

## Introduktion

Idag används centraliserade datacenter och cloud computing för insamling av data från överallt i världen, lagring av denna data och analys av datan. Detta tillvägagångssätt stöder dock inte den ökade distributionen av data från "edge devices" (enheter vid nätverkets utkant), då cloud computing vilket dessa enheter stöder sig på i dagsläget är begränsat av uppkopplingens bandbredd. Att flytta beräkningarna till molnet har tidigare varit en effektiv lösning men på grund av ökad användning av tidigare nämnda edge devices vilka över tid har börjat generera större mängder data. Detta i och med att mer och mer kraftfulla mikrodatorer utvecklats samt att dessa finns tillgängliga för ett överkomligt pris hos de flesta elektronikåterförsäljare. Baserat på dessa nya krav har en ny metod utvecklats, vilken är "edge computing" (när dataanalys sker vid ett nätverksutkant). Edge Computing syftar till att flytta tunga beräkningar från ett centraliserat datacenter till en edge device eller ett mikro-datacenter alternativt en cloudlet. Fördelen med detta är att cloud computing's flaskhals undviks då den data som skall skickas redan har analyserats och därmed kommer antalet bytes som skickas bli betydligt mindre, då den analyserade datan endast kommer innehålla exempelvis en prediktion av något slag vilken kan bestå av endast ett tal, jämfört med datasetet som denna analys baserats på vilket kan bestå utav flera miljoner rader av observationer. (Shi & Dusdar 2016)

Företaget Ride My Bike är intresserade av att undersöka om det är möjligt att konstruera en intelligent besöksräknare, och samarbetar i detta syfte med studenter vid högskolan i Borås.

Inom fältet besöksräknare finns redan smarta lösningar vilka agerar värd för en mängd funktioner, vilket bland annat innefattar att redovisa hur många som kommer befinna sig på en plats inom ett visst tidsfönster (tempwatch.se 2016) och redovisning av besökarnas rörelsemönster (viameetrics.com/ 2016). Denna uppsats kommer undersöka om det är möjligt att ta det ännu ett steg längre och göra besöksräknaren intelligent. Det som åsyftas med intelligent är att besöksvolymen för en given tidpunkt kan prediceras direkt i besöksräknaren, utan att skicka data till en extern entitet för dataanalys. Detta möjliggörs med hjälp av välkända maskininlärningsalgoritmer, av vilka random-forest classifiers (Berkley University, 2016) kommer vara mest central. Genom att övervaka besöksvolymen över tid och diverse miljörelaterade faktorer, skall denna prototyp sedan predicera hur många besökare som kommer befinna sig på platsen en timme från mättillfället.

Mikrodatorer är chip i kreditkortsstorlek men ändå levererar hög prestanda. De leverantörer av dessa som funnits lämpliga var Raspberry Pi (Raspberry Pi Foundation, 2016) och Arduino (Arduino, 2016). Av dessa var dock Raspberry Pi/Arduino bättre lämpad för kraven som detta utvecklingsprojekt ställer. De kriterier som undersöktes var följande och resultaten finns i nedanstående tabell. Plattformen som valts för den mjukvara som kommer utföra prediktionerna är Raspberry Pi. Detta val har baserats på att Raspberry Pi kan köra en mängd olika operativsystem och detta utan tilläggskort, Raspberry Pi korten har även hos majoriteten av modellerna mer ram minne än konkurrenterna. En buss är ett antal ledningar som används för kommunikation med sensorer och används i olika former i tabellen nedan.

