Madani: Jurnal Ilmiah Multidisiplin
Volume 2, Nomor 7, 2024, Halaman 276-283

E-ISSN: 2986-6340
DOI: https://doi.org/10.5281/zenodo.12562473

Mengoptimalkan Deteksi Intrusi Jaringan: Perbandingan SVM dan KNN Menggunakan Dataset KddCup99

Etha Felisya Br Purba[1], Rehana Putri Salsabilla[2], Nur Azka Rahadiansyah[3]
[123]Sepuluh Nopember Institute of Technology (ITS)
Email: felisyaetha@gmail.com[1], rehanaputri80@gmail.com[2], nurazkarahadian@gmail.com[3]

Abstrak

Keamanan jaringan menjadi semakin kritis dengan meningkatnya kompleksitas serangan siber. Sistem Deteksi Intrusi (IDS) berperan penting dalam memantau dan mengidentifikasi aktivitas mencurigakan secara real-time. Penelitian ini membandingkan dua algoritma pembelajaran mesin, Support Vector Machine (SVM) dan K-Nearest Neighbors (KNN), dalam mendeteksi serangan menggunakan dataset KDD Cup 99. Hasil eksperimen menunjukkan bahwa KNN unggul dalam hal akurasi, presisi, recall, dan F1-score dibandingkan SVM. KNN lebih baik dalam mengklasifikasikan data secara keseluruhan, sedangkan SVM efektif dalam mengelola false positives dan menangani data berdimensi tinggi. Kedua algoritma memiliki kelebihan dan kekurangan masing-masing, sehingga pemilihan algoritma harus disesuaikan dengan karakteristik data dan kebutuhan deteksi yang spesifik. Penelitian ini memberikan wawasan penting dalam memilih algoritma yang tepat untuk meningkatkan efektivitas deteksi intrusi jaringan di masa depan.
Kata kunci: Keamanan Jaringan, Sistem Deteksi Intrusi, Support Vector Machine, K-Nearest Neighbors, Pembelajaran Mesin, Deteksi Serangan

Abstract

Network security is becoming increasingly critical with the rising complexity of cyber attacks. Intrusion Detection Systems (IDS) play a crucial role in monitoring and identifying suspicious activities in real-time. This study compares two machine learning algorithms, Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), in detecting attacks using the KDD Cup 99 dataset. The experimental results show that KNN outperforms SVM in terms of accuracy, precision, recall, and F1-score. KNN is more effective in classifying overall data, while SVM is efficient in managing false positives and handling high-dimensional data. Both algorithms have their respective strengths and weaknesses, so the choice of algorithm should be tailored to the specific characteristics of the data and detection requirements. This research provides valuable insights into selecting the appropriate algorithm to enhance the effectiveness of network intrusion detection in the future.
Keywords: *Network Security, Intrusion Detection System, Support Vector Machine, K-Nearest Neighbors, Machine Learning, Attack Detection*

Article Info
Received date: 10 June 2024 Revised date: 18 June 2024 Accepted date: 23 June 2024

PENDAHULUAN

Dengan pesatnya pertumbuhan teknologi informasi, keamanan jaringan menjadi semakin penting mengingat serangan *cyber* yang semakin canggih dan beragam [1]. Serangan seperti pencurian data, perusakan sistem, atau penggangguan layanan dapat menimbulkan kerugian yang signifikan bagi perusahaan dan organisasi. Untuk melindungi sistem dan data mereka, banyak organisasi mengandalkan Sistem Deteksi Intrusi (IDS), yang berperan dalam memantau dan mengidentifikasi aktivitas mencurigakan dalam jaringan secara *real-time* [2].

