# Frühzeitige Serverausfallserkennung mittels Vorhersagemethoden und Outlieranalyse

Matthias Wastian[1], Dr. Felix Breitenecker[2], Michael Landsiedl[1]
[1]dwh GmbH, Neustiftgasse 57-59, 1070 Wien, Österreich
[2]Technische Universität Wien, Wiedner Hauptstraße 8-10, 1040 Wien, Österreich
*matthias.wastian@dwh.at*

In dem vorliegenden Paper werden unterschiedliche Ansätze diskutiert, um ungewöhnliche und außerordentliche Ereignisse, für die man eine separate und spezialisierte Betrachtung als wünschenswert erachtet, anhand von in Form einer multivariaten Zeitreihe regelmäßig mitprotokollierten Daten möglichst frühzeitig zu detektieren und bei entsprechender Möglichkeit sogar vorherzusagen. Nach einem kurzen Überblick über diverse Anwendungsbereiche, in denen diese Fragestellung der abnormalen Eventdetektion auftritt, wird der Fokus auf die frühzeitige Erkennung von Serverausfällen gelegt. Die zwei elementaren Bausteine des vorgestellten Modells sind - nach geeigneter Datenvorverarbeitung der großen Anzahl der durch Server Monitoring gemessenen Server Features - ein univariater Prädiktor und ein multivariater Ausreißerdetektor. Großen Einfluss auf diese Modellbausteine haben dabei Methoden aus den Bereichen des Data Mining, des Machine Learning und des Soft Computing.
Als Prädiktoren werden sowohl klassische SARIMA-Modelle als auch Neuroprädiktoren, die einen bestimmten Typ von künstlichen Neuronalen Netzen zur Vorhersage nutzen, vorgestellt. Auf eine kurze Präsentation von Werkzeugen zur Ausreißer- bzw. Novumserkennung, insbesondere einer One-Class Support Vector Machine und des Ansatzes der winkelbasierten Ausreißerdetektion, folgt eine abschließende Diskussion der bis dato erzielten Simulationsergebnisse. Der Text endet mit einem Ausblick auf mögliche Modellerweiterungen und zukünftige Arbeiten.

## 1 Grundbegriffe und Anwendungen von abnormaler Eventdetektion

**Definition 1** (Event)**.** Ein Ereignis wird als ein Vorkommnis definiert, das zu einer bestimmten Zeit an einem bestimmten Ort geschieht und eine bestimmte Dauer besitzt. Dieses Vorkommnis kann Teil einer Kette von Vorkommnissen sein und von einem vorherigen Event verursacht werden oder ein weiteres Event auslösen. Es ist möglich, dass mehrere Events zur gleichen Zeit und/oder am selben Ort auftreten.

**Definition 2** (Abnormales Event)**.** Ein abnormales Event wird als Ausreißer in einer Eventkette definiert und ist somit ein Event, dass so weit von den anderen Events abweicht, so dass ein begründeter Verdacht auftritt, dass es von etwas verursacht worden ist, dass nicht vom Normalverhalten des Systems herrührt, und dass es das gesamte Systemverhalten nachhaltig ändern könnte.

Ausgehend von obigen Definitionen ist es naheliegend, dass abnormale Eventdetektion vor allem bei der Betrachtung von derartigen Systemen von Interesse ist, bei denen die Gewährleistung eines Mindestmaßes an Sicherheit relevant ist. Neben dem diesem Paper zugrunde liegenden Beispiel der Serverausfallsvorhersage sind folgend einige weitere Anwendungsbereiche aufgelistet:

- Vorhersage/Früherkennung von Naturkatastrophen wie Überflutungen, Hurricanes, Erdbeben oder Tsunamis
- Vorhersage/Früherkennung von Aktienmarktein-

brüchen

- Erkennung von unbefugten Netzwerkzugriffen
- Audio- und Videoüberwachung: Erkennung von Massenpaniken, Verkehrsanalyse
- Ambient Assisted Living.

### 1.1 State of the Art

Es ist eingangs zu bemerken, dass abnormale Eventdetektion und vor allem die dafür sinnvollerweise verwendbaren Methoden massiv von der Struktur der gewonnenen Daten sowie von der Art des Systems abhängen.

