

Tahapan Dalam Desain dan Penerapan Model Machine Learning untuk Sistem Tertanam

Ady Chandra Nugroho
Teknik Komputer
ady_chandra@gmail.com

Abstrak

Pembelajaran mesin dan komputasi tepi saat ini menjadi teknologi populer yang digunakan dalam disiplin apa pun. Fleksibilitas dan beradaptasi dengan masalah adalah keunggulan utama dari teknologinya. Dalam tulisan ini, Penulis menjelaskan langkah demi langkah untuk membuat model pembelajaran mesin ringan yang khusus ditujukan untuk aplikasi sistem tertanam. Penulis menggunakan alat pembelajaran mesin open source yang disebut Weka untuk merancang model. Selain itu, Penulis melakukan eksperimen pengenalan tegangan sederhana untuk membuat dataset Penulis sendiri untuk evaluasi. Penulis mengevaluasi kompleksitas dan akurasi algoritma untuk pengklasifikasi terkenal yang berbeda seperti mesin vektor dukungan, logistik sederhana dan pohon hoeffding. Dalam model tersebut dapat direpresentasikan menjadi 3 baris snippet program, yang mana sangat memungkinkan untuk dijalankan pada end device yang memiliki kemampuan komputasi yang rendah sekalipun. Sedangkan model yang kompleksitasnya agak tinggi (berbasis tree) dapat direpresentasikan menjadi sebuah program dengan menggunakan syntax if-else atau case bercabang. Kompleksitas model yang Penulis contohkan memiliki kompleksitas sehingga dapat mudah untuk dijalankan. Hasil yang didapatkan akan berbeda setiap resource dan aplikasi yang dibuat. Contoh aplikasi-aplikasi yang memiliki workload yang besar seperti natural language processing (NLP). Maka prosesor yang digunakan untuk aplikasi tersebut memerlukan proses komputasi yang lebih tinggi contohnya menggunakan single board PC (SBC). Serta metode machine learning yang digunakan akan lebih kompleks dibanding dengan aplikasi yang sederhana.

Kata Kunci: Sistem Tertanam, Model, Teknologi.

PENDAHULUAN

An intelligence system atau sebuah sistem cerdas merupakan idaman bagi manusia di mana sistem tersebut dapat menganalisa permasalahan yang ada (perception) dan memberikan feedback secara natural (decision making). Penggunaan sistem cerdas saat ini meliputi banyak sektor seperti transportasi, ekonomi, pelayanan umum, pendidikan hingga dunia medis (Rahmanto et al., 2020), (S Samsugi & Suwanto, 2018), (F. Kurniawan & Surahman, 2021). Secara teknis untuk mewujudkan sebuah sistem cerdas beberapa metode dapat diimplementasikan. Saat ini terdapat 2 teknik yang populer untuk digunakan yakni machine learning (ML) dan deep learning (DL) (Surahman et al., 2021), (Ratnasari et al., n.d.). Gambar 1 memperlihatkan peningkatan jumlah publikasi ilmiah yang membahas kedua metode tersebut, hal ini menunjukkan kredibilitas metode yang disebutkan. Perbedaan mendasar dari machine learning dan deep learning ialah cara ekstraksi fitur. ML mengandalkan sebuah tahapan preprocessing untuk mendapatkan fitur yang diinginkan selanjutnya kita proses pada sebuah model (Hayatunnufus & Alita, 2020), (S Samsugi et al., 2018), (Ahmad et al., 2022). Sedangkan DL merupakan metode yang bersifat “featureless extraction” di mana fitur diproses di dalam sebuah model. Pada dasarnya, teori fundamental yang digunakan pada kedua metode tersebut meliputi probabilitas, statistika,

kalkulus, dan aljabar linear. Sehingga sebuah model tidak lain hanyalah beberapa persamaan matematis yang dapat direalisasikan (Selamet Samsugi, Mardiyansyah, et al., 2020), (Rahmanto et al., 2021), (Suaidah, 2021). Kedua metode tersebut memiliki keunggulan dan kelemahan masing-masing dan secara khusus untuk ML yang memiliki kelebihan yakni ukuran modelnya yang relatif lebih kecil dan tidak kompleks, (Puspaningrum et al., 2020), (S Samsugi & Burlian, 2019), (Anantama et al., 2020). Penggunaan sistem cerdas idealnya dapat dilakukan secara real time. Biasanya untuk sistem yang kompleks dan menggunakan model yang berukuran besar spesifikasi komponen komputasi harus mengikuti. Sedangkan, tren saat ini telah dikenal dan sering digunakan sebuah teknologi yang disebut edge computing, di mana semua proses komputasi dilakukan di tempat sehingga menyediakan informasi dengan cepat (Pratama et al., 2021), (Pratama Zanofa & Fahrizal, 2021), (Yuliana et al., 2021). Hanya saja, komponen pemroses yang digunakan terbatas pada beberapa parameter spesifikasi. Sehingga saat menggunakan sebuah model sistem cerdas sebaiknya dengan kompleksitas dan ukuran yang kecil. Padahal, implementasi sistem cerdas pada edge computing memiliki prospek aplikasi yang luas salah satunya digunakan untuk penerjemah Bahasa isyarat (Kristiawan et al., 2021), (Riski et al., 2021), (Hafidhin et al., 2020).



Gambar 1. Tren jumlah publikasi ilmiah machine learning dan deep learning

Berdasarkan urgensi masalah di atas dan karena juga sedikitnya literatur mengenai implementasi ML untuk edge computing dalam Bahasa Indonesia, pada tulisan ilmiah ini Penulis menjabarkan langkah-langkah dalam mendesain dan mengimplementasikan algoritma ML untuk aplikasi edge computing dalam hal ini Penulis menggunakan sistem tertanam. Tools yang Penulis gunakan pada penelitian ini bersifat open source sehingga dapat digunakan oleh semua pihak. Terlebih, Penulis berikan contoh sederhana implementasi ML model pada stress recognition berdasarkan langkah-langkah yang Penulis jabarkan. Sebagai evaluasi, Penulis akan menguji kompleksitas dari model ML ketika ditanamkan pada sistem embedded yang digunakan (Selamet Samsugi et al., 2021), (Dita et al., 2021), (Selamet Samsugi, Yusuf, et al., 2020).