IDS terdiri dari dua metode utama dalam mendeteksi serangan: deteksi berbasis anomali dan deteksi berbasis tanda tangan *(misuse detection)*. Deteksi anomali digunakan untuk membandingkan perubahan perilaku sistem untuk menemukan tindakan atau aktivitas yang tidak normal. Misalnya, jika terjadi lonjakan permintaan pengguna ke basis data yang tidak sebanding dengan pola historis, sistem akan mengeluarkan peringatan deteksi anomali [4]. Namun, kelemahan dari metode ini adalah saat berhadapan dengan lingkungan yang dinamis di mana pola permintaan pengguna berubah dari waktu ke waktu dan tidak sesuai dengan data historis [3], sehingga metode ini rentan terhadap kesalahan positif *(false positives)* [4].

Dalam upaya untuk meningkatkan kinerja IDS, pendekatan baru menggunakan teknik *Machine Learning* (ML) mulai digunakan. ML memungkinkan IDS untuk belajar dari data historis dan mengidentifikasi pola yang rumit dan tidak terstruktur yang mungkin sulit dideteksi oleh metode tradisional [5]. Dalam konteks ini, studi ini memfokuskan pada perbandingan dua algoritma ML populer, yaitu *Support Vector Machine* (SVM) dan *K-Nearest Neighbors* (KNN), dalam mendeteksi serangan terhadap dataset KddCup99. Dataset ini telah menjadi acuan dalam penelitian deteksi intrusi dan mencakup berbagai jenis serangan yang mungkin terjadi dalam lingkungan jaringan.

Penerapan KNN dan SVM dalam IDS memberikan keunggulan tersendiri. KNN tidak memerlukan pembelajaran yang rumit dan cocok digunakan untuk dataset dengan pola yang tidak jelas. Namun, KNN cenderung lambat dalam fase pengujian karena harus memeriksa setiap sampel dengan tetangga terdekatnya. Di sisi lain, SVM cenderung memberikan performa yang lebih baik dalam hal generalisasi terhadap data baru dan dapat menangani masalah klasifikasi pada data yang kompleks, meskipun memerlukan tuning parameter yang cermat.

Pada penelitian yang dilakukan oleh Rahul Vigneswaran et al. [6] KNN memberikan hasil evaluasi yang lebih baik daripada VSM, penelitian Li Yang et al. [7] juga menunjukkan bahwa hasil evaluasi KNN lebih baik daripada VSM begitu juga dengan penelitian yang dilakukan oleh Iram Abrar et al [8]. Namun, baik KNN maupun SVM perlu disesuaikan dengan karakteristik dataset yang spesifik dan tujuan deteksi serangan yang ingin dicapai.

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan performa SVM dan KNN dalam klasifikasi biner dan multikelas terhadap dataset KddCup99. Hasil dari penelitian ini diharapkan dapat memberikan wawasan yang lebih dalam tentang kelebihan dan kelemahan masing-masing pendekatan, serta membantu dalam memilih algoritma yang paling sesuai untuk meningkatkan efektivitas deteksi intrusi jaringan di masa depan.

Dalam bab-bab berikutnya, akan dibahas langkah-langkah metodologis yang digunakan, hasil evaluasi dari eksperimen yang dilakukan, serta analisis dan kesimpulan yang dapat ditarik dari studi ini. Diharapkan bahwa temuan dari penelitian ini dapat memberikan kontribusi signifikan dalam bidang keamanan jaringan dan pengembangan teknologi deteksi intrusi di masa depan.

METODE

Dataset KDD Cup 99

Sejak tahun 1999, KDD'99 [9] telah menjadi yang paling luas digunakan kumpulan data untuk evaluasi metode deteksi anomali. Kumpulan data ini disiapkan oleh Stolfo dkk. [10] dan dibangun berdasarkan data yang diambil dalam evaluasi IDS DARPA'98 program [11]. KDD Cup'99 membantu mengidentifikasi hubungan antara jenis serangan dan protokol yang dimanfaatkan oleh peretas, yang vital untuk deteksi intrusi.[12] Kami menggunakan subset dari dataset ini, yaitu Kdd_cup_10_percent untuk tahap pelatihan, sementara dataset *correct* digunakan untuk pengujian model.

