

Využití statistických metod k analýze výsledků výkonostního testování

Martina Kůrová*

Abstrakt

Tato práce se zabývá analýzou dat naměřených během výkonostního testování s cílem rozeznat ve výsledcích podezřelé hodnoty a identifikovat často se vyskytující výkonostní problémy. Užitím statistických metod, jako je například regresní analýza, je provedena analýza výsledků, identifikace výkonostních problémů a výsledkem je report o celkovém stavu systému z hlediska jeho výkonu. Uživatel se dozví, ke kterým potencionálním výkonostním problémům mohlo dojít a jak moc velké riziko představují. Implementací regresní analýzy je možné detekovat výkonostní problémy jako jsou například zhoršující se reakční čas odpovědi, nízká propustnost systému či odhalit únik paměti. Hlavním posláním práce je napomoci vývojářům k jednodušší analýze reportů z výkonostního testování a tak urychlit potřebnou dobu vynaloženou právě na tuto, často opomíjenou, část životního cyklu software. Implementovaný mechanismus obohatí report o vyhodnocení stability systému - jeho stavu z hlediska výkonu.

Klíčová slova: výkonostní testování — regresní analýza — detekce anomálií

Příložené materiály: [Downloadable Code](#)

*xkurov00@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Úvod

Na úvod je nutné říci, že testování výkonu není levnou záležitostí, a proto mnoho zejména malých firem jednoduše netestuje jejich aplikace a spoléhá především na zpětnou vazbu od uživatelů. Napjaté termíny, vývoj do posledního momentu před uvedením do produkce či akceptačním testováním. Firmy si pak často uvědomí, že na testování a analýzu výsledků prostě nemají dostatek času a nebo prostředků. To vše má za následek opomíjení právě té části vývojového cyklu software, která se zabývá testováním a ověřováním jeho kvality. Nicméně výkonostní testování dokáže upozornit právě na to, co je ještě třeba zlepšit před uvedením produktu na trh. U aplikací disponujících nízkým výkonem z důvodu špatného nebo žádného testování je velmi pravděpodobné, že si už od začátku pokazí reputaci a nesplní očekávané prodejní cíle. Na druhou stranu dokáže odhalit i řadu problémů, které se vyskytnou až po delší době používání aplikace - příkladem je postupná degradace sledovaných metrik jako čas odpovědi či propustnost. Co je důležitější,

výkonostní testování software je mimořádně významné pro mnoho velkých průmyslových projektů a nezbytné v kritických aplikacích, jako jsou například kosmické programy nebo záchranné zdravotnické zařízení. Zde je nutné zajistit, aby fungovali správně po dlouhou dobu a bez odchylek.

Jedním z hlavních indikátorů při výběru testovacího nástroje je způsob, jakým nám prezentuje své výsledky. Právě tento výsledný report je hlavním prostředkem k odhalení výkonostních problémů. Vygenerovaný report pomáhá vývojářům odhalovat podezřelé chování systému a navíc je to právě to, co chce management vidět, když přijde na posuzování kvality software. Některé nástroje nabízí pouze textové logy a jednoduchý přehled, další vykreslí z výsledků graf nebo dokonce poskytují možnost exportovat výsledky například do tabulkového procesoru, kde je vhodné místo po porovnávání jednotlivých běhů testu.

Námi vyvinutá komponenta poskytuje mechanismy, které urychlí a zjednoduší analýzu výsledného reportu z výkonostního testování. Úkolem je celková au-

tomatizace procesu zpracování dat – výsledků naměřených během testu, rozeznání podezřelých hodnot a z nich identifikace často se vyskytujících výkonnostních problémů. Je tak provedeno využitím statistických metod jako je lineární regresní analýza, pomocí které je možné detekovat výkonnostní problémy jako například zhoršující se reakční čas odpovědi a nízká propustnost systému či odhalit únik paměti. Ve finále dojde k vygenerování reportu z testování s přehledem vykreslených grafů z jednotlivých testů a vyhodnocení celkového stavu systému z hlediska výkonu.

