation (bandwidth, latency)
SousaRodrigues 2007 aScalableMulti-Agent [83] 3.2.2.7	performance metrics (DNS res Time, TCP connect, Whole Page, Vieo QoS), Content Checking (static and dynamic)	floating average	time series analysis	performance degradation forecasting, application level failures and operator errors, service outages
Wolski 1997 ForecastingNetwork [93] 3.2.2.8	Latency, CPU percentage, throughput	N/A	mean-based methods, median-based methods, autoregressive method	predict network performance (TCP/IP end-to-end throughput, latency), system resource
ZhangCohen 2005 EnsemblesofModels [97] 3.2.2.9	application-level performance data, system-level resource utilization metrics	pattern classification	TAN (Tree-Augmented Bayesian Network)	performance diagnosis and forecasting

Table 4. Part of the performance prediction methods.

- **2001Hmine [76]:** This algorithm is implemented on a Pentium 466MHz PC with 128 megabytes memory and a 20Gb hard disk. The H-mine algorithm is 5 times faster than FP-grow and 10 times faster than Apriori.
- **JiangChen2006DiscoveringLikely [55]:** "Our algorithm is implemented in Java programming language and runs on a Pentium 4 machine with 3.0 GHz CPU and 512KB L2 cache. It takes 1419 seconds execution time to process 1.5 hour data and construct all 6105 models in the first phase. Note that this process can be done offline with collected data. Sequential testing process is very fast and it averagely takes 2.5 seconds execution time for each testing phase that has half an hour data and 1158 models."
- **JiangChen2007EfficientandScalable [57]:** "All our algorithms are implemented in Java programming language and run on a Pentium 4 machine with a 3.0-GHz CPU and 512-Kbyte L2 cache. It takes 1,296 sec of execution time for Part I of the FullMesh algorithm to process 1.5 hours of monitoring data and construct all 6,105 models. We observe that 1,158 models (among the total 6,105 models) have fitness scores higher than 50. As mentioned earlier, Part II of the FullMesh algorithm runs very fast, and it only takes 2.5 sec of execution time on the average for each sequential testing phase to process half an hour of data and validate 1,158 models." "... the SimpleTree algorithm reduces computational time from 1,290 sec used in the FullMesh algorithm to 222 sec."
- **LandwehrHall2004LogisticModel [70]:** Compared with other tools, LMT creates lesser trees, but has a high com-

putional rate fitting the logistic regression models.

- **YamanishiTakeuchi2000OnlineUnsupervised [95]:** 97621 data – SS: 28.2 seconds, SS*: 244.8 seconds.

3.4 Evaluation

3.4.1 State of the art of different fields

3.4.1.1 Web application There are 14 articles which describe fault prediction or detection methods tested in web application environments, and 10 of these methods can be used online. Real-time usage allows the user to continuously process the new information in order to predict failure or performance degradation as early as possible. Another important feature of the methods used in real systems is the way the method handles the changing environment: whether it can or cannot refresh the model. 6 of the 10 methods can do it, but only two of them can do it online [38, 42].

Precision can show the accuracy of the prediction (detection) technique. The precision value of the above-mentioned two methods: [42] has a precision value of 92%, while [38] has a precision value of 96%. The recall value is not given in [38], but in [42] it is only 34.2%. This means that the method cannot find many failures, but the ones that are found are real.

Another important feature is scalability: how the method works in a large system. A method for large-scale systems is employed in [57], which describes two algorithms for reducing computational complexity. These algorithms are not only efficient but accurate as well: the precision values are 100% and 96%, while the recall values are 94% and 93%. The drawback of these high values is that the methods do not work efficiently in real time on real systems. The methods are online, but they do not employ model refreshment.

The authors develop a maintainable, extensible, and reusable architecture for complex detection and diagnosis in [39]. These features are important as well, because the method can be modified, some components can be changed independently, or it can be easily maintained. In another article [41], they employ TANs (Tree-Augmented Bayesian Networks) which can be interpreted and modified. This method is online, the model refreshment is supported. It can be used for performance forecasting with the precision of 83 - 94% .

3.4.1.2 Network topology There are four articles where the mentioned methods can be used in network topology. Active sampling [80] is a novel unique approach for hardware and software failure prediction and detection. The goal of this method is to select the minimal number of probes to improve the quality and speed of decision-making in system management.

The main advantage of [93] is the dynamic short-term prediction of network performance with the support of parallel application scheduling. This method dynamically chooses the best (with the least error statistics) of 3 different kinds of models to handle dynamic systems. The authors give a summary of the best forecasters for throughput and latency performance from the mentioned 9 methods: stochastic gradient and trimmed mean had almost the least prediction errors. The framework is extensible with other learning methods.

Finally, in two articles [83, 95], the employed methods can predict software errors. A tool called SmartSifter is described in [95]. This tool works online with online model refreshment to detect outliers. A score is given to each data, the higher this score is, the more possible it is that the data is an outlier. The top 10% of the scores predicted 97% of the outliers; this recall value is high enough to be regarded as efficient. This tool is used in [38] as well.

