International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Enhancing Multi-Step Cyber Attack Detection: The Impact of Hyperparameter Optimization on Deep and Ensemble Learning Models

L.Parthasarathi[1], Dr. N. Kamalraj[2]

[1]Research Scholar, Department of Computer Science, Park's College (Autonomous), Tirupur-641605, Tamil Nadu, India
[2]Associate Professor and Vice Principal, Park's College (Autonomous), Tirupur-641605, Tamil Nadu, India

Abstract
The increasing sophistication of multi step cyber attacks demands intrusion detection systems capable of modelling complex flow behaviours across large high dimensional network datasets. This study presents a comprehensive comparative evaluation of four learning architectures MLP, XGBoost, DNN, and DBN applied to the Multi Step Cyber Attack Dataset (MSCAD), which captures realistic multi stage attack scenarios. We analyse the influence of four hyperparameter optimisation strategies, namely Grid Search (GS), Random Search (RS), Bayesian Optimisation (BO), and Hyperband, on model effectiveness. Experimental results demonstrate that deep and ensemble based architectures achieve substantial performance gains after tuning, with BO and Hyperband consistently delivering the most significant improvements across all evaluation metrics. DBN and DNN show strong enhancements in hierarchical pattern extraction, while XGBoost maintains highly stable decision boundaries under tuned configurations. Findings highlight the critical role of both architectural design and optimisation methodology in improving detection accuracy for complex multi flow attack sequences. This study underscores the necessity of pairing advanced learning models with efficient hyperparameter optimisation to build scalable and robust intrusion detection systems for modern network environments.

Keywords: Intrusion Detection Systems, MSCAD, Hyperparameter Optimization, Deep Learning Models, Cyber Attack Detection

1. INTRODUCTION

The rising complexity of cyber attacks in modern networks makes it increasingly difficult for systems to maintain security and ensure dependable communication. Multi step intrusions now involve subtle shifts in flow behaviour, contextual traffic patterns, and layered attack sequences that traditional detection methods struggle to recognise. Although previous research has explored various machine learning and network analysis strategies, major challenges remain in model optimisation, scalability to large traffic environments, and generalisation across different attack types and network conditions.

Effective automated detection of advanced cyber attacks is essential for reducing operational risk, maintaining trust in digital systems, and safeguarding users from malicious activity. Improving intrusion detection can strengthen organisational resilience and enhance network reliability by using accurate,

efficient, and well tuned learning models. To support this goal, this work evaluates four models MLP, XGBoost, DNN, and DBN on the MSCAD dataset and investigates four hyperparameter optimisation methods Grid Search, Random Search, Bayesian Optimisation, and Hyperband to determine how tuning improves the recognition of complex multi-step attack behaviour.

Contributions:

1. A complete evaluation of four learning models MLP, XGBoost, DNN, and DBN on the MSCAD dataset to study their ability to detect multi step cyber attacks.
2. A systematic comparison of four hyperparameter tuning methods Grid Search, Random Search, Bayesian Optimisation, and Hyperband to measure their effect on detection performance.
3. A detailed analysis showing how tuning improves learning behaviour, strengthens feature representation, and enhances the detection of both common and rare attack classes.
4. Identification of model and optimisation combinations that deliver the best overall improvement in accuracy, precision, recall, and F1 score for real world intrusion detection settings.

This paper offers methodological insight and practical direction for building scalable and high performance intrusion detection systems capable of identifying complex cyber attack behaviour in modern network environments.

2. Literature Review

Zou et al. (2025) introduce a hyperparameter optimization approach that integrates Bayesian optimization with an EM-based framework using relative entropy to refine parameter updates. They reformulate Gaussian weight priors to derive clear E-Step/M-Step re-estimation equations, providing a mathematically grounded alternative to heuristic tuning. Experiments on synthetic data and the diabetes dataset show fast convergence (approx 10 iterations) and stable hyperparameter estimates. The method outperforms grid search and random search by reducing computational effort and improving tuning reliability. Their work emphasizes principled HPO for complex ML models where hyperparameters strongly affect performance. Overall, the study contributes a robust, theory-driven optimization strategy suitable for modern ML pipelines.

