

IMPLEMENTASIMACHINELEARNINGPADA DESAINSISTEMMONITORINGMOBIL MENGUNAKANON-BOARDDIAGNOSTIC-II

MACHINELEARNINGIMPLEMENTATIONFORCAR MONITORINGDESIGNSYSTEMWITHON-BOARD DIAGNOSTIC-II

Samudra Dzikri¹, Ahmad Tri Hanuranto², Nyoman Bogi Aditya Karna³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Telkom University, Bandung
Fakultas Teknik Elektro, Telkom University, Bandung

¹samudradzikri@student.telkomuniversity.ac.id,

²Athanuranto ³nyoman.bogi@telkomuniversity.ac.id

Abstrak

Banyak fitur dari mobil yang mencakup tentang keselamatan pengemudinya, namun masih banyak kecelakaan lalu-lintas terjadi dimana-mana. Salah satu faktor kecelakaan yang cukup umum yaitu malfungsi dari mobil itu sendiri. Salah satu penyebab malfungsi mesin pada mobil dapat terjadi karena kelalaian pengguna memonitor kondisi mobilnya.

On-Board Diagnostic (OBD II) merupakan standar internasional berkaitan diagnosa kendaraan yang memungkinkan pengambilan data kendaraan yang didapat dari sensor di mobil melalui Engine Control Unit (ECU). Dengan perangkat interface OBDII yaitu ELM327 memungkinkan pengambilan data, pengolahan data dan diagnosa kendaraan secara realtime. Data yang telah diambil, dikumpulkan pada server kemudian diolah oleh machine learning dimana pada perancangan sistem ini digunakan Decision Tree dengan algoritma Classification and Regression Tree (CART) untuk memprediksi terjadinya perubahan kondisi mesin yang tidak baik. Keluaran dari machine learning tersebut akan diteruskan ke user interface untuk menampilkan informasi kendaraan dan prediksi kondisi kendaraan.

Kata Kunci: OBD-II, Machine Learning, Internet of Things, Cloud Computing, Decision Tree

Abstract

Many manufacturer implement safety feature for car user, but traffic accident still occurred even with all of that kind of safety feature which often happened because of engine failure. User neglecting maintenance leaves car in a bad shape and can affect safety of driving.

On-Board Diagnostic (OBD II) are international standard which refers to diagnostic for vehicle which allows us to collect raw data from sensor mounted on car connected to Engine Control Unit (ECU). With interface device of OBD-II which is ELM327, data acquisition, data stream processing, and real time vehicle diagnostic become possible. Collected data are sent to server so it can be processed by machine learning with Decision Tree algorithm with Classification and Regression method which will predict car condition. Car information and predicted condition will be informed to user via web based interface.

Keywords: OBD-II, Machine Learning, Internet of Things, Cloud Computing, Decision Tree.

1. Pendahuluan

Seiring berkembangnya waktu, kendaraan bermotor khususnya mobil bukan lagi menjadi hal yang mewah untuk dimiliki. Banyaknya produksi mobil dan mobil yang terjual khususnya di Indonesia mencapai 1.151.284 pada tahun 2018[11]. Berkembang pesatnya pengguna mobil di Indonesia membutuhkan pengawasan lalu lintas yang lebih efisien dan akurat, dimana tingkat kecelakaan kemungkinan besar bertambah. Tingkat kecelakaan lalu lintas di Indonesia sendiri sudah mencapai 103.228 pada tahun 2017 [14]. Tidak dapat dipungkiri bahwa salah satu penyebab kecelakaan disebabkan oleh Engine Fault karena kelalaian pengguna terhadap Maintenance mesin secara sengaja maupun tidak sengaja. On-Board Diagnostic(OBD) merupakan standar untuk mendiagnosa kendaraan bermotor yang menggunakan berbagai sensor dari Electrical Control Unit (ECU) mulai dari kecepatan kendaraan, Revolution per Minute(RPM) mesin, *coolant temperature*, *fuel rate* dan lain-lain[22][9]. OBD-II merupakan versi terbaru dari OBD, dimana *International Standardization Organization* (ISO) menetapkan OBD II sebagai standar pengambilan data digital dari ECU ke *diagnostic scan tool*. Pada tahun 2008 pemerintah *European Union* mewajibkan mobil yang di produksi diatas tahun 2008 dilengkapi dengan *Control Area Network* (CAN) dan OBD-II untuk memudahkan memonitor kendaraan yang beredar di Eropa[7]. Penggunaan CAN dengan OBD II *diagnostic tools* pada mobil memungkinkan penambahan sensor yang kompleks dengan mudah pada mobil. Melihat kemampuan CAN dan OBD II yang dapat mendiagnosa kesehatan kendaraan secara kompleks, muncul cara-cara untuk mengolah data tersebut menjadi informasi yang penting bagi pengemudi. Data-data mentah seperti berikut dapat di proses secara kompleks dan realtime dengan *MachineLearning* (ML)[1]. *Machine Learning* dapat menganalisis data mentah dari berbagai sumber namun perlu dilatih terhadap data yang ada sebelum digunakan. Berbagai algoritma *Machine Learning* dikembangkan berdasarkan keperluannya, salah satunya yaitu *decision tree* yang akan digunakan pada tugas akhir ini.

