Klasifikasi Data Trafik Jaringan dengan Framework Big Data Analitik

I Made Suartana[1], Ricky Eka Putra[2], Aditya Prapanca[3]

[1,2,3] Jurusan Teknik Informatika, Universitas Negeri Surabaya
[1]madesuartana@unesa.ac.id
[2]rickyeka@unesa.ac.id
[3]adityaprapanca@unesa.ac.id

***Abstrak*— Pada era Big Data perkembangan data menjadi sangat pesat. Berbagai teknologi menghasilkan data sangat cepat dengan berbagai format dan struktur data. Masifnya data menjadi permasalahan dalam pengolahan dan analisis data untuk mendapatkan informasi terkait data tersebut. Masifnya data juga menjadi masalah dalam pengelolaan jaringan komputer. Dengan perkembangan teknologi jaringan yang mengarah ke virtualisasi jaringan dan semakin banyaknya penggunaan IoT menghasilkan data yang semakin beragam. Beragamnya data menjadi permasalahan sendiri dalam manajemen, monitoring dan keamanan jaringan. Data trafik jaringan yang dihasilkan berbagai perangkat memiliki struktur yang beragam dan jumlah data yang semakin besar menyulitkan mekanisme pemantauan dan pendeteksi serangan untuk menemukan pola serangan. Munculnya konsep Big Data Analitik menjadi satu solusi dalam manajemen data, Big data analitik adalah proses mendapatkan informasi yang bermanfaat lewat analisis berbagai jenis kumpulan data yang berukuran sangat besar. Pada penelitian ini menggunakan pendekatan big data analitik untuk mengolah data trafik jaringan. Tujuannya untuk mengetahui sejauh mana pemanfaatan big data analitik dalam mengelompokkan data normal atau yang diindikasikan serangan pada jaringan. Penelitian ini menggunakan dataset UNSW-NB15. Spark dipilih sebagai framework big data analitik yang digunakan dalam proses klasifikasi trafik jaringan. Pendekatan big data analitik dengan menggunakan framework Spark dpat digunakan sebagai environment dalam mengolaj data dengan uji-coba klasfikasi model yang dilakukan didapatkan akurasi 85% dengan efisiensi model (recall) sebesar 85%.**

***Kata Kunci*— Big Data Analitik, Klasifikasi, Keamanan Jaringan, Apache Spark, Decision tree.**

I. PENDAHULUAN

Kompleksitas jaringan modern menjadi semakin meningkat, perkembangan virtualisasi jaringan dan IoT menambah kompleksitas pada infrastruktur jaringan secara keseluruhan. Akibatnya, pemantauan dan mekanisme keamanan jaringan juga menjadi lebih kompleks. Semakin kompleksnya jaringan mengakibatkan proses pemantauan dan pendeteksian serangan berdasarkan trafik jaringan menjadi lebih sulit dilakukan. Sedangkan pemantauan dan mekanisme pendeteksi serangan dalam jaringan sangat dibutuhkan. Pemantauan dan analisis dapat membantu memecahkan masalah jaringan, memvalidasi konfigurasi, dan meningkatkan keamanan jaringan.

Pemantauan dan deteksi serangan berdasarkan data lalu lintas pada jaringan komputer adalah tugas yang rumit. Beberapa permasalahan seperti: volume lalu lintas yang besar, kecepatan sistem transmisi, perkembangan dari layanan, perkembangan jenis serangan, dan berbagai sumber data dan metode untuk memperoleh data trafik. Akibat kompleksitas jaringan terus meningkat, diperlukan lebih banyak titik untuk pengamatan dan berpotensi menghasilkan data yang heterogen yang harus dikumpulkan dan dievaluasi. Hal ini menyulitkan untuk menerapkan mekanisme yang efisien untuk melakukan analisis dan klasifikasi dari aliran data dalam skala besar.