2. Podobné práce

Algoritmů vycházejících ze statistických metod existuje velké množství. Je to dáno také tím, že se tyto metody využívají v širokém spektru oborů. Práce [?] pojednává o využití korelační a regresní analýzy k detekci bezpečnostních incidentů v rámci Internetu věcí. Podobně jako u výkonnostního testování jsou pro nás nejzajímavější odchylky od běžného předpokládaného scénáře, je tomu tak stejně i u analýzy síťového provozu s cílem detekovat anomálie a odhalit tak útok [?] [?]. Regresní analýza taktéž představuje významnou metodu v oblasti sociologického výzkumu.

3. Hledání informace v datech

Data mining – neboli dolování dat – je soubor metod sloužících ke zpracování různých dat bez ohledu na obor a původ a získání netriviálních informací, které jsou v nich obsažené. Dolování dat je v dnešní době důležitý marketingový nástroj a zcela určitě nejrychleji rostoucím segmentem business intelligence. S jeho pomocí se snažíme z ukládaných dat získat složitější a užitečnější informace než jen grafy a základní přehledy. Příkladem může být vytváření statistických modelů.

Ze statistického úhlu pohledu se jedná o hledání korelací, tedy vyšetřování vzájemných vztahů nebo vzorů v datech. Smyslem je tedy analyzovat datové závislosti, určit trendy, a pokud to typ dat umožňuje, předpovědět budoucí vývoj. Využívají se nejrůznější metody od jednoduchých tabelací a vizualizací až po sofistikované přístupy jako je genetické programování. Asi nejčastěji používanými metodami však jsou rozhodovací stromy, asociační pravidla, neuronové sítě, regresní a shluková analýza.

3.1 Regresní analýza

Metody regresní analýzy jsou využívány v situacích, kdy nás zajímá závislost jisté náhodné veličiny (závislé proměnné) na jedné nebo více nezávislých proměnných, tzv. regresorech. Předem je dáno, která proměnná je nezávislá a která je závislá. Cílem regresní

analýzy je popsat tuto závislost pomocí vhodného matematického modelu.

Podle počtu nezávislých proměnných rozlišujeme modely jednoduché regrese a vícenásobné regrese. Jednoduchá regrese popisuje závislost vysvětlované proměnné na jednom regresoru. Naproti tomu vícenásobná regrese řeší situaci, kde závisle proměnná závisí na více než jednom regresoru. Tato práce se zabývá jednoduchou regresí, kdy je vysvětlující proměnná závislá pouze na jednom regresoru. Závislost je zde přímková a tedy vztah obou veličin lineární. Lze sestavit regresní model.

$$y = b_0 + b_1x + e_i \quad (1)$$

Kde y (závislá proměnná) bude měřená veličina a x (nezávislá proměnná - regresor). Parametr, který určuje polohu přímky, zde značíme jako b_0 , směrnici přímky jako b_1 . e_i reprezentuje náhodnou chybu modelu.

3.1.1 Metoda nejmenších čtverců

Metoda nejmenších čtverců je metoda pro odhad parametrů regresní funkce a to pouze pro takové modely, které jsou lineární v parametrech. Podstatou je tedy aproximace zadaných (naměřených) hodnot nějakou funkcí z předepsaného prostoru. Nejjednodušším příkladem je proložení (aproximace) dat přímkou, tedy lineární funkcí. Cílem je nalezení takové přímky, aby součet druhých mocnin chyb e_i byl minimální. Znamená to, že parametry funkce $y'_i = b_0 + b_1x_i$ (hodnoty b_0 a b_1) hledáme tak, aby součet čtverců odchylek pozorovaných hodnot Y_i od hodnot \bar{Y}_i S_{rez} byl co nejmenší. Pro danou regresní funkci tento součet nazýváme reziduální součet čtverců.

$$S_{rez} = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - b_0 - b_1x_i)^2 \quad (2)$$

Z podmínky minimálnosti čtverců jsou vyvozeny normální rovnice, ze kterých se jejich řešením vypočtou neznámé parametry b_0 a b_1 .

$$b_0 = \bar{y} - b_1\bar{x} \quad (3)$$

$$b_1 = \frac{S_{xy}}{S_{xx}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (4)$$

3.2 Korelační analýza

Korelační analýza slouží k vyjádření síly závislosti, těsnosti dvou a více numerických proměnných. Odpovídá na otázku: Jak silná je závislost mezi proměnnými, respektive jak moc odpovídá model skutečnosti?

