Penerapan Autoencoder untuk Ekstraksi Fitur pada Model RF dan MLP dalam Deteksi Serangan SQL Injection

Application of Autoencoder for Feature Extraction in RF and MLP Models to Detect SQL Injection Attacks

Franki Setyo Wargo[1], **Much Aziz Muslim**[2]
Jurusan Ilmu Komputer, Fakultas Matematika dan IPA, Universitas Negeri Semarang
E-mail : frank@students.unnes.ac.id[1]*, a212muslim@mail.unnes.ac.id*[2]
Received 9 December 2025; Revised 29 December 2025; Accepted 6 Januari 2026

Abstrak - Serangan *SQL Injection* (SQLi) merupakan ancaman serius bagi aplikasi web karena dapat mengeksploitasi kerentanan pada akses *database* untuk memanipulasi data sensitif, sehingga diperlukan sistem deteksi serangan yang lebih adaptif dan akurat. Urgensi ini muncul karena metode deteksi berbasis *machine learning* yang ada masih mengandalkan rekayasa fitur manual dan kurang responsif terhadap pola serangan baru. Penelitian terdahulu menunjukkan bahwa model seperti *Random Forest* (RF) dan Multi-Layer Perceptron (MLP) memiliki ketergantungan tinggi pada kualitas fitur manual, sehingga performanya tidak stabil. Penelitian ini bertujuan mengembangkan model deteksi SQLi yang lebih adaptif, efisien, dan tidak bergantung pada *feature engineering* manual. *Autoencoder* (AE) diusulkan sebagai *automated feature extractor* untuk mempelajari representasi laten berdimensi rendah secara mandiri. Hasil representasi laten non-linear akan digunakan sebagai input untuk model RF dan MLP. Selain akurasi, penelitian ini mengevaluasi *latency* dan *throughput* untuk menilai kinerja dalam skenario *traffic* nyata. Hasil eksperimen menunjukkan kombinasi Model MLP dan AE mencapai kinerja tertinggi dengan nilai akurasi 99,57% dan AUC 99,88%, Kombinasi ini juga menunjukkan efisiensi komputasi tinggi, ditunjukkan oleh latency rendah dan throughput tinggi, sehingga layak diterapkan pada traffic nyata. Penelitian selanjutnya disarankan mengeksplorasi arsitektur lebih dalam seperti *encoder* berbasis *transformer* untuk menghadapi pola serangan *zero-day*.

Kata kunci - SQL Injection, Autoencoder, Random Forest, MLP, Machine Learning.

Abstract - *SQL Injection (SQLi) attacks pose a serious threat to web applications because they exploit vulnerabilities in database access mechanisms to manipulate sensitive data, thereby requiring a more adaptive and accurate detection system. This urgency arises because existing machine-learning-based detection methods still rely on manual feature engineering and are less responsive to emerging attack patterns. Previous studies indicate that models such as Random Forest (RF) and Multi-Layer Perceptron (MLP) have a strong dependence on the quality of manually crafted features, resulting in unstable performance. Therefore, this study aims to develop a more adaptive and efficient SQLi detection model that does not rely on manual feature engineering. An Autoencoder is proposed as an automated feature extractor capable of independently learning low-dimensional latent representations. These latent features are then used as inputs for the RF and MLP models. In addition to accuracy, this research evaluates latency and throughput to assess performance under realistic traffic conditions. The experimental results show that the combination of the MLP model and AE achieves the highest performance, with an accuracy of 99.57% and an AUC of 99.88%. This combination also demonstrates high computational efficiency, indicated by low latency and high throughput, making it suitable for deployment on real-world traffic. Future work is recommended to explore deeper architectures such as transformer-based encoders to address zero-day attack patterns.*

Keywords - *SQL Injection, Autoencoder, Random Forest, MLP, Machine Learning.*

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

1. PENDAHULUAN

Dalam era transformasi digital yang pesat, keamanan aplikasi web menjadi perhatian kritis bagi organisasi di seluruh dunia. Serangan *SQL Injection* (SQLi) merupakan salah satu ancaman keamanan siber yang paling berbahaya dan persisten [1]. Serangan ini mengeksploitasi kerentanan dalam mekanisme akses database aplikasi, sehingga memungkinkan penyerang untuk mengeksekusi *query* SQL yang tidak sah dan berpotensi mengakses, memodifikasi, atau bahkan menghapus data sensitif [2]. Sementara metode *Web Application Firewall* (WAF) tradisional yang mengandalkan *signature-based detection* atau *rule-based systems* menunjukkan keterbatasan dalam menghadapi serangan yang semakin canggih dan adaptif [3]. Kondisi keterbatasan ini mendorong kebutuhan akan pendekatan yang lebih adaptif dan dinamis. Integrasi *machine learning* dalam WAF muncul sebagai solusi yang menjanjikan karena mampu mengenali pola serangan yang belum terdokumentasi serta menurunkan tingkat *false positive* [4].

Deteksi serangan SQLi telah menjadi area penelitian yang intensif dengan berbagai pendekatan *machine learning* yang diusulkan. Akan tetapi setiap model memiliki kelebihan dan kekurangan yang perlu dianalisis secara kritis untuk mengidentifikasi celah penelitian yang dapat diatasi. Pada studi sebelumnya model *Support Vector Machine* (SVM) telah banyak digunakan dalam deteksi serangan SQLi karena kemampuannya dalam klasifikasi teks. SVM memiliki kelebihan dalam mencapai akurasi yang sangat tinggi ketika dikombinasikan dengan teknik *Natural Language Processing* (NLP) dan representasi teks yang sesuai, serta menunjukkan kinerja yang stabil pada ruang fitur berdimensi tinggi seperti *Term Frequency-Inverse Document Frequency* (TF-IDF) atau vektorisasi korpus [5], [6]. Namun, SVM memiliki kelemahan yaitu performa model sangat bergantung pada desain fitur manual yang tepat, dan tanpa fitur yang informatif, performanya akan menurun drastis [5], [6].

Penelitian [7] juga menyebutkan bahwa model *Long Short-Term Memory* (LSTM) memiliki kelebihan dalam menangkap pola urutan dan dependensi sekuensial dalam *query* SQL, yang membuatnya efektif untuk mendeteksi pola serangan SQLi yang bersifat sekuensial. Dan kombinasi LSTM dengan teknik seleksi fitur seperti *Principal Component Analysis* (PCA) telah terbukti meningkatkan akurasi dengan memasok fitur yang relevan dan membantu gerbang LSTM memilih informasi penting [8]. Namun, LSTM memiliki kelemahan yaitu memerlukan proses *feature selection* untuk mencapai performa optimal. Studi [8] menunjukkan bahwa LSTM memerlukan pemilihan dan penyaringan fitur untuk menghindari *noise* dan mencapai hasil terbaik.