Model I	RASPBERRY PI 3 MODEL B <a href="https://www.raspberrypi.org/products/raspberry-pi-3-model-b/">https://www.raspberrypi.org/products/raspberry-pi-3-model-b/</a>	Genuino Yún Shield ( Tilläggskort med kompabilitet för övriga Genuino kort ) <a href="https://www.arduino.cc/en/Main/Arduino">https://www.arduino</a>	Genuino Zero <a href="https://www.arduino.cc/en/Main/ArduinoBoardZero">https://www.arduino</a>	Genuino Uno <a href="https://www.arduino.cc/en/Main/ArduinoBoardUno">https://www.arduino</a>
------------	--	---	---	---

		<a href="#">YunShield</a>		
<b>Ram minne</b>	1GB	64 MB	32 KB	2 KB
<b>cpu/mcu</b>	1.2GHz 64-bit quad-core ARMv8 CPU	Atheros AR9331 MIPS 400 mhz	ATSAMD21G18, 32-Bit ARM Cortex M0+ 48 MHz	ATmega328P 16 MHz
<b>Bussar</b>	40 GPIO pins,	SPI, Hardware Serial	Digital I/O: 20st UART: native.programming analog I/O: 6 st 12-bit ADC kanaler, 1, 10-bit DAC	Digital I/O Pins: 14 (6 provide PWM output) Analog input pins: 6
<b>Lagrings möjligheter</b>	Micro SD	16 MB	256 KB flash minne,	32 KB flash minne
<b>Näverksadapter</b>	802.11n Wireless LAN, Ethernet port	Ethernet 802.3 10/100 Mbits/s, Wifi 802.11b/g/n	-	-
<b>Bluetooth</b>	Bluetooth 4.1, BLE(Bluetooth Low Energy)		-	-
<b>Operativ system</b>	Raspbian officiellt os, med möjlighet att köra andra operativsystem vilka installeras på SD kort innan det matas in i kortet samt diverse open-source operativ system.	Levereras med OpenWRT, Möjlighet att installera Linux	-	-

En microcontroller är ett system på ett chip med beräknings- och kommunikationsegenskaper. En sensor är en enhet som kan mäta en viss egenskap hos ett objekt och kommunicera detta till exempelvis en microcontroller. För att göra denna till en besöksräknare behöver den sensorer för att uppfatta när någon anländer till eventet, och en display för att visa hur många besökare som uppfattats. För att lägga till en intelligenta delen, som innebär att predicera hur många besökare som kommer befinna sig på platsen en timme efter observationstillfället. För detta används Random-Forest Classifier fokuset i denna uppsats är inte Random-Forest därför ska istället sci-kit learns ([sci-kit learn, 2016](#)) bibliotek användas vilket innehåller beprövade implementationer av Random-Forest klassificerare.

### Forskningsöversikt

En av de stora utmaningarna med att göra den applikation som gör prediktionerna på mikrodatorn är att det inte finns något generellt ramverk för hur data ska insamlas från sensorerna. Därför kommer möjligheten att utveckla ett sådant ramverk undersökas.

Utöver det så kommer undersökningen redogöra för om det är möjligt eller inte att skapa en AIM (Autonomous Intelligent Microcomputer) med en mikrodator som plattform. Viktiga parametrar kommer undersökas för att utröna vilka förutsättningar som krävs, för att erhålla en fristående mikrodator med tillhörande sensorer och mjukvara som kan göra prediktioner.

En implementation av en Random-Forest klassificerare kommer vi inte utveckla, utan har istället valt att använda ett tredje-parts bibliotek. Detta val har baserats på det faktum att det viktigaste med undersökningen är om det är prestandamässigt möjligt att göra denna typ av prediktion med hjälp av en mikrodator, och inte att undersöka någon aspekt av själva klassificeringsalgoritmen.

Från ett generellt perspektiv kan dataanalys ses som ett paraply som omfamnar forskningsområdet för frågorna som ska besvaras. Datainsamling i någon eller flera former sker innan dataanalysen kan ta vid. Dataanalys innefattar bland annat tolkning av data (Robson 2016, s. 403).

Tolkningen av data – eller snarare klassificering av densamma - kommer ett ramverk för maskininlärning att stå för. Datat levererat av sensorerna, som är kopplade till AIM:en, utgörs till exempel av en signal som sätts hög då någon passerat en rörelsesensor.