3.2.1 Pearsonův korelační koeficient r

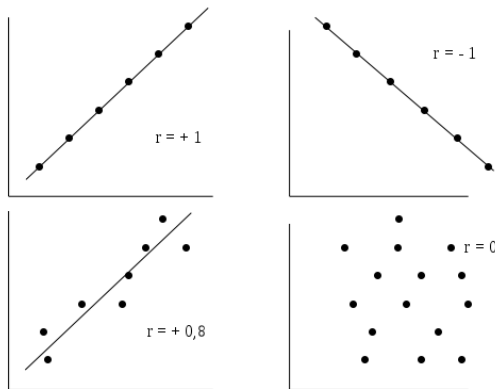
Nejčastěji se pro měření závislosti používá **Pearsonův korelační koeficient** r , který měří lineární závislost dvou náhodných veličin x a y s dvourozměrným normálním rozdělením.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (5)$$

kde:

- x : je x-ová souřadnice datového bodu (nezávislá proměnná)
- \bar{x} : je průměrná hodnota x-ových hodnot
- y : je y-ová souřadnice datového bodu (závislá proměnná)
- \bar{y} : je průměrná hodnota y-ových hodnot
- n : je počet hodnot

Nabývá hodnot od -1 do 1 , které značí perfektní lineární vztah (záporný nebo kladný), jak je znázorněno na následujícím obrázku:



V případě kladné korelace $r > 0$ hodnoty obou proměnných zároveň stoupají. V případě záporné korelace $r < 0$ hodnota jedné proměnné stoupá a druhé klesá. V případě neexistence lineárního vztahu je $r = 0$.

3.2.2 Koeficient determinace R^2

Druhá mocnina korelačního koeficientu R^2 se nazývá **koeficient determinace**. Vyjadřuje podíl, jakým je rozptyl závisle proměnné veličiny vysvětlen změnami nezávisle proměnné. Obvykle se násobí stem, čímž je tento podíl vyjádřen v procentech. Koeficientem determinace rozumíme veličinu

$$R^2 = 1 - \frac{S_{rez}}{S_{yy}} \quad (6)$$

kde:

- S_{yy} : je celkový součet čtverců odchylek dat od průměru.

$$S_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (7)$$

V modelu lineární regrese s absolutním členem leží hodnota R^2 v intervalu $< 0, 1 >$ a udává, jaký podíl rozptylu v pozorování závislé proměnné se podařilo regresí vysvětlit. Udává shodu modelu s daty. Větší hodnoty znamenají větší úspěšnost regrese.

3.3 Základní statistické testy v regresním modelu

Statistická hypotéza je předpoklad o hodnotě neznámého parametru nebo o zákonu rozdělení sledované veličiny. U modelů lineární regrese můžeme testovat buď celý model (pomocí F testu) nebo vliv jednotlivých prediktorů (t test). Nejjednodušší nulovou hypotézou je rovnost některého z regresních koeficientů nule — $H_0 : b_i = 0$. Testům významnosti koeficientů vyrovnávací přímky říkáme dílčí t-testy.

$$t = \frac{\text{odhad hodnoty parametru}}{\text{střední chyba odhadu parametru}} \quad (8)$$

Testujeme nulovou hypotézu, že model nic nevysvětluje (proměnné jsou nezávislé). Je-li parametr nevýznamný, potom platí že $b_i = 0$.

$$t_i = \frac{b_i}{S_{b_i}} \quad (9)$$

Hodnotu t_i v absolutní hodnotě porovnáme s $t_{krit}(1 - \alpha/2)$ Studentova rozdělení pro $(n-m)$ stupňů volnosti, kde n je počet hodnota a m je počet parametrů b_i .

$$t_i = \frac{b_i}{S_{b_i}} T(n - m) \quad (10)$$

Číslo α se nazývá hladina statistické významnosti testu. Určuje pravděpodobnost, že testovací charakteristika padne mimo obor přijetí. Obvykle nabývá hodnot od $0,001$ do $0,3$ v závislosti na povaze zkoumaného problému (doporučované hodnota je $0,05$). V případě, že získaná hodnota je menší než $0,05$ - zamítáme H_0 , tzn. koeficient b_i z modelu vypustit nemůžeme.

4. Detekce anomálních profilů

Detekce anomálií spočívá v definování normálního chování pomocí množiny vybraných proměnných, které jsou poté porovnávány s nově naměřenými. Pokud jsou výrazně odlišné od očekávaných hodnot, je hlášena anomálie. U výkonnostních problémů je cílem sledovat funkční závislost měřené metriky na čase a tak identifikovat její odchylky vůči normálnímu chování.