Sementara model *Random Forest* (RF) dan *Multi-Layer Perceptron* (MLP) memiliki kelebihan untuk mengatasi beberapa kelemahan yang dimiliki oleh SVM dan LSTM. Diantaranya kelebihan dalam memberikan akurasi yang kompetitif dan *robust* pada *dataset* dengan fitur manual yang baik, serta relatif tahan terhadap *overfitting* [9]. Dan juga kelebihan lain adalah dalam mencapai akurasi sangat tinggi (>99%) ketika fitur statistik atau *handcrafted* yang informatif tersedia, dengan implementasi yang relatif sederhana untuk *deployment* [10].

Baik RF maupun MLP memiliki kelemahan serupa dalam hal ekstraksi fitur. RF masih memerlukan fitur yang dirancang secara manual dan berkualitas tinggi agar dapat memberikan hasil yang optimal, sehingga tetap bergantung pada rekayasa fitur yang spesifik terhadap domain [9], [11]. Hal yang sama juga berlaku pada MLP, yang kinerjanya sangat dipengaruhi oleh kualitas fitur manual seperti fitur statistik atau heuristik, karena model ini tidak secara otomatis dapat menggantikan proses rekayasa fitur [10]. Hasil tinjauan sistematis menunjukkan bahwa performa MLP berbeda-beda di setiap penelitian dan sangat bergantung pada kualitas fitur serta tahap preprocessing yang diterapkan [12].

Autoencoder, termasuk varian RNN-*Autoencoder*, menawarkan solusi untuk mengatasi masalah *feature engineering* yang dihadapi oleh model RF dan MLP. AE memiliki kemampuan untuk belajar representasi otomatis dari input teks atau *query* dengan menghasilkan *encoding* berdimensi lebih rendah, sehingga menggantikan beberapa langkah ekstraksi fitur manual yang kompleks dan memakan waktu [12], [13], [14], [15]. Model *Autoencoder* dapat menangkap pola

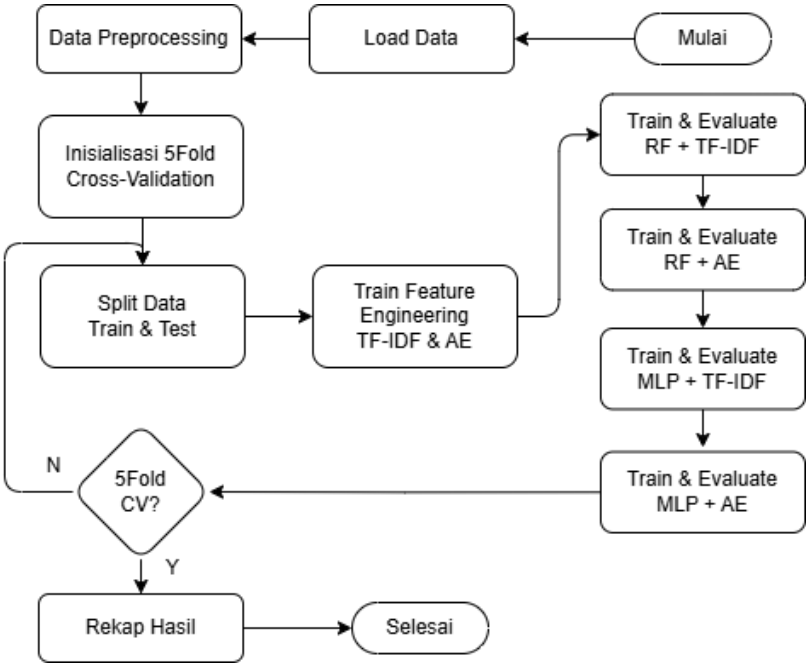
e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

nonlinier dan struktur sekuensial dalam *query* SQL tanpa memerlukan definisi fitur eksplisit, dan representasi yang dihasilkan dapat digunakan langsung oleh model seperti RF atau MLP sehingga *pipeline* menjadi lebih ringkas dan efisien [12] .

Penelitian ini menawarkan pendekatan baru berupa integrasi AE sebagai *automated feature extractor* untuk menggantikan rekayasa fitur manual pada RF dan MLP dalam deteksi SQLi. Berbeda dari penelitian sebelumnya yang umumnya memanfaatkan AE sebagai bagian dari arsitektur *deep learning end-to-end*, penelitian ini mengevaluasi AE sebagai ekstraktor fitur universal yang dapat digunakan secara efektif baik pada model *non-deep learning* seperti RF maupun pada model *neural network* seperti MLP dan penelitian ini juga menambahkan kontribusi penting dengan mengukur *latency* dan *throughput* guna menilai kelayakan implementasi model pada *traffic* nyata, sehingga hasilnya tidak hanya unggul secara akurasi tetapi juga relevan untuk kebutuhan sistem deteksi *real-time*.

2. METODE PENELITIAN

Metode penelitian yang diusulkan dalam studi ini dirancang untuk menggambarkan alur sistematis mulai dari tahap load dataset hingga analisis hasil akhir, dengan tujuan utama mengevaluasi efektivitas integrasi AE sebagai *automated feature extraction* pada model RF dan MLP dalam meningkatkan akurasi, stabilitas, serta efisiensi deteksi serangan SQLi termasuk performa operasionalnya melalui pengukuran *latency* dan *throughput*. Metode ini memastikan seluruh proses berjalan secara terukur, konsisten, dan dapat direplikasi secara ilmiah.



Gambar 1 Flowchart Alur Penelitian

Pada Gambar 1 menunjukkan bahwa penelitian ini dimulai dengan tahap pemanggilan *dataset*. *Dataset* yang digunakan bersumber dari SQLi Dataset milik Syed Saqlain Hussain Shah yang tersedia pada platform Kaggle. Dataset ini dipilih karena memiliki struktur data yang jelas, mencakup variasi *query* normal dan *malicious* dengan label yang terdefinisi dengan baik, serta telah banyak digunakan dalam penelitian keamanan siber berbasis kecerdasan buatan. Karakteristik tersebut menjadikan dataset ini representatif untuk mengevaluasi kinerja sistem deteksi SQLi berbasis *machine learning*.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

Tahap selanjutnya adalah pra-pemrosesan data, yang bertujuan menyiapkan *dataset* agar sesuai dengan kebutuhan model. Proses ini mencakup pembersihan karakter yang tidak relevan, konversi teks menjadi huruf kecil, serta normalisasi struktur *query* agar lebih konsisten. Langkah ini berperan penting untuk memastikan bahwa data yang masuk ke tahap pemodelan memiliki kualitas yang memadai dan bebas dari *noise* yang dapat mengganggu proses pembelajaran [16].