2. Dasar Teori

2.1 Engine Control Unit (ECU)

Engine Control Unit (ECU) merupakan sebuah komponen elektronik pada mobil yang bertujuan untuk mengontrol kinerja mesin sesuai dengan kondisi lingkungan agar mesin tetap bekerja secara optimal. Berbagai macam sensor dipasang pada mobil seperti sensor suhu, *throttle position* sensor, O₂ sensor dan lain-lain, lalu ECU akan menyesuaikan kinerja mesin (*ignition*, *fuel injector*, *heater*, dll) terhadap keluaran sensor-sensor tersebut. Pada mobil yang lebih modern ECU dilengkapi dengan CAN Bus yang memungkinkan komunikasi data pada ECU lebih terpusat dan dapat diakses melewati *microcontroller*[4].

2.2 On-Board Diagnostic (OBD)

On Board Diagnostic mulai dipergunakan secara umum pada pertengahan tahun 1990 dengan maksud mengontrol emisi kendaraan roda empat yang diresmikan oleh Environmental Protection Agency (EPA) melalui kongres mengenai *Clean Air Act*. OBD dikembangkan menjadi OBD-II dengan fitur yang lebih modern dibandingkan OBD yang kemudian diresmikan menjadi *standar diagnostic system* kendaraan. Kendaraan roda empat yang diproduksi masa kini diwajibkan memiliki OBD-II yang siap digunakan. Seiring berkembangnya OBD-II, transmisi antar tiap sensor dan ECU menjadi lebih kompleks sehingga jaringan yang digunakan saat itu menjadi tidak efisien. Pada tahun 1991 BOSCH memperkenalkan CAN yang telah dikembangkan dari tahun 1983 di mobil Mercedes-Benz W140 untuk mengatasi masalah kurang efisiennya data transfer dari sensor ke ECU. CAN Bus dapat mentransfer data dengan kecepatan 1 Mbps lebih cepat dibandingkan teknologi terdahulunya. CAN Bus mengirim data dengan mengerim data-data kecil (RPM, Temperatur, dll.) ke semua *Node* sehingga data diBroadcast ke semua koneksi dengan konsisten ke semua Node. CAN Bus Berfungsi untuk mengatasi *complex wiring* pada ECU dengan hanya dua kabel Bus[22][9].

2.3.1 Protokol Data OBD-II

Walaupun OBD-II sudah menjadi standar dari diagnostic system namun protokol data OBD-II itu sendiri bervariasi contoh nya manufaktur di amerika kebanyakan menggunakan protokol standar *Society of Automotive Engineers* (SAE) dan di Asia dan Eropa sendiri menggunakan *protocol The International Organization for Standardization* (ISO). Mobil yang diproduksi diatas tahun 2008 namun kebanyakan sudah menggunakan protokol ISO yang kompatibel dengan CAN Bus, yaitu ISO 11898 CAN di physical dan datalink OSI layer sedangkan ISO 15765 di network dan transport OSI layer[15].

2.4 Cloud Computing

Cloud computing teknologi yang menjadikan internet sebagai pusat pengelolaan data dan aplikasi, dimana pengguna komputer diberikan hak akses terhadap platform tersebut. Seiring berkembangnya jaman, *cloud computing* dapat menjadi server dengan performansi tinggi. Berdasarkan model deployment *cloud computing* dibagi menjadi 3 yaitu *public*, *private* dan *hybrid* dimana *public cloud* penggunaannya dalam satu server terdapat banyak pengguna dan pada *private cloud* terdapat satu pengguna dalam server. *Hybrid cloud* dapat digunakan untuk *public* atau *private cloud*[16].