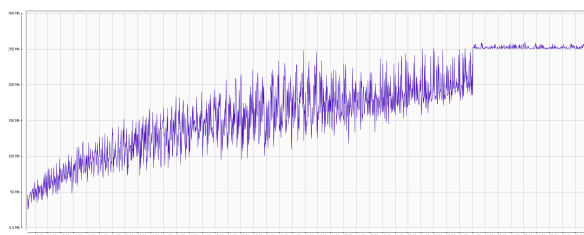
Zaměřujeme se na statistické metody, jejichž výhodou je schopnost učit se přímo z pozorovaných dat, a tudíž není nutné procházet počáteční trénovací fází, jako je tak u náročnějších metod založených například na strojovém učení. Jedná se tedy hlavně o různé přístupy analýzy trendu v grafu.

4.1 Výkonnostní problémy v grafech

Z průzkumu často se vyskytujících výkonnostních problémů v serverových aplikacích jsme vybrali takové, které se vyznačují projevy specifických vzorů a u kterých budeme zřejmě schopni identifikovat odchylky lišící se od normálního chování.

4.1.1 Únik paměti

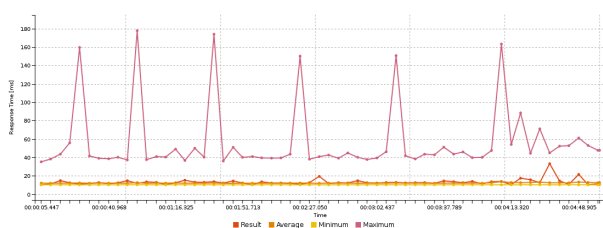
Zdrojem problému s výkonem může být případ, kdy nedochází k uvolňování již nevyužívané paměti. Únik paměti nastává vinou chyby v programu, ať už se jedná o službu napsanou v jazyce s automatickou správou paměti či nikoliv. Jako příklad nám poslouží graf na obrázku 1 znázorňující postupné narůstající množství využití paměti.



Obrázek 1. Využití paměti

4.1.2 Výkyvy v času odpovědi

Pravidelné výkyvy v času odpovědi mohou být způsobené činností garbage collectoru (GC). Není to takový problém, pokud GC běží pouze po dobu pár sekund, například každou minutu, jako je typické pro aplikace, které často ukládají data z databází (disku) do JVM (paměti). Problém nastává, když GC trvá déle než jen několik vteřin. V tomto případě se už konečnému uživateli může zdát aplikace pomalá. Příkladem může být graf na obrázku 2, kdy čas odpovědi dosahuje v určitých momentech až 2 vteřin.

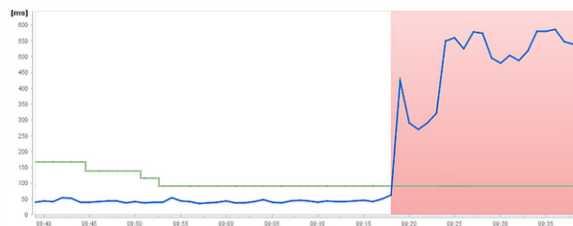


Obrázek 2. Výkyvy v času odpovědi

4.1.3 "Provozní špička"

Dalším příkladem, který má přímý vliv na výkon systému, je náhlá degradace dané metriky - ať už se jedná o čas odpovědi, propustnost nebo využití zdrojů, způsobená velkým počtem uživatelů v provozních špičkách. Příkladem je znázornění času odezvy systému na obrázku 3.

Existují ale i velmi specifické případy, kdy nám korelace proměnných napříč celého záznamu nic ne-



Obrázek 3. Degradace času odezvy

napoví a je nutné zkoumaný rozsah zmenšit na několik málo posledních vzorků a nebo zvážit použití sofistikovanějších metod, zmíněných v závěrečné kapitole 6.

4.2 Identifikace výkonnostních problémů s využitím regresní a korelační analýzy

Analýza výkonu aplikace v kontextu regresní analýzy představuje zkoumání závislosti dané výkonnostní metriky (času odezvy, propustnosti, množství spotřebovaného zdroje) na čase — tedy hlavně to, jak se naměřené hodnoty v průběhu času mění. To ovlivňuje vztah závislosti mezi těmito dvěma hodnotami, který lze definovat řadou veličin pomocí korelační analýzy. Do vzorce regresní analýzy bychom tedy za závislou proměnnou dosadili zkoumanou výkonnostní metriku a za nezávislou proměnnou čas.