KAJIAN PUSTAKA

Arduino UNO

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (datasheet). Uno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega yang diprogram sebagai konverter USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB to serial. Nama "Uno" berarti *satu*

dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. Uno adalah yang terbaru dalam serangkaian board USB Arduino, dan sebagai model referensi untuk platform Arduino, untuk perbandingan dengan versi sebelumnya, lihat indeks board Arduino (Isnain et al., 2021), (S Samsugi, 2017), (S Samsugi & Silaban, 2018b).

Sistem Cerdas (Intelligent System)

Sistem cerdas (intelligent system) adalah sistem yang dibangun dengan menggunakan teknik kecerdasan buatan. Kursus ini terdiri dari dua bagian. Bagian pertama meliputi: konsep dasar logika fuzzy, teori himpunan fuzzy, fuzzy, defuzzifikasi, inferensi dan pemetaan fuzzy untuk kontrol. Bagian kedua meliputi: pengetahuan dasar jaringan saraf tiruan, jaringan saraf tiruan perceptron dan pola backpropagation, dan aplikasi jaringan saraf tiruan terutama untuk sistem kontrol dan metode lainnya. Setelah mengikuti mata kuliah ini mahasiswa akan memahami kontrol cerdas, khususnya kontroler logika fuzzy dan jaringan syaraf tiruan (Nurdiansyah et al., 2020), (Setiawan et al., 2021), (Rumalutur & Ohoiwutun, 2018).

Sistem Tertanam

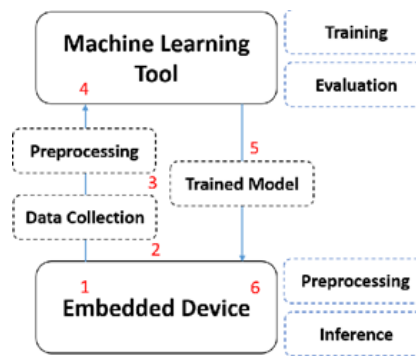
Sistem tertanam adalah kombinasi perangkat keras dan perangkat lunak komputer, tetap atau dapat diprogram, dirancang untuk fungsi atau fungsi tertentu dalam sistem yang lebih besar. Mesin industri, peralatan industri pertanian, mobil, peralatan medis, kamera, peralatan rumah tangga, pesawat terbang, mesin penjual otomatis dan mainan, serta perangkat seluler, adalah contoh-contoh yang mungkin dari implementasi sistem tertanam (Jayadi et al., 2021), (S Samsugi & Silaban, 2018a), (Selamet Samsugi et al., 2018). Secara umum, definisi sistem tertanam adalah sistem komputer, tetapi sistem tertanam berbeda dari tidak memiliki antarmuka pengguna (UI) - misalnya, pada perangkat di mana sistem dirancang untuk melakukan satu tugas - hingga grafik yang kompleks antarmuka pengguna (GUI), seperti pada perangkat seluler. Antarmuka pengguna dapat mencakup tombol, lampu, sensor layar sentuh, dan banyak lagi. Beberapa sistem juga menggunakan antarmuka pengguna jarak jauh (Budioko, 2016), (Yulianti et al., 2021), (Ahdan et al., 2019).

Photoplethysmograph (PPG)

Photoplethysmograph (PPG) adalah metode yang digunakan untuk mengukur perubahan volume darah pada suatu organ atau tubuh dengan menangkap sinyal yang terbentuk ketika sumber cahaya ditransmisikan ke jaringan kulit melalui sensor optik (Borman et al., 2018), (Imani & Ghassemian, 2019), (Pindrayana et al., 2018).

METODE

Secara sederhana proses desain dan implementasi algoritma machine learning (ML) pada sistem embedded dapat diilustrasikan pada Gambar 2. Pada dasarnya, 2 perangkat keras dibutuhkan untuk merealisasikan sistem ini, yakni embedded device (sistem yang akan kita implementasikan) dan host computer (digunakan untuk menjalankan machine learning tool untuk training dan evaluasi model) (D. E. Kurniawan et al., 2019), (Zanofa et al., 2020), (Wantoro et al., 2021).



Gambar 2. Implementasi model machine learning untuk sistem tertanam

Workflow dari implementasi model ML, secara berurutan ialah secara berikut :

Desain perangkat keras (hardware) dari sistem tertanam sesuai dengan tujuan dan spesifikasi yang dibutuhkan.

Kumpulkan data dengan baik dan benar dan merepresentasikan penggunaan sebenarnya.

Mendapatkan fitur dari data yang terkumpul (features extraction) dan memberi label atau class pada kumpulan data tersebut (labelling).

Train model pada host computer dengan menggunakan ML tool, pilih classifier yang cocok digunakan.

Test model yang didapat dengan beberapa metrik pengujian (misal K-fold validation).

Lakukan optimasi dengan memilih fitur mana yang dibutuhkan dan memiliki korelasi dengan permasalahan yang dihadapi.

Implementasikan model yang sudah dilatih ke dalam sistem embedded yang telah dibuat.

Dalam penggunaan secara real time datastream harus diproses terlebih dahulu untuk mendapatkan fitur yang sama fitur yang digunakan untuk training model.

Terakhir untuk memprediksi class yang sesuai, proses inference dapat dilakukan

Berikut detail dari Gambar 2 mengenai cara implementasi model ML ke dalam sistem tertanam (Susanto, n.d.), (Yurnama & Azman, 2009).