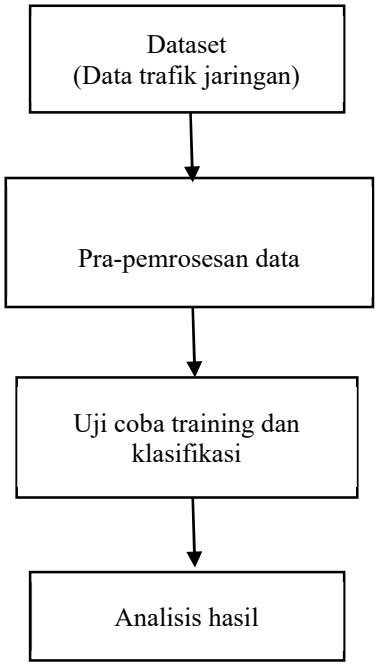
Dengan munculnya teknologi big data, yaitu situasi di mana volume data, kecepatan, kebenaran, dan variasi adalah tantangan utama untuk memungkinkan ekstraksi nilai dari data. Memungkinkan menjadi suatu solusi dari kompleksnya klasifikasi data lalu lintas jaringan. Beberapa peneliti menggunakan pendekatan teknologi big data untuk memproses informasi dalam berbagai bidang seperti [1] Melakukan analisis arsitektur ekologi Hadoop, proses dasar analisis data besar berdasarkan Hadoop. [2] Memanfaatan analitik data besar yang efektif dan berkelanjutan dalam pembuatan kebijakan dan penciptaan inovasi digital. Dalam bidang keamanan jaringan penggunaan framework big data pada penelitian [3] yang fokus pada *Volume, Veracity*, dan *Variety* karakteristik data besar dalam lalu lintas jaringan dan serangan. Penelitian [4] membahas keamanan jaringan dan strategi perlindungan big data. Tantangan dan solusi potensial untuk melindungi big data dalam komputasi awan dibahas pada [5]. Penggunaan arsitektur big data dalam monitoring dan keamanan jaringan pada penelitian [6] dan[7]. Sedangkan penerapan pembelajaran mesin dalam keamanan jaringan seperti pada penelitian [8]. Framework Big Data, dan pembelajaran mesin adalah beberapa contoh teknologi yang membantu aplikasi untuk menangani kumpulan data. Diperlukan eksplorasi lebih lanjut apakah proses monitoring dan keamanan jaringan dapat sepenuhnya dapat memanfaatkan potensi teknologi Big Data.

Dalam penelitian ini mencoba memanfaatkan potensi framework Big Data Analitik untuk mekanisme monitoring dan keamanan pada jaringan. Uji coba penelitian dengan melakukan klasifikasi trafik jaringan dengan menggunakan framework spark. Library Mlib pada framework spark mendukung kalsifikasi dan metode analisis data menggunakan pembelajaran mesin. Dataset yang digunakan pada penelitian ini menggunakan UNSW-NB 15.

II. METODOLOGI

Penelitian ini untuk menerapkan big data analitik framework dalam melakukan klasifikasi terhadap data trafik jaringan, untuk melakukan analisis terhadap pola-pola serangan

keamanan yang pada data trafik jaringan. Proses klasifikasi menggunakan Spark framework dan dataset sebagai data uji. Tahapan penelitian digambarkan pada Gbr.1, dimulai dari persiapan dataset yang digunakan selanjutnya tahap pre-processing dataset untuk pemilihan fitur data yang akan dipakai pada proses klasifikasi. Selanjutnya tahap klasifikasi dengan menggunakan algoritma pembelajaran mesin. Tahap terakhir dilakukan evaluasi terhadap hasil klasifikasi. Semua tahapan penelitian tersebut dibagun dengan pendekatan big data analitik dengan menggunakan framework Apache Spark.



Gbr. 1 Tahapan penelitian

A. *Pendekatan Big Data Analitik*

Analisis big data besar mengidentifikasi tren, pola, dan korelasi dalam jumlah besar data yang tidak terstruktur untuk proses analisis dan pengambilan keputusan berbasis data. Teknik ini menggunakan teknologi modern untuk menerapkan teknik seperti pengelompokan dan regresi, ke kumpulan data. Proses analitik big data, meliputi proses:

1) pengumpulan data terstruktur dan tidak terstruktur: dari berbagai sumber, seperti log trafik jaringan, aplikasi paket analisis jaringan dan aplikasi monitoring jaringan. Selanjutnya data akan disimpan dalam format penyimpanan data.