Efter dataanalysen får man en produkt som kan tolkas. Analys innebär att bryta upp något stort objekt i mindre delar och förklara hur delarna hänger ihop, samt egenskaperna för delarna (Robson 2016, ss. 405 - 408).

Produkten, som människor lätt kan tolka i detta arbetet, kommer bli i form av en prediktion av till exempel hur många besökare som förväntas komma in till företaget nästföljande timme. Egenskaperna för delarna och hur dessa hänger ihop sker i en algoritm, vilken viktar olika inputs för sensorer och genererar en output i form av en önskad prediktion.

För att kunna analysera datat kan man skapa ett dataset för datorn att analysera. Datasetet kan skapas automatisk med hjälp av en teknisk produkt(exempelvis en maskin som användaren väljer knapp på för ett visst svar – 'a' för ja, 'b' för nej etc.), inläsning av en ifylld papperskopia till en datafil eller mata in datasetet manuellt (Robson 2016, s. 411).

AIM:en kommer att läsa av sensorerna automatiskt eller via kommando och bilda flertalet dataset, som därefter kan bearbetas via maskininlärningsalgoritmerna och generera output. Produkten, som Robson (2016) refererar till ovan, är själva AIM:en utan inblandning av mänsklig intervention i detta fallet.

Det finns flera saker som kan gå snett vid behandlingen av data. Saknad data kan hanteras med hjälp av en markör. Missad data är inte lätt att undvika helt - ofta sker detta då respondenten inte vet vad denne ska fylla i. Datat behöver troligtvis rensas efter inmatning. Man kan låta två personer mata in det fullständiga datat var för sig och se vilka positioner datat skiljer sig på efteråt, samt ta ställning för ett av dem. Outliers brukar vara lätta att se då de står ut i mängden genom att vara kraftigt avvikande punkter i exempelvis ett diagram. De kan då med fördel tas bort från den fortsatta dataanalysen (Robson 2016, s. 413).

Problemen med Outliers och saknad data är högst relevant även för en AIM. En sensor skulle kunna gå sönder och generera output som skiljer sig markant från tidigare avläsningar eller inte generera någon data alls. Problem av allvarigare art borde det bli om en sensor får för sig att generera värden som

systematiskt ligger lågt eller högt, utan att AIM:en upptäcker det som värden som borde ligga utanför sensorns mätområde.

Man kan använda scatter-plots för att visa på 'Deviant points', som utmärker sig genom att inte följa ett visst mönster som de andra punkterna i diagrammet följer. Säg att till exempel de andra punkterna i det tvådimensionella diagrammet ser ut att följa en böjd banan, men en enskild punkt ligger långt ifrån de andra. Då kan man dubbelkolla denna punkt och kanske avgöra om den ska vara med i den fortsatta dataanalysen (Robson 2016, s. 414).

Styrkan med maskininlärningsalgoritmer är deras förmåga att klassificera data. 'Deviant points' borde förstås även kunna vara ett problem här - om punkten ligger långt utanför de andra punkterna, som bildar mönstret. Robsons (2016) metod att dubbelkolla punkter och avgöra om de ska vara med har som styrka att det är människor som avgör detta (troligtvis med hjälp av någon statistisk metod).

Dataanalysen kan delas upp i två grenar - exploratory och confirmatory. Exploratory innebär att undersöka vad datat säger. Ofta använder man denna gren i kvalitativ dataanalys. Undersök datat med hjälp av olika diagram, använd Excel för att ta ut medelvärden och varians etc. Lek med siffrorna och försök se vad de säger i sin enklaste form. Mer avancerade program kan snabbt mata ut obegripligheter om man matar in datat felaktigt och motverka en djupare förståelse för vad datat skulle kunna säga. Confirmatory bekräftar om man fick det som förväntades av datat. Man kan med fördel använda denna gren i kvantitativa analyser. Många kvantitativa dataanalyser söker att verifiera att förutsägelser och hypoteser kan konfirmeras via datat. Confirmatory dataanalys används för det mesta i statistisk analys (Robson 2016, ss. 414 – 415).