Tabel 1 Distribusi kelas pada dataset NSL-KDD

Label Kelas	Training Set	Percentage	Test Set	Percentage
Normal	81.,814	75,611%	60.593	19.481%
Serangan DoS	247.267	23,002%	229.853	73,901%
Serangan Probe	13.860	1,289%	4.166	1,339%
Serangan R2L	999	0,093%	16.189	5,205%
Serangan U2R	52	0,005%	228	0,073%
Total	1.074.992	100%	311.029	100%

Persiapan dan Pra-Pemrosesan Dataset

1. Inisialisasi Dataset
   Dataset yang digunakan terdiri dari dua bagian utama: data latih dan data uji. Data ini diinisialisasi menggunakan file yang diberikan sebagai input saat inisialisasi kelas IDS.

2. Perubahan Label Kelas
   Label kelas dalam dataset awal diubah untuk membedakan antara kelas "normal" dan "serangan". Langkah ini dilakukan untuk memudahkan proses klasifikasi.
3. Encoding Fitur
   Fitur-fitur dalam dataset di encoding menggunakan *DictVectorizer* untuk mengubah data dalam format kamus menjadi matriks fitur biner.
4. Transformasi Data
   Setelah encoding, data latih dan data uji diubah bentuknya menggunakan PCA *(Principal Component Analysis)* untuk mereduksi dimensi fitur menjadi 27 komponen utama. Langkah ini membantu dalam meningkatkan efisiensi komputasi dan mengurangi kompleksitas model.
5. Normalisasi Dataset
   Data yang telah direduksi kemudian dinormalisasi menggunakan *StandardScaler* untuk memastikan bahwa semua fitur memiliki skala yang seragam dan memudahkan konvergensi algoritma *machine learning*.

Implementasi Algoritma Pembelajaran Mesin

1. Support Vector Machine (SVM)

SVM *(Support Vector Machine)* adalah salah satu algoritma *Machine Learning* yang digunakan untuk klasifikasi dan regresi. Prinsip utama dari SVM adalah mencari *hyperplane* terbaik yang memisahkan dua kelas data dalam ruang fitur. Algoritma SVM menghasilkan *hyperplane* yang akan membantu dalam klasifikasi. Mungkin ada dua SVM kelas atau multi kelas [22]. *Hyperplane* harus melakukannya ditemukan berjarak sama dari kelas klasifikasi, jika tidak kesalahan klasifikasi mungkin terjadi. Ada margin untuk ini kedua kelas itu ditemukan menggunakan vektor dukungan. Ini vektor adalah vektor-vektor yang jaraknya berdekatan. Itu jarak antara margin kedua kelas dan *hyperplane* harus sama.

Implementasi algoritma SVM *(Support Vector Machine)* dimulai dengan memuat dan memproses dataset KDD Cup '99 yang telah dipersiapkan sebelumnya. Dataset ini meliputi rekaman transaksi jaringan yang mencakup berbagai jenis serangan dan aktivitas normal. Setelah memuat data, langkah pertama adalah menggunakan teknik PCA *(Principal Component Analysis)* untuk mereduksi dimensi fitur dari dataset. Proses reduksi dimensi dilakukan untuk mengurangi kompleksitas data dan mempercepat proses pelatihan model SVM.

Setelah fitur-fitur direduksi, data yang telah direduksi kemudian dinormalisasi menggunakan Standard Scaler untuk memastikan semua fitur memiliki skala yang serupa. Langkah normalisasi ini penting untuk menghindari bias dalam proses pembelajaran mesin.

Selanjutnya, SVM diterapkan menggunakan perpustakaan *scikit-learn* dengan parameter kernel linear. Model SVM dilatih menggunakan data latih yang telah direduksi dan dinormalisasi, dan kemudian diuji dengan data uji yang sama prosesnya. Hasil evaluasi model termasuk akurasi dari model SVM dan laporan klasifikasi yang menunjukkan matrik evaluasi seperti presisi, *recall*, dan F1-score untuk setiap kelas.

