Sains Data Jurnal Studi Matematika dan Teknologi
ISSN 2986-903X
v3i2(2025) pp. 62-71
https://pub.nuris.ac.id/sainsdata

Klasifikasi URL Phishing untuk SIEM: Perbandingan Model Machine Learning XGBoost dan Deep Learning TabNet dalam Deteksi Ancaman Siber

Azza Farichi Tjahjono[1], Hasan[2], Randist Prawandha Putera[3],
Dionisius Marcell Putra Indranto[4], Abhirama Triadyatma Hermawan[5]
[1,2,3,4,5]Teknologi Informasi, Institut Teknologi Sepuluh Nopember
E-mail: [1] 5027231071@student.its.ac.id , [2] 5027231073.@student.its.ac.id,
[3] 5027231059.@student.its.ac.id , [4] 5027231044@student.its.ac.id , [5] 5027231061@student.its.ac.id

DOI: https://doi.org/10.52620/sainsdata.v3i2.227

Abstrak
Deteksi phishing merupakan komponen krusial dalam sistem *Security Information and Event Management* (SIEM) modern, yang menuntut akurasi tinggi dan kinerja waktu nyata *(real-time)*. Penelitian ini menyajikan perbandingan komprehensif antara model Gradient-Boosted Decision Tree, yaitu XGBoost, dengan arsitektur deep learning, TabNet, untuk klasifikasi URL phishing. Kedua model dioptimalkan secara sistematis menggunakan teknik penalaan hyperparameter tingkat lanjut, Randomized Search untuk XGBoost dan Optuna dengan pruning untuk TabNet guna memastikan evaluasi yang adil dan kokoh. Model-model tersebut dilatih dan diuji menggunakan Dataset of Suspicious Phishing URL Detection, sebuah koleksi fitur URL yang baru dan relevan. Hasil penelitian menunjukkan bahwa model XGBoost yang telah ditala secara signifikan mengungguli model TabNet yang telah ditala di semua metrik utama. Lebih lanjut, analisis kecepatan inferensi mengungkapkan bahwa XGBoost secara substansial lebih efisien pada perangkat keras CPU maupun GPU, dengan waktu inferensi GPU lebih dari 33 kali lebih cepat daripada TabNet. Temuan ini mengarah pada kesimpulan bahwa untuk tugas ini, XGBoost menawarkan kombinasi akurasi, kecepatan, dan kepraktisan implementasi yang superior, menjadikannya arsitektur yang lebih sesuai untuk diintegrasikan ke dalam sistem SIEM.
Kata Kunci: Deteksi Phishing, XGBoost, TabNet, SIEM, Kecepatan Inferensi

Abstract
Phishing detection is a critical component of modern Security Information and Event Management (SIEM) systems, requiring both high accuracy and real-time performance. This study conducts a comprehensive comparison between a Gradient-Boosted Decision Tree model, XGBoost, and a deep learning architecture, TabNet, for classifying phishing URLs. Both models were systematically optimized using advanced hyperparameter tuning techniques, Randomized Search for XGBoost and Optuna with pruning for TabNet to ensure a fair and robust evaluation. The models were trained and tested on the "Dataset of Suspicious Phishing URL Detection," a recent and relevant collection of URL features. The results demonstrate that the tuned XGBoost model significantly outperforms the tuned TabNet model across all key metrics. Furthermore, inference speed analysis revealed XGBoost to be substantially more efficient on both CPU and GPU hardware, with a GPU inference time over 33 times faster than TabNet. These findings lead to the conclusion that for this task, XGBoost offers a superior combination of accuracy, speed, and practical deployability, making it the more suitable architecture for integration into a SIEM system.
Keywords: *Phishing Detection, XGBoost, TabNet, SIEM, Inference Speed*

PENDAHULUAN

Serangan siber terus berkembang seiring meningkatnya ketergantungan pada teknologi digital. Salah satu ancaman yang paling sering ditemui dan berdampak signifikan adalah phishing, yaitu metode yang digunakan penyerang untuk memperoleh informasi sensitif seperti kredensial login atau data finansial dengan menyamar sebagai entitas tepercaya melalui situs web, email, atau pesan palsu (Schmitt & Flechais, 2024). Dalam konteks keamanan siber, kemampuan untuk mendeteksi phishing secara cepat dan akurat sangat penting, terutama dalam sistem seperti *Security Information and Event Management* (SIEM), yang bertujuan untuk memantau, mengidentifikasi, dan merespons ancaman secara real-time (Naqvi et al., 2023; Basit et al., 2021).

Tantangan utama dalam deteksi phishing terletak pada keragaman teknik yang digunakan pelaku, seperti pemalsuan domain agar menyerupai situs resmi, penggunaan karakter khusus untuk mengaburkan tampilan URL, serta modifikasi struktur tautan agar dapat lolos dari sistem deteksi otomatis (Jain & Gupta, 2022). Serangan ini semakin kompleks seiring munculnya teknik phishing berbasis deepfake dan kecerdasan buatan, yang memperbesar tantangan dalam upaya pencegahan dan deteksi dini (de Rancourt-Raymond & Smaili, 2023). Menurut Jain & Gupta (2022) phishing merupakan salah satu vektor serangan yang paling umum digunakan dalam pelanggaran data, dengan tingkat keberhasilan yang tinggi karena memanfaatkan kelemahan manusia sebagai faktor keamanan. Serangan dunia maya dimulai dengan aktivitas phishing, menjadikannya prioritas utama dalam mitigasi risiko keamanan siber (Ashfaq et al., 2023; Park & Kim, 2025).