Embedded Devices

Saat ini komponen komputasi yang digunakan pada sistem tertanam memiliki banyak pilihan, dari berbagai macam tipe mikrokontroler hingga beberapa tipe mikrokomputer yang populer digunakan di Indonesia antara lain ialah :

Arduino berbasis ATmega328 – RISC 8 bit mikrokontroler berjalan pada kecepatan 16 MHz.

ESP32 – 32 bit Xtensa LX6 SoC dengan clock source 160 MHz.

ARM seri M – RISC 32 bit mikrokontroler dengan basis clock 72 MHz.

ARM seri A – arm64 arch mikrokomputer yang berjalan dengan menggunakan sistem operasi (misal Raspberry Pi).

Dari beberapa contoh di atas, pertimbangan pemilihan komponen komputasi tergantung kepada permasalahan yang ingin di atasi, karena komponen tersebut memiliki kelebihan dan kekurangan masing- masing. Parameter yang dapat menjadi pertimbangan seperti interface I/O, kecepatan (clock speed semata bukan menjadi benchmark), kapasitas memori, tipe data yang dapat digunakan (penting untuk ML) dan konsumsi daya (penting untuk edge computing)

Data Collection and Preprocessing

Melatih sebuah model ML membutuhkan data yang cukup banyak untuk mendapatkan hasil yang lebih terpercaya. Oleh sebab itu pengumpulan data dengan baik dan benar perlu

dilakukan. Data yang dikumpulkan merupakan data dari sensor, contoh: suhu, kelembapan, inertial motion unit, elektrokardiograf, dan banyak lainnya. Pengumpulan data harus dilakukan sesuai dengan kriteria sinyal dan sistem yang diinginkan contoh yang menjadi konstrain adalah frekuensi sampling dan tipe data yang didapat (signed, unsigned, fixed point, floating point). Terlebih, data tersebut mewakili beberapa atribut dan class yang akan diprediksi (S Samsugi et al., 2021), (Rikendry & Navigasi, 2007).

Setelah data yang diinginkan terkumpul, proses selanjutnya yaitu memproses data tersebut untuk mendapatkan fitur ML yang baik. Preprocessing akan berbeda sesuai dengan spesifikasi dan kebutuhan masing-masing sinyal. Salah satu parameter yang perlu diperhatikan ialah panjang sebuah segmen atau window dari sebuah sinyal. Membuat fitur dari satu data stream tidaklah direkomendasi karena apabila sinyal tersebut sebuah noise akan membuat fitur yang kurang bagus. Dengan demikian biasanya sebuah segmen (kumpulan dari beberapa n poin sinyal, buffer) sinyal dibuat. Ukuran segmen akan berpengaruh dengan respon dan akurasi prediksi. Semakin panjang segmen maka akan kurang responsif, namun memiliki banyak informasi di dalamnya. Segmen tidak hanya dalam bentuk time domain, namun bisa juga dalam bentuk frequency domain. Terlebih lagi bisa dalam bentuk dari gabungan kedua domain tersebut, seperti yang dilakukan pada karya ilmiah (Widodo et al., 2020), (Selamet Samsugi & Wajiran, 2020).

Selanjutnya salah satu proses yang paling penting ialah mengekstraksi sebuah segmen sinyal menjadi fitur ML. Fitur yang paling umum digunakan ialah statistika fitur seperti rata-rata, median, modus, maksimum, minimum, standar deviasi, kuartil dan banyak lainnya. Fitur-fitur tersebut dikombinasikan agar mendapatkan keunikan yang lebih terlihat pada sebuah sinyal. Setelah perhitungan fitur selesai, kita dapat memberi label pada fitur tersebut dan membuat sebuah database fitur beserta labelnya yang biasa disebut dataset. Format dataset bermacam, yang umum digunakan kolom merepresentasikan beberapa fitur, dan juga label dari fitur tersebut.

Model Training and Evaluation

Dari dataset yang sudah dibuat, selanjutnya kita dapat melakukan proses training model dengan menggunakan ML tool pada host komputer kita, spesifikasi host komputer kita berpengaruh terhadap lamanya proses training. Terdapat beberapa pilihan yang dapat digunakan dari yang gratis hingga berbayar, diantaranya yang populer ialah,

TensorFlow – free, berbasis python, script interface

PyTorch – free, berbasis python, script interface

Caffe – free, berbasis C++, script interface

Weka – free, berbasis Java, GUI interface

Rapid Miner – paid, berbasis Java, GUI interface

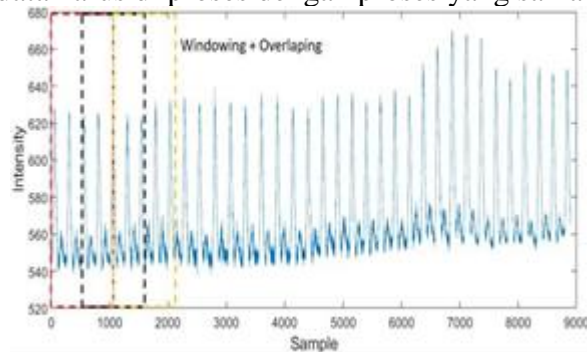
Matlab – paid, berbasis Java, GUI interface

ML dapat dibagi menjadi 2 kategori learning, yakni supervised dan unsupervised learning. Penggunaannya tergantung dari aplikasi yang diimplementasikan. Jenis kategori tersebut berpengaruh pada tipe classifier yang harus digunakan, dan juga konfigurasi parameter pada classifier tersebut juga harus diperhatikan. Setelah model kita train, selanjutnya kita dapat mengevaluasi kinerja dari model tersebut secara offline. Yakni dengan cara split data ataupun cara umum yang digunakan yakni K-fold cross validation (K adalah angka bulat menandakan pembagian data, misal 5, 10, 20) (Amarudin et al., 2020), (Kholidi dkk., 2015), (Subandi, 2016). Hal ini perlu dilakukan untuk mengetahui apakah model yang dihasilkan memiliki performa yang bagus. Apabila tidak, maka kita dapat melakukan optimasi dengan cara memilih classifier yang lain, mengganti konfigurasi parameter pada