2.5 Machine Learning

Machine Learning merupakan sebuah metode data analisis dimana sistem yang dibuat otomatis melakukan analisis data dari trainset yang dipelajari sehingga menghasilkan keluaran yang diinginkan dengan bantuan manusia seminimal mungkin. Algoritma *machine learning* dibagi menjadi tiga yaitu *supervised learning*, *unsupervised learning* dan *reinforcement learning*[18].

2.6 Decision Tree

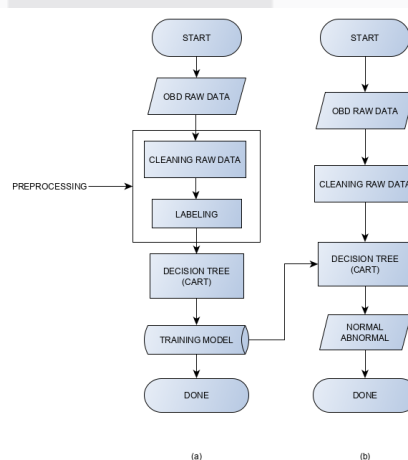
Decision tree merupakan salah satu algoritma machine learning dengan arsitektur seperti pohon dimana terdapat cabang, daun, dan akar. Akar atau root node merupakan awal dari algoritma bekerja dimana data berawal masuk, data lalu diteruskan dan diklasifikasikan di cabang atau branch node dengan menghasilkan keluaran yang akan di simpan di daun atau leaf node. Ada 4 algoritma yang dapat dipakai di *Decision tree* yaitu *Classification and Regression Tree (CART)* dan *Iterative Dichotomiser 3* dimana dikembangkan menjadi *C4.5* dan *C5.0*. Pada tugas akhir ini digunakan algoritma *CART* untuk mengklasifikasi dan prediksi data Input[3].

2.6.1 Classification and Regression Tree (CART)

CART merupakan cabang algoritma dari *decision tree* yang digunakan untuk mengklasifikasikan data dengan nilai *gini impurity* sebagai faktor pengklasifikasian. *Gini impurity* yaitu alat ukur classification disorder dimana nilai tersebut menunjukkan bahwa data yang diamati dengan sempurna terklasifikasi atau tidak. Nilai *gini impurity* dihitung pada parameter yang akan dipelajari oleh algoritma yang kemudian diurutkan dari terbesar ke kecil, parameter dengan *gini impurity* terbesar akan menjadi *root node* yang akan dicabangkan kemudian dihitung kembali *gini impurity* nya untuk menentukan parameter yang akan menjadi bidang pemisah setelahnya, dilakukan secara recursive sampai syarat kedalaman atau *maxdepth* dari *tree* itu terpenuhi. Data akan terus dicabangkan jika masih terdapat *gini impurity*, dengan kata lain data selesai diklasifikasikan jika *gini impurity* bernilai nol dan *gini impurity* bernilai satu jika data terdistribusi merata[15].

3. Perancangan Sistem

Perancangan sistem adalah untuk mengidentifikasi apakah mobil mengalami abnormalitas dari data OBD-II yang telah disimpan di database. Sistem ini menggunakan *decision tree* sebagai algoritma mengklasifikasi dan memprediksi berdasarkan data yang telah direkam. Data yang sudah diklasifikasi dan di prediksi akan ditampilkan pada *user interface* sebagai info kendaraan. Berikut ini merupakan diagram alir dari sistem:

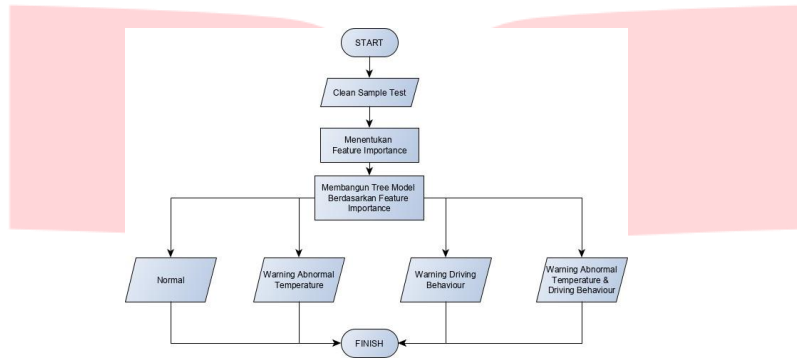


Gambar 2 : Diagram alur sistem.