Mengingat keterbatasan metode berbasis aturan yang cenderung statis, pendekatan berbasis machine learning (ML) dan deep learning (DL) semakin banyak diterapkan karena kemampuannya dalam mengenali pola serangan yang lebih kompleks dan dinamis (Catal et al., 2022; Dewis & Viana, 2022, Metwaly et al., 2024; Sahingoz et al., 2019). Oleh karena itu perlu pendekatan hibrid antara ML dan DL untuk meningkatkan akurasi deteksi phishing (Bountakas & Xenakis, 2023; Karim et al., 2023; Nayak et al., 2025). Pendekatan berbasis ensemble seperti Random Forest dan Gradient Boosting juga sering digunakan bersama teknik deep learning untuk hasil yang lebih optimal (Chiew et al., 2019 ; Wang et al., 2021).

Selain pendekatan tradisional, berbagai arsitektur deep learning juga mulai menunjukkan performa yang mengesankan dalam tugas deteksi phishing. Namun demikian, sejumlah studi terkini menunjukkan bahwa algoritma berbasis boosting, khususnya XGBoost, secara konsisten mengungguli baik model machine learning maupun deep learning dalam hal akurasi dan kecepatan inferensi pada beragam dataset phishing (Anitha et al., 2022; Do et al., 2022; Lin et al., 2021; Yang et al., 2021).

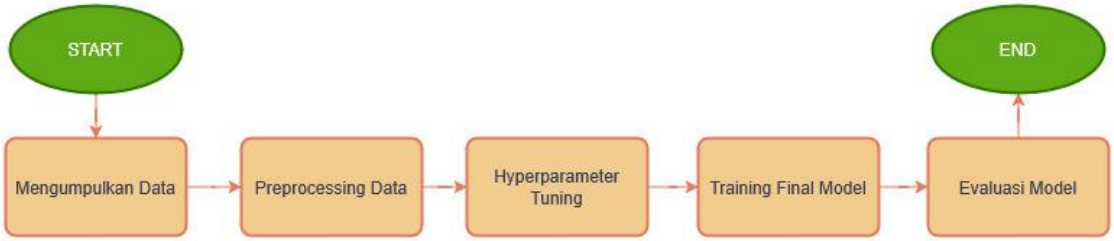
Dengan meningkatnya kompleksitas serangan dan evolusi teknik phishing , diperlukan solusi yang tidak hanya akurat tetapi juga adaptif terhadap perubahan lingkungan ancaman. Oleh karena itu, penelitian ini bertujuan untuk menyediakan basis empiris dalam memilih model yang paling efektif untuk integrasi ke dalam sistem manajemen keamanan berbasis real-time analytics.

Penelitian ini membandingkan model XGBoost, algoritma berbasis pohon keputusan yang dikenal dengan efisiensi dan akurasinya dalam klasifikasi data, dengan model deep learning seperti TabNet dan Deep Neural Network (DNN), yang memiliki keunggulan dalam menangkap pola non-linear pada data tabular. Selain itu, TabNet juga menunjukkan performa yang baik dalam interpretasi fitur, menjadikannya alternatif kuat dalam aplikasi analitik keamanan siber. Perbandingan dilakukan berdasarkan metrik evaluasi seperti akurasi, tingkat false positive , serta kecepatan prediksi, guna menentukan pendekatan yang paling sesuai untuk diterapkan dalam sistem SIEM. Dataset yang digunakan dalam penelitian ini merupakan kumpulan URL mencurigakan yang telah diverifikasi secara manual, sehingga memberikan representasi realistis terhadap variasi pola phishing yang ada di dunia nyata (NaliniPriya et al., 2023; Zhu et al., 2022).

Penelitian ini bertujuan untuk membandingkan kinerja model XGBoost dan TabNet dalam mendeteksi URL phishing menggunakan dataset yang kaya fitur terkait karakteristik URL. Pemilihan kedua model didasarkan pada perbedaan pendekatan dalam memproses data tabular, XGBoost sebagai model berbasis pohon keputusan yang unggul dalam menangani data dengan fitur eksplisit dan Tabnet merupakan model deep learning yang mampu menangkap pola non-llinear dengan mekanisme perhatian (attention mechanism).