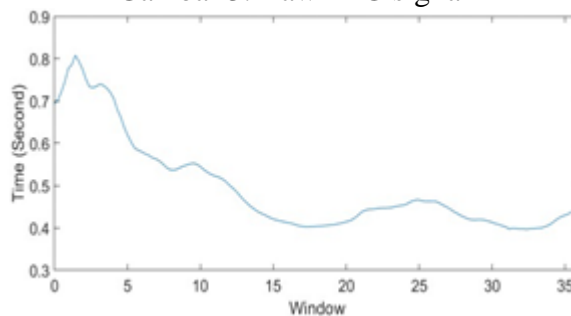
classifier tersebut, ataupun dapat dilakukan dengan cara mencari fitur yang sesuai dan dibutuhkan untuk menyelesaikan masalah yang dihadapi.

Model Implementation on Embedded Device

Apabila model yang didapat sudah memiliki performa yang baik, maka model tersebut dapat kita implementasikan ke sistem tertanam kita. Cara implementasi dapat dilakukan dengan cara mencari persamaan atau model matematika yang terdapat di dalam model tersebut. Selanjutnya model tersebut dapat dijalankan secara real time. Sebelum melakukan proses inference, raw data harus di proses dengan proses yang sama saat membuat dataset.



Gambar 3. Raw PPG signal



Gambar 4. Fitur: mean RR

HASIL DAN PEMBAHASAN

Implementasi menggunakan sinyal photoplethysmography (PPG) yang Penulis rekam untuk mengetahui tingkat stress yang dialami oleh pengguna. Implementasi menggunakan mouse PPG dan sebuah stress test game yang dapat dilihat detailnya pada referensi ilmiah. Terdapat 3 kelas yang diukur yakni, “no stress”, “time pressure”, dan “interruption”. Machine learning tool yang digunakan pada tahap evaluasi ialah Weka dengan bantuan Matlab untuk tahap pengolahan sinyal nya. Berikut detail dari masing-masing tahapan yang Penulis ilustrasikan seperti Gambar 2,

Embedded Device

Mikrokontroler yang digunakan yakni NUC120 berbasis ARM®Cortex™-M0. Mouse tersebut dilengkapi dengan Bluetooth modul dan 4 sensor PPG dengan analog front end (AFE) meliputi filter dan programmable-gain amplifier (PGA117). Sensor menggunakan frekuensi sampling 250 Hz.

Data Collection and Preprocessing

Data diambil dari beberapa subjek, tetapi untuk penelitian ini Penulis hanya mengambil satu sebagai contoh. Subjek diminta untuk meletakkan telapak tangan di mouse (seperti menggunakan mouse) sembari melakukan aktivitas sesuai dengan aplikasi yang Penulis

buat. Di dalam eksperimen kita dapatkan 4 sinyal PPG dari 4 sensor berbeda, tetapi untuk tujuan kali ini Penulis gunakan salah satu sensor yang memiliki signal-to-noise ratio (SNR) paling besar yakni pada sensor yang terletak pada jari. Selanjutnya sinyal Penulis analisa dan kita proses untuk membuat fitur ML-nya. Pada tahap ini yang kita miliki ialah raw signal yang dapat dilihat pada Gambar 3. Pada sinyal 1 dimensi seperti sinyal PPG sinyal setiap fitur diambil dari sinyal yang sudah melalui proses windowing dan overlapping. Proses windowing yakni mengambil beberapa sample dari sinyal secara real-time untuk mendapatkan resolusi yang tinggi pada fitur-fitur yang dibuat. Panjang window berbeda antara karakteristik sinyal dan sistem yang digunakan. Penulis menggunakan 3 detik window (750 samples) dengan panjang overlapping 1 detik (250 samples). Lalu kita dapat mengekstrak fitur dari setiap window sinyal. Untuk penelitian ini Penulis memberi contoh penggunaan 4 statistical features yang sering digunakan dalam stress recognition yakni “Mean RR”, “RMSSD”, “LF”, dan “HF”. Dengan mengaplikasikan operasi pada persamaan (1) sampai (4) secara berturut-turut. Di mana, PR ialah pulse rate yang merupakan nilai denyut jantung dalam satu menit, RR ialah peak-to-peak interval dari sinyal PPG yang memberikan informasi periode pada sinyal, RMSSD ialah root-mean square dari RR interval, LF ialah total dari power spectrum di frekuensi 0.04 sampai 0.15 Hz, HF ialah total power spectrum di frekuensi 0.15 sampai 0.4 Hz. Contoh gambaran fitur pada fungsi waktu dapat dilihat pada Gambar 4 yang merupakan nilai fitur dari “Mean RR”.

$$RR(second) = \frac{60}{PR} \quad (1)$$

$$RMSSD = \sqrt{\frac{1}{r-1} \sum_{k=1}^{r-1} (RR_{k+1} - PR)^2} \quad (2)$$

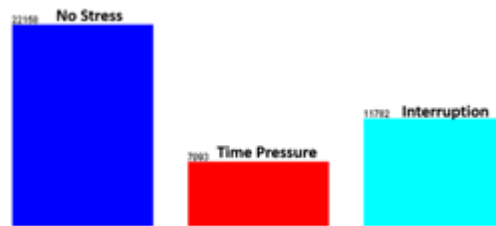
$$LF = \frac{\sum_{f1=0.04}^{0.15} P_{NN}(f1)}{\sum_{f2=0}^{f2=0.15} P_{NN}(f2)} \cdot 100 \quad (3)$$

$$HF = \frac{\sum_{f1=0.15}^{0.4} P_{NN}(f1)}{\sum_{f2=0}^{f2=0.4} P_{NN}(f2)} \cdot 100 \quad (4)$$

Fitur yang sudah dikalkulasi dan diekstrak lalu dibentuk menjadi sebuah dataset untuk dapat digunakan di dalam machine learning tool.