The other article [83] describes only one approach (it has not been implemented yet) to detect and forecast remote failures in web-sites. The authors give a features comparison between their tool (GOA-Net) and 4 other similar, widely used commercial tools. Their tool seems to have almost every feature which the others have, but they enhance the existing tools in three main aspects: high scalability, pro-activity in failure forecasting, and Internet Video QoS monitoring.

3.4.1.3 Telecommunication Telecommunication is the assisted transmission of signals over a distance for the purpose of communication. A basic telecommunication system consists of three elements:

- a transmitter that transmits information and converts it to a signal,
- a transmission medium that carries the signal,
- a receiver that receives the signal and converts it back into usable information.

We have found articles related to all three elements.

Moreover, we found articles which deal with the faults of transmitter systems, like those describing the Pinpoint algorithm. Pinpoint is presented in [60]. This tool materializes a fault detection model which works on the components of complex web applications. This online fault detection tool detects 89% - 100% of the major faults.

Additionally, some articles describe the methods which detect or predict fault before software release. The most elaborated method is probably the one described in [88], which uses the correlation of metrics.

Among the articles, we have found one which presents results on the transmission element. Article [67] shows the use of the neural network for video traffic prediction. A complex monitoring system (QMON) is presented in [30], which is a basis for good prediction.

Article [61] deals with the receiver element. This project describes a method which can predict file changes in mobile computers, and it is capable to predict the downloaded online contents.

We have examined whether the methods in the articles can be used in the telecommunication area or not. Based on our evaluation models, we think that all of the methods used in web application or networks can be used in telecommunication as well. Software error detection and/or prediction methods used on web applications can predict/detect the faults of the transmitter system. The quality of the communication medium (bandwidth, throughput, latency, etc.) can be predicted by the methods used in the network.

3.4.1.4 Finance In the field of finance, we have processed only one article and one framework. The article [54] presents a new data mining approach which works with the relational links of huge relational data bases. The framework in finance is SPSS [52]. One telecommunication provider uses SPSS predictive analytics solutions to better understand its business customers and develop new products targeted at specific subscriber groups.

3.5 Open research questions

How will the original system be affected after the integration of the forecasting system? It is really important how a potential solution or method affects the original system. Undesirable side effects can arise because of the integration of the forecasting system, e.g., lower performance or errors derived from the forecasting system. The examination of such consequences is an open question now, the previously presented papers do not discuss it.

How scalable is a forecasting method? The usability of a method is significantly leveraged by its scalability. Principally, two fields are especially affected by scalability. First of all, the scalability of the input data is very important, e.g., can a method which works with 5 MB input data also handle 5 GB input data correctly? Furthermore, the scalability of the forecasting feature of a method cannot be neglected either, e.g., a method which can forecast the subsequent 5 minutes with good results does not obviously forecast 5 days with acceptable results. The scalability of the previously analyzed methods are not emphasized.

How can the methods be compared in practice? The practical comparison of the certain steps of the forecasting process (preprocessing and transformation of the data as well as the creation of the learning model) could give a complete picture about which combination of these steps gives the best method and solution. Such a comparison could also give aspects to be considered while choosing between two or more potential algorithms to be implemented. Most of the presented articles does not discuss these kinds of questions, only few of them are concerned with the comparison of the learning models, but none of them gives a full practical comparison.

Can the predictors be determined effectively? Regarding the learning model, the selected predictors have critical significance. Do any selection strategies that can search and select the most effective predictors exist?

Optimally, when should the learning model be refreshed? The refresh rate of the model is really important in the case of a forecasting component which is integrated into a real-life software system. Too frequent model refreshments can overload the original software system, while rare model refreshments can decrease the efficiency of prediction. On the basis of these facts, the aim is to determine the optimal refresh rate of the model.

Which frequency is the best for pattern sampling? The frequency of pattern sampling is important from many aspects. First of all, in the case of a real-life software, the system can be overloaded by too frequent pattern sampling. On the other hand, the quality of forecasting can be decreased by an improper (too frequent or too rare) pattern sampling. Thus, it is not an easy task to find the appropriate sampling rate. Most of the articles do not deal with this question.

How are the certain steps of the forecasting process named? When writing the survey, we experienced that the articles used different terminologies for the forecasting process steps. This can lead to serious problems and misunderstandings, therefore it would be very important to define a common naming standard for the steps of the whole process.

4 Relationship between CEP and PA

As we presented in this survey, *complex event processing* engines process events in the operation of large scale software systems to understand systems, detect patterns, find relationship (causality, membership or timing) between events. *Predictive analytics*, on the other hand, deals with the prediction of future events based on the previously experienced historical events and predictors to predict or detect faults. So, in a sense, CEP is more complete as it deals with all the details of collecting, filtering and processing the data for a specific goal in a specific environment, and it deals not only with prediction. At the same time PA is more general in the sense that the methods can be applied in a variety of applications and configurations where prediction based on past data is needed. In this section, we summarize our view on the relationship of these two areas. This is rather challenging, since on one hand there is a large overlap between the approaches in terms of the goals and the basic techniques applied, and on the other hand the methods are sometimes approached quite differently with different terminology and matureness of the solution.