2) Pra-pemrosesan Data: Setelah data dikumpulkan dan disimpan, data tersebut harus dipilah sesuai dengan kebutuhan proses selanjutnya perubahan data dari data tidak terstruktur menjadi data terstruktur sesuai dengan framework big data.

3) *Pembersihan Data:* pembersihan data mengacu pada proses untuk menentukan data yang tidak akurat, tidak lengkap, atau tidak masuk akal dan kemudian mengubah atau menghapus data tersebut untuk meningkatkan kualitas data. Kerangka kerja umum untuk pembersihan data terdiri dari lima langkah:

- Mendefinisikan dan menentukan jenis kesalahan,
- Cari dan identifikasi contoh kesalahan,
- Perbaiki kesalahan,
- Mendokumentasikan contoh kesalahan dan jenis kesalahan, dan
- Memodifikasi prosedur entri data untuk mengurangi kesalahan di masa mendatang.

4) *Analisis Data:* Algoritma analitik data digunakan pada tahap ini untuk menerjemahkan sejumlah besar data menjadi informasi yang berguna. Proses ini memilah-milah kumpulan data yang sangat besar untuk mengidentifikasi pola dan asosiasi, hal ini dicapai dengan pengelompokan atau klasifikasi data. AI dan metode pembelajaran mesin juga digunakan dalam tahap ini untuk menemukan pola dalam data yang paling kompleks dan abstrak.

B. *Dataset*

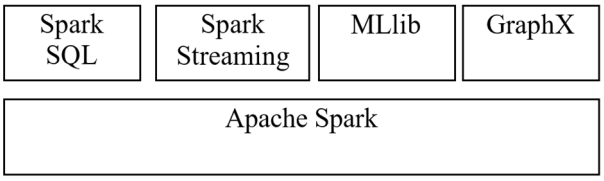
Dataset yang digunakan adalah UNSW-NB 15. UNSW-NB adalah kumpulan data yang dibuat di Cyber Range Lab Australia menggunakan alat IXIA-PerfectStorm untuk mengekstraksi gabungan dari aktivitas alami modern yang ntaya dan simulasi perilaku serangan yang dihasilkan oleh lalu lintas jaringan. Dataset ini berisi 49 fitur yang dikategorikan ke dalam lima kelompok [9]. Tabel 1 berikan kategori jenis serangan yang diklasifikasikan ke dalam sembilan kelompok

TABEL I
JENIS SERANGAN UNSW-NB15.

Tipe	Deskrips
Fuzzers	Penyerang mencoba membuat program atau jaringan ditangguhkan dengan memberinya data yang dihasilkan secara acak.
Analysis	Serangan menembus aplikasi web melalui port (port scan), skrip web (file HTML), dan email (spam).
Backdoor	Suatu teknik di mana mekanisme keamanan sistem dilewati untuk mengakses komputer atau datanya.
DoS	Intrusi yang mencoba membuat sumber daya jaringan atau server tidak tersedia bagi pengguna, umumnya dengan menangguhkan sementara layanan dari host yang terhubung ke Internet.
Exploit	Itu memanfaatkan kesalahan, bug, atau kerentanan yang disebabkan oleh perilaku yang tidak disengaja pada jaringan atau host.
Generic	Sebuah teknik menetapkan terhadap setiap block-cipher menggunakan fungsi hash untuk bertabrakan tanpa konfigurasi block-cipher.
Reconnaissance	Itu mengumpulkan informasi tentang jaringan komputer untuk menghindari kontrol keamanannya.
Shellcode	Penyerang menembus sedikit kode mulai dari shell untuk memeriksa mesin yang disusupi.
Worm	Penyerang mereplikasi dirinya sendiri untuk maju di komputer lain. Seringkali, ia menggunakan jaringan komputer untuk menyebarkan dirinya sendiri, bergantung pada kegagalan keamanan pada komputer target untuk mengaksesnya.
Fuzzers	Penyerang mencoba membuat program atau jaringan ditangguhkan dengan memberinya data yang dihasilkan secara acak.