Setelah tahap praproses, dilakukan inisialisasi 5-Fold *Cross-Validation* (CV) untuk menjamin evaluasi model yang adil dan menghindari bias akibat pembagian data [17]. Pada setiap fold, dataset dibagi menjadi data pelatihan (*train set*) dan data pengujian (*test set*) secara stratifikasi.

Untuk pendekatan TF-IDF, vektorisasi fitur dilakukan dengan cara fit hanya pada data pelatihan, kemudian transform diterapkan pada data pengujian pada fold yang sama. Dengan demikian, informasi dari data uji tidak digunakan dalam proses pembentukan fitur. Dan dengan membatasi proses pembelajaran fitur hanya pada data pelatihan di setiap fold, penelitian ini memastikan bahwa evaluasi model bebas dari data leakage. Pendekatan ini menghasilkan estimasi performa yang lebih realistis dan dapat digeneralisasikan pada skenario penggunaan data *real*. TF-IDF digunakan sebagai baseline karena sifatnya sederhana, umum digunakan dalam berbagai penelitian *machine learning*, dan dapat memberikan gambaran awal performa model berbasis fitur manual [18], [19]. TF-IDF digunakan sebagai baseline dan nilainya dihitung menggunakan rumus:

$$\text{TFIDF}(t,d) = \text{TF}(t,d) \times I\text{DF}(t) \text{ , dengan } \text{TF}(t,d) = \frac{f_{t,d}}{\max(f_d)} \text{ dan } \text{IDF}(t) = \log\left(\frac{N}{df_t}\right)$$

Dimana :

$f_{t,d}$ = frekuensi term *t* dalam dokumen *d*
df_t = jumlah dokumen yang mengandung term *t*
N = total dokumen

Sementara itu, AE berfungsi sebagai automated feature extractor yang mempelajari representasi laten berdimensi rendah dari *query* SQL secara nonlinier [20], [21]. Model Autoencoder dilatih hanya menggunakan data pelatihan pada setiap fold. Setelah proses pelatihan selesai, bagian encoder dari model AE digunakan untuk mengekstraksi representasi fitur dari data pelatihan dan data pengujian. Pendekatan ini memastikan bahwa pembelajaran representasi fitur tidak melibatkan data uji, sehingga menghindari potensi data *leakage*. AE mempelajari representasi laten berdimensi rendah dari *query* SQL dengan komponen utama:

a. Fungsi rekonstruksi
   $\hat{x} = f_\theta(x)$
b. Loss rekonstruksi - Mean Squared Error (MSE)
   $L(x,\hat{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x_i})^2$
c. Loss rekonstruksi - Binary Cross Entropy (BCE)
   $L(x,\hat{x}) = -\sum_{i=1}^{n}[x_i \log(\hat{x_i}) + (1-x_i)\log(1-\hat{x_i})]$

Representasi laten diperoleh dari output bottleneck layer:
$z = \text{Encoder}(x)$
Representasi z inilah yang menjadi input untuk model RF dan MLP.

Fitur hasil ekstraksi kemudian digunakan sebagai masukan untuk proses pelatihan dan pengujian *classifier*. Tahap inti penelitian dimulai dengan pengujian kombinasi model dan fitur. Empat konfigurasi diuji, yaitu: RF dengan TF-IDF, RF dengan AE, MLP dengan TF-IDF, dan MLP dengan AE. Pada setiap *fold*, model dilatih untuk mengenali pola *query* normal dan

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

malicious berdasarkan representasi fitur yang diberikan. Evaluasi dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score* [22], dimana metrik evaluasi diperoleh dari *confusion matrix* TP = *True Positive*, TN = *True Negative*, FP = *False Positive*, dan FN = *False Negative*. Sehingga didapatkan :

a. $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$

b. $\text{Precision} = \frac{TP}{TP+FP}$

c. $\text{Recall} = \frac{TP}{TP+FN}$

d. $\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Sebagai bagian dari kontribusi kebaruan penelitian, evaluasi juga mencakup pengukuran *latency* (waktu inferensi per *instance*) [23], [24], dengan rumus :

$$\text{Latency} = \frac{T_{\text{inference}}}{N}$$

dan *throughput* (jumlah prediksi per detik) untuk menilai kelayakan model dalam skenario implementasi pada trafik nyata [24], dengan rumus :

$$\text{Throughput} = \frac{N}{T_{\text{inference}}}$$

Pada penelitian ini RF dipilih karena kemampuannya menangani data *non-linear* dan mengurangi *overfitting* melalui mekanisme agregasi dari banyak *decision tree* [25], [26]. Sementara itu, MLP digunakan karena kemampuannya menangkap pola *non-linear* yang kompleks menggunakan beberapa lapisan tersembunyi dan fungsi aktivasi *ReLU* [27], [28], [29], [30]. Kedua model dilatih dengan konfigurasi parameter dan porsi data yang identik guna memastikan evaluasi berlangsung secara adil *(fair comparison*) dan memungkinkan analisis yang objektif terhadap pengaruh representasi laten hasil AE.

Hasil evaluasi dari setiap fold dicatat dan mencakup metrik kinerja klasifikasi seperti accuracy, precision, recall, F1-score, dan AUC, serta metrik performa komputasi seperti *latency* dan *throughput*. Setelah seluruh *fold* selesai diproses, dilakukan tahap rekap hasil untuk menghitung nilai rata-rata dan standar deviasi antar fold.

Sebagai tambahan, untuk menilai signifikansi perbedaan kinerja antara pendekatan AE dan TF-IDF, dilakukan uji statistik berpasangan menggunakan Wilcoxon signed-rank test pada nilai metrik per *fold*, serta perhitungan *effect size* menggunakan *Cohen's d*. Tahap ini bertujuan untuk memastikan bahwa perbedaan performa yang diperoleh tidak hanya bersifat numerik, tetapi juga signifikan secara statistik.

3. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dan pembahasan dari penelitian yang diperoleh melalui empat kelompok eksperimen yang memadukan dua algoritma pembelajaran mesin, yaitu RF dan MLP, dengan dua pendekatan representasi fitur, yaitu TF-IDF dan AE. Untuk setiap metode, kinerja dievaluasi menggunakan *5-Fold* CV dan seluruh metrik dilaporkan dalam bentuk nilai per fold serta nilai rata-rata dan simpangan baku (*mean* ± *std*) guna merepresentasikan variasi performa antar fold. Evaluasi mencakup metrik akurasi, *precision, recall, F1-Score*, waktu inferensi (*latency*), *throughput*, analisis *confusion matrix*, serta ROC *Curve* dan nilai AUC. Sub bab berikut memaparkan hasil secara sistematis sesuai urutan eksperimen yang dilakukan.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

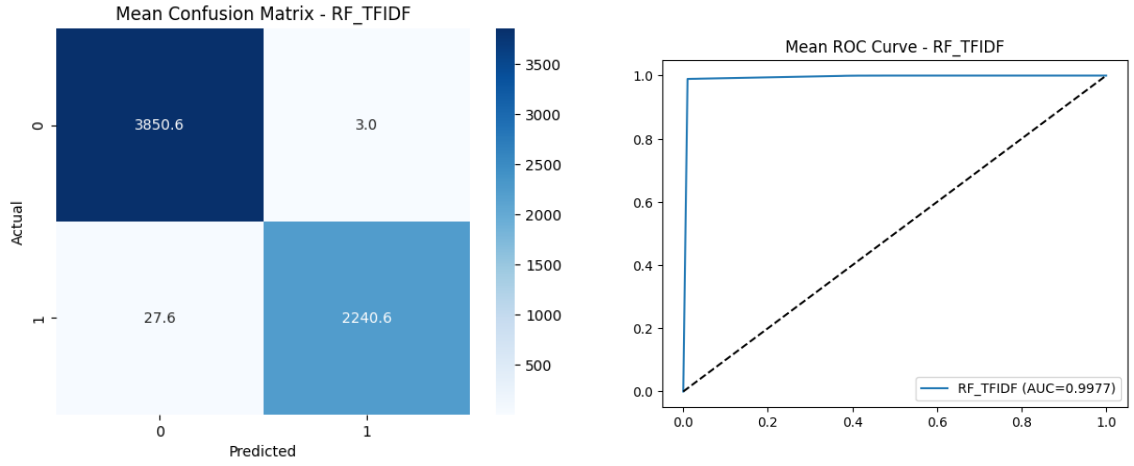
3.1 RF dengan Fitur TF-IDF

Eksperimen pertama menggunakan model RF yang dilatih pada representasi tekstual berbasis TF-IDF. Hasil *5-Fold* CV ditunjukkan pada tabel 1, menghasilkan akurasi rata-rata sebesar ±99,5% dengan simpangan baku yang rendah, precision mendekati 100%, serta recall yang stabil di kisaran 98,7%. Nilai F1-score yang berada di atas 0,99 mengindikasikan keseimbangan yang baik antara kemampuan mendeteksi serangan SQLi dan meminimalkan kesalahan klasifikasi. Rendahnya variasi nilai metrik antar fold menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting* terhadap subset data tertentu.

Tabel 1 Hasil Evaluasi RF fitur TF-IDF

Fold	Train Time (s)	Pred Latency (s)	Throughput (samples/s)	Acc	Prec	Rec	F1
1	18,48	0,000168	5934,96	0,9954	1,0000	0,9877	0,9938
2	17,22	0,000142	7060,16	0,9948	0,9973	0,9885	0,9929
3	18,03	0,000147	6810,79	0,9951	0,9996	0,9872	0,9933
4	18,64	0,000150	6652,44	0,9949	0,9982	0,9881	0,9931
5	17,79	0,000141	7095,42	0,9948	0,9982	0,9877	0,9929
mean ± std		0,00015 ± 0,000011	6710,75 ± 470,46	0,9950 ± 0,0003	0,9987 ± 0,0011	0,9878 ± 0,0005	0,9932 ± 0,0004

Dari sisi efisiensi komputasi, RF–TFIDF memiliki waktu pelatihan yang relatif singkat (sekitar 18 detik per *fold*), serta *latency* prediksi yang sangat rendah, yang dihitung berdasarkan *throughput* pengujian. Dengan kemampuan memproses lebih dari 6000 sampel per detik, model ini tergolong efisien untuk digunakan pada skenario pemantauan trafik SQL secara berkelanjutan. Kombinasi performa tinggi dan efisiensi ini menjadikan RF–TFIDF sebagai baseline eksperimental yang kuat dalam eksperimen deteksi serangan SQLi.



Gambar 2 *Confusion Matrix* dan *ROC Curve* RF fitur TF-IDF

Analisis *confusion matrix* menunjukkan bahwa model RF–TFIDF mampu membedakan kelas Normal dan SQLi dengan tingkat kesalahan yang sangat rendah, ditandai oleh jumlah *false*

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

negative sekitar 27–28 sampel dan *false positive* sekitar 2–3 sampel. Kurva ROC menghasilkan nilai AUC sebesar 0,9977, yang mengindikasikan kemampuan diskriminasi antar kelas berada pada kategori sangat baik.

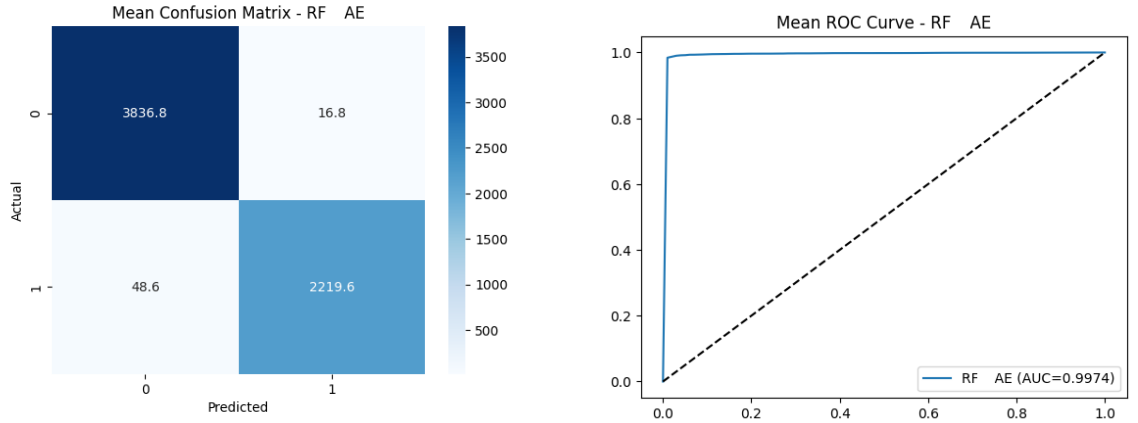
3.2 RF dengan Fitur AE

Eksperimen kedua menerapkan model RF yang menggunakan fitur hasil reduksi dimensi dari AE. Hasil 5-Fold Cross-Validation yang disajikan pada Tabel 2 menunjukkan bahwa RF–AE mampu mencapai akurasi rata-rata di kisaran 98–99% dengan simpangan baku yang masih terkendali, serta nilai precision dan recall yang relatif konsisten. Meskipun nilai akurasi dan F1-score sedikit lebih rendah dibandingkan pendekatan berbasis TF-IDF.