Liyew et al. (2025) investigate hyperparameter-optimized machine learning models for predicting actual evapotranspiration using meteorological and soil variables. They evaluate deep learning architectures such as LSTM, GRU, and CNN, along with classical models including SVR and Random Forest. Bayesian Optimization is employed for tuning, showing superior efficiency and accuracy compared to grid search. The LSTM-BO model delivers the best performance, achieving RMSE is 0.0230 and R^2 is 0.8861 with the full feature set. Even with reduced features, LSTM maintains strong generalization and outperforms SVR and other baselines. The study demonstrates that optimized DL models can effectively replace costly direct evapotranspiration measurements for environmental prediction.

Bohang Li et al. (2025) present a novel method for image steganalysis that combines active learning with off-policy deep-reinforcement learning (DRL), reducing the need for large labeled datasets. The model uses DRL to dynamically select which unlabeled images to request labels for optimizing sample selection and improving learning efficiency. They add hyperparameter tuning via a Differential Evolution (DE) algorithm to stabilize and optimize the DRL model's performance. On benchmark datasets (BossBase 1.01 and BOWS-2), their method achieves a high average F-measure is 93.15% and is 91.83%(approx), outperforming traditional steganalysis baselines. The work demonstrates that

intelligent data-selection and optimization can yield strong detection accuracy with minimal labeled data. This research underscores how combining active learning, DRL, and optimization can advance security-critical detection tasks under labeling constraints.

Madhurika and Naga Malleswari (2025) propose SentiNet, a hybrid deep-learning model combining parallel CNN-based feature extractors, bidirectional LSTM (BiLSTM), and a channel-wise attention fusion mechanism for sentiment classification of customer reviews. The text is first converted to dense word-embeddings, then parallel 1D-convolutions capture local n-gram features; the concatenated features are fed to BiLSTM to model sequence context, and attention emphasizes sentiment-relevant tokens. Evaluated on multiple benchmark datasets (IMDb, Twitter, Yelp), SentiNet achieves up to 98.7% accuracy with high precision/recall, outperforming baseline CNN, LSTM, and BiLSTM models. Ablation experiments show that removing attention, embeddings, or optimized dropout significantly degrades performance, indicating that every module contributes to SentiNet's robustness. The authors demonstrate that SentiNet balances performance, generalizability, and interpretability, making it suitable for real-world noisy, short-text sentiment analysis tasks. They also highlight its cross-domain robustness and potential for deployment in applications like e-commerce reviews and social-media analytics.

3. Methodology

This methodology begins with an analysis of the Multi Step Cyber Attack Dataset to understand its traffic features and attack patterns. The study then applies four hyperparameter tuning methods Grid Search, Random Search, Bayesian Optimisation, and Hyperband to optimise the selected models. Together, these steps provide a structured basis for evaluating detection performance.

3.1 Multi-Step Cyber-Attack Dataset Description

The Multi-Step Cyber-Attack Dataset (MSCAD) is a labelled network-traffic dataset designed for evaluating IDS models on realistic multi-stage attacks. It contains 77 engineered flow features, covers both normal and malicious traffic, and includes sequential attack chains such as port scanning → web crawling → brute force, as well as multi-phase DDoS behaviour. The data has no missing or duplicate records, is available in PCAP/CSV formats, and includes modern attack types like HTTP Slowloris, ICMP Flood, and password-cracking attempts. MSCAD's structure supports training IDS models that must recognise attack progression across multiple steps.
(https://www.kaggle.com/datasets/drjamailalsawwa/mscad)

Table.1. MSCAD Dataset Description

File Name	Description	Traffic Type
MSCAD.xlsx	Full labelled dataset with 77 features	Normal & Malicious
N-0	Normal network traffic	Normal
Scan-1	Port-scan traffic	Malicious (Full, SYN, FIN, UDP)
App-01	HTTP Slowloris DDoS	Malicious
App-02	ICMP Flood (Volume DDoS)	Malicious
W-B-01	Web Crawling	Malicious
W-B-02	Password Cracking (Brute Force)	Malicious

3.2 Hyperparameter Tuning

Effective hyperparameter optimisation is crucial for improving the performance and generalisation capacities of deep learning models. This study employed four common methodologies: GS, RS, BO, and Hyperband each with distinct characteristics for exploring the hyperparameter space.