C. *Big Data Analitik Framework (Spark)*

Apache Spark, adalah mesin pemrosesan data hibrida yang kuat, dapat diskalakan, dan terdistribusi dengan cepat, proyek yang bersifat *open-source* paling aktif untuk Big Data. Spark dikembangkan di UC Berkeley pada tahun 2009. Spark menyediakan API dalam bahasa Scala, Java, Python, dan R. Untuk mengolah data yang sangat besar, Spark harus cukup cepat dengan memproses data besar sekaligus. Oleh karena itu, arsitektur Spark tersedia dalam mode *cluster*, bukan di satu mesin. Hasil proses yang dijalankan oleh Spark tidak ditulis ke disk tetapi disimpan di memori. Kemampuan *all-in-memory* ini adalah teknik komputasi performa tinggi untuk analitik tingkat lanjut, membuat Spark 100 kali lebih cepat daripada Hadoop. Spark juga memiliki ekosistem pustaka yang dapat digunakan untuk Pembelajaran Mesin, kueri interaktif yang dapat memiliki implikasi penting untuk produktivitas. Spark telah diperkaya secara progresif untuk menyediakan ekosistem yang lengkap saat ini dukungan pustaka pada Spark ditunjukkan pada gbr 2[10].



Gbr 2. Ekosistem Apache Spark

D. *Proses Training dan Klasifikasi*

Proses Training dan klasifikasi serangan dilakukan Setelah persiapan data atau dataset yang dapat dieksekusi sebagai data input. Hasil dari eksekusi ini telah diolah terlebih dahulu untuk mendapatkan vektor fitur numerik. Vektor ini kemudian diteruskan sebagai input untuk proses klasifikasi, selanjutnya proses klasifikasi serangan ke dalam salah satu kelas serangan yang telah ditentukan sebelumnya [11].

Dalam uji coba proses klasifikasi dalam penelitian ini menggunakan metode Decision Tree. Decision Tree memperoleh seperangkat aturan keputusan untuk membangun pohon keputusan yang dapat digunakan untuk memprediksi label numerik dari suatu pengamatan. Sebuah pohon keputusan berbeda dari grafik dalam hal tidak ada loop; node non-leaf disebut sebagai node internal atau split, dan node daun disebut sebagai node terminal. Metode untuk pohon keputusan dimulai pada simpul akar dan turun ke simpul terminal. Algoritma pohon keputusan melakukan serangkaian tes pada fitur untuk memprediksi label. Meskipun pohon keputusan dapat digunakan untuk regresi dan klasifikasi, dalam hal ini digunakan untuk klasifikasi. Menggunakan dataset pelatihan, metode pohon keputusan memperoleh seperangkat aturan keputusan. Ini membangun pohon keputusan yang dapat digunakan untuk meramalkan label numerik dari suatu pengamatan. Node dan tepi pohon terstruktur secara hierarkis. Sebuah pohon keputusan berbeda dari grafik dalam hal tidak ada *loop*; *node non-leaf* disebut kembali sebagai node internal atau split, dan node daun disebut sebagai node terminal. Metode untuk pohon keputusan dimulai pada simpul akar dan turun ke simpul terminal. Algoritme pohon keputusan "melakukan serangkaian tes pada fitur untuk memprediksi label". Meskipun pohon keputusan dapat digunakan untuk regresi dan klasifikasi, dalam hal ini digunakan untuk klasifikasi.

E. *Metrik Akurasi*

Evaluasi kinerja proses learning dan klasifikasi berdasarkan pada metrik [6]

1) Akurasi: Didefinisikan sebagai persentase dari data yang terklasifikasi dengan benar terhadap jumlah total data.

$$A = TP + TN/TP + FP + FN + TN \quad (1)$$

2) Precision (P): Didefinisikan sebagai rasio % dari jumlah data true positive (TP) dibagi dengan jumlah true positive (TP) dan false positive (FP) data yang terklasifikasi.

$$P=TP/((TP+FP)) \times 100\% \quad (2)$$

3) Recall (R): Didefinisikan sebagai rasio % jumlah data true positive (TP) dibagi dengan jumlah true positive (TP)dan false negative (FN) yang diklasifikasikan catatan.

$$P=TP/((TP+FN)) \times 100\% \quad (3)$$

4) *F-Measure (F):* Didefinisikan sebagai rata-rata harmonik dari Precision (P) dan Recall (R) dan mewakili keseimbangan di antara keduanya.