The most notable difference lies with the fact that complex event processing engines require rules or patterns to alert the upcoming problematic events, which usually have to be given manually by the administrators of the system, while the goal of predictive analytics is usually to automate these processes. Based on this, there is an assumption that administrators have the required preliminary knowledge, which sometimes is not available (in the case of unknown patterns) or not so precise. So the manual setting of the rules and patterns are in a sense a weak point of complex event processing, which can readily be supported by the techniques of predictive analytics. Predictive analytics is not a key part of most complex event processing engines, but current CEP engines can probably be extended (manually) with PA.

Furthermore, predictive analytics applies diverse disciplines such as probability and statistics, machine learning, artificial intelligence, and other areas of fundamental computer science to business problems; it employs a variety of methods and techniques from data mining and statistics that explore current and historical data to make predictions about future events. On the other hand, (complex) event processing often employs relatively simple techniques for similar tasks. Hence, with the help of PA, CEP could be made more automatic and intelligent leading to improved accuracy, speed, consistency, and easier maintenance.

The next issue is the point in time the data is processed. Since, in the case of PA, the learning phase takes place before the usage of the model (possibly real-time), these techniques require certain computations in advance. On the other hand, CEP techniques are most often applied in real-time. In the case of PA, the emphasis is often on how to *build* a suitable model for prediction. In many cases the solution is "academic" meaning that remains on the level of a research prototype, and it is often only a building block and not a complete solution. CEP techniques, on the other hand, concentrate on *using* the model and thus usually include a complete solution (also involving data storage and transfer, monitoring, filtering, language, action, architecture, etc.), and are targeted more in industrial applications than in academia.

A further difference is that while complex event processing monitors all of the events come continuously (also including a lot of unnecessary information), predictive analytics monitors only some predefined metric data. CEP examines the events with their context, it needs all of the information because of the event is considered as context dependent. Also, according to CEP, the events have a hierarchy, contrary to PA, where data is usually treated as independent from each other.

Finally, time between events is an important factor with CEP techniques; it is measured and processed due to their context dependency. On the other hand, time is usually not an important factor with PA because metric values are sampled at certain points in time (however, the elapsed time between events is sometimes used as a predictor).

In Table 8, we summarize the similarities and differences of PA and CEP.

5 Conclusions and possible future work

In this paper, we have examined the research areas of Complex Event Processing (CEP) and Predictive Analytics (PA). The two fields are in many aspects similar but show some differences as well, as we outlined in the previous section. Regarding CEP, it is clear that this area is going to be very important in the future in the fields of software engineering, software maintenance and IT operations, which view is supported by other related surveys as well. However, this area is relatively young; it is not standardized in terms of its terminology, solutions, and lots of other factors. Event processing in general is developing rapidly, but it has many deficiencies as well. A major deficiency, the lack of automation and the need for human interaction, could be supported by applying the techniques of predictive analytics. Due to this and the similarity of the two fields, we present them jointly in this survey, although the primary focus is on event processing.

In this paper, we evaluated the state of the art of CEP and PA by assessing the relevant research papers, practical implementations and tools, and other sources of information like web articles. Although the assessment was general, some domains were more in focus, like telecommunication. Sections 2.5, 2.6, 3.4 and 3.5 provide the summaries of evaluation and the list of currently open research questions, respectively. The open questions sections can serve as possible future research areas. We envision that the integration of these two research fields could be a major improvement over state of the art of both areas.

There are several other opportunities for future work as well. First, a practical comparison of the presented papers, solutions, methods, and tools should be performed by using or defining a benchmark designed for this purpose [48, 84, 50, 49]. A practical comparison could show the concrete differences between the different tools in terms of speed, memory consumption, accuracy, and other aspects.

Another future direction could be to participate in the development of an open source CEP solution, e.g., Esper, thus gaining valuable practical experience. Although this survey tries to comprehensively overview the ongoing research results of the event processing research area, it could be extended with real experiments coming from the development of practical CEP applications. There are several directions for such further development. First, enhanced CEP engines could deal with uncertain data – when the events are not certain to happen – as presented in two papers [81, 90]. Another development direction could be to handle incomplete data, e.g., when an event misses some of its attributes. Finally, a specific development direction could be to automate the manual phases of event processing, like we already suggested with predictive analytics. An example is to use ECA (Event Condition Action [74]), or machine learning algorithms [90] as a support technique.