- **Grid Search**

Grid Search formalizes hyperparameter tuning as exhaustive traversal of a discretized parameter manifold, where the optimisation landscape is treated as a finite combinatorial structure. The search space is constructed as the Cartesian product $G = H_1 \times H_2 \times \cdots \times H_k$, and each point in this space is evaluated independently using a validation metric. The computational demand is described by

$$\boldsymbol{G = \prod_{i=1}^{k} |H_i|}$$

where H_i denotes the domain of the i^{th} hyperparameter and $|H_i|$ the discretization granularity. If the objective function is J(θ), where θ∈ G, then Grid Search aims to compute $\theta^* = \arg min_{\theta \in G} J(\theta)$. This formulation explicitly treats the optimisation as a discrete minimisation problem over an enumerated set, making the evaluation deterministic and reproducible. The method essentially transforms continuous tuning spaces into structured grids, enabling systematic traversal while providing a direct mapping between parameter scaling and search complexity.[5]

- **Random Search**

Random Search models hyperparameter optimisation as stochastic sampling over a continuous or discrete parameter distribution. If the hyperparameter vector lies in the space θ, a trial configuration is drawn according to $\theta^{(t)} \sim p(\theta)$, where $p(\theta)$ may be uniform, log-uniform, or any parameterised prior reflecting expected importance. The optimisation objective can be expressed as estimating $\theta^* = \arg min_{\theta \in G} J(\theta)$, which implies that Random Search approximates the true optimum by Monte Carlo sampling. The probability of sampling a near-optimal configuration relates to the measure of its region in θ, which can be formalised as

$$\boldsymbol{P(\in) = \int_{\{\theta : J(\theta) \le J^* + \in\}} p(\theta) d\theta}$$

This expression quantifies the likelihood of obtaining a configuration within an ϵ -neighborhood of the global optimum. By treating optimisation as statistical exploration rather than deterministic enumeration, Random Search leverages the high-dimensional geometry of the search space, often revealing high-performance regions with far fewer evaluations.[6]

- **Bayesian Optimization**

Bayesian Optimization frames hyperparameter tuning as sequential probabilistic inference over an unknown objective function J(θ). A surrogate model, commonly a Gaussian Process, estimates the posterior distribution $p(J(\theta) \mid D_t)$, where D_t is the set of evaluated points at iteration t. The acquisition function guides exploration, with Expected Improvement defined as $EI(\theta) = E[\max(J^* - J(\theta), 0)]$, where J^* is the best observed value. For a Gaussian posterior with mean μ(θ) and variance $\sigma^2(\theta)$, this becomes

$$\boldsymbol{EI(\theta) = (J^* - \mu(\theta))\Phi(z) + \sigma(\theta)\phi(z)}$$

where $z = J^* - \mu(\theta)/\sigma(\theta)$, Φ is the Gaussian CDF, and ϕ the PDF. This formulation captures both exploitation (low mean) and exploration (high variance). The optimisation objective becomes $\theta_{t+1} = argmax_\theta \, EI(\theta)$, defining a principled decision-making rule under uncertainty. Bayesian Optimization

therefore builds a continuously refined probabilistic model of the hyperparameter landscape, using predictive uncertainty as a mathematical lever to guide sampling trajectories.[7]

- **Hyperband**

Hyperband conceptualises hyperparameter optimisation as a resource allocation problem governed by the multi-armed bandit framework. Configurations are sampled and assigned partial budgets, with underperforming configurations pruned using successive halving. If the total resource budget is B and the maximum per-configuration resource is R, the number of configurations evaluated in a bracket s is given by $n=\lfloor B/R \cdot s \rfloor$. The successive-halving step evaluates n configurations with resource r and retains only the top $\lfloor n/\eta \rfloor$, where $\eta>1$ is the reduction factor. This reduction is expressed as

$$n_{J+1} = \left\lfloor \frac{n_j}{\eta} \right\rfloor$$

and the corresponding resource allocation grows as $r_{j+1}=r_j\eta$. The optimisation objective becomes identifying $\theta^* = \arg min_{\theta \in S} J(\theta)$, where S is the subset that survives all halving rounds. Hyperband therefore formalises tuning as repeated partial evaluation and elimination cycles, using mathematical scheduling rules to concentrate computational expenditure on the most promising configurations. [8]

4. Results and Discussion

This section presents the outcomes of the four tuned models namely MLP, XGBoost, DNN, DBN and examines how Grid Search, Random Search, Bayesian Optimisation, and Hyperband influence their performance across key metrics. The discussion compares these results to identify which optimisation strategies provide the most consistent gains and explains their impact on detecting complex cyber attack behaviour.