AIM:en kommer att arbeta med kvantitativ dataanalys – siffror säger allt och används av maskininlärningsalgoritmerna. Confirmatory borde därmed vara den gren som AIM:en vilar sig på genom att förutsäga hur många som kommer besöka företaget inom den närmaste timmen.

En algoritm för ett artificiellt neuralt nätverk har implementerats i en billig MCU 8051, en Micro Controller Unit av familjen 8051, för att bygga ett autonomt intelligent trådlöst sensornätverk. En online strategi, dvs beräkning på MCU:n, för inlärning användes vid prediceringen av rumstemperaturen, som mättes för ett antal noder, utan att spara historiska data i en databas. Några problem som dock kan uppstå är att prediktionen kan bli osäker då den grundar sig på ett fixt antal mätvärden bakåt i tiden, vilket kan tacklas med att läsa av sensorer för exempelvis ventilation, samt andra källor som påverkar temperaturen i rummet. MCU:n har upp till 4kB RAM och upp till 32 kB flashminne för programmet (Pardo, Zamora-Martinez & Botella-Rocamora 2015).

MCU:n 8051 **verkar arbeta** på 24MHz frekvens enligt Texas Instruments (2016). Den produkt som refereras till i Pardo, Zamora-Martinez och Botella-Rocamora (2015) - CC1110F32 – finns inte att läsa om direkt på den hemsidan, men den familj som produkten tillhör finns att läsa om och där anges 24MHz.

Arbetet som Pardo, Zamora-Martinez och Botella-Rocamora (2015) utfört visar att det är fullt möjligt att predicera en egenskap med hjälp av en microcontroller. Raspberry Pi är mycket mer välutrustad vad gäller arbetsfrekvens och minne, samt tillgång till avancerade algoritmer för predicering. Det är uppmuntrande och visar att detta arbete har förutsättningarna att lyckas.

Wireless Sensor Networks är trådlösa sensornätverk - vilket är lika med en microcontroller plus radio plus sensorer. Många användare av Wireless Sensor Networks vill ofta komma åt data från sensorerna kontinuerligt för senare dataanalys - en kostsam process i form av energiåtgång. Genom att samla sensorerna i kluster, där det finns ett klusterhuvud för en samling sensorer som har till uppgift att predicera sensorvärden, kan energiåtgången reduceras kraftigt. Även sensorerna använder sig av predicering och skickar bara mätvärden till klusterhuvudet, om mätvärdet skiljer sig från en satt felmarginal. Det finns en avvägning mellan kostnaden för kommunikation och prediktionskostnad. För att effektivt avväga dessa kostnader kan man använda sig av ett adaptivt schema för när prediktionsoperationerna ska utföras. Även ett schema för standby/vakna upp kan appliceras på nätverket. Mätvärdena skickas effektivt samlade mellan kluster till kluster och undviker därmed den dyra nod till nod kommunikationen av aggregerade mätvärden (Jiang, Jin & Wang 2011).

Här har Jiang, Jin och Wang (2011) en arkitektur av klustren som påminner om den som vårt arbete inriktat sig på - att prediceringen ligger närmare sensorerna. Enligt Texas Instruments (2016) har dessa sensorer en mycket låg effekt och skiljer sig markant åt jämför med Raspberry Pi:s energiförbrukning. Här ligger en utmaning att försörja Raspberry Pi:n med ström från batterier och få den fristående från elnätet - om det skulle behövas för projektet.