Tabel 2 Hasil Evaluasi RF fitur AE

Fold	Train Time (s)	Pred Latency (s)	Throughput (samples/s)	Acc	Prec	Rec	F1
1	1037,87	0,000072	13926,8	0,99	0,9955	0,9775	0,9864
2	1304,86	0,000062	16140,22	0,9864	0,984	0,9793	0,9817
3	1267,53	0,000058	17217,26	0,9881	0,9924	0,9753	0,9838
4	1004,58	0,000054	18683,79	0,9909	0,9964	0,9788	0,9875
5	1188,58	0,000061	16339,9	0,9912	0,9942	0,9819	0,988
mean ± std		0,000061 ± 0,000007	16461,60 ± 1735,83	0,9893 ± 0,0020	0,9925 ± 0,0050	0,9786 ± 0,0024	0,9855 ± 0,0027

Dari aspek komputasi, RF–AE memiliki waktu pelatihan yang jauh lebih tinggi dibandingkan pendekatan TF-IDF, namun latency prediksi tetap rendah dan throughput berada pada tingkat menengah. Pendekatan ini menunjukkan bahwa peningkatan kualitas fitur diperoleh dengan



konsekuensi waktu pelatihan yang lebih lama.

Gambar 3 *Confusion Matrix* dan *ROC Curve* RF fitur AE

Analisis *confusion matrix* menunjukkan bahwa model RF–AE masih mampu memisahkan kelas normal dan serangan dengan baik, namun tingkat kesalahan prediksi sedikit lebih tinggi dibandingkan RF–TFIDF, ditunjukkan oleh peningkatan *false negative* (≈49 sampel) dan *false positive* (≈17 sampel). Kurva ROC menghasilkan nilai AUC sebesar 0,9974, yang tetap berada

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

pada kategori sangat baik, meskipun sedikit lebih rendah dibandingkan RF–TFIDF. Hasil ini mengindikasikan meskipun fitur AE mampu mempertahankan kemampuan diskriminatif tinggi, representasi TF-IDF masih memberikan performa klasifikasi yang lebih stabil pada model RF.

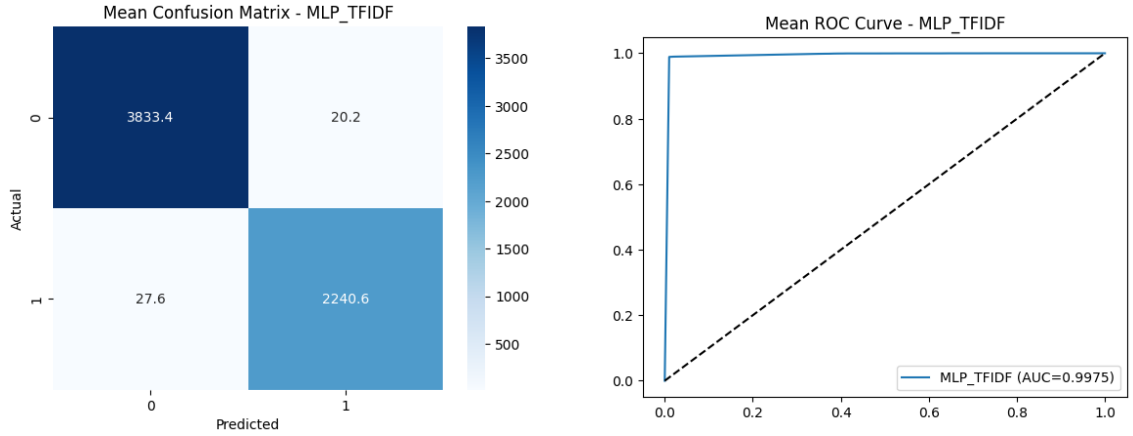
3.3 MLP dengan Fitur TF-IDF

Eksperimen ketiga menggunakan model MLP yang dilatih menggunakan representasi fitur berbasis TF-IDF. Hasil *5-Fold* CV yang ditunjukkan pada Tabel 3 menghasilkan akurasi rata-rata di atas 99% dengan nilai simpangan baku yang relatif kecil, serta precision dan recall yang stabil antar fold. Nilai F1-score yang mendekati 0,99 menunjukkan bahwa model MLP–TFIDF mampu mencapai performa klasifikasi yang kompetitif dalam mendeteksi serangan SQLi. Variasi metrik yang sedikit lebih besar dibandingkan RF–TFIDF tercermin dari simpangan baku yang lebih tinggi, namun masih menunjukkan kemampuan generalisasi yang baik.

Tabel 3 Hasil Evaluasi MLP fitur TF-IDF

Fold	Train Time (s)	Pred Latency (s)	Throughput (samples/s)	Acc	Prec	Rec	F1
1	65,92	0,000005	212592,24	0,9928	0,9929	0,9877	0,9903
2	94,98	0,000003	309047,6	0,9943	0,9964	0,9881	0,9923
3	111,54	0,000003	325340,88	0,9899	0,9859	0,9868	0,9863
4	68,59	0,000003	310055,17	0,9918	0,9898	0,9881	0,989
5	80,68	0,000003	315746,34	0,9922	0,9903	0,9885	0,9894
mean ± std		0,000003 ± 0,000001	294556,45 ± 46273,72	0,9922 ± 0,0016	0,9911 ± 0,0039	0,9878 ± 0,0007	0,9894 ± 0,0021

Dari sisi efisiensi komputasi, MLP–TFIDF memerlukan waktu pelatihan yang lebih lama dibandingkan RF–TFIDF, dengan rata-rata sekitar puluhan detik per *fold*. Namun demikian, model ini memiliki *latency* prediksi yang sangat rendah serta *throughput* yang sangat tinggi, mampu memproses ratusan ribu sampel per detik. Karakteristik ini menjadikan MLP–TFIDF sangat sesuai untuk skenario inferensi real-time, khususnya pada volume data yang besar.