4.1. MLP and XGBoost – After Hyperparameter Tuning

Hyperparameter tuning enhances the learning behavior of both MLP and XGBoost by optimizing key parameters that control model complexity and generalization, enabling each model to better capture structured patterns in MSCAD's network-flow features.

Table.2.MLP and XGBoost – After Hyperparameter Tuning

Metric	MLP				XG Boost			
	GS	RS	BO	Hyperband	GS	RS	BO	Hyperband
Accuracy	0.958	0.963	0.972	0.969	0.976	0.979	0.985	0.983
Precision	0.957	0.964	0.973	0.969	0.976	0.980	0.986	0.984
Recall	0.952	0.958	0.969	0.965	0.971	0.974	0.982	0.980
F1-Score	0.954	0.961	0.971	0.967	0.973	0.977	0.984	0.982

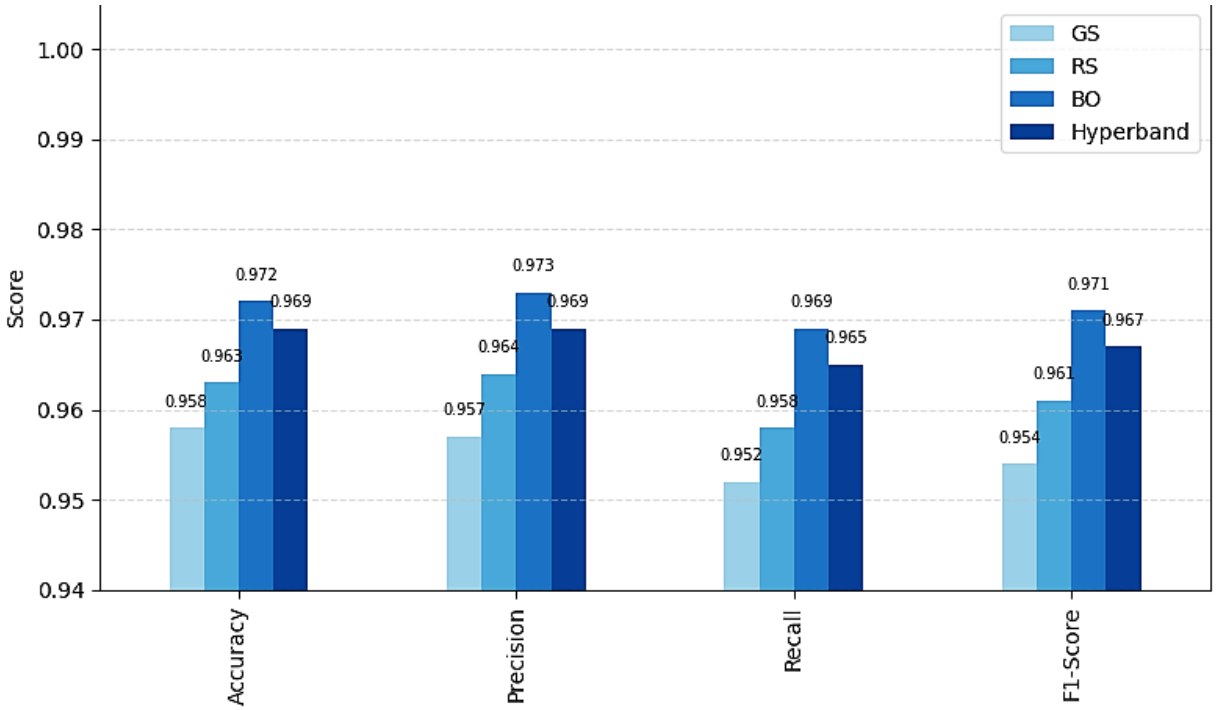

Fig.1.MLP after Hyperparameter tuning

The above figure shows consistent metric improvements across all methods, with BO achieving the highest overall performance. The bar chart highlights a clear upward pattern from GS to BO, indicating more effective optimization of learning rate and hidden-layer parameters. Hyperband also delivers strong gains, particularly in recall and F1-score, confirming that early-stopping–based tuning benefits MLP stability.