References

[1] Homepage of Aleri CEP 3.0. http://www.aleri.com/.
[2] Homepage of Aurora. http://www.cs.brown.edu/research/aurora/.
[3] Homepage of BEA WebLogic. http://edocs.bea.com/.
[4] Homepage of Borealis. http://www.cs.brown.edu/research/borealis/public/.
[5] Homepage of Cayuga. http://www.cs.cornell.edu/database/cayuga/.
[6] Homepage of Coral8. http://www.coral8.com/.
[7] Homepage of Esper/NEsper. http://www.espertech.com/.
[8] Homepage of Event Zero. http://www.event-zero.com/.
[9] Homepage of IBM Active Middleware Technology (Amit). http://www.haifa.il.ibm.com/dept/services/soms_ebs.html.
[10] Homepage of IBM InfoSphere Streams (supposed System S). http://www-01.ibm.com/software/data/infosphere/streams/.
[11] Homepage of IBM WebSphere Business Events. http://www-01.ibm.com/software/integration/wbe/.
[12] Homepage of Intelligent Event Processor (IEP). https://open-esb.dev.java.net/IEPSE.html.
[13] Homepage of Oracle Complex Event Processing 10g. http://www.oracle.com/technology/products/event-driven-architecture/complex-event-processing.html.
[14] Homepage of PADRES. http://research.msrg.utoronto.ca/Padres/.
[15] Homepage of PIPES. http://dbs.mathematik.uni-marburg.de/Home/Research/Projects/PIPES/.
[16] Homepage of Progress Apama. http://www.progress.com/apama/index.ssp.
[17] Homepage of RealTime Monitoring. http://www.realtime-monitoring.com/.
[18] Homepage of RuleCore CEP Server. http://www.rulecore.com/.
[19] Homepage of RulePoint. http://www.agentlogic.com/products/rulepoint.html.
[20] Homepage of Senactive InTime. http://www.senactive.com/index.php?id=113&L=1.
[21] Homepage of Sopera. http://www.sopera.de/en/home/.
[22] Homepage of STREAM. http://www-db.stanford.edu/stream/.
[23] Homepage of Stream-based And Shared Event Processing (SASE). http://sase.cs.umass.edu/.

[24] Homepage of StreamBase. http://www.streambase.com/.
[25] Homepage of TelegraphCQ. http://telegraph.cs.berkeley.edu/.
[26] Homepage of TIBCO BusinessEvents. http://www.tibco.com/.
[27] Homepage of Truviso. http://truviso.com/.
[28] Simon fraser university database and data mining lab. URL: http://ddm.cs.sfu.ca.
[29] Streambase: Cep standards. http://streambase.typepad.com/streambase_stream_process/cep_standards/.
[30] I. Agarwala, Y. Chen, D. Milojicic, and K. Schwan. Qmon: Qos- and utility-aware monitoring in enterprise systems. 2006. doi: http://dx.doi.org/10.1109/ICAC.2006.1662390.
[31] C. C. Aggarwal. On classification and segmentation of massive audio data streams. 2008. URL: http://dx.doi.org/10.1007/s10115-008-0174-y.
[32] Aleri STAC certification. http://www.aleri.com/files/STAC_0.pdf.
[33] A. Arasu, M. Cherniack, E. Galvez, D. Maier, A. S. Maskey, E. Ryvkina, M. Stonebraker, and R. Tibbetts. Linear road: a stream data management benchmark. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 480–491. VLDB Endowment, 2004.
[34] R. S. Barga, J. Goldstein, M. H. Ali, and M. Hong. Consistent streaming through time: A vision for event stream processing. *CoRR*, abs/cs/0612115, 2006.
[35] A. P. Barros, G. Decker, and A. Grosskopf. Complex events in business processes. In W. Abramowicz and W. Abramowicz, editors, *BIS*, volume 4439 of *Lecture Notes in Computer Science*, pages 29–40. Springer, 2007. URL: http://dx.doi.org/10.1007/978-3-540-72035-5_3.
[36] R. Bent and P. V. Hentenryck. Online stochastic optimization under time constraints, 2004.
[37] P. Bizarro. Bicep - benchmarking complex event processing systems. In M. Chandy, O. Etzion, and R. von Ammon, editors, *Event Processing*, number 07191 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
[38] H. Chen, G. Jiang, C. Ungureanu, and K. Yoshihira. Failure detection and localization in component based systems by online tracking. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 750–755, New York, NY, USA, 2005. ACM. doi: http://doi.acm.org/10.1145/1081870.1081968.
[39] M. Y. Chen, A. Accardi, E. Kiciman, J. Lloyd, D. Patterson, O. Fox, and E. Brewer. Path-based failure and evolution management. In *In Proceedings of the International Symposium on Networked Systems Design and Implementation (NSDI'04*, pages 309–322, 2004.
[40] Y. Chen, G. Dong, J. Han, J. Pei, B. W. Wah, and J. Wang. Regression cubes with lossless compression and aggregation. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1585–1599, 2006. doi: http://doi.ieeecomputersociety.org/10.1109/TKDE.2006.196.
[41] I. Cohen, M. Goldszmidt, T. Kelly, J. Symons, and J. S. Chase. Correlating instrumentation data to system states: a building block for automated diagnosis and control. In *OSDI'04: Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation*, pages 231–244, Berkeley, CA, USA, 2004. USENIX Association. URL: http://www.usenix.org/events/osdi04/tech/full_papers/cohen/cohen.pdf.
[42] I. Cohen, S. Zhang, M. Goldszmidt, J. Symons, T. Kelly, and A. Fox. Capturing, indexing, clustering, and retrieving system history. *SIGOPS Oper. Syst. Rev.*, 39(5):105–118, 2005. doi: http://doi.acm.org/10.1145/1095809.1095821.
[43] M. Eckert and F. Bry. Complex event processing (cep). *Informatik Spektrum*, 32(2):163–167, 2009. doi: http://dx.doi.org/10.1007/s00287-009-0329-6.
[44] The Event Processing Technical Society. http://www.ep-ts.com/.
[45] E. A. Feinberg and D. Genethliou. Load forecasting. In *Springer '05: Applied Mathematics for Restructured Electric Power Systems*, pages 269–285. Springer, 2005. doi: http://dx.doi.org/10.1007/0-387-23471-3_12.
[46] T. Greiner, W. Düster, F. Pouatcha, R. von Ammon, H.-M. Brandl, and D. Guschakowski. Business activity monitoring of norisbank taking the example of the application easycredit and the future adoption of complex event processing (cep). In *PPPJ '06: Proceedings of the 4th international symposium on Principles and practice of programming in Java*, pages 237–242, New York, NY, USA, 2006. ACM. URL: http://doi.acm.org/10.1145/1168054.1168090.
[47] Z. Guo, G. Jiang, H. Chen, and K. Yoshihira. Tracking probabilistic correlation of monitoring data for fault detection in complex systems. In *DSN '06: Proceedings of the International Conference on Dependable Systems and Networks*, pages 259–268, Washington, DC, USA, 2006. IEEE Computer Society. doi: http://dx.doi.org/10.1109/DSN.2006.70.
[48] Homepage of BiCEP. http://bicep.dei.uc.pt/index.php/Main_Page.
[49] Homepage of Linear Road. http://www.cs.brandeis.edu/ linearroad/index.html.
[50] Homepage of NEXMark. http://datalab.cs.pdx.edu/niagara/NEXMark/.
[51] Homepage of PetStore. http://java.sun.com/developer/releases/petstore/.