4.2.1 Sklon regresní přímky

Z grafu na obrázku 1 se dá vyčíst jakási růstová tendence přímky odhadnuté regresní funkce znázorňující množství využití paměti vzhledem k času provádění testu. Z matematického hlediska bychom tuto tendenci, míru těsnosti závislosti dvou proměnných kvantifikovali parametrem regresní funkce b_1 , jehož odhad dokážeme získat podle vzorce 4 metodou nejmenších čtverců. Parametr b_1 zde představuje směrnici — sklon odhadované regresní přímky, a tudíž lze podle ní odhadnout, zda dochází k růstové tendenci v grafu. Na základě znalosti tohoto parametru lze tedy odhalit degradaci měřené výkonnostní metriky, ať už chvilkové, zapříčiněné přetíženým provozem či z dlouhodobého hlediska.

4.2.2 Úroveň významnosti korelace sklonu

Podmínky lineárního regresního modelu je nutno v rámci regresní analýzy ověřit. Využijeme k tomu základní statistické testy v regresním modelu podle kapitoly 3.3. Existenci lineárního vztahu mezi dvěma veličinami zjišťujeme tak, že se formálně ptáme, zda je směrnice b_1 rovna nule. Pokud je odpověď na tuto otázku kladná, znamená to, že směrnice regresní přímky se liší od nuly pouze náhodně, tzn., že vztah mezi sledovanými veličinami není lineární. Když se opět přesuneme do kontextu detekce výkonnostních problémů, můžeme usuzovat, že pokud vztah mezi sledovanými veličinami není lineární, vyskytují se v sadě namě-

řených výsledků hodnoty, které jsou výrazně odlišné od průměru a představovat například podstatné výkyvy, jako je znázorněno na obrázku 2. Tímto způsobem tedy identifikujeme potenciální anomální profil, který může být podle závažnosti označen za výkonnostní problém.

4.2.3 Kvalitu regresního modelu

Kvalitu regresního modelu můžeme hodnotit pomocí koeficientu determinace R^2 . Jak je vysvětleno v kapitole 3.2.2, koeficient determinace udává, kolik procent rozptylu vysvětlované proměnné je vysvětleno modelem. V praxi to znamená, jak moc jsou jednotlivé sledované hodnoty odlišné od střední hodnoty. Tento poznatek využijeme už jen k potvrzení předchozích hypotéz o výskytu anomálního profilu.

5. Využití v praxi

Výše popsané přístupy lze využít pro detekci anomálních profilů ve výsledcích provedeného výkonnostního testu.

5.1 Nová komponenta v nástroji PerfCake

Vznikla nová komponenta typu `Destination` v reportovací části open-source nástroje pro výkonnostní testování a generování zátěže **PerfCake**¹. Vyvinutá komponenta po skončení testu provádí analýzu naměřených výsledků pomocí statistických metod a na závěr vygeneruje souhrnný report, ve kterém je vyhodnocen celkový stav systému vzhledem k provedeným testům definovaným v testovacím scénáři.

5.1.1 PerfCake

Nástroj PerfCake poskytuje provádění testů na propustnost systému, čas odezvy, dokáže monitorovat náročnost inicializace aplikace či paměť za účelem detekce úniku paměti. K tomu také byly přizpůsobeny algoritmy detekující anomálie právě v těchto metrikách, které nástroj dokáže měřit.

5.1.2 Detaily implementace

Byla implementována jednoduchá regresní analýza pomocí knihovny Apache Commons Math² a pravidla pro identifikaci anomálních profilů definována v kapitole 4.2.