Für den verwendeten Prädiktor, der auf den statistischen ARIMA-Modellen beruht, liefert [1] eine gute Zusammenfassung der theoretischen Grundlagen, anhand derer direkte zeitliche Abhängigkeiten, Saisonalität, Differenzenbildung u.Ä. modelliert werden. Die Verwendung von künstlichen Neuronalen Netzen (NN) als Vorhersagetool handeln u.a. [2] und [3] ab.

Im Bereich der durchaus auch in einem hochdimensionalen Raum notwendigen Ausreißeranalyse zur abnormalen Eventdetektion wird die Anwendung einer One-Class Support Vector Machine (OC-SVM) u.a. in [4], [5], [6], [7] und [8] vorgeschlagen, diskutiert und evaluiert. Alternativ stellen wir die winkelbasierte Ausreißerdetektion vor, deren Grundlagen in [9] erarbeitet wurden. Ein beschleunigter Algorithmus findet sich in [10].

Weiters stechen als Methoden zur abnormalen Eventdetektion insgesamt die Verwendung von statistischen Methoden ([11]), Clustering ([12]), Replicator Neural Networks ([13]), Sparse Reconstruction Cost ([14]) und Wavelet Decomposition ([15]) heraus.

## 2 Daten

Die für die Simulationsläufe verwendeten Server sind IBM Lotus Notes Server, von denen mittels Server Monitoring bis zu 1439 unterschiedliche Features mit einer konstanten Abtastfrequenz zwischen einmal pro Minute und einmal pro 15 Minuten gemessen wurden. Neben historischem Datenmaterial stand auch ein Software Tool von IBM zur Verfügung, dass künstliche Datensets generieren kann, indem jeder Client eine simulierte Benutzerlast zum Server schickt, der die mitprotokollierten Statistiken retourniert.

Um diese enorme Datenmenge einzuschränken, wurde mittels Expertenwissen der Serveradministratoren eine Kategorisierung aller Server Features in vier unterschiedliche Prioritätsklassen vorgenommen. 14 Features erhielten die höchstögliche Priorität 0, 73 entweder Priorität 0 oder Priorität 1. Die meisten Simulationsläufe wurden mit diesen beiden Gruppen durchgeführt.

Akkumulierte Werte (z.B. die Anzahl der verschickten Mails seit Start des Server Monitoring) wurden differenziert und nach einem kurzen Check auf offensichtlich falsche Messdaten wurden diese sowie Messungen, die keinen Wert (bzw. einen NaN-Wert) geliefert hatten, aus dem Datensatz gelöscht. Keiner der vorgeschlagenen Algorithmen kann nämlich mit NaN-Values umgehen.

Da sich die Wertebereiche der Messgrößen teilweise gravierend voneinander unterscheiden, wurde eine Normalisierung der Daten durchgeführt. Vor Verwendung des Neuro-Prädiktors wurde insbesondere ein sog. Minmax-Mapping angewandt:

**Algorithmus 1** (Minmax-Mapping)**.** Gegeben sei der Datensatz $\{x_1,\ldots,x_n\}$ mit Minimum $x_{min}$ und Maximum $x_{max}$. Das folgende Mapping transformiert diesen affin zu $\{y_1,\ldots,y_n\}$ im Intervall $[y_{min},y_{max}]$.

$$f_{minmax} : [x_{min},x_{max}] \to [y_{min},y_{max}],$$
$$x \mapsto y = \frac{(y_{max}-y_{min})(x-x_{min})}{x_{max}-x_{min}} + y_{min}$$

$y_{min}$ und $y_{max}$ sind im Prinzip frei wählbar; da damit Inputgrößen für ein NN generiert werden, wurden $y_{min} = -0.6$ und $y_{max} = 0.6$ gesetzt. Als Alternative zum Minmax-Mapping ist auch ein klassischer Z-Score denkbar.