[52] Homepage of SPSS.
http://www.spss.com/.

[53] N. Jain, S. Mishra, A. Srinivasan, J. Gehrke, J. Widom, H. Balakrishnan, U. Çetintemel, M. Cherniack, R. Tibbetts, and S. Zdonik. Towards a streaming sql standard. *Proc. VLDB Endow.*, 1(2):1379–1390, 2008. URL: http://doi.acm.org/10.1145/1454159.1454179.

[54] D. Jensen and J. Neville. Data mining in social networks. In *National Academies Press, '03: Dynamic Social Network Modeling and Analysis*, pages 289–301. National Academies Press, 2003.

[55] G. Jiang, H. Chen, and K. Yoshihira. Discovering likely invariants of distributed transaction systems for autonomic system management. *Cluster Computing*, 9(4):385–399, October 2006. doi: http://dx.doi.org/10.1007/s10586-006-0008-1.

[56] G. Jiang, H. Chen, and K. Yoshihira. Modeling and tracking of transaction flow dynamics for fault detection in complex systems. *IEEE Transactions on Dependable and Secure Computing*, 3(4):312–326, 2006.
doi: http://doi.ieeecomputersociety.org/10.1109/TDSC.2006.52.

[57] G. Jiang, H. Chen, and K. Yoshihira. Efficient and scalable algorithms for inferring likely invariants in distributed systems. *IEEE Trans. on Knowl. and Data Eng.*, 19(11):1508–1523, 2007. doi: http://dx.doi.org/10.1109/TKDE.2007.190648.

[58] M. Jiang, M. A. Munawar, T. Reidemeister, and P. A. Ward. Diagnosis of recurrent faults by invariant analysis in enterprise software.

[59] D. A. K. Metaxiotis, A. Kagiannas and J. Psarras. Artificial intelligence in short term electric load forecasting. 2002.
doi: http://dx.doi.org/10.1016/S0196-8904(02)00148-6.

[60] E. Kiciman and A. Fox. Detecting application-level failures in component-based internet services. *Neural Networks, IEEE Transactions on*, 16(5):1027–1041, 2005. doi: http://dx.doi.org/10.1109/TNN.2005.853411.

[61] G. H. Kuenning, G. J. Popek, and P. L. Reiher. An analysis of trace data for predictive file caching in mobile computing, 1994.

[62] P. L. Li, J. Herbsleb, and M. Shaw. Forecasting field defect rates using a combined time-based and metrics-based approach: A case study of openbsd. *Software Reliability Engineering, International Symposium on*, 0:193–202, 2005.
doi: http://doi.ieeecomputersociety.org/10.1109/ISSRE.2005.19.

[63] D. Luckham and R. Schulte, editors. *Event Processing Glossary – Version 1.1*. Event Processing Technical Society, 2008. URL: http://www.ep-ts.com/component/option,com_docman/task,doc_download/gid,66/Itemid,84/.

[64] D. C. Luckham and B. Frasca. Complex event processing in distributed systems. Technical Report CSL-TR-98-754, 1998. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.876.

[65] M. R. N. Mendes, P. Bizarro, and P. Marques. A framework for performance evaluation of complex event processing systems. In *DEBS '08: Proceedings of the second international conference on Distributed event-based systems*, pages 313–316, New York, NY, USA, 2008. ACM. URL: http://doi.acm.org/10.1145/1385989.1386030.