Gambar 4 *Confusion Matrix* dan *ROC Curve* MLP fitur TF-IDF

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

Analisis *confusion matrix* menunjukkan bahwa model MLP dengan fitur TF-IDF mampu mengklasifikasikan kelas normal dan serangan dengan tingkat kesalahan yang sangat rendah, ditunjukkan oleh jumlah *false positive* yang relatif kecil (≈20 sampel) dan *false negative* sekitar 28 sampel. Kurva ROC menghasilkan nilai AUC sebesar 0,9975, yang menandakan kemampuan diskriminasi antar kelas berada pada kategori sangat baik.

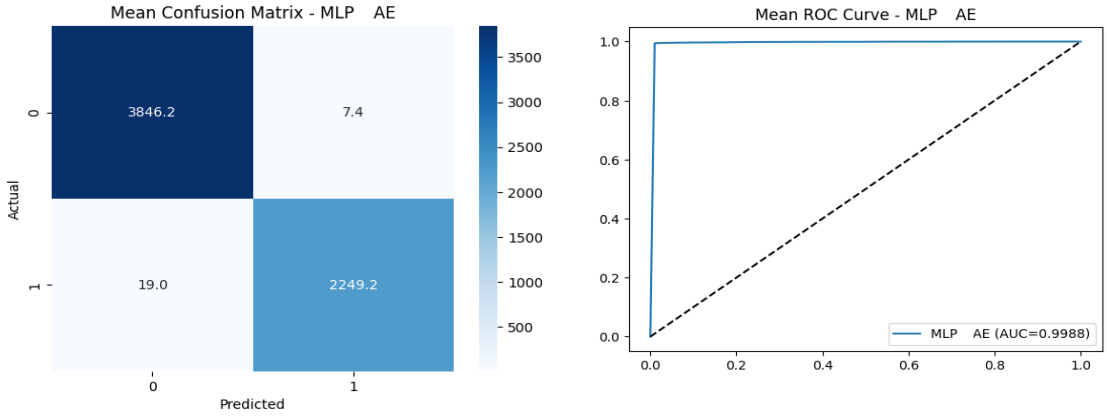
3.4 MLP dengan Fitur AE

Eksperimen keempat mengombinasikan fitur hasil AE dengan model MLP. Berdasarkan hasil *5-Fold* CV pada Tabel 4, pendekatan MLP–AE menghasilkan akurasi rata-rata mendekati 99,6% dengan simpangan baku yang rendah, serta nilai precision, recall, dan F1-score yang konsisten di atas 0,99. Rendahnya simpangan baku antar fold mengindikasikan bahwa kombinasi AE dan MLP mampu menghasilkan model yang robust dan memiliki kemampuan generalisasi yang sangat baik dalam mendeteksi serangan SQLi.

Tabel 4 Hasil Evaluasi MLP fitur AE

Fold	Train Time (s)	Pred Latency (s)	Throughput (samples/s)	Acc	Prec	Rec	F1
1	975,26	0,000003	322505,04	0,9943	0,9942	0,9903	0,9923
2	1199,69	0,000005	204574,11	0,9958	0,9978	0,9907	0,9942
3	1165,74	0,000003	363159,12	0,9953	0,9969	0,9903	0,9936
4	937,82	0,000003	350135,39	0,9962	0,9987	0,9912	0,9949
5	1103,47	0,000003	348666,15	0,9969	0,996	0,9956	0,9958
mean ± std		0,000003 ± 0,000001	317807,97 ± 64996,13	0,9957 ± 0,0010	0,9967 ± 0,0017	0,9916 ± 0,0022	0,9942 ± 0,0013

Dari sisi efisiensi komputasi, MLP–AE memiliki waktu pelatihan paling tinggi di antara seluruh skenario eksperimen akibat proses pembelajaran AE dan jaringan MLP yang lebih kompleks. Namun demikian, model ini menunjukkan *latency* prediksi yang sangat rendah dan *throughput* yang sangat tinggi, sehingga tetap layak digunakan pada fase inferensi.



Gambar 5 *Confusion Matrix* dan *ROC Curve* MLP fitur AE

Analisis *confusion matrix* menunjukkan bahwa kombinasi MLP–AE menghasilkan tingkat kesalahan prediksi yang paling rendah dibandingkan seluruh skema yang diuji, dengan *false positive* yang sangat kecil (≈7 sampel) dan *false negative* sekitar 19 sampel. Hal ini

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

mengindikasikan kemampuan model yang sangat baik dalam membedakan trafik normal dan serangan SQLi. Kurva ROC memperlihatkan nilai AUC sebesar 0,9988, yang merupakan nilai tertinggi di antara semua kombinasi model, menandakan kemampuan diskriminasi antar kelas berada pada kategori sangat unggul.

3.5 Ringkasan Komparatif

Tabel 5 Perbandingan Kinerja Model (Mean ± Std, 5-Fold CV)

Eva	Model Fitur	Pred Latency (s)	Throughput (samples/s)	Acc	Prec	Rec	F1
1	RF TF-IDF	0,000150 ± 0,000011	6710,75 ± 470,46	0,9950 ± 0,0003	0,9987 ± 0,0011	0,9878 ± 0,0005	0,9932 ± 0,0004
2	RF AE	0,000061 ± 0,000007	16461,60 ± 1735,83	0,9893 ± 0,0020	0,9925 ± 0,0050	0,9786 ± 0,0024	0,9855 ± 0,0027
3	MLP TF-IDF	0,000003 ± 0,000001	294556,45 ± 46273,72	0,9922 ± 0,0016	0,9911 ± 0,0039	0,9878 ± 0,0007	0,9894 ± 0,0021
4	MLP AE	0,000003 ± 0,000001	317807,97 ± 64996,13	0,9957 ± 0,0010	0,9967 ± 0,0017	0,9916 ± 0,0022	0,9942 ± 0,0013

Secara keseluruhan, kombinasi model dan fitur menghasilkan performa klasifikasi yang sangat tinggi, dengan nilai akurasi dan *F1-score* mendekati atau melebihi 0,99. Pada model RF, penggunaan TF-IDF memberikan performa klasifikasi yang lebih baik dibandingkan AE, terutama pada metrik akurasi dan F1-score. Sebaliknya, pada model MLP, integrasi AE menghasilkan peningkatan kinerja yang konsisten dibandingkan TF-IDF, dengan nilai akurasi, *precision, recall, dan F1-score* tertinggi di antara seluruh konfigurasi yang diuji.