¹<http://www.perfcake.org>

²<http://commons.apache.org/math>

Algorithm 1 Detekce anomálních profilů

Data: regresní model

Result: výkonnostní problém/normální profil

if hodnota směrnice regresní přímky je výrazně odlišná od nuly **then**

| detekována degradace testované metriky

else

| **if** hypotéza o významnosti koeficientů regresní přímky je zamítnuta **then**

| | detekovány výkyvy v profilu

| **else**

| | normální profil

| **end**

end

Reportovací část, která zkompileje všechny výsledky do jednoho souhrnného reportu a obohatí jej sdělením, ke kterým potenciálním problémům mohlo v systému dojít a tipy čím mohly být způsobené, je nyní v procesu vývoje a bude k ní využita knihovna C3.js³

5.2 Simulace výkonnostních problémů

K simulaci anomálních profilů pro účely testování a optimalizace navržených algoritmů je využíváno několik služeb. Jedou z nich je open-source služba **Weaver**⁴, která komunikuje s okolním světem přes HTTP. Je tak možné spustit současně několik vláken — označovány jako `Workers`, které zpracovávají požadavky na základě různé konfigurace jejich chování a tak napodobit negativní testovací scénáře. Například čas odezvy s daným zpožděním, simulace úniku paměti, přepínání mezi dvěma různými konfiguracemi, ale i simulaci normálního chování. Dalším jsou webové stránky `mocky.io`⁵ a `httpbin.org`⁶ pro HTTP "Request & Response" simulaci služeb fungující na podobném principu.

5.3 Průběh experimentování

Na základě znalosti výkonnostních problémů a schopnosti rozpoznání jejich charakteristických profilů z grafu naměřených hodnot byly provedeny experimenty s vyvinutou metodou pro detekci anomálních profilů za účelem zvýšení její přesnosti.

Byly vytvořeny jednotlivé konfigurace služby **Weaver** s vlastnostmi charakteristickými pro zkoumané výkonnostní problémy a adekvátní konfigurace testovacího scénáře pro testování této metriky. Z této fáze byly získané grafy použité v kapitole 4.1.

³<http://c3js.org/>

⁴<http://github.com/PerfCake/Weaver>

⁵<http://www.mocky.io/>

⁶<http://httpbin.org/>

Poté bylo provedeno spuštění služby pro simulaci serverové aplikace s konfigurací dané PI (s takovou konfigurací, aby generovala hodnoty charakteristické pro daný anomální profil) a její otestování nástrojem PerfCake s daným testovacím scénářem. Po skončení testování a získání všech naměřených hodnot byly provedeny nad výsledky heuristiky námi vytvořenou komponentou a shromážděny výsledky korelací těchto hodnot.

Následovalo zkoumání výsledků korelačních funkcí a hledání souvislostí mezi těmito hodnotami a projevenými PI. Na základě pár vzorových případů byla vytvořena heuristická pravidla pro rozpoznání dané PI zmíněná v kapitole 4.2 a následně byla tato pravidla testována na větším vzorku dat a postupně vylepšována.

6. Shrnutí/Závěr, rozšíření práce

V tomto článku pojednáváme o detekci podezřelých hodnot ve výsledcích výkonnostního testování na základě regresní analýzy, která spadá do množiny statistických metod.

V budoucnu je možné práci rozšířit použitím dalších metod založených na dolování dat či strojovém učení. V případě strojového učení nemusíme chápat podstatu procesů generující data, protože tyto metody jsou založeny na postupném učení se a vylepšování své výkonnosti pro danou úlohu na základě předchozích výsledků. Tuto metodu by tedy bylo vhodné použít pro případy dlouhodobého testování software z hlediska jeho výkonu a tím pádem odhalovat i jiné typy výkonnostních problémů, především postupnou degradaci jednotlivých měřených metrik systému.

V oblasti dolování dat jsou vhodné metody založené na klasifikaci, které rozdělují vstupní data do několika tříd na základě sady pravidel, vzorů nebo podobných technik. V případě detekce anomálií se obvykle jedná o binární klasifikaci, tedy rozdělení pouze do dvou tříd - normální data a anomálie. Další možností je shluková analýza. Shluková analýza představuje souhrnný název pro metody vyhledávání shluků v nepopsaných datech o mnoha dimenzích. Hlavní výhoda této metody je její schopnost učit se z dat a vyhledávat v nich anomálie bez toho, abychom museli poskytnout popis různých typů těchto anomálií. Množství trénovacích dat, které je nutno systému dodat, je také menší, než u jiných metod. Výsledkem shlukové analýzy je definování tzv. Outliers, které jsou z pohledu této metody objekty, které nenáleží do žádného shluku, a v kontextu detekce anomálií tedy pravděpodobně reprezentují výkonnostní problémy.