## Problem Diskussion

Ett behov av att flytta dataanalys från centraliserade datacenter, detta då mikrodatorer blir allt kraftfullare och kan hantera större mängder data. Detta visar sig vara ett problem i dagsläget då dagens molnlösningar inte kan tillhandahålla den nödvändiga bandbredden för att stödja smidig överföring av data från edge devices då den data dessa producerar blir allt större lagringsmässigt, och komplex. Ett problem med att utveckla dessa edge devices är dock att det inte finns någon generell metod för att läsa av dessa enheters sensorer och lagra observationerna, på ett sätt som möjliggör djupare analys. Det är därmed intressant att undersöka om det är möjligt att utveckla en generell metod för detta ändamål. I samband med detta vore det även intressant att undersöka om dagens mikrodatorer har tillräcklig hårdvara för att exekvera maskininlärningsalgoritmerna och få en prediktion utan att behöva vänta orimliga tider.

## Syfte och forskningsmål

Syftet är att undersöka möjligheterna med att utföra tunga beräkningar på enheter nära användaren och därmed undvika att sända stora mängder sensordata till en central enhet. De forskningsmål som studien avser att uppfylla är:

1. Utvärdera hur en AIM med maskininlärningsalgoritmer presterar som autonom besöksräknare.
  - a. Skapa en prototyp av en AIM.
  - b. Utvärdera tiden det tar för prediktioner av besöksfrekvens.
2. Ta fram designprinciper för ett generellt ramverk för insamling av sensordata från olika bussar på en mikrocontroller.

- a. Skapa en prototyp för ramverket.
- b. Utvärdering av ramverkets användbarhet.

## Informationsbehov

För att konstruera den kunskapsbas som studien bygger på har främst *Summon* (som är bibliotekets på Högskolan i Borås generella sökverktyg) använts. Där sökningar på nyckelbegrepp såsom "framework database microcontroller sensor" och "Edge Computing". De källor med flest citeringar har ansetts vara mer trovärdiga än källor med färre citeringar och dessa har därför valts som källor till rapporten.

För att välja plattform för prototypen behövde kandidater utses, för att göra detta behövdes information om vilka plattformar som var lämpliga för arbetet. Denna information delgavs till viss del av handledare och uppdragsgivare som föreslog Arduino/Raspberry pi som utgångspunkt. Utifrån detta valdes kandidater från dessa två leverantörer som sedan ställdes mot varandra. Den information som behövdes för detta hittades på respektive leverantörs hemsida.

För att utveckla prototypen fanns ytterligare behov av information, delvis för teknisk förståelse av mikrodatorn, och delvis för utvecklingen av prediktionsmotorn.

## Design av studien

Design Science innebär att besvara frågor som människor har angående ett problem via design av en artefakt. Denna artefakt ska bidra till ny kunskap som kan adderas till kunskapskroppen inom vetenskapen. För att förstå problemet ska den designade artefakten vara både användbar och grundläggande. Artefakten är central inom design science och kunskap ska erhållas under konstruktionen av denna artefakt. Man kan också förbättra redan konstruerade lösningar och bygga vidare på dessa eller skapa en helt unik lösning för ett viktigt problem. Resurserna som står till förfogande är problemen, möjligheterna och de tekniska systemen inom domänen. Designcykeln inom design science har två steg som utförs iterativt: byggande av artefakten samt evalueringen av densamma. Tidigare forskning samt resultat från referensdiscipliner bygger de grundläggande teorierna och ger stringens till forskningen. Resultatet vid användandet av design science är en artefakt med demonstrerad nytta (Recker 2013, ss. 106 - 108).

De nyckelkriterier som ska vara uppfyllda vid användandet av design science är (Recker 2013, ss. 106 - 108):

- Den demonstrerade nyttan för en artefakt ska vara en nyhet.
- En positiv differens för bidraget av nyttan för en artefakt i jämförelse med existerande arbeten.
- En genomgripande utvärdering som bidrar med avgörande bevis för överlägsen nytta av en artefakt.