$$F\ (2.P.R)/\ ((P+R)) \times 100\% \quad (4)$$

III. HASIL DAN PEMBAHASAN

Dalam uji coba penelitian ini sepenuhnya menggunakan lingkungan pengembangan dengan framework Spark. Tipe data Spark juga digunakan untuk mengunggah dan menganalisis kumpulan data dengan algoritma Decision Tree. Dalam penelitian ini menggunakan dataset untuk membentuk dan mengevaluasi model yang disediakan dalam pusaka Spark. Dataset uji kemudian digunakan untuk membuat prediksi. Penelitian ini menggunkan Jupyter Notebook dan Python API(PySpark). Framework Spark mendukung beberapa format data seperti *Resilent Distributed Dataset* (RDD) dan DataFrame. Pada proses berikut menggunakn Dataframe sebagai objek penyimpanan data sebelum data diproses pada tahap selanjutnya. Tahap pengumpulan data awal dilakukan dengan load dataset UNSW-NB15 dalam format CSV dan mengubahnya ke format Dataframe. Proses load data dan hasilnya bisa dilihat pada ilustrasi Gbr 3.

```
# Load data
initial_data = pd.read_csv('./UNSW_NB15_training-set.csv')
```

```
initial_data.head(n=5)
```

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_ltm	ct_dst_src_ltm
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	...	1	2
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	...	1	2
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	...	1	3
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	...	1	3
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	...	1	3

Gbr 3. Load dataset

Proses selanjutnya memeriksa apakah ada nilai yang hilang (*missing value*) dalam dataset. Salah satu Teknik pre-prosesing data untuk dataset yang tidak lengkap adalah membuang seluruh baris dan/atau kolom yang berisi nilai kosong. Pada penelitian ini akan menggunakan salah satu teknik pemrosesan awal data dengan menghilangkan baris dengan nilai yang hilang (*missing value*). Apache spark mendukung beberapa teknik untuk pre-prosesing data. Pada pembahasasn berikut di tampilkan satu contoh proses tersebut pada framework Spark seperti yang diilustrasikan pada gbr 4.

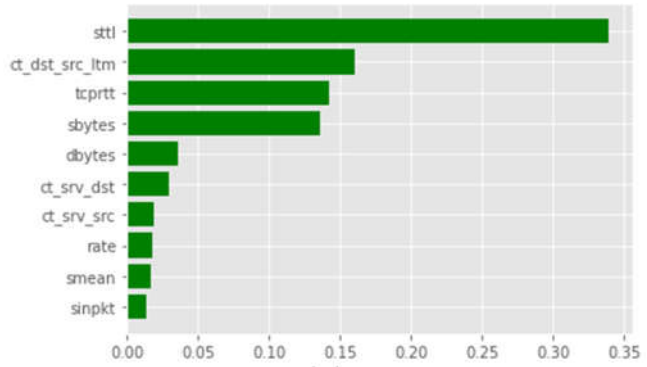
```
# hapus baris dengan nilai null
data_to_use = initial_data.dropna()

#Bentuk data: jumlah baris tetap sama karena tidak ada nilai nol yang dilaporkan
data_to_use.shape
```

```
(82332, 45)
```

Gbr 4. Menghapus missing value pada dataset

Fungsi *dropna()* dapat juga digunakan untuk menghilangkan redudansi pada data. Penghapusan duplikat (menghapus semua redudansi data) membantu proses klasifikasi serangan karena membuat bias sistem berkurang dan membuat perhitungan lebih cepat karena harus menangani lebih sedikit data [12].