Dari sisi efisiensi komputasi, model berbasis MLP menunjukkan *throughput* yang jauh lebih tinggi dibandingkan RF, dengan *latency* prediksi yang sangat rendah. Kombinasi MLP–AE menjadi konfigurasi paling unggul karena mampu mencapai keseimbangan terbaik antara performa klasifikasi dan efisiensi inferensi, sehingga lebih sesuai untuk skenario deteksi serangan SQLi secara *real-time* dan berskala besar.

3.6 Analisis Statistik Antar Metode

Untuk memastikan bahwa perbedaan performa antara pendekatan TF-IDF dan AE tidak hanya disebabkan oleh variasi data antar fold, dilakukan analisis statistik berbasis evaluasi antar fold. Setiap metrik performa dihitung pada lima fold menggunakan skema *5-fold* CV, kemudian dirangkum dalam bentuk nilai rata-rata (*mean*) dan simpangan baku (*standard deviation*). Selanjutnya, dilakukan uji *Wilcoxon signed-rank* sebagai *paired non-parametric test* untuk membandingkan performa TF-IDF dan AE pada setiap *classifier*. Uji ini dipilih karena jumlah *fold* yang relatif kecil dan tidak mengasumsikan distribusi normal. Selain signifikansi statistik (*p-value*), penelitian ini juga melaporkan *effect size* menggunakan *Cohen's d* guna mengukur besarnya pengaruh perbedaan metode secara praktis.

Tabel 6 Hasil Uji Statistik Wilcoxon dan Effect Size (Cohen's d)

Classifier	Perbandingan Fitur	p-value (Wilcoxon)	Cohen's d	Effect Size
RF	AE vs TF-IDF	0,0625	-4,012	Efek sangat besar
MLP	AE vs TF-IDF	0,0625	2,634	Efek besar

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

Hasil uji statistik menunjukkan bahwa perbedaan performa antara TF-IDF dan AE pada model RF dan MLP tidak signifikan secara statistik pada taraf signifikansi 5% ($p > 0,05$). Namun demikian, nilai Cohen's d yang besar mengindikasikan adanya perbedaan efek praktis yang kuat, khususnya pada kombinasi MLP dengan AE. Hal ini menunjukkan bahwa meskipun variasi performa antar fold relatif kecil, penggunaan AE tetap memberikan keuntungan performa yang bermakna dalam konteks praktis.

4. KESIMPULAN

Berdasarkan hasil eksperimen *5-Fold* CV, kombinasi MLP dengan fitur AE menunjukkan performa terbaik secara keseluruhan. Model ini secara konsisten menghasilkan nilai akurasi, *precision, recall*, dan *F1-score* tertinggi, serta nilai AUC mendekati sempurna. Analisis *confusion matrix* dan *ROC curve* memperlihatkan tingkat kesalahan klasifikasi yang sangat rendah dan kemampuan diskriminasi antar kelas yang sangat baik. Meskipun AE membutuhkan waktu pelatihan yang lebih besar, MLP–AE tetap menunjukkan efisiensi inferensi yang tinggi dengan *throughput* besar dan *latency* prediksi yang sangat rendah, sehingga layak untuk diterapkan pada skenario deteksi SQLi berbasis trafik nyata.

Hasil eksperimen juga menunjukkan bahwa akurasi RF dengan fitur TF-IDF sedikit lebih tinggi dibandingkan RF dengan fitur AE. Hal ini disebabkan oleh karakteristik RF yang bekerja optimal pada fitur sparsity tinggi dan bersifat diskrit seperti TF-IDF, di mana informasi token spesifik serangan SQLi tetap dipertahankan secara eksplisit. Sebaliknya, fitur AE merupakan representasi laten hasil kompresi nonlinier yang bersifat lebih padat, sehingga sebagian informasi leksikal penting dapat tereduksi dan kurang optimal dimanfaatkan oleh mekanisme pemisahan berbasis pohon keputusan pada RF. Dengan demikian, meskipun AE mampu mengekstraksi pola global, TF-IDF tetap lebih sesuai untuk model RF.

Analisis statistik menggunakan uji *Wilcoxon signed-rank* pada *nilai F1-score* antar *fold* menunjukkan bahwa perbedaan performa antara pendekatan AE dan TF-IDF tidak signifikan secara statistik pada tingkat signifikansi 5%. Namun, perhitungan *effect size* (*Cohen's d*) menunjukkan efek yang besar, terutama pada kombinasi MLP–AE dibandingkan MLP–TFIDF. Temuan ini mengindikasikan bahwa meskipun perbedaan rata-rata metrik relatif kecil, penggunaan AE memberikan dampak praktis yang substansial pada model *berbasis neural network*, sehingga mendukung kesimpulan bahwa pemilihan teknik *feature engineering* harus disesuaikan dengan karakteristik model klasifikasi yang digunakan. Penelitian selanjutnya disarankan untuk mengeksplorasi arsitektur representasi yang lebih dalam, *seperti Variational Autoencoder* atau *encoder* berbasis *Transformer*, guna menghadapi pola serangan SQLi yang semakin kompleks dan bersifat zero-day. Selain itu, evaluasi lintas dataset dengan karakteristik dan distribusi serangan yang berbeda juga menjadi arah penelitian selanjutnya untuk menguji generalisasi dan ketahanan model dalam skenario dunia nyata.

DAFTAR PUSTAKA

[1] J. Koman and M. Janiszewski, "SCAnME - scanner comparative analysis and metrics for evaluation," *Int J Inf Secur*, vol. 24, no. 3, p. 147, 2025, doi: 10.1007/s10207-025-01054-8.

[2] C. Y. Daramola, S. A. Akinpelu, E. O. Akinyemi, S. A. Ajagbe, G. A. Akinpelu, and M. O. Adigun, "Malicious Query Recognition Using Chosen Machine Learning Techniques," *SN Comput Sci*, vol. 6, no. 3, p. 281, 2025, doi: 10.1007/s42979-025-03745-4.

[3] S. Applebaum, T. Gaber, and A. Ahmed, "Signature-based and Machine-Learning-based Web Application Firewalls: A Short Survey," *Procedia Comput Sci*, vol. 189, pp. 359–367, 2021, doi: https://doi.org/10.1016/j.procs.2021.05.105.

[4] T. S. Oyinloye, M. O. Arowolo, and R. Prasad, "Enhancing cyber threat detection with an improved artificial neural network model," *Data Science and Management*, vol. 8, no. 1, pp. 107–115, Mar. 2025, doi: 10.1016/J.DSM.2024.05.002.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

[5] J. Triloka, H. Hartono, and S. Sutedi, "Detection of SQL Injection Attack Using Machine Learning Based On Natural Language Processing," *International Journal of Artificial Intelligence Research*, vol. 6, no. 2, Aug. 2022, doi: 10.29099/ijair.v6i2.355.