## 3 Prädiktoren

Die Prädiktoren des vorgestellten Serverausfallserkennungsmodelles arbeiten univariat. Für jedes der $m$ Features liegt eine Zeitreihe mit einer gewissen Anzahl von in der Vergangenheit erfolgreich gemessenen

Werten vor. Der aktuelle Wert von Feature $i$ wird mit $x_{i,n}$ bezeichnet. Es gilt also, die nächsten Messwerte $x_{i,n+1}, i = 1,\ldots,m$ möglichst akkurat vorherzusagen. Von einer multivariaten Vorhersage wurde aufgrund der deutlich schlechteren Performance (längere Rechenzeiten, größere Vorhersagefehler) und der Tatsache, dass die kausale Abhängigkeit der Features untereinander sehr sparse ist, nach einigen Tests abgesehen.

Grundlegende Modellannahme ist, dass die Prädiktoren im gewünschten Normalbetrieb (i.e., wenn keine abnormalen Events auftreten) gute Vorhersagen liefern; die Vorhersagefehler zumindest zu Beginn eines abnormalen Events aber einer anderen Verteilung entspringen.

Alle Prädiktoren müssen relativ starke saisonale Effekte berücksichtigen. So hängt für einen durchschnittlichen Büroserver die Anzahl der eingeloggten User an einem Montagmorgen um 8:00 sicherlich stärker von dem entsprechenden Wert exakt eine Woche zuvor ab als von jenem am selben Montag um 7:00.

## 3.1 SARIMA-Modell

Durch Analyse der partiellen (PACF) und der Autokorrelationsfunktion (ACF) einer univariaten Zeitreihe werden zunächst die Differenzenordnungen $d$ und $D$ sowie anschließend die restlichen Modellparameter bestimmt, um das $(p,d,q)\times(P,D,Q)_s$-SARIMA-Modell zu erstellen. Parameter, die mit Großbuchstaben bezeichnet werden, gehören dabei zum saisonalen Teil mit Periode $s$. $p$ und $P$ bezeichnen die Anzahlen der autoregressiven Terme, $q$ und $Q$ die der Moving-Average-Terme. Wie genau so eine Analyse vonstattengeht, beschreibt [1].

## 3.2 Neuro-Prädiktor-Modell

Künstliche Neuronale Netze eignen sich aufgrund ihrer nichtlinearen, datengesteuerten Struktur sehr gut als Vorhersagemodell. Auch wenn sie Saisonalität implizit mitabbilden können, wurde eine explizite Modellierung bevorzugt, um die Anzahl der Inputneuronen klein zu halten. Ein Neuro-Prädiktor sucht also eine möglichst gute Funktion $f$ mit $x_{n+1} = f(x_n, x_{n-1},\ldots,x_{n-d+1}, x_{n-s},\ldots,x_{n-2s},\ldots)$.

Anhand der gegebenen Zeitreihe ist es einfach, gelabellte Daten für das NN zu erstellen, um dieses überwacht lernen zu lassen. Die vorhandenen Daten werden dem Konzept der Cross-Validierung folgend aufgeteilt: 70% bilden den Trainings-, 15% den Test- und 15% den Validierungsdatensatz. Laut [16] ist es sinnvoll, die Kardinalität des Trainingssets in etwa als Quotient der Anzahl der Gewichte des NN und des Klassifizierungsfehlers zu wählen. Das vorgeschlagene NN besitzt einen Input-Layer mit so vielen ($n_i$) Neuronen, wie Variable in der obigen Funktion $f$ auftreten, einen Output-Layer mit einem Neuron ($n_o = 1$), das den Vorhersagewert $x_{n+1}$ ausspuckt, sowie eine versteckte Schicht, deren Neuronenanzahl mittels der geometrischen Pyramidenregel als $n_h = \alpha\sqrt{n_i n_o}, \alpha \in [0.5, 2]$, festgelegt wird. Aktivierungsfunktion im versteckten Layer ist der tanh, jene in der Output-Schicht die lineare. Der Levenberg-Marquardt-Trainingsalgorithmus stoppt, wenn eine der Abbruchbedingungen erfüllt ist:

- Die Anzahl der Trainingsepochen überschreitet das vorgegebene Maximum.
- Die Anzahl aufeinanderfolgender Epochen mit steigendem Fehler im Validierungsset überschreitet den vorgegebenen Wert (etwa 6).
- Der Fehler im Testdatensatz liegt unter einem vorgegebenen Zielwert.