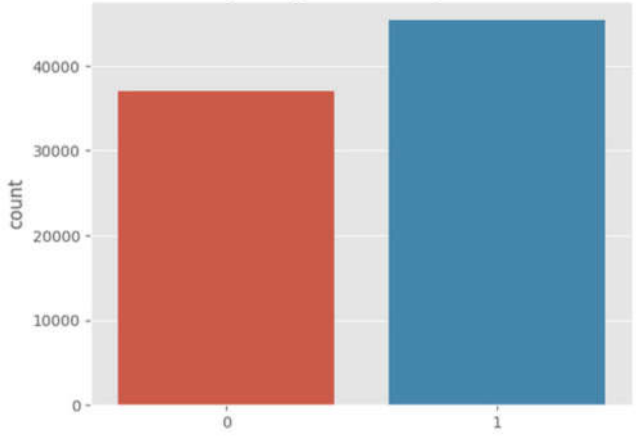
Fase selanjutanya pemilihan fitur yang bertujuan untuk memilih atribut yang relevan yang diperlukan untuk pengambilan keputusan. Penelitian ini menerapkan ekstraksi fitur *Importance* Decision Tree. Hasil dari ekstraksi fitur ini memberikan empat fitur terbaik yaitu: sttl, ct_dst_src_ltm, tcprtt, dan sbytes dan Gbr 5 memberi label representasi grafis dari fitur dan empat fitur teratas yang diproses dengan menggunakan Spark.



Gbr 5. Menghapus missing value pada dataset

Proses klasifikasi menggunakan algoritma Decision Tree yang tersedia pada lingkungan Pengembang Spark (MLlib). Dengan klasifikas akan memprediksi apakah data trafik jaringan termasuk normal atau termasuk serangan. Melatih klasifikasi Decision Tree menggunakan MLlib memerlukan beberapa parameter seperti: *training data, num classes, Categorical features info, Impurity metric, Tree maximum depth, dan tree maximum number of bins. Categorical features info* merupakan pemetaan dari kolom ke variable kategori. Variabel ini bersifat opsional, meskipun seharusnya meningkatkan akurasi model. Namun mengharuskan kita mengetahui level dalam variabel kategori terlebih dahulu. kedua kita perlu mem-parsing data untuk mengonversi label menjadi nilai integer dalam rentang arity.

Dalam melatih pohon klasifikasi nilai maxDepth dijaga agar tetap kecil sehingga akan menghasilkan akurasi yang lebih kecil, namun akan mengakibatkan lebih sedikit pemisahan sehingga nantinya dapat menginterpretasikan *tree* dengan lebih baik. Klasifikasi data akan menggunkan sejumlah data, karakteristik data terdiri dari data normal dan intrusi. Sebaran data normal dan serangan seperti terlihat pada Gbr 6.



Gbr 6. pengelompokan *record* data normal dan serangan

Apache Spark Machine Learning menyediakan rangkaian metrik untuk mengevaluasi kinerja model Machine Learning [13]. Untuk mengukur kinerja klasifikasi pada penelitian ini menggunakan: akurasi, *precision, recall*, dan *f1-score*. Hasil pengukuran matrik seperti terlihat pada gbr 7.

```
Accuracy: 0.8510901614568184
Reporting for ['Decision Tree Classifer', 'RegLog']:
              precision    recall  f1-score   support

           0       0.69      0.98      0.81     56000
           1       0.99      0.79      0.88    119341

    accuracy                           0.85    175341
   macro avg       0.84      0.88      0.84    175341
weighted avg       0.89      0.85      0.86    175341
```

Gbr 6. Hasil Matrik evaluasi

Berdasarkan hasil matrik konvolusi didapat nilai akurasi sebesar 0.85, ini berarti berdasarkan proses klasifikasi yang dilakukan total data yang diklasifikasikan dengan benar dibagi dengan jumlah keseluruhan data sebesar 85%. Hasil matriks

presisi untuk data normal 0.89, dalam klasifikasi yang dilakukan prediksi benar aktual dibagi dengan prediksi total benar yang dibuat oleh model 89%. Nilai *recall* untuk prediksi jumlah positif benar dibagi jumlah total kasus positif sejati dari hasil yang didapatkan 0.85. Untuk nilai *F1-Score* sebagai rata-rata dari presisi dan recall sebesar 0.86.

Pendekatan big data analitik dengan menggunakan framework Spark dpat digunakan sebagai environment dalam mengolaj data dengan uji-coba klasfikasi model yang dilakukan didapatkan akurasi 85% dengan efisiensi model (recall) sebesar 85%.