[66] J. Mihaeli and O. Etzion. Detecting event processing patterns in event databases. 2007. URL: http://www.dcs.bbk.ac.uk/~ptw/vldbw07/edaps/mihaeli.pdf.

[67] M. P. Milos Oravec and F. Pilka. Video traffic prediction using neural networks, 2008.

[68] M. A. Munawar and P. A. S. Ward. Adaptive monitoring in enterprise software systems, 2006.

[69] M. A. Munawar and P. A. S. Ward. A comparative study of pairwise regression techniques for problem determination. In *CASCON '07: Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*, pages 152–166, New York, NY, USA, 2007. ACM. doi: http://doi.acm.org/10.1145/1321211.1321227.

[70] M. H. Niels Landwehr and E. Frank. Logistic model trees. In *Springer '05: Machine Learning*, pages 161–205. Springer Netherlands, 2005. doi: http://www.springerlink.com/content/q7816655u1g42715.

[71] T. J. Ostrand, E. J. Weyuker, and R. M. Bell. Predicting the location and number of faults in large software systems. *IEEE Transactions on Software Engineering*, 31(4):340–355, 2005. doi: http://doi.ieeecomputersociety.org/10.1109/TSE.2005.49.

[72] T. J. Owens. Survey of event processing. Technical report, Air Force Research Laboratory, Information Directorate, Dec 2007. URL: http://www.dtic.mil/cgi-bin/GetTRDoc/?AD=ADA475386&Location=U2\&doc=GetTRDoc.pdf.

[73] M. Palmer. Seven Principles of Effective RFID Data Management. *Progress Software's Real Time Division, http://www.psdn.progress.com/realtime/docs/articles/7principles_rfid_mgmnt.pdf, Tech. Rep*, 2004.

[74] A. Paschke. A homogenous reaction rule language for complex event processing. In *In Proc. 2nd International Workshop on Event Drive Architecture and Event Processing Systems (EDA-PS*, 2007.

[75] A. Paschke. Design patterns for complex event processing. *CoRR*, abs/0806.1100, 2008. URL: http://arxiv.org/abs/0806.1100.

[76] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang. H-mine: Hyper-structure mining of frequent patterns in large databases. *Data Mining, IEEE International Conference on*, 0:441, 2001.
doi: http://doi.ieeecomputersociety.org/10.1109/ICDM.2001.989550.

[77] J. O. Per Gustas, Per Magnusson and N. Storm. Real-time performance monitoring and optimization of cellular systems. 2002.

[78] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43(5):51–58, 2000. URL: http://doi.acm.org/10.1145/332833.332838.

[79] C. S. Rainer v. Ammon and C. Wolff. Domain specific reference models for event patterns – for faster developing of business activity monitoring applications. 2007. URL: http://www.citt-online.com/downloads/ReferenceModelsEventPatterns_with_Appendix_v3.pdf.

[80] I. Rish. Active sampling approaches in systems management applications.

[81] S. Rizvi. Complex event processing beyond active databases: Streams and uncertainties. Master's thesis, EECS Department, University of California, Berkeley, Dec 2005.

[82] K. U. Schmidt, D. Anicic, and R. Stühmer. Event-driven Reactivity: A Survey and Requirements Analysis. In *3rd International Workshop on Semantic Business Process Management*, pages 72–86, 2008.

[83] D. Sousa, N. Rodrigues, L. Silva, and A. Andrzejak. A scalable multi-agent architecture for remote failure detection in websites. Technical Report TR-0072, Institute on Architectural issues: scalability, dependability, adaptability, CoreGRID - Network of Excellence, July 2007. URL: http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0072.pdf.

[84] STAC Benchmark Council. http://www.stacresearch.com/a1.

[85] M. Stonebraker, U. Çetintemel, and S. Zdonik. The 8 requirements of real-time stream processing. *SIGMOD Rec.*, 34(4):42–47, December 2005. doi: http://dx.doi.org/10.1145/1107499.1107504.

[86] M. Suntinger, H. Obweger, J. Schiefer, and M. E. Gröller. The event tunnel: Interactive visualization of complex event streams for business process pattern analysis. Technical Report TR-186-2-07-07, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, May 2007. URL: http://dx.doi.org/10.1109/PACIFICVIS.2008.4475466.

[87] Tutorials on the 3rd ACM International Conference on Distributed Event-Based Systems. http://debs09.isis.vanderbilt.edu/tutorials1.php.

[88] D. Wahyudin, A. Schatten, D. Winkler, A. M. Tjoa, and S. Biffl. Defect prediction using combined product and project metrics - a case study from the open source "apache" myfaces project family. *Software Engineering and Advanced Applications, Euromicro Conference*, 0:207–215, 2008. doi: http://doi.ieeecomputersociety.org/10.1109/SEAA.2008.36.

[89] F. Wang, S. Liu, P. Liu, and Y. Bai. Bridging physical and virtualworlds: Complex event processing for rfid data streams. In *10th International Conference on Extending Database Technology (EDBT'2006)*, 2006.