## 3.3 Vergleich

Stehen mehrere Prädiktor-Modelle vergleichbarer Qualität zur Auswahl, sollte jenes mit dem niedrigsten AIC- oder BIC-Wert verwendet werden.

Ob SARIMA-Modelle oder Neuro-Prädiktoren effizienter sind, beantwortet die Fachliteratur unterschiedlich. In [16] wurden folgende Aspekte herausgearbeitet: Haben Zeitreihen ein langes Gedächtnis, sind die Ergebnisse ähnlich. Bei einem kurzen Gedächtnis liefern NN manchmal deutlich bessere Resultate. Bei Zeitreihen unterschiedlicher Komplexität ist ein optimal getuntes NN ebenfalls effizienter.

Im konkreten Anwendungsbeispiel der Serverausfallserkennung kommt noch hinzu, dass gemeinsame Parameter für alle Zeitreihen das Modell deutlich vereinfachen. Dies ist mit Neuro-Prädiktoren sicherlich effizienter umsetzbar.

# 4 Ausreißerdetektoren

Grundlage für die Ausreißerdetektoren ist eine multivariate Analyse der Vorhersagefehler durch einen der beiden oben vorgestellten Prädiktoren. Modelliert wird die Ausreißeranalyse als $(1+x)$-Klassifizierungs-Task, d.h. modelliert wird nur die Klasse der normalen Daten. Der Zugang entspricht halbüberwachtem Lernen (vgl. [17]). Neben diesem Modellierungsansatz war auch die Dimension der Vorhersagefehler entscheidend für die Methodenwahl; rein distanzbasierte Verfahren scheitern in hochdimensionalen Räumen aufgrund des Fluchs der Dimensionalität.

Noch nicht implementiert wurde der vielversprechend klingende Ansatz von [20], der auf der Verwendung von Genetischen Algorithmen für Projektionen in niedrigdimensionale Räume beruht.

## 4.1 Winkelbasierte Ausreißererkennung

Seien $A, B$ und $C$ beliebige Punkte in einem Datensatz $\mathscr{D}$ mit $n$ $m$-dimensionalen Vektoren. Der winkelbasierte Outlier-Faktor $ABOF$ von einem Punkt $A$ sei definiert als:

$$ABOF(A) = \underset{B,C\in\mathscr{D}}{\mathbb{VAR}} \left( \frac{\langle \vec{AB}, \vec{AC} \rangle}{\|\vec{AB}\|^2 \|\vec{AC}\|^2} \right)$$

Je kleiner $ABOF(A)$, desto größer die Zugehörigkeit von $A$ zur Klasse der Outlier. Auch wenn $ABOF$ ebenfalls Distanzen verwendet, so ist deren Einfluss beim winkelbasierten Ansatz deutlich geringer als bei rein distanzbasierten Methoden.

Um den Rechenaufwand zu reduzieren, kann anstatt der Varianz aller möglichen Winkel mit Scheitel $A$ auch nur die Varianz jener Winkel hergenommen werden, die durch die $k$ nächsten Nachbarn von $A$ entstehen. Der Algorithmus wird damit $O(m^2 + k^2 m)$ statt $O(m^3 n)$.

## 4.2 One-Class Support Vector Machine

Die OC-SVM kann als reguläre Zwei-Klassen-SVM interpretiert werden, bei der alle Trainingsdaten in der ersten Klasse liegen und der Ursprung das einzige Element der zweiten Klasse ist. Es wird versucht jene Hyperebene $W$ mit Normalvektor $w$ und Bias $b$ zu finden, die die meisten, aber nicht alle Elemente der ersten Klasse mit maximalem Abstand vom Ursprung abtrennt. Dass nicht alle Elemente abgetrennt werden, verhindert den Overfitting-Effekt.

$$wx - b = 0$$

Neben einem Gauss-Kernel $\kappa$ (vgl. [18]) werden Schlupfvariable $\xi_i$ verwendet, um die Fehlerterme unter einem vorgegebenen Budget zu halten.