3.1 Cleaning dan Labeling Data

Cleaning dan Labeling data adalah proses awal dari seluruh sistem. Pada proses ini data mentah yang diambil dari OBD-II yang dimodifikasi dikumpulkan pada database. Data yang dikumpulkan kemudian dilakukan cleaning sehingga menghasilkan data yang relevan agar machine learning dapat dengan mudah mengenali pattern dari data tersebut. Pada proses cleaning data terdapat perubahan data berdasarkan data mentah yang diambil.

3.2 Klasifikasi



Gambar 3.2 : Diagram Alir Klasifikasi

Prinsip dari klasifikasi CART yaitu menghitung nilai gini impurity atau bidang pemisah antara class. Pada proses pengujian Decision Tree CART menggunakan persamaan sebagai berikut.

$$Gini = 1 - \sum_{i=1}^c (p_i)^2 \tag{3.1}$$

Terdapat proses learning dari Training set yang menyimpan ciri latih dari masing-masing data OBD-II yang digunakan untuk referensi data OBD-II untuk menentukan kelas.

3.3 Performansi Sistem

Evaluasi kerja sistem dilakukan setelah seluruh tahap penelitian dan pengujian selesai. Evaluasi ini bertujuan untuk mengetahui keberhasilan sistem yang telah dirancang untuk memprediksi keadaan mobil abnormal atau normal. Berikut parameter evaluasi kerja sistem yang prediksi keadaan mobil pada tugas akhir ini yaitu performansi dengan parameter seperti:

1. Akurasi

Akurasi merupakan banyaknya klasifikasi yang berhasil pada sistem ini. Akurasi dapat diandalkan jika distribusi data klasifikasi di sistem rata, dalam artian jumlah kelas satu dan kelas lainnya seimbang.

$$Akurasi = \left(\frac{\Sigma \text{prediksi benar}}{\Sigma \text{data}} \times 100 \right) \% \tag{3.2}$$

2. Presisi

Presisi merupakan data yang berhasil diprediksi dalam suatu klasifikasi pada sistem ini, dimana kelas yang akan diamati yaitu mobil dengan keadaan abnormal. Akurasi digunakan jika distribusi data tidak terdistribusi secara merata.

$$Presisi = \left(\frac{\Sigma \text{prediksi suatu kelas}}{\Sigma \text{data suatu kelas oleh algoritma}} \times 100 \right) \% \tag{3.3}$$

3. Sensitivitas

Sensitivitas merupakan proporsi data yang berhasil diprediksi terhadap semua data aktual yang telah di klasifikasi.

$$\text{Presisi} = \left(\frac{\Sigma \text{prediksi suatu kelas}}{\Sigma \text{data aktual suatu kelas}} \times 100 \right) \% \quad (3.4)$$

4. F1 Score

F1 Score merupakan harmonic means dari presisi dan sensitivitas dimana dalam sebuah kasus, kedua parameter ini perlu digabungkan agar merepresentasikan keduanya. F1 score dapat dijadikan parameter penentu sebuah sistem.

$$\text{F1 Score} = \left(\frac{2 \times \text{presisi} \times \text{sensitivitas}}{\text{presisi} + \text{sensitivitas}} \right) \quad (3.5)$$

4. Hasil dan Analisis

4.1 Alat dan Perangkat

Berdasarkan perancangan yang sudah dijelaskan sebelumnya, maka dibutuhkan perangkat yang merupakan aspek pendukung seperti perangkat keras. Perangkat yang digunakan sebagai berikut:

1. ELM 327 OBD-II interface
2. Arduino Nano
3. SIM 800L

Perancangan sistem tidak hanya membutuhkan perangkat keras, tetapi perangkat lunak untuk menunjang kebutuhan dari sistem. Perangkat lunak yang digunakan sebagai berikut:

1. Cloud Computing Domainsia
2. Scikit-learn Library

4.2 Pengujian Sistem

Pengujian sistem merupakan langkah yang dilakukan untuk mengetahui apakah sistem tersebut layak atau tidak untuk diimplementasikan. Tujuan dari pengujian sistem ini adalah :

1. Mengetahui keluaran yang dihasilkan dari sistem ini apakah sesuai dengan prediksi dari keluaran sebenarnya.
2. Mengetahui parameter-parameter yang mempengaruhi sistem dan output yang dihasilkan agar dapat mengetahui kekurangan dan kelebihan dari sistem.

