**Definition**

AIM: Autonomous Intelligent Microcomputer – myntat av uppdragsgivaren

IoT: Internet of Things.(ref)

Edge Computing: När dataanalys sker vid ett nätverksutkant.(ref)

Edge Device: enhet vid nätverkets utkant

MCU 8051: Micro Controller Unit av familjen 8051

Wireless Sensor Networks: Trådlösa sensornätverk = microcontroller + radio + sensorer

Microcontroller: ett system på ett chip med beräknings- och kommunikationsegenskaper

Buss: ett antal ledningar som används för kommunikation med sensorer

Sensor: en enhet som kan mäta en viss egenskap hos ett objekt och kommunicera detta till exempelvis en microcontroller

**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, 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 flest elektronikåterförsäljare. Baserat på dessa nya krav har en ny metod utvecklats, vilken är edge computing. 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.(<http://ieeexplore.ieee.org.lib.costello.pub.hb.se/document/7469991/?part=1> 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 besökräknare 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(viametrics.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 (https://www.stat.berkeley.edu/~breiman/RandomForests/cc\_home.htm 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(www.raspberrypi.org/ 2016) och Arduino(www.arduino.cc/ 2016). Av dessa var dock Raspberry Pi/Arduino bättra 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,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modell | RASPBERRY PI 3 MODEL B <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> | Genuino Yún Shield ( Tilläggskort med kompabilitet för övriga Genuino kort ) <https://www.arduino.cc/en/Main/ArduinoYunShield> | Genuino Zero <https://www.arduino.cc/en/Main/ArduinoBoardZero> | Genuino Uno <https://www.arduino.cc/en/Main/ArduinoBoardUno> |
| Ram minne | 1GB | 64 MB | 32 KB | 2 KB |
| cpu/mcu | 1.2GHz 64-bit quad-core ARMv8 CPU | Atheros AR9331 MIPS 400 mhz | ATSAMD21G18, 32-Bit ARM Cortex M0+ 48 MHz | ATmega328P 16 MHz |
| Bussar | 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 |
| Lagrings möjligheter | Micro SD | 16 MB | 256 KB flash minne, | 32 KB flash minne |
| Näverksadapter | 802.11n Wireless LAN, Ehternet port | Ethernet 802.3 10/100 Mbits/s, Wifi 802.11b/g/n | - | - |
| Bluetooth | Bluetooth 4.1, BLE(Bluetooth Low Energy) |  | - | - |
| Operativ system | 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 | - | - |

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 Classifer fokuset i denna uppsats är inte Random-Forest därför ska istället sci-kit learns (scikit-learn.org 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\* 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 datatsetet 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 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(ex vis talet –99). 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. Outliars 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 Outliars 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 allvarligare 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 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 confimatory. 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, s. 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.

**Relaterat arbete**  
En algoritm för ett artificiellt neuralt nätverk har implementerats I en billig MCU 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 et al, 2015).

MCU:n 8051 I projektet för Pardo et al(2015) verkar arbeta på 24MHz frekvens enligt Texas Instruments (2016). Den produkt som refereras till I Pardo et al(2015) - CC1110F32 – finns inte att läsa om direkt på den hemsidan, men den familj som produkten tillhör finns att läsa om.

Arbetet som Pardo et al(2015) utfört visar att det är fullt möjligt att predicera en egenskap med relativt enkel och svag hårdvara I form 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.

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 et al, 2011).

Här har Jiang et al(2011) en arkitektur av klustren som påminner om den som vårat 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.

**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 producerarar 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.

**Problemformulering och syfte**

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 frågor som studien syftar till att besvara är:

1. Hur kan en AIM konstrueras för att självständigt kunna predicera någon parameter via utläsning av värden från en eller flera sensorer?
2. Hur kan ett generellt ramverk utformas för utläsning av sensordata från olika bussar på en microcontroller?

**Källförteckning**

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

<https://www.raspberrypi.org/> [29-11-2016]

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

<http://viametrics.com/sv/besoksrakning/kopcentrum/> [28-11-16]

<http://scikit-learn.org/stable/index.html> [02-12-2016]

<http://ieeexplore.ieee.org.lib.costello.pub.hb.se/document/7469991/?part=1> [01-12-16]

<https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm> [02-12-16]

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.

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

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]