2. K-Nearest Neighbour (KNN)

KNN *(K-Nearest Neighbour)* adalah algoritma *Machine Learning* yang digunakan untuk klasifikasi dan regresi. Prinsip dasar dari KNN adalah mengklasifikasikan sebuah titik data berdasarkan kelas mayoritas dari k-titik data terdekat di ruang fitur. Algoritma ini bergantung pada tetangga terdekat[20]. Label juga diberikan pada kumpulan data pelatihan sehingga KNN yang diarahkan dapat diperiksa[21].

Implementasi algoritma KNN (K-Nearest Neighbors) dimulai dengan memuat dan memproses dataset yang sama, yaitu KDD Cup '99. Seperti pada SVM, data diolah dengan menggunakan teknik PCA untuk mereduksi dimensi fitur. Reduksi dimensi bertujuan untuk mempercepat komputasi dan mempertahankan informasi yang relevan dalam data.

Setelah fitur-fitur direduksi, dilakukan normalisasi data menggunakan Standard Scaler untuk menstabilkan distribusi data. Data yang telah dinormalisasi kemudian dibagi menjadi subset pelatihan dan pengujian.

Pada langkah selanjutnya, model KNN diimplementasikan dengan menggunakan pendekatan yang disederhanakan dalam kelas SimpleKNN. Model KNN ini dilatih menggunakan subset data pelatihan dan diuji menggunakan subset data pengujian yang telah dipersiapkan sebelumnya. Evaluasi

model KNN mencakup perhitungan akurasi berdasarkan prediksi terhadap kelas target yang sebenarnya.

Flowchart Alur Kerja

Gambar berikut ini menunjukkan *flowchart* dari proses metodologi yang digunakan dalam penelitian ini:

Gambar 1. Flowchart proses pada metodologi

HASIL DAN PEMBAHASAN

Pada bagian ini, kami akan membahas hasil dari implementasi model KNN dan SVM untuk deteksi intrusi menggunakan dataset yang disiapkan dengan beberapa metrik evaluasi. Metrik evaluasi yang digunakan untuk mengevaluasi kinerja algoritma Machine Learning terdiri dari empat faktor utama: *True Positive* (TP), *False Negative* (FN), *False Positive* (FP), dan *True Negative* (TN).

Akurasi (*Accuracy*)

Akurasi adalah salah satu metrik evaluasi paling umum digunakan untuk menilai kinerja algoritma ML. Hal ini disebabkan oleh kesederhanaannya dan kemudahannya dalam implementasi [13]. Akurasi mengukur seberapa banyak data uji yang diklasifikasikan dengan benar, biasanya disajikan dalam bentuk persentase. Namun, tidak disarankan untuk mengukur akurasi pada data pelatihan karena dapat menyebabkan *overfitting* di mana tingkat akurasi terlihat lebih tinggi daripada yang sebenarnya, menghasilkan hasil yang tidak dapat diandalkan [14]. Secara matematis, akurasi didefinisikan sebagai:

$$Accuracy = \frac{TP + TN}{P + N}$$

Presisi (*Precision*) dan *Recall*

Presisi dan *recall* sering kali diperlakukan bersamaan karena keduanya saling terkait. Presisi mengukur jumlah prediksi positif yang benar dari kelas positif. Sementara *recall* mengukur jumlah prediksi positif yang benar dari seluruh instance yang sebenarnya positif [15]. Persamaan matematisnya adalah sebagai berikut:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{P}$$

Seperti halnya akurasi, presisi dan recall populer karena kecenderungan implementasi yang sederhana dan pemahaman yang jelas. Namun, kelemahan utamanya adalah bahwa TN tidak dipertimbangkan [16], yang berarti bahwa negatif yang diklasifikasikan dengan benar tidak berdampak pada skor keseluruhan dari kedua metrik ini.