4.3 Tahap Pengujian Sistem

Dalam pengujian penelitian ini, terdapat dataset dari OBD-II yang telah di preprocess agar dimengerti oleh algoritma. Data sensor yang mempengaruhi keluaran yaitu RPM mobil, temperatur mesin, durasi mobil menyala, dan engine load

Berikut merupakan pengujian berdasarkan tujuan yang telah dijelaskan sebagai berikut:

1. Tahap Preprocessing

Tahap pertama ini, dataset yang telah diperoleh akan dilakukan proses pre-processing. Dalam proses ini pada dataset mentah akan dilakukan cleaning data yang tidak diperlukan sehingga menyisakan data yang berpengaruh pada keluaran prediksi yaitu RPM, temperatur mesin, durasi mobil menyala dan engine load. Labeling dilakukan setelahnya untuk memberi label pada data sehingga sistem tau bahwa data tersebut masuk ke golongan klasifikasi yang ada.

2. Tahap Training

Pada tahap kedua ini dataset yang telah di pre-processing dijadikan data latih dimana *pattern* dari suatu data akan dipelajari oleh algoritma, dari *pattern* tersebut kemudian dibuatlah training model untuk memprediksi keluaran data. Nilai *gini impurity* dikalkulasikan terhadap parameter yang berpengaruh kemudian dibandingkan satu dengan lainnya dengan nilai tertinggi berbanding jumlah sampel akan menjadi prioritas bidang pemisah untuk klasifikasi data, jika dari sebuah data *gini impurity* = 0, maka data tersebut tidak bisa dicabangkan lebih jauh lagi sehingga dapat disimpulkan semua data sudah terklasifikasi dengan sempurna. Jika masih terdapat nilai *gini impurity*, sistem akan kembali mengklasifikasikan data berdasarkan referensi data latih hingga syarat *maxdepth* ataupun toleransi *threshold gini impurity* tercapai. *Maxdepth* merupakan

batas seberapa dalam *decision tree* dapat berkembang untuk mengklasifikasikan, klasifikasi dilakukan dengan melihat parameter- parameter yang mempengaruhi label data.

4.4 Pengujian Sistem dan Analisis

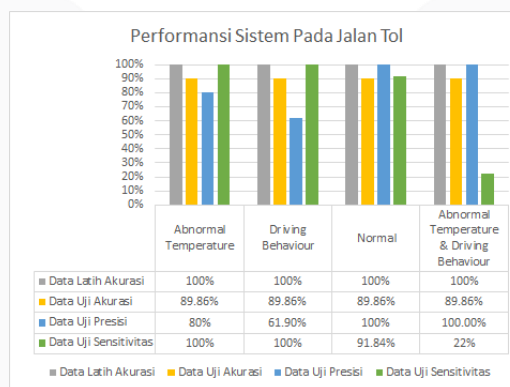
Pengujian sistem pada Tugas Akhir ini dilakukan pelatihan untuk menghasilkan sistem yang dapat memprediksi keadaan mobil dengan ketepatan yang maksimal.

1. Tahap awal dengan melakukan pengambilan data latih pada satu mobil dengan berbagai macam medan seperti jalan tol, jalan kota, dan rural area. Pada tahap ini data diambil selama dua hari untuk durasi pengambilan data yaitu dua jam setiap harinya. Mobil yang digunakan yaitu 2 macam mobil dengan transmisi manual dan transmisi auto.
2. Tahap kedua dengan melakukan perhitungan akurasi sistem terhadap prediksi data latihnya sendiri untuk menjadi alat ukur apakah training sistem bekerja sebagaimana harusnya.
3. Tahap ketiga adalah mengambil data uji dengan skenario yaitu pada jalan tol dimulai dari Gerbang Tol Buah Batu sampai Gerbang Tol Pasteur dan jalan kota dimulai dari Gerbang Tol Pasteur sampai Jalan Supratman. Data diambil selama 20 menit pada masing-masing skenario yang kemudian akan dihitung akurasinya untuk menjadi alat ukur sistem apakah sistem dapat memprediksi data yang tidak ada pada data latihnya dan menjadi acuan apakah performansi sistem bagus atau tidak.