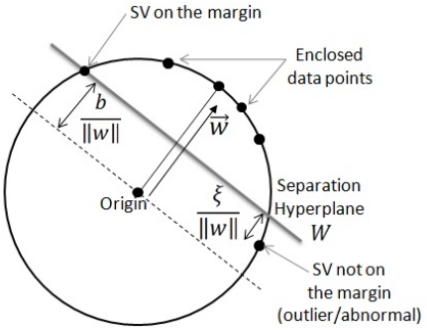


Abbildung 1: One-Class Support Vector Machine ([8])

Das OC-SVM Optimierungsproblem ist äquivalent zum dualen quadratischen Programmierungsproblem mit Lagrange-Multiplikatoren $\alpha$, das mit Standardmethoden und unter Verwendung der Karush-Kuhn-Tucker-Bedingungen gelöst werden kann. Ein neuer Vorhersagefehlervektor wird dann anhand der Entscheidungsfunktion

$$f(x) = \sum_i \alpha_i \kappa(x_i, x) - b$$

bewertet, die positiv für Inlier und negativ für Outlier ist.

# 5 Simulationsergebnisse und Ausblick

Eingangs soll bemerkt werden, dass eine exakte Definition des Terminus *Serverausfall* auch in Expertenkreisen nicht existiert. Jedwede Einschränkung

des Betriebs ist aber definitiv unerwünscht. Während mehrere Testläufe erkannten beide Ausreißerdetektoren mit Leichtigkeit, wenn der Status eines Servers von idle zu busy wechselte oder umgekehrt. Eine umfangreiche Modellvalidierung kann aber letztlich erst bei Veröffentlichung der entsprechenden Software und ausreichenden Tests im Alltagsbetrieb erfolgen. Als Benchmark-Datensatz wurden zudem die Ölpreise über mehrere Jahrzehnte hergenommen: Die Ölkrise von 1979 wurde als abnormales Event eindeutig erkannt.

Mittels mehrerer Tuning Parameter kann die Sensitivität der Serverausfallserkennung vom jeweiligen Systemadministrator bei Bedarf individuell bestimmt werden. Da mehrere Modellteile erst trainiert werden müssen, braucht die Gesamtsoftware im Anwendungsfall eine gewisse Zeit, bis sie sinnvolle Warnungen ausgeben kann.

Noch nicht umgesetzt, aber naheliegend sind eine Kombination der Ergebnisse der beiden Ausreißerdetektoren sowie eine Unifizierung der Outlierscores (vgl. [19]). Alternativ scheint auch der Ansatz von [20] mittels Projektionen in niedrigdimensionale Räume und der Verwendung eines Genetischen Algorithmus vielversprechend.

# Literatur


[1] Robert Nau. *Forecasting - Decision 411*, Online-Kurs, verfügbar unter `people.duke.edu/~rnau/411home.htm`, 2005.

[2] G. Peter Zhang, Douglas Kline. *Quarterly Time-Series Forecasting With Neural Networks*. *IEEE Transactions on Neural Networks*, Volume 8, Nummer 6, IEEE Computational Intelligence Society, S. 1800-1814, 2007.

[3] Sven Crone, Rohit Dhawan. *Forecasting Seasonal Time Series with Neural Networks: A Sensitivity Analysis of Architecture Parameters*. Tagungsband *International Joint Conference on Neural Networks 2007*, IEEE, Orlando, Florida, S. 2099-2104, 2007.

[4] Katherine Heller, Krysta Svore, Angelos Keromytis, Salvatore Stolfo. *One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses*. Tagungsband *Workshop on Data Mining for Computer Security* der *IEEE International Conference on Data Mining 2003*, Melbourne, Florida, S. 2-9, 2003.

[5] Paul Evangelista, Piero Bonnisone, Mark Embrechts, Boleslaw Szymanski. *Fuzzy ROC Curves for the 1 Class SVM: Application to Intrusion Detection*. Tagungsband *13th European Symposium on Artificial Neural Networks 2005*, d-side, Bruges, Belgien, S. 345-350, 2005.