Skor F (F-Score)

Juga dikenal sebagai Skor F1 atau F-Measure, Skor F adalah rata-rata tertimbang dari presisi dan *recall* yang memberikan penilaian keseluruhan tunggal untuk keduanya. Matematis, Skor F didefinisikan sebagai:

$$F = \frac{2(Precision x Recall)}{(Precision + Recall)}$$

Skor F dianggap populer karena mampu memberikan gambaran keseluruhan dari presisi dan *recall*, mengatasi situasi di mana algoritma memiliki presisi tinggi namun *recall* rendah, atau sebaliknya [15]. Namun, penggunaan Skor F sebagai metrik evaluasi harus dilakukan dengan hati-hati

[17], karena metrik ini menggunakan rata-rata tertimbang dari presisi dan *recall*, yang bobotnya bervariasi tergantung pada konteks evaluasi spesifik.

Perbandingan Hasil Model SVM dan KNN

Pada bagian ini, hasil model SVM dan KNN akan dibahas dan dibandingkan berdasarkan metrik-metrik yang telah dijelaskan di atas.

1. Hasil Model SVM

KNN Accuracy: 0.91

	precision	recall	f1-score	support
0	0.89	0.99	0.94	1297
1	0.99	0.77	0.86	703
accuracy			0.91	2000
macro avg	0.94	0.88	0.90	2000
weighted avg	0.92	0.91	0.91	2000

Gambar 2. Hasil model SVM

Metrik rata-rata tertimbang *(weighted avg)* menunjukkan bahwa model SVM memiliki presisi sebesar 0.86, recall sebesar 0.86, dan F1-score sebesar 0.86. Meskipun akurasi dan presisi keseluruhan model SVM cukup baik, performa model ini untuk kelas 1 tidak sebaik KNN, terutama terlihat dari nilai *recall* dan F1-score yang lebih rendah.

2. Hasil Model KNN

SVM Accuracy: 0.86

	precision	recall	f1-score	support
0	0.89	0.95	0.92	250436
1	0.71	0.52	0.60	60593
accuracy			0.86	311029
macro avg	0.80	0.73	0.76	311029
weighted avg	0.86	0.86	0.86	311029

Gambar 3. Hasil Model KNN

Secara keseluruhan, metrik rata-rata tertimbang *(weighted avg)* menunjukkan bahwa model KNN memiliki presisi sebesar 0.92, *recall* sebesar 0.91, dan F1-score sebesar 0.91. Hasil ini menunjukkan bahwa KNN sangat efektif dalam mengklasifikasikan data dengan keseimbangan yang baik antara presisi dan *recall*, terutama terlihat dari nilai F1-score yang tinggi.

3. Perbandingan Kinerja Model SVM dan KNN

Tabel berikut menunjukkan perbandingan kinerja antara model SVM dan KNN berdasarkan metrik-metrik yang digunakan.

Tabel 2 Perbandingan kinerja model SVM dan KNN

Metrik	**KNN**	**SVM**
Akurasi	0.91	0.86
Presisi (0)	0.89	0.89
Recall (0)	0.99	0.95
F1-Score (0)	0.94	0.92
Presisi (1)	0.99	0.71
Recall (1)	0.77	0.52
F1-Score (1)	0.86	0.60

Rata-rata Makro Presisi	0.94	0.80
Rata-rata Makro Recall	0.88	0.73
Rata-rata Makro F1-Score	0.90	0.76
Rata-rata Tertimbang Presisi	0.92	0.86
Rata-rata Tertimbang Recall	0.91	0.86
Rata-rata Tertimbang F1-Score	0.91	0.86

K-Nearest Neighbors (KNN) menunjukkan performa yang lebih unggul dibandingkan dengan *Support Vector Machine* (SVM) dalam semua metrik evaluasi yang digunakan. KNN tidak hanya memiliki akurasi yang lebih tinggi, tetapi juga *recall* yang lebih baik, menunjukkan kemampuan yang lebih besar dalam mengenali true positives. Selain itu, F1-score yang lebih tinggi pada KNN menunjukkan bahwa model ini memiliki keseimbangan yang lebih optimal antara presisi dan recall, dibandingkan dengan SVM. Meskipun SVM menunjukkan performa yang baik terutama dalam mengklasifikasikan kelas 0, kelemahan terlihat pada kemampuannya dalam mengenali kelas 1, seperti yang diperlihatkan oleh nilai recall dan F1-score yang lebih rendah. Secara keseluruhan, hasil ini menegaskan bahwa KNN lebih efektif dalam mengklasifikasikan data secara menyeluruh dibandingkan dengan SVM.