4.4.1 Pengujian Sistem di Jalan Tol Terhadap Performansi

Skenario ini merupakan pengujian sistem dengan medan jalan tol dimulai dari Gerbang Tol Buah Batu sampai Gerbang Tol Pasteur dengan *Maxdepth* = 3 terhadap performansi yang dijelaskan sebelumnya. Adapun grafik performansi pada skenario ini sebagai berikut:

Tabel 4.1: Analisis Parameter Performansi Terhadap Klasifikasi Pada Skenario Jalan Tol



Dari grafik diatas dapat dihitung nilai F-1 score sebagai pembandingan performansi sistem terhadap sistem lainnya. Hasil dari perhitungan F-1 score dapat dilihat dari tabel berikut:

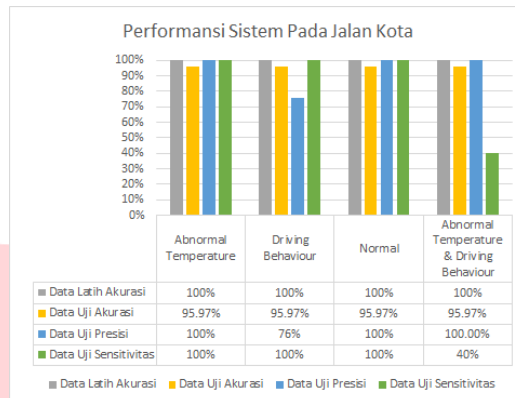
Tabel 4.2: F-1 Score pada skenario jalan tol

Kondisi	F1-Score
Abnormal Temperature	89%
Driving Behaviour	76%
Normal	95.74%
Abnormal Temperature & Driving Behaviour	36%
Average	74.37%

4.4.2 Pengujian Sistem di Jalan Kota Terhadap Performansi

Skenario ini merupakan pengujian sistem dengan medan jalan kota dimulai dari Gerbang Tol Pasteur sampai Jalan Supratman dengan *Maxdepth* = 3 terhadap performansi yang dijelaskan sebelumnya. Adapun grafik performansi pada skenario ini sebagai berikut:

Tabel 4.2: Analisis Parameter Performansi Terhadap Klasifikasi Pada Skenario Jalan Kota



Berdasarkan hasil grafik diatas dapat dihitung nilai F-1 score sebagai pembanding performansi sistem terhadap sistem lainnya. Hasil dari perhitungan F-1 score dapat dilihat dari tabel berikut:

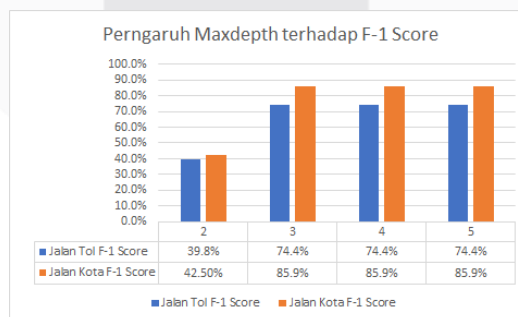
Tabel 4.2: F-1 Score pada skenario jalan kota

Kondisi	F1-Score
Abnormal Temperature	100%
Driving Behaviour	86.36%
Normal	100%
Abnormal Temperature & Driving Behaviour	57.14%
Average	85.88%

4.4.3 Pengaruh Maxdepth terhadap Performansi

Pada tahap ini akan diberikan hasil dari pengujian pengaruh *Maxdepth* terhadap performansi sistem yang diwakilkan dengan nilai rata-rata F-1 Score dengan data sample pada skenario jalan tol dan jalan kota. Berikut gambar pengaruh *maxdepth* terhadap perbandingan F-1 Score pada skenario jalan tol dan kota:

Tabel 4.4: Pengaruh Maxdepth terhadap Sensitifitas maxdepth



Hasil yang didapat menunjukkan bahwa rata-rata F-1 Score pada maxdepth 3,4, dan 5 pada skenario jalan kota maupun jalan tol hasilnya konstan dengan nilai 74.4% untuk jalan tol dan 85.9% untuk jalan kota, hal tersebut terjadi dikarenakan tree tidak dapat tumbuh lebih dalam karena tidak adanya data yang dapat diklasifikasi kembali. Pada maxdepth = 2, F-1 Score turun pada kedua skenario dikarenakan under-fitting terjadi. Under-

fitting dapat terjadi karena sistem terlalu sederhana untuk mengklasifikasikan data ataupun data latih yang kurang menunjukkan variasi terhadap distribusi klasifikasi.