METODE

Metodologi penelitian ini mencakup beberapa tahapan utama yaitu pemilihan dataset, pra-pemrosesan data, implementasi model, serta evaluasi performa menggunakan metrik yang relavan



Gambar 1. Flowchart Proses pada Metodologi Penelitian

Datasets

Dataset yang digunakan dalam penelitian ini adalah "Dataset of Suspicious Phishing URL Detection" yang dipublikasikan di Frontiers in Computer Science. Dataset ini dikembangkan untuk memfasilitasi penelitian dalam mendeteksi phishing berdasarkan analisis karakteristik URL. Dataset ini dibuat berdasarkan data pada tahun 2024, jadi masih tergolong bari dan bisa dibilang akuran untuk training model Machine Learning dan Deep Learning

Dataset yang digunakan dalam penelitian ini terdiri dari 247,950 entri URL dengan 42 fitur atau kolom numerik yang diekstrak dari URL dan domainnya. Dataset besar dan seimbang sangat penting dalam memastikan validitas model deteksi phishing (López et al., 2014).

Pra-Pemrosesan Data

Sebelum memulai pelatihan model, langkah-langkah pra-pemrosesan yang cermat dilakukan untuk memastikan memastikan kualitas data yang digunakan. Tujuan dari proses ini adalah untuk meningkatkan akurasi model dan meminimalkan potensi bias dalam pembelajaran.

Tahap pra-pemrosesan mencakup beberapa langkah penting:

Pembersihan Data

Langkah pertama adalah memeriksa adanya nilai yang hilang atau duplikasi data yang bisa mempengaruhi hasil prediksi. Selanjutnya, entri yang tidak relevan atau yang mengandung noise akan dihapus untuk meningkatkan performa model.

Transformasi Fitur

Fitur numerik perlu dinormalisasi atau distandarisasi agar memiliki skala yang seragam, terutama ketika menggunakan model deep learning yang sangat sensitif terhadap skala data (Maulani et al., 2025). Dalam proses ini, penerapan scaling Min-Max Scaler untuk XGBoost dan Standard Scaler untuk TabNet.

Kami menggunakan Standard Scaler pada TabNet karena model ini sangat sensitif terhadap skala fitur. Standard Scaler mengubah distribusi fitur menjadi rata-rata 0 dan standar deviasi 1.

Untuk fitur seperti url_length, path_length, dan entropy_of_url yang memiliki rentang nilai bervariasi, diterapkan MinMaxScaler agar semua fitur berada dalam skala 0-1. Meski XGBoost tidak memerlukan penskalaan, langkah ini dilakukan untuk menyeragamkan seluruh fitur dan menghindari bias akibat perbedaan skala.

Pembagian Dataset

Dataset awalnya dibagi menjadi dua bagian utama, yaitu dataset_training.csv dan dataset_testing.csv dengan rasio 9:1. Kemudian, dataset_training kembali dibagi menjadi subset sementara untuk validasi. Yakni 80% sebagai data train dan 20% sebagai data validasi. Sedangkan dataset_test.csv digunakan pada saat evaluasi akhir model XGBoost dan TabNet.

Penanganan Data Imbalance

Strategi penanganan data imbalance bisa mencakup SMOTE atau stratified sampling jika dataset menunjukkan distribusi target yang tidak seimbang (Bharuka et al., 2024). Berdasarkan analisis, diketahui bahwa distribusi dataset cukup seimbang, sehingga langkah-langkah lebih lanjut untuk penanganan data yang tida seimbang tidak diperlukan.

Dengan mengikuti tahapan pra-pemrosesan tersebut, diharapkan model yang dibangun bekerja dengan optimal

Implementasi, Pelatihan dan Optimisasi Model

Setelah tahap pra-pemrosesan data, penelitian ini dilanjutkan ke tahap implementasi, optimisasi dan pelatihan model. Kami memilih XGBoost karena teknik ensemble seperti boosting memiliki potensi performa lebih tinggi dalam deteksi phishing jika dipadukan dengan fitur domain dan HTML URL. Dua arsitektur utama, XGBoost dan TabNet, dipilih untuk perbadingan. Untuk memastikan perbandingan yang adil dan mencapai performa puncak dari masing masing model, proses hyperparameter tuning yang sistematis diterapkan pada kedua model sebelum training model final.

Pelatihan dan Tuning XGBoost

Model XGBoost atau *Xtreme Gradient Boost* diimplementasikan dengan pustaka *xgboost* di python. Proses optimisasi dilakukan dengan metode *Randomized Search* dengan validasi dengan validasi silang atau *cross-validation 3-fold*. Penelitian terdahulu menunjukkan bahwa GridSearchCV dan RandomizedSearch merupakan metode populer dalam tuning model berbasis pohon seperti XGBoost (Ahammad tl al., 2022; Chen & Guestrin, 2016). Fine-tuning XGBoost dengan RandomizedSearchCV secara signifikan meningkatkan akurasi deteksi phishing, dan model ini juga cocok untuk implementasi cloud real-time (Anggoro et al., 2021).

Parameter-parameter utama yang dioptimalkan mencakup:

Tabel 1. Parameter XGBoost

NO	PARAMETER	DESKRIPSI
1	n_estimator	Jumlah pohon
2	max_depth	Kedalaman maksimum pohon
3	learning_rate	Laju pembelajaran
4	gamma	Parameter regulasi
5	subsample	Mengontrol dan mencegah ovefitting
6	colsample_bytree	Mengontrol dan mencegah overfitting

Metrik F1-Score digunakan sebagai target optimisasi selama proses *cross-validation* atau validasi silang.