[90] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin. Complex event processing over uncertain data. In *DEBS '08: Proceedings of the second international conference on Distributed event-based systems*, pages 253–264, New York, NY, USA, 2008. ACM. doi: http://doi.acm.org/10.1145/1385989.1386022.

[91] W. White, M. Riedewald, J. Gehrke, and A. Demers. What is "next" in event processing? In *PODS '07: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 263–272, New York, NY, USA, 2007. ACM. URL: http://doi.acm.org/10.1145/1265530.1265567.

[92] A. Widder, R. v. Ammon, P. Schaeffer, and C. Wolff. Identification of suspicious, unknown event patterns in an event cloud. In *DEBS '07: Proceedings of the 2007 inaugural international conference on Distributed event-based systems*, pages 164–170, New York, NY, USA, 2007. ACM. URL: http://doi.acm.org/10.1145/1266894.1266926.

[93] R. Wolski. Forecasting network performance to support dynamic scheduling using the network weather service. pages 316–325, Aug 1997. doi: http://dx.doi.org/10.1109/HPDC.1997.626437.

[94] E. Wu, Y. Diao, and S. Rizvi. High-performance complex event processing over streams. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 407–418, New York, NY, USA, 2006. ACM. URL: http://doi.acm.org/10.1145/1142473.1142520.

[95] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 320–324, New York, NY, USA, 2000. ACM. doi: http://doi.acm.org/10.1145/347090.347160.

[96] C. Zang and Y. Fan. Complex event processing in enterprise information systems based on rfid. *Enterp. Inf. Syst.*, 1(1):3–23, 2007. URL: http://dx.doi.org/10.1080/17517570601092127.

[97] S. Zhang, I. Cohen, J. Symons, and A. Fox. Ensembles of models for automated diagnosis of system performance problems. In *DSN '05: Proceedings of the 2005 International Conference on Dependable Systems and Networks*, pages 644–653, Washington, DC, USA, 2005. IEEE Computer Society. doi: http://dx.doi.org/10.1109/DSN.2005.44.

Article (cite, section)	*raw data generation*	*filtering, transformations*	*learning*	*testing, forecasting*
ChenJiang 2005 FailureDetection [38] 3.2.1.2	interactions between components	Hotelling T^2 score, SPE (squared prediction error)	SDEM (sequentially discounting expectation maximization) with EWMA filter (exponentially weighted moving average)	failure detection and localization
ChenAccardi 2004 Path-BasedFailure [39] 3.2.1.3	path (collection of observations: timestamp, component name, host name, version number)	aggregation (SQL operations), clustering, classification	decision tree, association rules	detect and diagnose failures (correctness and performance problems), and understand the evolution of a system
CohenZhang 2005 CapturingIndexing [42] 3.2.1.4	application-level performance data, system-level resource utilization metrics	pattern classification	TAN (Tree-Augmented Bayesian Network)	identify when an observed system state is similar to a previously-observed state and evolve to the failure
GuoJiang 2006 TrackingProbabilistic [47] 3.2.1.5	measure user request flow (transaction flow intensity) at multiple checkpoints	Gaussian mixture model	modified Expectation-Maximization (EM)	fault detection
JiangChen 2006 DiscoveringLikely [55] 3.2.1.6	measure user request flow (transaction flow intensity) at multiple checkpoints	fitness score $>$ 50	ARX (AutoRegressive models with eXogenous inputs)	fault detection
JiangChen 2006 ModelingandTracking [56] 3.2.1.7	measure user request flow (transaction flow intensity) at multiple checkpoints	fitness score $>$ 50	ARX (AutoRegressive models with eXogenous inputs)	fault detection
JiangChen 2007 EfficientandScalable [57] 3.2.1.8	measure user request flow (transaction flow intensity) at multiple checkpoints	fitness score $>$ 50, SmartMesh algorithm, SimpleTree algorithm	ARX (AutoRegressive models with eXogenous inputs)	fault detection
JiangMunawar 0000 DiagnosisofRecurrent [58] 3.2.1.9	data reflects performance, state, and errors of components of the application and the application server	SLR (Simple Linear Regression), $R^2 > 0.9$	neural network	reoccured failure detection, diagnosis
KicimanFox 2005 DetectingApplication-Level [60] 3.2.1.10	components	N/A	ID3 (Decision Tree)	
MunawarWard 2006 AdaptiveMonitoring [68] 3.2.1.11	J2EE components	linear correlation between metric pairs	linear regression	activity, performance, state and errors of components and services in the server
MunawarWard 2007 aComparativeStudy [69] 3.2.1.12	components' metrics	$R^2 > 0.6$	SLR (simple linear regression), SLRT (SLR using Transformed Data), SLRS (SLR using Smoothed Data), ARX (AutoRegressive models with eXogenous inputs), LWR (Local Weighted Regression)	problem determination
OstrandWeyuker 2005 PredictingtheLocation [71] 3.2.1.13	number of lines of code, file age, the number of prior faults	logarithm, square root	Negative Binomial Regression	fault detection
WahyudinSchatten 2008 DefectPrediction [88] 3.2.1.14	Project and Product Metrics, normalized number of defects	Pearson correlation	linear regression	fault detection

Table 5. Parts of the detection methods.