Model Training and Evaluation

Dataset yang dibuat bisa kita import ke dalam Weka. Sebaiknya dalam membuat dataset seharusnya dalam jumlah yang sama (balanced) sehingga dapat meningkatkan akurasi dari model yang akan dibuat. Sedangkan pada penelitian ini Penulis menggunakan imbalanced dataset dengan distribusi instance dapat dilihat pada Gambar 5, di mana kelas “No Stress” memiliki jumlah yang paling banyak diikuti dengan kelas “Interruption” dan kelas “Time Pressure”. Tiga classifier Penulis gunakan sebagai contoh, yakni support vector machine, simple logistic, dan hoeffding tree, dengan konfigurasi masing-masing classifier sebagai berikut,



Gambar 5. Distribusi Dataset

Hoeffding tree

- decimal number = 2
- batch size = 100
- grace period = 200
- hoeffding tie threshold = 0.05

Setelah dilakukan proses training model yang dihasilkan terlihat seperti pada Gambar 6. Model hoeffding membentuk pohon logika.

```

LF <= 1521.861:
| LF <= 227.334: no stress (414.500) NB1 NB adaptive1
| LF > 227.334:
| | MEAN_RR <= 948.797:
| | | MEAN_RR <= 587.016: time pressure (121.000) NB2 NB adaptive2
| | | MEAN_RR > 587.016:
| | | | MEAN_RR <= 784.427: no stress (4619.401) NB3 NB adaptive3
| | | | MEAN_RR > 784.427:
| | | | | MEAN_RR <= 799.741: no stress (474.500) NB4 NB adaptive4
| | | | | MEAN_RR > 799.741: no stress (2045.500) NB5 NB adaptive5
| | | MEAN_RR > 948.797: time pressure (2201.207) NB6 NB adaptive6
| LF > 1521.861: interruption (3515.001) NB7 NB adaptive7

```

Gambar 6. Model Hoeffding Tree

```

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      22587          55.0459 %
Incorrectly Classified Instances    18446          44.9541 %
Kappa statistic                    0.0508
Mean absolute error                 0.3561
Root mean squared error            0.456
Relative absolute error             89.6178 %
Root relative squared error        102.291 %
Total Number of Instances         41033

```

Gambar 7. Hasil Testing Hoeffding

Support vector machine (SVM):

- one-versus-one SVM
- decimal number = 2
- seed number = 1
- batch size = 100.

Setelah dilakukan proses training model yang dihasilkan terlihat seperti pada Gambar 8. Pada gambar terdapat 3 persamaan berbeda yakni (dari atas ke bawah) persamaan untuk “no stress” vs “time pressure”; “no stress” vs “interruption”; “time pressure” vs “interruption”. Pada model ini semua persamaan dioperasikan dan dicari nilai maksimum sehingga didapatkan kelas yang diprediksi.


```

2.3807 * (normalized) MEAN_RR
+ 2.5474 * (normalized) RMSSD
+ -3.7529 * (normalized) LF
+ 0.0106 * (normalized) HF
- 2.1626

0.002 * (normalized) MEAN_RR
+ -0.0042 * (normalized) RMSSD
+ 0.0057 * (normalized) LF
+ 0.0004 * (normalized) HF
- 1.0001

-5.6278 * (normalized) MEAN_RR
+ 2.3668 * (normalized) RMSSD
+ 0.5107 * (normalized) LF
+ -2.6365 * (normalized) HF
+ 2.1102

```

Gambar 8. Model SVM

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      25147      61.2848 %
Incorrectly Classified Instances    15886      38.7152 %
Kappa statistic                    0.2518
Mean absolute error                 0.3644
Root mean squared error             0.4247
Relative absolute error             91.7018 %
Root relative squared error         95.2832 %
Total Number of Instances          41033

```

Gambar 9. Hasil Testing SVM

Simple Logistic

batch size = 100.
heuristic stop = 50
max boosting iterations = 500
decimal number = 2

Setelah dilakukan proses training model yang dihasilkan terlihat seperti pada Gambar 10. Pada model ini pula semua persamaan yang ada di model dioperasikan sehingga mendapatkan nilai maksimum yang berkorelasi dengan kelas yang diprediksi.

```

Class no stress :
3.74 +
[MEAN_RR] * -0 +
[RMSSD] * -0.01

Class time pressure :
-2.66 +
[MEAN_RR] * 0 +
[LF] * -0

Class interruption :
-0.75 +
[LF] * 0

```

Gambar 10. Model Simple Logistic

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      25297      61.6504 %
Incorrectly Classified Instances    15736      38.3496 %
Kappa statistic                    0.3094
Mean absolute error                 0.3213
Root mean squared error             0.4137
Relative absolute error             80.8592 %
Root relative squared error         92.8007 %
Total Number of Instances          41033