Pelatihan dan Tuning TabNet

Implementasi TabNet dilakukan menggunakan pustaka pytorch-tabnet, sebuah kerangka kerja berbasis PyTorch yang dirancang khusus untuk arsitektur ini. Kami menggunakan framework Optuna yang mengimplementasikan algoritma optimisasi Bayesian untuk mencari ruang parameter secara cerdas. Penggunaan Optuna untuk tuning TabNet memberikan fleksibilitas eksplorasi parameter yang cerdas secara otomatis (Hanifi et al., 2024). Untuk meningkatkan performa klasifikasi pada model TabNet, proses pelatihan dilakukan dengan memperhatikan pemilihan fitur dan parameter yang relevan terhadap pola URL phishing.

Pendekatan ini selaras dengan temuan Naseer et al. (2025), yang menunjukkan bahwa kombinasi feature selection dan ensembel TabNet dapat meningkatkan kemampuan deteksi terhadap URL yang disamarkan (obfuscated), sekaligus mempertahankan aspek explainability dari model berbasis perhatian seperti TabNet (Naseer et al., 2025).

Proses tuning difokuskan pada hyperparameter yang paling berpengaruh mencakup:

Tabel 2. Parameter TabNet

NO	PARAMETER	DESKRIPSI
1	n_d	Dimensi fitur
2	n_a	Dimensi attention
3	n_steps	Jumlah langkah keputusan
4	gamma	Parameter regulasi atau faktor penyeimbang
5	learning_rate	Laju pembelajaran

Untuk mempercepat proses pencarian, kami juga mengimplementasikan mekanisme pruning yang secara otomatis menghentikan trial yang tidak menjanjikan diawal. Metrik AUC (Area Under ROC Curve) dipilih sebagai target utama untuk dimaksimalkan selama proses tuning. Studi oleh Norton dan Uryasev (2019) menunjukkan bahwa optimasi langsung terhadap nilai AUC maupun Buffered AUC dapat meningkatkan performa klasifikasi, khususnya dalam konteks dataset dengan distribusi kelas yang tidak seimbang (Norton & Uryasev, 2019).

Training Model Final

Setelah kombinasi *hyperparameter* terbaik untuk masing-masing model ditemukan melalui proses tuning, sebuah model final dilatih dari awal. Proses pelatihan final ini menggunakan keseluruhan data dari *dataset_training.csv* untuk memaksimalkan pemanfaatan informasi data yang tersedia.

Evaluasi Model

Tahap evaluasi dilakukan untuk mengukur dan membandingkan kinerja model XGBoost dan TabNet yang telah dilatih secara final. Evaluasi ini menggunakan set data uji yang sepenuhnya terpisah dan belum pernah digunakan selama proses training maupun tuning, untuk memastikan penilaian yang subjektif.

Metrik performa utama yang diukur adalah Accuracy, Precision, Recall, dan F1-Score untuk menilai efektivitas klasifikasi. Selain itu, kecepatan inferensi juga diukur untuk setiap model pada lingkungan CPU dan GPU secara terpisah. Pengukuran kecepatan ini dilakukan dengan menghitung rata-rata waktu eksekusi dari 50 kali prediksi pada seluruh data uji untuk mendapatkan hasil yang stabil dan andal. Hasil dari evaluasi komprehensif ini menjadi dasar untuk menarik kesimpulan mengenai model mana yang paling unggul untuk tugas deteksi URL phishing dalam konteks implementasi SIEM.

HASIL DAN PEMBAHASAN

Lingkungan Eksperimen

Seluruh eksperimen, termasuk pelatihan dan evaluasi model, dijalankan pada platform Google Colaboratory yang dilengkapi dengan satu GPU NVIDIA T4. Perangkat lunak utama yang digunakan meliputi Python, dengan pustaka-pustaka inti seperti Scikit-learn untuk pra-pemrosesan dan evaluasi, XGBoost (versi 2.1.4) untuk model gradient boosting, serta PyTorch (versi 2.6.0+cu124) dan Pytorch-TabNet (versi 4.1.0) untuk model deep learning. Proses optimisasi hyperparameter difasilitasi oleh pustaka Optuna (versi 4.3.0).

Hasil Performa Model Klasifikasi

Evaluasi utama difokuskan pada kemampuan masing-masing model dalam mengklasifikasikan URL phishing secara akurat setelah melalui proses hyperparameter tuning. Performa kedua model pada set data uji

dirangkum dalam Tabel 3.

Tabel 3. Perbandingan Metrik Performa Model Final

Model	Accuracy	Precision	Recall	F1-Score
XGBoost (Tuned)	0.95503	0.96791	0.939	0.953
TabNet (Tuned)	0.9059	0.9294	0.873	0.900

Secara keseluruhan, model XGBoost menunjukkan keunggulan yang signifikan di semua metrik evaluasi yang diukur. Model ini berhasil mencapai F1-Score sebesar 0.9533, jauh melampaui TabNet yang mencatatkan skor 0.900. Keunggulan serupa juga tercermin pada metrik akurasi, di mana XGBoost memperoleh nilai 0.9550 dibandingkan dengan 0.9059 dari Tabnet.