Implikasi Praktis

Hasil penelitian ini memiliki beberapa implikasi praktis dalam konteks deteksi intrusi jaringan:

1. Pemilihan Algoritma Berdasarkan Karakteristik Data: Pengambil keputusan perlu mempertimbangkan karakteristik data yang ada. Jika dataset relatif sederhana dengan jumlah fitur yang tidak terlalu besar, KNN dapat menjadi pilihan yang baik karena kesederhanaannya dalam implementasi dan kemampuannya untuk memberikan akurasi yang tinggi dalam kasus ini. Namun, jika data kompleks dengan dimensi tinggi, SVM mungkin lebih sesuai karena kemampuannya untuk menangani ruang fitur yang kompleks dengan menggunakan kernel non-linear.
2. Optimasi dan Tuning Model: Untuk KNN, fokus pada optimasi hyperparameter seperti jumlah tetangga dan metrik jarak yang sesuai dapat membantu meningkatkan performa deteksi. Hal ini mengingat KNN sensitif terhadap pemilihan k yang optimal. Sementara itu, SVM dapat dioptimalkan dengan mempertimbangkan penggunaan kernel *non-linear* atau tuning parameter regulasi untuk meminimalkan *overfitting* dan meningkatkan akurasi klasifikasi.
3. Manajemen dan Respons Terhadap *False Positives*: SVM cenderung memiliki keunggulan dalam mengelola *false positives* dibandingkan KNN, terutama ketika digunakan dalam konteks deteksi intrusi jaringan yang memerlukan kehati-hatian dalam mengidentifikasi serangan yang sebenarnya. Penggunaan SVM dengan kernel RBF dapat membantu mengklasifikasikan data dengan lebih baik, namun memerlukan proses training yang lebih panjang dan konfigurasi parameter yang tepat.
4. Implementasi dalam Lingkungan Jaringan: KNN dapat memberikan respons cepat terhadap serangan dalam lingkungan jaringan dengan volume data yang relatif kecil dan memerlukan deteksi yang real-time. Di sisi lain, SVM lebih cocok untuk lingkungan jaringan dengan volume data yang besar dan memerlukan analisis yang lebih mendalam terhadap pola serangan yang kompleks.

SIMPULAN

Penelitian ini mengevaluasi performa SVM dan KNN dalam mendeteksi serangan pada dataset KDD Cup 99, dengan fokus pada klasifikasi biner dan multikelas. Berdasarkan hasil evaluasi, KNN menunjukkan performa yang lebih unggul dibandingkan SVM dalam semua metrik evaluasi yang digunakan, termasuk akurasi, presisi, *recall*, dan F1-score. KNN lebih efektif dalam mengklasifikasikan data secara keseluruhan dan memiliki keseimbangan yang lebih baik antara presisi dan recall. Meskipun SVM memiliki kekuatan dalam mengelola *false positives* dan menangani data

berdimensi tinggi, kelemahannya terlihat pada pengenalan kelas tertentu.Implikasi praktis dari penelitian ini menekankan pentingnya pemilihan algoritma berdasarkan karakteristik data dan kebutuhan deteksi. Selain itu, optimasi dan tuning model juga penting untuk meningkatkan performa deteksi intrusi jaringan. Hasil ini memberikan panduan bagi pengambil keputusan dalam memilih dan mengimplementasikan IDS yang efektif.