5. Kesimpulan

1. Sistem dapat dikatakan baik karena performansi dari sistem terbilang tinggi dimana akurasi training mencapai 100%, namun belum bisa memprediksi dengan baik pada kelas "Driving Behaviour" dan "Abnormal Temperature & Driving Behaviour" dimana data belum cukup untuk menjelaskan karakteristik pada kelas tersebut.
2. Berdasarkan perhitungan Presisi dan F1-Score, terjadi *underfitting* pada maxdepth 2 yang berarti sistem terlalu sederhana untuk memprediksi data *sample* yang lebih kompleks dan *tree* tidak bisa tumbuh lebih besar diatas $maxdepth = 3$ dikarenakan kurangnya parameter sebagai bidang pemisah.
3. Pada scenario jalan tol inputan parameter lebih dinamik dibandingkan dengan inputan parameter skenario jalan kota sehingga sistem mengalami kesulitan untuk memprediksi data uji pada scenario jalan tol.

6. Daftar Pustaka

- [1] M. Amarasinghe, S. Kottegoda, A. L. Arachchi, S. Muramudalige, H. D. Bandara, and A. Azeez, "Cloud-based driver monitoring and vehicle diagnostic with obd2 telematics," in 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2015, pp. 243–249.
- [2] A. Arroyo. Business intelligence and its relationship with the big data, data analytics and data science. Accessed: 2019-07-21. [Online]. Available: <https://www.linkedin.com/pulse/business-intelligence-its-relationship-big-data-geekstyle>
- [3] L. Breiman, Classification and regression trees. Routledge, 2017.
- [4] D. A. Chawla, "Control system in automobile," p. 36, 2017.
- [5] T. M. Cover, P. Hart et al., "Nearest neighbor pattern classification," IEEE transactions on information theory, vol. 13, no. 1, pp. 21–27, 1967.
- [6] dewangNautiyal. Underfitting and overfitting in machine learning. Accessed: 2019-07-21. [Online]. Available: <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>
- [7] E. Directive, "98/69/ec of the european parliament and of the council of 13 october 1998 relating to measures to be taken against air pollution by emissions from motor vehicles and amending council directive 70/220/eec," Official Journal of the European Communities L, vol. 350, no. 28, p. 12, 1998.
- [8] C. Donalek, "Supervised and unsupervised learning," in Astronomy Colloquia. USA, 2011.
- [9] E. ELETRONICS, "Elm327-obd to rs232 interpreter," ELM Electronics Datasheets, 2015.
- [10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "Density-based spatial clustering of applications with noise," in Int. Conf. Knowledge Discovery and Data Mining, vol. 240, 1996, p. 6.
- [11] GAIKINDO, "Gaikindo wholesales data jan-dec 2018," 2018.
- [12] S. S. Haykin et al., Neural networks and learning machines/Simon Haykin. New York: Prentice Hall, 2009.
- [13] R. A. Howard, "Dynamic programming and markov processes." 1960.
- [14] B. P. S. Indonesia, "Statistik transportasi darat 2017" 2017.
- [15] ISO, "11898-1: 2015–road vehicles–controller area network (can)–part 1: Data link layer and physical signalling," International Organization for Standardization, 2015.
- [16] A. D. JoSEP, R. KATz, A. KonWinSKi, L. Gunho, D. PAttERSON, and A. RABKin, "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, 2010.
- [17] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 7, pp. 881–892, 2002.
- [18] Q. V. Le, A. J. Smola, and S. Vishwanathan, "Bundle methods for machine learning," in Advances in neural information processing systems, 2008, pp. 1377–1384.
- [19] T. Miller. Elm327: Top 10 adapters, apps, software review 2019 [and more]. Accessed: 2019-07-21. [Online]. Available: <https://www.obdadvisor.com/elm327/>
- [20] R. H. Myers and R. H. Myers, Classical and modern regression with applications. Duxbury press Belmont, CA, 1990, vol. 2.
- [21] H. Saputra, T. Atmaja, and D. Subagio, Teknologi Sensor Otomotif, 01 2017.
- [22] O. B. D. Solution. What is obd? Accessed: 2019-07-19. [Online]. Available: <https://www.obdsol.com/knowledgebase/on-board-diagnostics/what-is-obd/>
- [23] A. Zheng, N. Shelby, and E. Volckhausen, "Evaluating machine learning models," 2015.