Dari perspektif keamanan siber, nilai presisi yang lebih tinggi pada XGBoost yaitu 0.9679, mengindikasikan tingkat false positive yang lebih rendah, aertinya mdoel ini lebih jarang salah mengklasifikasikan URL yang aman sebagai phishing, sehingga mengurangi potensi gangguan bagi pengguna. Sementara itu, nilai recall yang juga lebih superior sekitar 0.9391, menunjukkan kemampuan XGBoost yang lebih baik dalam mengidentifikasi URL phishing yang sebenarnya dan meminimalkan jumlah ancaman yang lolos dari deteksi.

Temuan ini mengindikasikan bahwa untuk dataset dan tugas klasifikasi ini, arsitektur gradient-boosted decision tree (GBDT) yang telah dioptimalkan lebih efektif dalam membedakan pola URL phishing dibandikan dengan arsitektur deep learning TabNet. Bahkan setelah kedua model melalui proses hyperparameter tuning yang sistematis, kekuatan inheren XGBoost dalam menangani data tabular terstruktur terbukti lebih dominan.

Analisis Kecepatan Inferensi

Selain akurasi, kecepatan deteksi *(inference time)* adalah faktor krusial untuk implementasi pada sistem SIEM yang beroperasi secara real-time. Tabel 4 menyajikan perbandingan kecepatan inferensi rata-rata per sampel untuk kedua model yang telah di optimalkan, yang diukur secara terpisah pada lingkungan CPU dan GPU.

Tabel 4. Perbandingan Kecepatan Inferensi Rata-rata per Sampel (ms)

Model	CPU	GPU
XGBoost (Tuned)	0.034551	0.026675
TabNet (Tuned)	0.026744	0.000804

Pada lingkungan CPU, yang merepresentasikan skenario deployment paling umum, XGBoost menunjukkan efisiensi yang lebih tinggi dengan waktu inferensi 0.034 ms. Hal ini menempatkan XGBoost sebagai pilihan yang lebih praktis untuk sistem yang tidak bergantung pada akselerasi perangkat keras khusus.

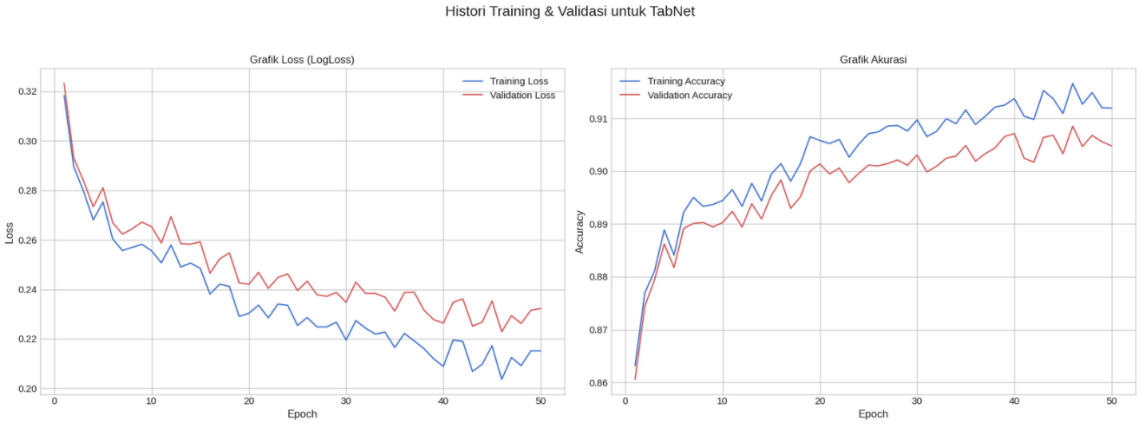
Ketika akselerasi GPU dimanfaatkan, keunggulan kecepatan XGBoost menjadi semakin dominan. Dengan waktu inferensi hanya 0.0008 ms, XGBoost tercatat **33 kali lebih cepat** daripada TabNet (0.0266 ms) pada perangkat keras yang sama. Meskipun model deep learning seperti TabNet dirancang untuk mendapatkan keuntungan dari GPU, hasil ini menunjukkan tingkat optimisasi yang sangat matang pada pustaka XGBoost untuk komputasi paralel, yang secara signifikan melampaui efisiensi TabNet pada tugas ini.

Secara konsisten pada kedua platform pengujian, XGBoost terbukti menjadi model yang jauh lebih efisien. Kecepatan superior ini, ketika digabungkan dengan performa akurasi yang lebih tinggi dari analisis sebelumnya, memperkuat posisinya sebagai kandidat model yang lebih unggul untuk aplikasi deteksi ancaman secara real-time dalam sistem SIEM.

Analisis Proses Pelatihan Model

Untuk mendapatkan wawasan yang lebih dalam mengenai perilaku kedua model selama proses pembelajaran, kami memvisualisasikan histori pelatihan mereka. Gambar 2 dan Gambar 3 masing-masing menampilkan kurva loss dan akurasi pada set data training dan validation untuk model XGBoost dan TabNet dengan *hyperparameter* terbaiknya.