Riktlinjerna för design science som detta arbete lutade sig tillbaka på, men kanske inte följdes till punkt och pricka, var de sju som Hevner, March, Park & Ram (2004) beskrev i sitt arbete:

1. Design som en artefakt: Design-science forskning måste producera en genomförbar artefakt i form av en konstruktion, en modell, en metod eller en instantiering.
2. Problemrelevans: Målet med design-science forskning är att utveckla lösningar baserat på teknologi som löser viktiga och relevanta affärsproblem.
3. Designutvärdering: Nyttan, kvaliteten och effektiviteten för en designartefakt måste stringent demonstreras via väl genomförda utvärderingsmetoder.
4. Forskningsbidrag: Effektiv design-science forskning måste leda till klara och verifierbara bidrag inom området för designartefakten, grunderna för designen och/eller metoderna för designen.
5. Forskningsstringens: Design-science forskning vilar på appliceringen av stringenta metoder för både konstruktionen samt evalueringen av designartefakten.
6. Design som en sökprocess: Letandet efter en effektiv artefakt kräver att använda tillgängliga medel för att uppnå önskat resultat samtidigt som lagarna inom problemmiljön följs.
7. Kommunikeringen av forskningsresultat: Design-science forskning måste kommuniceras effektivt både till teknologiorienterad så väl som till ledningsorienterad publik.

Design science var den metod som användes och verkade mest passande för denna uppsats - även om delar av andra metoder skulle kunna tas in under arbetets gång och få en mixad strategi.

En enkätundersökning har även utförts för att utvärdera användbarheten av ramverket som utvecklats. Enkäten innehöll frågor angående användarnas upplevelse under användandet av ramverket. Denna gjordes sedan tillgänglig på Raspberry pi's/Arduino's online forum. Detta gjordes då det ansågs vara det rimligaste sättet inom arbetets ramar att distribuera enkäten till personer som har tillgång till en Raspberry pi/Arduino enhet.

För att uppnå de forskningsmål denna rapport strävar mot har design science använts. Den grundläggande principen inom design science är att förståelse för design problem och deras lösningar tillägnas genom iterativ utveckling och användning av artefakter (Recker, 2013, ss. 106 - 108).

Design science har en central design cykel, som innebär iteration mellan design och utvärdering.(Recker, 2013, ss. 106 - 108). Vilket för denna rapport är ett naturligt tillvägagångssätt då alla specifikationer för prototypen inte fanns tillgängliga vid utvecklingens start.

De test av prototypen som utförts har varit montering av prototypen i entrén hos Tugg i Borås, vi har valt denna typ av miljö för att simulera ett scenario där en besöksräknare skulle kunna påträffas. Just denna plats har valts då handledande lärare har haft tidigare kontakt med ägaren av Tugg. Därigenom har tillåtelse givits att testa prototypen i Tuggs lokaler.

Empiri har samlats in under de testkörningar av prototypen som utförts hos Tugg. Den data som prototypen producerar är en datarad för varje besökare hos Tugg samt relevanta miljörelaterade attribut som läses av från sensorerna på mikrodatorn när en besökare passerat entrén.

Den data som samlas in klassificeras direkt av random forest implementationen på kortet. Detta presenteras sedan som en prediktion av besökarantalet en timme efter mättillfället.

Kvalitativa eller kvantitativa studier skulle kunna vara möjliga att använda tillsammans med design science, för att erhålla data för artefaktens användbarhet. En artefakt som inte är användbar, men har

uppenbar nytta, behöver antagligen designas om så att även användbarheten blir bättre. Användbarheten skulle kunna omfatta att det är enkelt att använda ramverket för sensorinsamling och användarna förstår sig på de funktioner som finns tillhanda inom ramverket. Att enbart använda sig av en kvantitativ eller kvalitativ studie för att ta fram prototypen skulle troligtvis gå, men då skulle de tillfrågade få undersöka prototypen i varje iteration och svara på frågor om den. Det låter som det kan bli ett klumpigt arbetssätt eller i varje fall ett utdraget sätt att samla in data på.