```

Gambar 11. Hasil Testing Simple Logistic

Setelah model di training, model tersebut dievaluasi sehingga kita dapat mengetahui performa dari model tersebut. Pada penelitian ini Penulis menggunakan 10- Fold Cross Validation untuk melakukan evaluasi model. Hasil dari masing-masing model terlihat pada Gambar 7, 9, dan 11 untuk model hoeffding tree, SVM, dan simple logistic secara berurutan. Rata-rata hasil yang didapatkan keseluruhan model menunjukkan akurasi di bawah 70 %, yang berarti kurang baik untuk diimplementasikan. Peningkatan akurasi dari sebuah model dapat dilakukan dengan beberapa cara, yakni salah satunya dengan menyeleksi fitur mana yang berdampak besar pada performa prediksi dan dapat menghilangkan fitur yang sifatnya redundan. Feature selection dapat dilakukan di dalam weka, rule-of-thumb dari seleksi fitur ialah melihat korelasi fitur tersebut dengan fitur yang lain sehingga dapat mempersempit dimensi fitur pada model. Ilustrasi pemilihan fitur pada weka terlihat pada Gambar 12. Penulis memberi contoh penggunaan principle component analysis (PCA) yang difungsikan sebagai penyeleksi fitur dengan mencari korelasi antara fiturnya. Setelah model dievaluasi dan berfungsi dengan baik. Maka bisa diimplementasikan ke dalam embedded device.

```

Correlation matrix
1      0.33  0.08  -0.55
0.33   1      0.89  0.33
0.08   0.89  1      0.32
-0.55  0.33  0.32   1

eigenvalue  proportion  cumulative
2.08711     0.52178   0.52178
1.54486     0.38622   0.90799
0.33176     0.08294   0.99093
-0.664RMSSD-0.653LF-0.355HF-0.072MEAN_RR
-0.758MEAN_RR+0.621HF-0.19RMSSD-0.062LF
0.644HF+0.541MEAN_RR-0.528LF+0.116RMSSD

Eigenvectors
V1      V2      V3
-0.072  -0.7579  0.5414 MEAN_RR
-0.6645 -0.1896  0.1164 RMSSD
-0.6534 -0.0617 -0.5282 LF
-0.3555  0.6212  0.6437 HF

Ranked attributes:
0.47822  1  -0.664RMSSD-0.653LF-0.355HF-0.072MEAN_RR
0.09201  2  -0.758MEAN_RR+0.621HF-0.19RMSSD-0.062LF
0.00907  3  0.644HF+0.541MEAN_RR-0.528LF+0.116RMSSD

```

Gambar 12. Feature Selection

Model Implementation

Cara paling sederhana menanamkan model ke dalam end device dengan cara membuat program persamaan dari model tersebut. Sebagai contoh seperti Gambar 13 yang merupakan bentuk snippet program (Bahasa C++) dari model simple logistic. Ataupun cara lain yakni dengan menggunakan API dari machine learning tools yang dipakai, karena pada contoh kali ini Penulis menggunakan weka, maka dapat menggunakan API yang weka miliki yang berbasis Java.

SIMPULAN

Dari pemilihan jenis device yang digunakan hingga bagaimana cara mengimplementasikan model ke dalam sebuah program. Di dalam menggunakan machine learning memang tujuan utamanya ialah performa dari model tersebut bukan kompleksitasnya. Tetapi dalam end device kita memiliki limitasi pada sumber komputasinya. Eksperimen dengan menggunakan 3 classifier, dan salah satu implementasi model terlihat pada Gambar 13 (simple logistic classifier). Dalam model tersebut dapat direpresentasikan menjadi 3 baris snippet program, yang mana sangat memungkinkan untuk dijalankan pada end device yang memiliki kemampuan komputasi yang rendah sekalipun. Sedangkan model yang kompleksitasnya agak tinggi (berbasis tree) dapat direpresentasikan menjadi sebuah

program dengan menggunakan syntax if-else atau case bercabang. Kompleksitas model yang Penulis contohkan memiliki kompleksitas sehingga dapat mudah untuk dijalankan. Hasil yang didapatkan akan berbeda setiap resource dan aplikasi yang dibuat. Contoh aplikasi-aplikasi yang memiliki workload yang besar seperti natural language processing (NLP). Maka prosesor yang digunakan untuk aplikasi tersebut memerlukan proses komputasi yang lebih tinggi contohnya menggunakan single board PC (SBC).

REFERENSI

- Ahdan, S., Susanto, E. R., & Syambas, N. R. (2019). Proposed Design and Modeling of Smart Energy Dashboard System by Implementing IoT (Internet of Things) Based on Mobile Devices. *2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 194–199.
- Ahmad, I., Samsugi, S., & Irawan, Y. (2022). Penerapan Augmented Reality Pada Anatomi Tubuh Manusia Untuk Mendukung Pembelajaran Titik Titik Bekam Pengobatan Alternatif. *Jurnal Teknoinfo*, *16*(1), 46. <https://doi.org/10.33365/jti.v16i1.1521>
- Amarudin, A., Saputra, D. A., & Rubiyah, R. (2020). Rancang Bangun Alat Pemberi Pakan Ikan Menggunakan Mikrokontroler. *Jurnal Ilmiah Mahasiswa Kendali Dan Listrik*, *1*(1), 7–13.
- Anantama, A., Apriyantina, A., Samsugi, S., & Rossi, F. (2020). Alat Pantau Jumlah Pemakaian Daya Listrik Pada Alat Elektronik Berbasis Arduino UNO. *Jurnal Teknologi Dan Sistem Tertanam*, *1*(1), 29–34.
- Borman, R. I., Syahputra, K., Jupriyadi, J., & Prasetyawan, P. (2018). Implementasi Internet Of Things pada Aplikasi Monitoring Kereta Api dengan Geolocation Information System. *Seminar Nasional Teknik Elektro, 2018*, 322–327.
- Budioko, T. (2016). Sistem monitoring suhu jarak jauh berbasis internet of things menggunakan protokol mqtt. *Seminar Nasional Riset Teknologi Informasi*, *1*(30 July), 353–358.
- Dita, P. E. S., Al Fahrezi, A., Prasetyawan, P., & Amarudin, A. (2021). Sistem Keamanan Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino UNO R3. *Jurnal Teknik Dan Sistem Komputer*, *2*(1), 121–135.
- Hafidhin, M. I., Saputra, A., Ramanto, Y., & Samsugi, S. (2020). Alat Penjemuran Ikan Asin Berbasis Mikrokontroler Arduino UNO. *Jurnal Teknik Dan Sistem Komputer*, *1*(2), 26–33.
- Hayatunnufus, H., & Alita, D. (2020). SISTEM CERDAS PEMBERI PAKAN IKAN SECARA OTOMATIS. *Jurnal Teknologi Dan Sistem Tertanam*, *1*(1), 11–16.
- Imani, M., & Ghassemian, H. (2019). Electrical Load Forecasting Using Customers Clustering and Smart Meters in Internet of Things. *9th International Symposium on Telecommunication: With Emphasis on Information and Communication Technology, IST 2018*, 113–117. <https://doi.org/10.1109/ISTEL.2018.8661071>