Gambar 2. Grafik Loss dan Akurasi XGBoost: (Dokumentasi penulis, 2025)



Gambar 3. Grafik Loss dan Akurasi TabNet: (Dokumentasi penulis, 2025)

Dari kedua gambar tersebut, dapat ditarik beberapa observasi penting. Pada **Gambar 2**, model XGBooost menunjukkan konvergensi yang sangat cepat dan stabil. Terlihat bahwa performa pada set validasi (kurva merah) meningkat secara drastis dalam 10-20 epoch pertama dan kemudian mencapai titik plateau, menandakan bahwa model telah dengan cepat mempelajari pola-pola utama dari data. Kesenjangan (gap) antara kurva training dan validation yang stabil menunjukkan bahwa model berhasil menggeneralisasi pengetahuan dengan baik tanpa mengalami overfitting yang merusak, berkat mekanisme regulasi yang efektif.

Di sisi lain, pada Gambar 3, model TabNet menampilkan karakteristik proses belajar yang berbeda. Kurva pelatihan dan validasinya terlihat lebih fluktuatif dan noisy, yang merupakan perilaku umum pada model deep learning yang menggunakan optimizer berbasis stochastic gradient descent. Proses konvergensinya lebih gradual dan tidak mencapai plateau yang stabil seperti XGBoost, melainkan berosilasi di sekitar titik performa terbaiknya. Hal ini menyoroti pentingnya mekanisme early stopping yang telah diterapkan, yang berhasil menangkap odel pada epoch dengan performa validasi optimal sebelum adanya potensi degradasi performa.

Secara komparatif, analisis visual ini memperkuat hasil kuantitatif sebelumnya, efisiensi dan stabilitas proses pembelajaran XGBoost selaras dengan keunggulannya dalam metrik performa akhir. Model ini tidak hanya mencapai hasil yang lebih superior tetapi juga melakukannya dengan cara yang lebih cepat dan dapat diprediksi selama fase pelatihan.

Pembahasan Implikasi untuk Implementasi SIEM

Berdasarkan analisis performa, kecepatan, dan perilaku pelatihan, kita dapat menarik kesimpulan praktis mengenai implikasi implementasi kedua model pada sistem Security Information and Event Management (SIEM). Hasil penelitian secara konsisten menunjukkan bahwa XGBoost tidak hanya unggul dalam hal akurasi deteksi, tetapi juga memiliki efisiensi komputasi yang jauh lebih superior.

Dalam konteks SIEM yang menuntut pemrosesan data secara real-time untuk deteksi ancaman yang cepat, kecepatan inferensi menjadi faktor penentu. Kemampuan XGBoost untuk memberikan prediksi dengan latensi yang sangat rendah, terutama pada lingkungan CPU (0.0267 ms/sampel), menjadikannya sangat ideal untuk diintegrasikan ke dalam alur kerja analisis keamanan tanpa menjadi bottleneck komputasi. Bahkan pada lingkungan GPU, optimisasi XGBoost terbukti luar biasa, menghasilkan kecepatan lebih dari 33 kali lipat dibanding dengan TabNet.

Keunggulan pada metrik presisi dan recall juga memiliki implikasi operasional yang penting. Presisi yang lebih tinggi pada XGBoost (0.967) secara langsung mengurangi beban kerja analis keamana dengan meminimalkan alarm palsu (*false positive*), sementara recall yang lebih tinggi (0.939) memastikan cakupan deteksi ancaman yang lebih luas dan anda, mengurangu risiko ancaman phishing lolos dari pemantauan.

Meskipun TabNet menunjukkan performa yang layak setelah tuning, ketergantungannya pada akselerasi GPU untuk mencapai kecepatan yang optimal yang bahkan masih kalah dari XGBoost menjadi pertimbangan biasa dan infrastruktur yang signifikan. Performa tinggi XGBoost pada CPU menjadikannya solusi yang lebih fleksibel, hemat biaya, dan mudah diimplementasikan pada berbagai skala infrastruktur perusahaan yang mungkin tidak memiliki perangkat keras GPU khusus untuk setiap komponen SIEM. Stabilitas yang ditunjukkan selama proses pelatihan juga mengindikasikan bahwa XGBoost adalah model yang lebih matang dan dapat diandalkan untuk aplikasi keamanan siber yang kritis.

Dengan demikian, menimbang secara holistik dantara akurasi, kecepatan, dan kepraktisan implementasi, model XGBoost secara tegas direkomendasikan sebagai arsitektur yang lebih unggul dan siap untuk diimplementasikan pada sistem SIEM untuk tugas deteksi URL phishing.