## **Metodreflektion**

Nackdelar med design science är att val av vilken nytta som är lämplig att utveckla är vagt definierat i design science. Utvärdering av denna nytta är också upp till de som designat artefakten och hur det ska gå till. En fallstudie eller experiment kan vara lämpligt att utföra för utvärdering av artefakten (Recker 2013, s.108).

Denna studie använde sig av en fallstudie för utvärderingen av prototypen i form av inriktningen mot att predicera besökarantal vid restaurangen Tugg. Själva prediceringen kan vara nyttan för slutkunden som ska använda sig av mikrodatorn. Men artefakten skulle lika gärna kunnat predicera när ett vindkraftverk är på väg att haverera, via avläsning av vibrationsgrad och temperatur på rullager – som kanske skulle kunna vara tänkbara parametrar som indikerar när rullager är på väg att rasa, även om de inte är verifierade med någon källa. Metoden att predicera med random forest gör att man skulle kunna få mikrodatorn att lära sig när ett lagerras är på gång. Fast det blir ändå först när lagerraset har skett, som man kan klassificera sensordatat som indikation på felet. Så utvärderingen av prototypen kan bli svår beroende på vad man vill predicera. Tidshorisonten för att få med tillräckligt antal prediceringar och inläring av sensordata beror på hur ofta man vill predicera samt hur ofta man kan läsa av sensorerna. Besöksräknaren borde generera flera dataset per dag kring lunchtid och således bli bättre och bättre på att förutspå besökarantalet följande timme. Det verkar rimligt att få tillräckligt med sensordata för besöksräknaren för att nyttan med den ska kunna påvisas för slutkunden.

Fördelar med design science är friheten att själva med uppdragsgivarens problem, välja hur problemet ska tacklas. Att inkrementellt designa en implementation av prototypen och utvärdera denna i varje cykel, gör att man tidigt får feedback på hur nyttan skulle kunna upplevas av användarna.

För att försäkra studiens reliabilitet samt validitet, har "Design Science in Information Systems Research" av Hevner, March, Park & Ram (2004) och "Scientific research in information systems: a beginner's guide" (Recker, 2013) använts för studiens utformning.





## Källförteckning

Arduino (2016). <https://www.arduino.cc/> [29-11-2016]

Berkley University (2016). *Random Forests*.

[https://www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm](https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm) [02-12-16]

Hevner, A.R., March, S.T., Park, J. & Ram, S. (2004). *Design Science in Information Systems Research*. MIS Quarterly, vol. 28, no. 1, ss. 75-105.

Jiang, H., Jin, S. & Wang, C. (2011), *Prediction or Not? An Energy-Efficient Framework for Clustering-Based Data Collection in Wireless Sensor Networks*. IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 6, ss. 1064-1071.

Pardo, J., Zamora-Martinez, F. & Botella-Rocamora, P. (2015). *Online Learning Algorithm for Time Series Forecasting Suitable for Low Cost Wireless Sensor Networks Nodes*. SENSORS, vol. 15, no. 4, ss. 9277-9304.

Raspberry Pi Foundation (2016). <https://www.raspberrypi.org/> [29-11-2016]

Recker, J. (2013). *Scientific research in information systems: a beginner's guide*. Springer, New York;Berlin.

Robson, C. & McCartan, K. (2016). *Real world research: a resource for users of social research methods in applied settings*. Fourth edition. Hoboken: Wiley.

Sci-kit Learn(2016). <http://scikit-learn.org/stable/index.html> [02-12-2016]

Shi, W. & Dusdar, S. (2016). *The Promise of Edge Computing*. IEEE Computer vol:49, no. 5, ss. 0018-9186.

TempWatch (2016). <http://www.tempwatch.se/en/> [28-11-2016]

Texas Instruments (2016). *CC1110-CC1111 Sub-1 GHz wireless MCU with up to 32 kB Flash memory*. <http://www.ti.com/product/cc1110-cc1111> [01-12-16]

Viametrics AB (2016). <http://viametrics.com/sv/besoksrakning/kopcentrum/> [28-11-16]