Article (cite, section)	*raw data generation*	*filtering, transformations*	*learning*	*testing, forecasting*
LiHerbsleb 2005 ForecastingField [62] 3.2.1.1	145 predictors (product, development, deployment and usage, software and hardware configurations metrics)	statistics	Exponential model (Decision Tree)	fault detection
AgrawalaChen 2006 QMON [30] 3.2.3.2	CPU, memory, network, block I/O	N/A	N/A	performance diagnosis
BentHentenryck 2004 OnlineStochasticOptimization [36] 3.2.3.3	Requests/Weights	N/A	Regret algorithm	Packet scheduling, Vehicle routing
2006 RegressionCube [40] 3.2.3.4	N/A	NCR (nonlinear compression representation)	N/A	Loosless compression of raw data for analysis used by the reducated measures
LandwehrHall 2004 LogisticModel [70] 3.2.3.6	Nominal and numeric value	CART (pruning) LogitBoost (fitting models)	LMT (Logistic Model Tree)	Create accurate single tree for prediction and classification
2001 Hmine [76] 3.2.3.7	N/A	H-struct/FP-tree	H-mine/FP-grown	Effective and low resource used, memory based data mining technique
YamanishiTakeuchi 2000 OnlineUnsupervised [95] 3.2.3.8	categorical and continuous attributes	aggregation	SDLE (Sequentially Discounting Laplace Estimation), SDEM (Sequentially Discounting Expectation and Maximizing)	outlier detection (network intrusion detection, rare event detection)

Table 6. Parts of the other and fault prediction methods.

Article [cite]	*test*	*live*	*precision*	*recall*
AgrawalaChen2006QMON [30]	X	X		
BentHentenryck2004OnlineStochasticOptimization [36]	X			
ChenJiang2005FailureDetection [38]	X		depend on threshold (50: 96%)	
ChenAccardi2004Path-BasedFailure [39]	X	X	decision tree: 77%, association rules: 50%	
2006RegressionCube [40]	X			
CohenGoldszmidt2004CorrelatingInstrumentation [41]	X		83% - 94%	
CohenZhang2005CapturingIndexing [42]	X	X	92%	34.2%
FeinbergGenethliou2005LoadForecasting [45]		X		
KuenningPopek1994anAnalysisofTrace [61]				
GuoJiang2006TrackingProbabilistic [47]	X		99%	
JensenNeville2002DataMininginSocialNetworks [54]	X	X		
JiangChen2006DiscoveringLikely [55]	X			
JiangChen2006ModelingandTracking [56]	X			
JiangChen2007EfficientandScalable [57]	X		SmartMesh: 100%, SimpleTree: 96%	SmartMesh: 94%, SimpleTree: 93%
JiangMunawar0000DiagnosisofRecurrent [58]	X		80.1% - 96.2%	
KicimanFox2005DetectingApplication-Level [60]	X	X	detecting 89% – 100% of major faults	
LiHerbsleb2005ForecastingField [62]	X	X		
OravecPetras2008VideoTrafficPrediction [67]	X			
MunawarWard2006AdaptiveMonitoring [68]	X			
MunawarWard2007aComparativeStudy [69]	X		97% - 98%	
LandwehrHall2004LogisticModel [70]	X		92% classification accuracy	
OstrandWeyuker2005PredictingtheLocation [71]	X		71% – 93%	
2001Hmine [76]	X			
Rish0000ActiveSampling [80]		X		
SousaRodrigues2007aScalableMulti-Agent [83]				
WahyudinSchatten2008DefectPrediction [88]	X	X		
Wolski1997ForecastingNetwork [93]		X		
YamanishiTakeuchi2000OnlineUnsupervised [95]		X		10% of highest scores: 97%
ZhangCohen2005EnsemblesofModels [97]	X		with one model: 77.4%, with 2 models: 87%	with one model: 67.4%, with 2 models: 90%

Table 7. Measure of testing.

Keyword	*Complex event processing*	*Predictive analytics*
Input information	Continuously detected events	Periodically measured predefined metrics
Steps	(1) event, (2) correlation, (3) assessment, (4) decision, (5) action	(1) data, (2) preprocessing, (3) learning the model, (4) using the model
Input overhead	Yes (every event is processed)	No (only predefined metrics are processed)
Emphasis on	Using the model	Computing the model
Model	Given by an administrator manually in form of rules and patterns	Automatically learned by algorithms in forms of decision trees, neural networks, etc.
Preprocessing	Highly supported, full and effective solutions to filter data	Supported, but not so effectively
Reaction	Reaction to certain events or situations are supported by ECA rules	It could be extended manually, but basically not supported
Technology readiness	Complete solution	Only algorithms (correlation, learning model)
Scientific or industrial	Mainly industrial development	Mainly academic development

Table 8. Comparing some aspects of complex event processing and predictive analytics