- Isnain, A. R., Sintaro, S., & Ariany, F. (2021). *Penerapan Auto Pump Hand Sanitizer Berbasis Iot*. 2(2), 63–71.
- Jayadi, A., Susanto, T., & Adhinata, F. D. (2021). Sistem Kendali Proporsional pada Robot Penghindar Halangan (Avoider) Pioneer P3-DX. *Majalah Ilmiah Teknologi Elektro*, 20(1), 47. <https://doi.org/10.24843/mite.2021.v20i01.p05>
- Kholidi dkk. (2015). Rancang Bangun Alat Pemberi Pakan dan Pengatur Suhu Otomatis untuk Ayam Pedaging Berbasis Programmable Logic Controller pada Kandang Tertutup. *Rekayasa Dan Teknologi Elektro Rancang*, 86–95.
- Kristiawan, N., Ghafaral, B., Borman, R. I., & Samsugi, S. (2021). Pemberi Pakan dan Minuman Otomatis Pada Ternak Ayam Menggunakan SMS. *Jurnal Teknik Dan Sistem Komputer*, 2(1), 93–105.
- Kurniawan, D. E., Iqbal, M., Friadi, J., Borman, R. I., & Rinaldi, R. (2019). Smart Monitoring Temperature and Humidity of the Room Server Using Raspberry Pi and Whatsapp Notifications. *Journal of Physics: Conference Series*, 1351(1). <https://doi.org/10.1088/1742-6596/1351/1/012006>
- Kurniawan, F., & Surahman, A. (2021). SISTEM KEAMANAN PADA PERLINTASAN KERETA API MENGGUNAKAN SENSOR INFRARED BERBASIS MIKROKONTROLLER ARDUINO UNO. *Jurnal Teknologi Dan Sistem Tertanam*, 2(1), 7–12.
- Nurdiansyah, M., Sinurat, E. C., Bakri, M., & Ahmad, I. (2020). Sistem Kendali Rotasi Matahari Pada Panel Surya Berbasis Arduino UNO. *Jurnal Teknik Dan Sistem Komputer*, 1(2), 7–12.
- Pindrayana, K., Borman, R. I., Prasetyo, B., & Samsugi, S. (2018). Prototipe Pemandu Parkir Mobil Dengan Output Suara Manusia Menggunakan Mikrokontroler Arduino Uno. *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro*, 2(2).
- Pratama, M. A., Sidhiq, A. F., Rahmanto, Y., & Surahman, A. (2021). Perancangan Sistem Kendali Alat Elektronik Rumah Tangga. *Jurnal Teknik Dan Sistem Komputer*, 2(1), 80–92.
- Pratama Zanofa, A., & Fahrizal, M. (2021). Penerapan Bluetooth Untuk Gerbang Otomatis. *Portaldata.Org*, 1(2), 1–10.
- Puspaningrum, A. S., Firdaus, F., Ahmad, I., & Anggono, H. (2020). Perancangan Alat Deteksi Kebocoran Gas Pada Perangkat Mobile Android Dengan Sensor Mq-2. *Jurnal Teknologi Dan Sistem Tertanam*, 1(1), 1–10.
- Rahmanto, Y., Burlian, A., & Samsugi, S. (2021). SISTEM KENDALI OTOMATIS PADA AKUAPONIK BERBASIS MIKROKONTROLER ARDUINO UNO R3. *Jurnal Teknologi Dan Sistem Tertanam*, 2(1), 1–6.
- Rahmanto, Y., Rifaini, A., Samsugi, S., & Riskiono, S. D. (2020). Sistem Monitoring pH Air Pada Aquaponik Menggunakan Mikrokontroler Arduino UNO. *Jurnal Teknologi Dan Sistem Tertanam*, 1(1), 23–28.