REFERENSI

[1] Guijarro, E. (2023). Network Infrastructure Security: Challenges and Protection Strategies. Revista VICTEC. https://doi.org/10.61395/victec.v4i7.127.

[2] Louvieris, P., Clewley, N., & Liu, X. (2013). Effects-based feature identification for network intrusion detection. Neurocomputing, 121, 265-273. https://doi.org/10.1016/j.neucom.2013.04.038.

[3] G. G. Liu, "Intrusion detection systems," in Applied Mechanics and Materials, vol. 596. Trans Tech Publ, 2014, pp. 852–855.

[4] C. Chio and D. Freeman, Machine Learning and Security: Protecting Systems with Data and Algorithms. " O'Reilly Media, Inc.", 2018.

[5] Liu, H., & Lang, B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Applied Sciences. https://doi.org/10.3390/app9204396.

[6] Vigneswaran, R. K., Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2018, July). Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security. In 2018 9th International conference on computing, communication and networking technologies (ICCCNT) (pp. 1-6). IEEE.

[7] Yang, L., Shami, A., Stevens, G., & De Rusett, S. (2022, December). LCCDE: a decision-based ensemble framework for intrusion detection in the internet of vehicles. In GLOBECOM 2022-2022 IEEE Global Communications Conference (pp. 3545-3550). IEEE.

[8] Abrar, I., Ayub, Z., Masoodi, F., & Bamhdi, A. M. (2020, September). A machine learning approach for intrusion detection system on NSL-KDD dataset. In 2020 international conference on smart electronics and communication (ICOSEC) (pp. 919-924). IEEE.

[9] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Ocotber 2007.

[10] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Costbased modeling for fraud and intrusion detection: Results from the jam project," discex, vol. 02, p. 1130, 2000.

[11] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," discex, vol. 02, p. 1012, 2000.

[12] Siddiqui, M., & Naahid, S. (2013). Analysis of KDD CUP 99 Dataset using Clustering based Data Mining. *International journal of database theory and application*, 6, 23-34. https://doi.org/10.14257/IJDTA.2013.6.5.03.

[13] N. Japkowicz, "Why question machine learning evaluation methods," in AAAI workshop on evaluation methods for machine learning, 2006, pp. 6–11.

[14] M. A. Hall, "Correlation-based feature selection for machine learning," 1999.

[15] J. Brownlee, How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification, 2020.

[16] M. Doring, ¨ The Case Against Precision as a Model Selection Criterion, 2018.

[17] D. Hand and P. Christen, "A note on using the f-measure for evaluating record linkage algorithms," Statistics and Computing, vol. 28, no. 3, pp. 539–547, 2018.

[18] D. A. Cieslak and N. V. Chawla, "A framework for monitoring classifiers' performance: when and why failure occurs?" Knowledge and Information Systems, vol. 18, no. 1, pp. 83–108, 2009.

[19] Stolfo, S., Fan, W. and Lee, W., KDD-CUP-99 Task Description. 1999- 10-28)[2009-05-08]. http://KDD. ics. uci. edu/databases/kddcup99/task, html.

[20] M. R. Al-Hadidi, A. Alarabeyyat and M. Alhanahnah, " Breast Cancer Detection Using K-Nearest Neighbor Machine Learning Algorithm," 2016 9th International Conference on Developments in eSystems Engineering (DeSE), Liverpool, 2016.

[21] K. Jothi A. and P. Mohan, "A Comparison between KNN and SVM for Breast Cancer Diagnosis Using GLCM shape and LBP Features," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020, pp. 1058-1062, doi: 10.1109/ICSSIT48917.2020.9214235. keywords: {Feature extraction;Support vector machines;Mammography;Breast cancer;Shape;Conferences;SVM;KNN;GLCM;LBP},
[22] P. Brata Chanda and S. Kumar Sarkar, ”Detection And Classification Technique Of Breast Cancer Using Multi Kernal SVM Classifier Approach,” 2018 IEEE Applied Signal Processing Conference (ASPCON), Kolkata, India, 2018.