KESIMPULAN DAN SARAN

Penelitian ini melakukan perbandingan komprehensif antara model machine learning XGBoost dan model deep learning TabNet untuk tugas deteksi URL phishing. Hasil evaluasi setelah proses optimisasi hyperparameter yang sistematis menunjukkan bahwa model XGBoost secara konsisten mengungguli TabNet di semua metrik performa, termasuk F1-Score (0.9533 vs 0.9007), serta memiliki kecepatan inferensi yang jauh lebih superior baik pada lingkungan CPU maupun GPU. Berdasarkan temuan ini, XGBoost terbukti menjadi model yang lebih akurat, efisien, dan andal untuk diimplementasikan pada sistem SIEM. Kebaruan penelitian ini terletak pada analisis perbandingan yang tidak hanya berfokus pada akurasi, tetapi juga mempertimbangkan efisiensi komputasi dan stabilitas pelatihan sebagai faktor krusial untuk aplikasi keamanan siber real-time. Kelebihan utama dari studi ini adalah penerapan metodologi tuning yang adil dan relevan untuk masing-masing arsitektur.

Meskipun penelitian ini telah memberikan perbandingan yang komprehensif antara XGBoost dan TabNet untuk tugas deteksi URL phishing, terdapat beberapa keterbatasan yang perlu diakui dan dapat menjadi arahan untuk penelitian di masa depan antara lain: Pertama, kesimpulan yang ditarik didasarkan pada analisis dari satu dataset spesifik, yaitu "Dataset of Suspicious Phishing URL Detection". Meskipun dataset ini relevan dan modern, performa relatif dari kedua model mungkin dapat bervariasi pada dataset phishing lain yang memiliki distribusi fitur, karakteristik URL, atau jenis serangan yang berbeda. Kedua, proses hyperparameter tuning yang menggunakan RandomizedSearchCV untuk XGBoost dan Optuna untuk TabNet telah menjelajahi sebagian besar ruang parameter yang relevan. Namun, proses ini tidak menjamin penemuan hyperparameter optimal secara global. Ada kemungkinan bahwa dengan pencarian yang lebih ekstensif atau menggunakan teknik optimisasi yang berbeda, performa salah satu atau kedua model dapat ditingkatkan lebih jauh, meskipun hal tersebut akan membutuhkan sumber daya komputasi yang jauh lebih besar. Ketiga, studi ini secara spesifik membatasi

perbandingan pada dua model representatif: XGBoost sebagai standar industri untuk GBDT dan TabNet sebagai arsitektur deep learning yang dirancang untuk data tabular. Penelitian ini tidak menyertakan model-model kuat lainnya seperti varian GBDT lain (misalnya, LightGBM, CatBoost) atau arsitektur deep learning yang lebih baru (misalnya, berbasis Transformer) yang juga menunjukkan hasil menjanjikan pada data tabular. Keterbatasan-keterbatasan ini tidak mengurangi validitas temuan dalam konteks penelitian ini, namun membuka peluang untuk investigasi lebih lanjut di masa depan, seperti melakukan studi komparatif yang lebih luas pada berbagai dataset dan arsitektur model.

Oleh karena itu, penelitian di masa depan direkomendasikan untuk: 1) Melakukan validasi temuan ini pada dataset yang lebih beragam untuk menguji generalisasi model. 2) Memperluas perbandingan dengan menyertakan model-model state-of-the-art lainnya, seperti varian Transformer untuk data tabular. 3) Mengeksplorasi dampak dari rekayasa fitur (feature engineering) terhadap performa kedua jenis model.

REFERENSI

Ahammad, S. H., Kale, S. D., Upadhye, G. D., Pande, S. D., Babu, E. V., Dhumane, A. V., & Bahadur, M. D.K. J. (2022). Phishing URL detection using machine learning methods. Advances in Engineering Software, 173, 103288. https://doi.org/10.1016/j.advengsoft.2022.103288

Anggoro, D. A., & Mukti, S. S. (2021). Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure. International Journal of Intelligent Engineering & Systems, 14(6). https://doi.org/10.22266/ijies2021.1231.19

Anitha, J., & Kalaiarasu, M. (2022). A new hybrid deep learning-based phishing detection system using MCS-DNN classifier. Neural Computing and Applications, 34(8), 5867-5882. https://doi.org/10.1007/s00521-021-06717-w

Arik, S. Ö., & Pfister, T. (2021, May). Tabnet: Attentive interpretable tabular learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 8, pp. 6679-6687). https://doi.org/10.1609/aaai.v35i8.16826

Ashfaq, S., Chandre, P., Pathan, S., Mande, U., Nimbalkar, M., & Mahalle, P. (2023). Defending against vishing attacks: A comprehensive review for prevention and mitigation techniques. In International Conference on Recent Developments in Cyber Security (pp. 411-422). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-9811-1_33

Basit, A., Zafar, M., Liu, X., Javed, A. R., Jalil, Z., & Kifayat, K. (2021). A comprehensive survey of AI enabled phishing attacks detection techniques. Telecommunication Systems, 76, 139-154. https://doi.org/10.1007/s11235-020-00733-2

Bountakas, P., & Xenakis, C. (2023). Helphed: Hybrid ensemble learning phishing email detection. Journal of Network and Computer Applications, 210, 103545. https://doi.org/10.1016/j.jnca.2022.103545

Catal, C., Giray, G., Tekinerdogan, B., Kumar, S., & Shukla, S. (2022). Applications of deep learning for phishing detection: a systematic literature review. Knowledge and Information Systems, 4(6), 1457-1500. https://doi.org/10.1007/s10115-022-01672-x