- Ratnasari, T. D., Samsugi, S., Kom, S., & Eng, M. (n.d.). *SETUP MIKROTIK SEBAGAI GATEWAY SERVER PADA SMK PELITA GEDONGTATAAN*.
- Rikendry, & Navigasi, S. (2007). *Sistem kontrol pergerakan robot beroda pemadam api. 2007(Snati)*, 1–4.
- Riski, M., Alawiyah, A., Bakri, M., & Putri, N. U. (2021). Alat Penjaga Kestabilan Suhu Pada Tumbuhan Jamur Tiram Putih Menggunakan Arduino UNO R3. *Jurnal Teknik Dan Sistem Komputer*, 2(1), 67–79.
- Rumalutur, S., & Ohoiwutun, J. (2018). Sistem Kendali Otomatis Panel Penerangan Luar Menggunakan Timer Theben Sul 181 H Dan Arduino Uno R3. *Electro Luceat*, 4(2), 43–51. <https://doi.org/10.32531/jelekn.v4i2.143>
- Samsugi, S. (2017). Internet of Things (iot): Sistem Kendali jarak jauh berbasis Arduino dan Modul wifi Esp8266. *ReTII*.
- Samsugi, S., & Burlian, A. (2019). Sistem penjadwalan pompa air otomatis pada aquaponik menggunakan mikrokontrol Arduino UNO R3. *PROSIDING SEMNASTEK 2019*, 1(1).
- Samsugi, S., Neneng, N., & Aditama, B. (2018). *IoT: kendali dan otomatisasi si parmin (studi kasus peternak Desa Galih Lunik Lampung Selatan)*.
- Samsugi, S., Neneng, N., & Suprpto, G. N. F. (2021). Otomatisasi Pakan Kucing Berbasis Mikrokontroler Intel Galileo Dengan Interface Android. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 5(1), 143–152.
- Samsugi, S., & Silaban, D. E. (2018a). PROTOTIPE CONTROLLING BOX PEMBERSIH WORTEL BERBASIS MIKROKONTROLER. *ReTII*.
- Samsugi, S., & Silaban, D. E. (2018b). Purwarupa Controlling Box Pembersih Wortel Dengan Mikrokontroler. *Prosiding Nasional Rekayasa Teknologi Industri Dan Informasi*, 13, 1–7.
- Samsugi, S., & Suwanto, A. (2018). Pemanfaatan Peltier dan Heater Sebagai Alat Pengontrol Suhu Air Pada Bak Penetasan Telur Ikan Gurame. *Conf. Inf. Technol*, 295–299.
- Samsugi, Selamat, Ardiansyah, A., & Kastutara, D. (2018). Arduino dan Modul Wifi ESP8266 sebagai Media Kendali Jarak Jauh dengan antarmuka Berbasis Android. *Jurnal Teknoinfo*, 12(1), 23–27.
- Samsugi, Selamat, Mardiyansyah, Z., & Nurkholis, A. (2020). Sistem Pengontrol Irigasi Otomatis Menggunakan Mikrokontroler Arduino UNO. *Jurnal Teknologi Dan Sistem Tertanam*, 1(1), 17–22.
- Samsugi, Selamat, Nurkholis, A., Permatasari, B., Candra, A., & Prasetyo, A. B. (2021). Internet of Things Untuk Peningkatan Pengetahuan Teknologi Bagi Siswa. *Journal of Technology and Social for Community Service (JTSCS)*, 2(2), 174.
- Samsugi, Selamat, & Wajiran, W. (2020). IOT: Emergency Button Sebagai Pengaman

- Untuk Menghindari Perampasan Sepeda Motor. *Jurnal Teknoinfo*, 14(2), 99–105.
- Samsugi, Selamat, Yusuf, A. I., & Trisnawati, F. (2020). Sistem Pengaman Pintu Otomatis Dengan Mikrokontroler Arduino Dan Module Rf Remote. *Jurnal Ilmiah Mahasiswa Kendali Dan Listrik*, 1(1), 1–6.
- Setiawan, M. B., Susanto, T., & Jayadi, A. (2021). PENERAPAN SISTEM KENDALI PID PESAWAT TERBANG TANPA AWAK UNTUK KESETABILAN ROLL, PITCH DAN YAW PADA FIXED WINGS. *The 1st International Conference on Advanced Information Technology and Communication (IC-AITC)*.
- Suaidah, S. (2021). Teknologi Pengendali Perangkat Elektronik Menggunakan Sensor Suara. *Jurnal Teknologi Dan Sistem Tertanam*, 02(02). <https://ejurnal.teknokrat.ac.id/index.php/jtst/article/view/1341>
- Subandi. (2016). *PEMBASMI HAMA SERANGGA MENGGUNAKAN CAHAYA LAMPU BERTENAGA SOLAR CELL*. 9(1), 86–92.
- Surahman, A., Aditama, B., Bakri, M., & Rasna, R. (2021). Sistem Pakan Ayam Otomatis Berbasis Internet Of Things. *Jurnal Teknologi Dan Sistem Tertanam*, 2(1), 13–20.
- Susanto, E. R. (n.d.). *Sistem Penunjang Keputusan Cerdas Spasial Pengendalian Avian Influenza H5n1 Pada Unggas Peternakan Rakyat Non Komersial: Studi Kasus Provinsi Lampung*. Bogor Agricultural University (IPB).
- Wantoro, A., Samsugi, S., & Suharyanto, M. J. (2021). Sistem Monitoring Perawatan dan Perbaikan Fasilitas PT PLN (Studi Kasus : Kota Metro Lampung). *Jurnal TEKNO KOMPAK*, 15(1), 116–130.
- Widodo, T., Irawan, B., Prastowo, A. T., & Surahman, A. (2020). Sistem Sirkulasi Air Pada Teknik Budidaya Bioflok Menggunakan Mikrokontroler Arduino Uno R3. *Jurnal Teknik Dan Sistem Komputer*, 1(2), 1–6.
- Yuliana, Y., Paradise, P., & Kusriani, K. (2021). Sistem Pakar Diagnosa Penyakit Ispa Menggunakan Metode Naive Bayes Classifier Berbasis Web. *CSRID (Computer Science Research and Its Development Journal)*, 10(3), 127. <https://doi.org/10.22303/csrid.10.3.2018.127-138>
- Yulianti, T., Samsugi, S., Nugroho, P. A., & Anggono, H. (2021). Rancang Bangun Pengusir Hama Babi Menggunakan Arduino dengan Sensor Gerak. *JTST*, 2(1), 21–27.
- Yurnama, T. F., & Azman, N. (2009). Perancangan Software Aplikasi Pervasive Smart Home. *Snati*, 2009(Snati), E2–E5.
- Zanofa, A. P., Arrahman, R., Bakri, M., & Budiman, A. (2020). Pintu Gerbang Otomatis Berbasis Mikrokontroler Arduino UNO R3. *Jurnal Teknik Dan Sistem Komputer*, 1(1), 22–27.