[6] A. Rattrout, M. Jaradat, and R. Jayousi, "Machine Learning Advancements in SQL Injection Detection: NLP and Feature Engineering Strategies," Oct. 18, 2023. doi: 10.21203/rs.3.rs-3446830/v1.

[7] M. Alshammari, "Deep learning approaches to SQL injection detection: evaluating ANNs, CNNs, and RNNs," in *Proc.SPIE*, Dec. 2023, p. 129360H. doi: 10.1117/12.3012620.

[8] D. Stiawan *et al.*, "An Improved LSTM-PCA Ensemble Classifier for SQL Injection and XSS Attack Detection," *Computer Systems Science and Engineering*, vol. 46, no. 2, pp. 1759–1774, 2023, doi: 10.32604/csse.2023.034047.

[9] E. Hosam, H. Hosny, W. Ashraf, and A. S. Kaseb, "SQL Injection Detection Using Machine Learning Techniques," in *2021 8th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, 2021, pp. 15–20. doi: 10.1109/ISCMI53840.2021.9654820.

[10] P. Tang, W. Qiu, Z. Huang, H. Lian, and G. Liu, "Detection of SQL injection based on artificial neural network," *Knowl Based Syst*, vol. 190, p. 105528, Feb. 2020, doi: 10.1016/J.KNOSYS.2020.105528.

[11] W. B. Demilie and F. G. Deriba, "Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques," *J Big Data*, vol. 9, no. 1, p. 124, 2022, doi: 10.1186/s40537-022-00678-0.

[12] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Deep Learning Architecture for Detecting SQL Injection Attacks Based on RNN Autoencoder Model," *Mathematics*, vol. 11, no. 15, Aug. 2023, doi: 10.3390/math11153286.

[13] A. E. E. Rashed, M. Badawy, M. A. Elhosseini, and W. M. Bahgat, "An autoencoder-based approach for feature engineering in cardiovascular disease prediction," *Neural Comput Appl*, vol. 37, no. 28, pp. 23185–23218, 2025, doi: 10.1007/s00521-025-11484-z.

[14] A. Alqahtani, "Optimized deep autoencoder and BiLSTM for intrusion detection in IoTs-Fog computing," *Multimed Tools Appl*, vol. 84, no. 8, pp. 4907–4943, 2025, doi: 10.1007/s11042-024-18919-0.

[15] S. B, S. M, M. K, and L. B, "Ensemble of feature augmented convolutional neural network and deep autoencoder for efficient detection of network attacks," *Sci Rep*, vol. 15, no. 1, p. 4267, 2025, doi: 10.1038/s41598-025-88243-6.

[16] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Preprocessing Tools for Data Preparation," in *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, O. A. Montesinos López, A. Montesinos López, and J. Crossa, Eds., Cham: Springer International Publishing, 2022, pp. 35–70. doi: 10.1007/978-3-030-89010-0_2.

[17] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Inf Process Manag*, vol. 39, no. 1, pp. 45–65, Jan. 2003, doi: 10.1016/S0306-4573(02)00021-3.

[18] R. K. Dey and A. K. Das, "Modified term frequency-inverse document frequency based deep hybrid framework for sentiment analysis," *Multimed Tools Appl*, vol. 82, no. 21, pp. 32967–32990, 2023, doi: 10.1007/s11042-023-14653-1.

[19] P. Li, Y. Pei, and J. Li, "A comprehensive survey on design and application of autoencoder in deep learning," *Appl Soft Comput*, vol. 138, p. 110176, May 2023, doi: 10.1016/J.ASOC.2023.110176.

[20] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: a survey," *Artif Intell Rev*, vol. 57, no. 2, p. 28, 2024, doi: 10.1007/s10462-023-10662-6.

[21] Y. Abdulmalik, "An Improved SQL Injection Attack Detection Model Using Machine Learning Techniques," *International Journal of Innovative Computing*, vol. 11, no. 1, pp. 53–57, Apr. 2021, doi: 10.11113/ijic.v11n1.300.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272

[22] G. H. Luu *et al.*, "XSShield: A novel dataset and lightweight hybrid deep learning model for XSS attack detection," *Results in Engineering*, vol. 24, p. 103363, Dec. 2024, doi: 10.1016/J.RINENG.2024.103363.
[23] B. Huang *et al.*, "Stateless Q-learning algorithm for service caching in resource constrained edge environment," *Journal of Cloud Computing*, vol. 12, no. 1, p. 132, 2023, doi: 10.1186/s13677-023-00506-7.
[24] Y. Hammad, A. Al-Said Ahmad, and P. Andras, "An empirical study on the performance overhead of code instrumentation in containerised microservices," *Journal of Systems and Software*, vol. 230, p. 112573, Dec. 2025, doi: 10.1016/J.JSS.2025.112573.
[25] C. I. Allen Akselrud, "Random forest regression models in ecology: Accounting for messy biological data and producing predictions with uncertainty," *Fish Res*, vol. 280, p. 107161, Dec. 2024, doi: 10.1016/J.FISHRES.2024.107161.
[26] J. Zhang, "Impact of an improved random forest-based financial management model on the effectiveness of corporate sustainability decisions," *Systems and Soft Computing*, vol. 6, p. 200102, Dec. 2024, doi: 10.1016/J.SASC.2024.200102.
[27] H. Hruschka, "Interpretation Aids for Multilayer Perceptron Neural Nets," in *Data Analysis and Decision Support*, D. Baier, R. Decker, and L. Schmidt-Thieme, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 57–64. doi: 10.1007/3-540-28397-8_7.
[28] T. Sananmuang, K. Mankong, and K. Chokeshaiusaha, "Multilayer perceptron and support vector regression models for feline parturition date prediction," *Heliyon*, vol. 10, no. 6, p. e27992, Mar. 2024, doi: 10.1016/J.HELIYON.2024.E27992.
[29] A. Chatterjee, J. Saha, and J. Mukherjee, "Clustering with multi-layered perceptron," *Pattern Recognit Lett*, vol. 155, pp. 92–99, Mar. 2022, doi: 10.1016/J.PATREC.2022.02.009.
[30] M. Rezaei, A. Soltani, and H. N. Kashkooli, "Predicting active travel durations in Tehran: A multilayer perceptron approach," *Journal of Urban Mobility*, vol. 7, p. 100126, Jun. 2025, doi: 10.1016/J.URBMOB.2025.100126.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i1.15272