[6] Riu Zhang, Shaoyan Zhang, Yang Lan, Jianmin Jiang. *Network Anomaly Detection Using One Class Support Vector Machine*. Tagungsband *MultiConference of Engineers and Computer Scientists 2008*, Volume 1, IAENG, Hong Kong, 2008.

[7] Stephan Dreiseitl, Melanie Osl, Christian Scheibböck, Michael Binder. *Outlier Detection with One-Class SVMs: An Application to Melanoma Prognosis*. Tagungsband *AMIA Annual Symposium 2010*, S. 172-176, 2010.

[8] Sébastian Lecomte, Régis Lengellé, Cédric Richard, Francois Capman, Bertrand Ravera. *Abnormal Events Detection using Unsupervised One-Class SVM - Application to Audio Surveillance and Evaluation*. Tagungsband *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance 2011*, IEEE, Klagenfurt, Österreich, S. 124-129, 2011.

[9] Hans-Peter Kriegel, Matthias Schubert, Arthur Zimek. *Angle-Based Outlier Detection in High-Dimensional Data*. Tagungsband *14th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining 2008*, Las Vegas, Nevada, ACM, New York, S. 444-452, 2008.

[10] Ninh Pham, Rasmus Pagh. *A Near-Linear Time Approximation Algorithm for Angle-Based Outlier Detection in High-Dimensional Data*. Tagungsband *18th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining 2012*, ACM, New York, S. 877-885, 2012.

[11] Valery Guralnik, Jaideep Srivastava. *Event Detection from Time Series Data*. Tagungsband *5th ACM SIGKDD International Conference on*

*Knowledge Discovery and Data Mining*, ACM New York, USA, S. 33-42, 1999.

[12] Fan Jiang, Ying Wu, Aggelos Katsaggelos. *Abnormal Event Detection Based on Trajectory Clustering by 2-Depth Greedy Search*. Tagungsband *IEEE International Conference on Acoustics, Speech and Signal Processing 2008*, IEEE, Las Vegas, Nevada, S. 2129-2132, 2008.

[13] Simon Hawkins, Hongxing He, Graham Williams, Rohan Baxter. *Outlier Detection Using Replicator Neural Networks*. Tagungsband *4th International Conference on Data Warehousing and Knowledge Discovery 2002*, Aix-en-Provence, Frankreich, veröffentlicht in *Lecture Notes in Computer Science 2454*, Springer, S. 113-123, 2002.

[14] Yang Cong, Junsong Yuan, Ji Liu. *Sparse Reconstruction Cost for Abnormal Event Detection*. Tagungsband *24th IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Colorado Springs, Colorado, S. 3449-3456, 2011.

[15] Mitsutoshi Suzuki, Hitoshi Ihara. *Development of Safeguards System Simulator Composed of Multi-Functional Cores*. *Journal of Power and Energy Systems*, Volume 2, Nummer 2, J-Stage, Japan, S. 899-907, 2008.

[16] Aloy Palit, Dobrivoje Popovic. *Computational Intelligence in Time Series Forecasting - Theory and Engineering Applications*. Springer Verlag, London, 2005.

[17] Victoria Hodge, Jim Austin. *A Survey of Outlier Detection Methodologies*. *Artificial Intelligence Review*, Volume 22, Ausgabe 2, Kluwer Academic Publishers, Niederlande, S. 85-126, 2004.

[18] Bernhard Schölkopf, Alexander Smola. *Learning with Kernels*. MIT Press, Cambridge, Massachusetts, 2002.

[19] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, Arthur Zimek. *Interpreting and Unifying Outlier Scores*. Tagungsband *11th SIAM International Conference on Data Mining*, Mesa, Arizona, 2011.

[20] Charu Aggarwal, Philip Yu. *An Effective and Efficient Algorithm for High-Dimensional Outlier Detection*. *The VLDB Journal 14*, Springer-Verlag, S. 211-221, 2005.