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp 885-794). https://doi.org/10.1145/2939672.2939785

Chiew, K. L., Tan, C. L., Wong, K., Yong, K. S., & Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Information Sciences, 484, 153-166. https://doi.org/10.1016/j.ins.2019.01.064

de Rancourt-Raymond, A., & Smaili, N. (2023). The unethical use of deepfakes. Journal of Financial Crime, 30(4), 1066-1077. https://doi.org/10.1108/JFC-04-2022-0090

Dewis, M., & Viana, T. (2022). Phish responder: A hybrid machine learning approach to detect phishing and spam emails. Applied System Innovation, 5(4), 73. https://doi.org/10.3390/asi5040073

Do, N. Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., & Fujita, H. (2022). Deep learning for phishing detection: Taxonomy, current challenges and future directions. Ieee Access, 10, 36429-36463. https://doi.org/10.1109/ACCESS.2022.3151903

El-Metwaly, A. E. S., Bedair, M. R., Abdallah, S. T., Mahmoud, A. M., Mohamed, M. E., Mahmoud, M. E., & Takieldeen, A. E. (2024, July). Detection of Phishing URLs Based on Machine Learning and Cybersecurity. In 2024 International Telecommunications Conference (ITC-Egypt) (pp. 394-398). IEEE. https://doi.org/10.1109/ITC-Egypt61547.2024.10620574

Hanifi, S., Cammarono, A., & Zare-Behtash, H. (2024). Advanced hyperparameter optimization of deep learning models for wind power prediction. Renewable Energy, 221, 119700. https://doi.org/10.1016/j.renene.2023.119700

Jain, A. K., & Gupta, B. B. (2022). A survey of phishing attack techniques, defence mechanisms and open research challenges. Enterprise Information Systems, 16(4), 527-565. https://doi.org/10.1080/17517575.2021.1896786

Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S. B., & Joga, S. R. K. (2023). Phishing detection system through hybrid machine learning based on URL. IEEE Access, 11, 36805-36822. https://doi.org/10.1109/ACCESS.2023.3252366

Lin, Y., Liu, R., Divakaran, D. M., Ng, J. Y., Chan, Q. Z., Lu, Y., ... & Dong, J. S. (2021). Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 3793-3810).

López, V., Fernández, A., & Herrera, F. (2014). On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. Information Sciences, 257, 1-13. https://doi.org/10.1016/j.ins.2013.09.038

Maulani, G., Hasan, F. N., Setiawan, D., Bowo, I. T., Ardhana, V. Y. P., Ramdhani, Y., Safitri, R. (2025). Machine Learning. MEGA PRESS NUSANTARA.

NaliniPriya, G., Damoddaram, K., Gopi, G., & Nitish Kumar, R. (2023, February). Phishing attack detection using machine learning. In International Conference on Emerging Trends in Expert Applications & Security (pp. 301-312). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-1946-8_27

Naqvi, B., Perova, K., Farooq, A., Makhdoom, I., Oyedeji, S., & Porras, J. (2023). Mitigation strategies against the phishing attacks: A systematic literature review. Computers & Security, 132, 103387. https://doi.org/10.1016/j.cose.2023.103387

Naseer, M., Ullah, F., Saeed, S., Algarni, F., & Zhao, Y. (2025). Explainable TabNet ensemble model for identification of obfuscated URLs with features selection to ensure secure web Browse. Scientific Reports, 15(1), 9496. https://doi.org/10.1038/s41598-025-93286-w

Nayak, G. S., Muniyal, B., & Belavagi, M. C. (2025). Enhancing Phishing Detection: A Machine Learning Approach With Feature Selection and Deep Learning Models. IEEE Access. https://doi.org/10.1109/ACCESS.2025.3543738

Norton, M., & Uryasev, S. (2019). Maximization of auc and buffered auc in binary classification. Mathematical Programming, 174, 575-612. https://doi.org/10.1007/s10107-018-1312-2

Park, J. Y., & Kim, T. S. (2025). An Automated Scenario Generation Model for Anti-phishing using Generative AI. In 2025 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 368-370). IEEE. https://doi.org/10.1109/BigComp64353.2025.00073

Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345-357. https://doi.org/10.1016/j.eswa.2018.09.029

Schmitt, M., & Flechais, I. (2024). Digital deception: Generative artificial intelligence in social engineering and phishing. Artificial Intelligence Review, 57(12), 1-23. https://doi.org/10.1007/s10462-024-10973-2

Yang, R., Zheng, K., Wu, B., Wu, C., & Wang, X. (2021). Phishing website detection based on deep convolutional neural network and random forest ensemble learning. Sensors, 21(24), 8281. https://doi.org/10.3390/s21248281

Zhu, E., Chen, Z., Cui, J., & Zhong, H. (2022). MOE/RF: a novel phishing detection model based on revised multiobjective evolution optimization algorithm and random forest. IEEE Transactions on Network and Service Management, 19(4), 4461-4478. https://doi.org/10.1109/TNSM.2022.3162885