KESIMPULAN

Penelitian ini berfokus pada metode dan teknologi untuk mengelola data trafik jaringan, dengan pembahasan terkait klasifikasi data dengan algoritma pembelajaran mesin. Pendekatan big data digunakan untuk mengetahui seberapa potensi big data analitik dapat dimanfaatkan dalam mengelola data trafik jaringan. Dari percobaan didapat hasil Pendekatan framework Big Data Analitik dapat digunakan untuk proses transformasi berbagai tipe dan struktur data (contoh dalam penelitian ini mengguakan dataset dan data log jaringan) untuk diubah ke data terstruktur dengan menggunakan Apache Spark (berbasis Hadoop). Proses klasifikasi data pada framework Spark menggunkan library Mllib dengan beberapa pre-define algoritma pembelajaran mesin dapat melakukan klasifikasi pada data untuk keperluan analitik dan pengambilan keputusan dalam manajemen jaringan. Pendekatan big data analitik dengan menggunakan framework Spark dpat digunakan sebagai environment dalam mengolaj data dengan uji-coba klasfikasi model yang dilakukan didapatkan akurasi 85% dengan efisiensi model (*recall)* sebesar 85%.

UCAPAN TERIMA KASIH

Terimakasih kepada Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) UNESA melalui skema penelitian kebijakan fakultas teknik, yang telah membrikan dukungan pembiayaan dalam pelaksanaan penelitian ini.

REFERENSI

[1] H. Jiang, “Research and Practice of Big Data Analysis Process Based on Hadoop Framework,” in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Mar. 2019, pp. 2044–2047. doi: 10.1109/ITNEC.2019.8729522.

[2] P. Thamjaroenporn and T. Achalakul, “Big Data Analytics Framework for Digital Government,” in *2020 1st International Conference on Big Data Analytics and Practices (IBDAP)*, Sep. 2020, pp. 1–6. doi: 10.1109/IBDAP50342.2020.9245461.

[3] L. Wang and R. Jones, “Big Data Analytics in Cyber Security: Network Traffic and Attacks,” *Journal of Computer Information Systems*, vol. 61, no. 5, pp. 410–417, Sep. 2021, doi: 10.1080/08874417.2019.1688731.

[4] I. el Alaoui and Y. Gahi, “Network Security Strategies in Big Data Context,” *Procedia Comput Sci*, vol. 175, pp. 730–736, 2020, doi: 10.1016/j.procs.2020.07.108.

[5] G. Manogaran, C. Thota, and M. V. Kumar, “MetaCloudDataStorage Architecture for Big Data Security in Cloud Computing,” *Procedia Comput Sci*, vol. 87, pp. 128–133, 2016, doi: 10.1016/j.procs.2016.05.138.

[6] S. Marchal, X. Jiang, R. State, and T. Engel, “A Big Data Architecture for Large Scale Security Monitoring,” in *2014 IEEE International Congress on Big Data*, Jun. 2014, pp. 56–63. doi: 10.1109/BigData.Congress.2014.18.

[7] P. Casas, A. D’Alconzo, T. Zseby, and M. Mellia, “Big-DAMA,” in *Proceedings of the 2016 workshop on Fostering Latin-American Research in Data Communication Networks*, Aug. 2016, pp. 1–3. doi: 10.1145/2940116.2940117.

[8] I Made Suartana, “Analisis Penerapan Deep Learning untuk Klasifikasi Serangan Terhadap Keamanan Jaringan,” *KLIK-KUMPULAN JURNAL ILMU KOMPUTER*, vol. 9, no. 1, pp. 100–109, 2022.

[9] M. Albanese *et al.*, “Recognizing Unexplained Behavior in Network Traffic,” 2014, pp. 39–62. doi: 10.1007/978-1-4614-7597-2_3.

[10]“https://spark.apache.org/.”

[11]R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, “Malware classification with recurrent networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 1916–1920. doi: 10.1109/ICASSP.2015.7178304.

[12]M. F. Umer, M. Sher, and Y. Bi, “Flow-based intrusion detection: Techniques and challenges,” *Comput Secur*, vol. 70, pp. 238–254, Sep. 2017, doi: 10.1016/j.cose.2017.05.009.

[13]R. Verma, M. Kantarcioglu, D. Marchette, E. Leiss, and T. Solorio, “Security Analytics: Essential Data Analytics Knowledge for Cybersecurity Professionals and Students,” *IEEE Secur Priv*, vol. 13, no. 6, pp. 60–65, Nov. 2015, doi: 10.1109/MSP.2015.121.