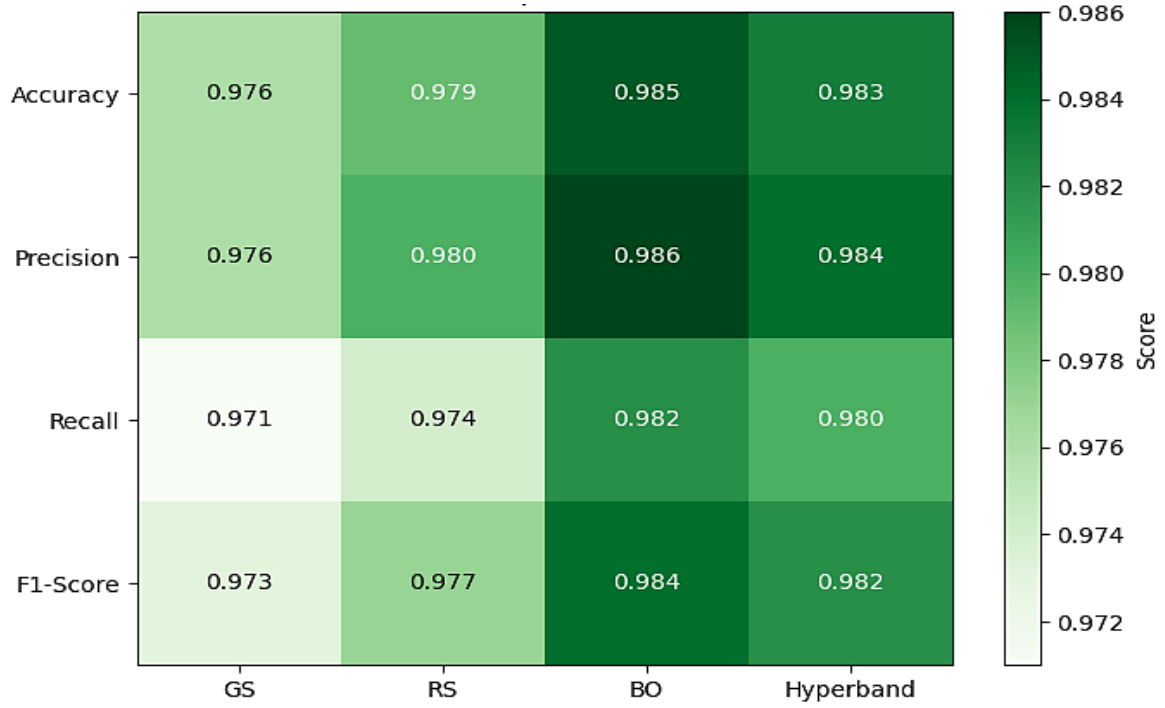

Fig.2.XGBoost after Hyperparameter tuning

This reveals uniformly high performance across all tuning strategies, with BO and Hyperband producing

the strongest concentration of high-value cells. The gradient intensifies across the table, demonstrating that optimized tree depth, learning rate, and regularization significantly enhance the model's discrimination across MSCAD's attack classes. BO achieves the highest cluster of peak values.

4.2. DNN and DBN – After Hyperparameter Tuning

Tuned DNN and DBN models gain improved hierarchical feature extraction through refined depth, learning rates, and regularization settings, strengthening their ability to represent complex traffic characteristics in the MSCAD dataset.

Table.3.DNN and DBN - after Hyperparameter Tuning

Metric	DNN				DBN			
	GS	RS	BO	Hyperband	GS	RS	BO	Hyperband
Accuracy	0.967	0.972	0.981	0.984	0.974	0.978	0.986	0.987
Precision	0.967	0.972	0.982	0.985	0.974	0.978	0.986	0.987
Recall	0.963	0.968	0.980	0.982	0.971	0.975	0.984	0.986
F1-Score	0.965	0.970	0.981	0.983	0.972	0.976	0.985	0.986

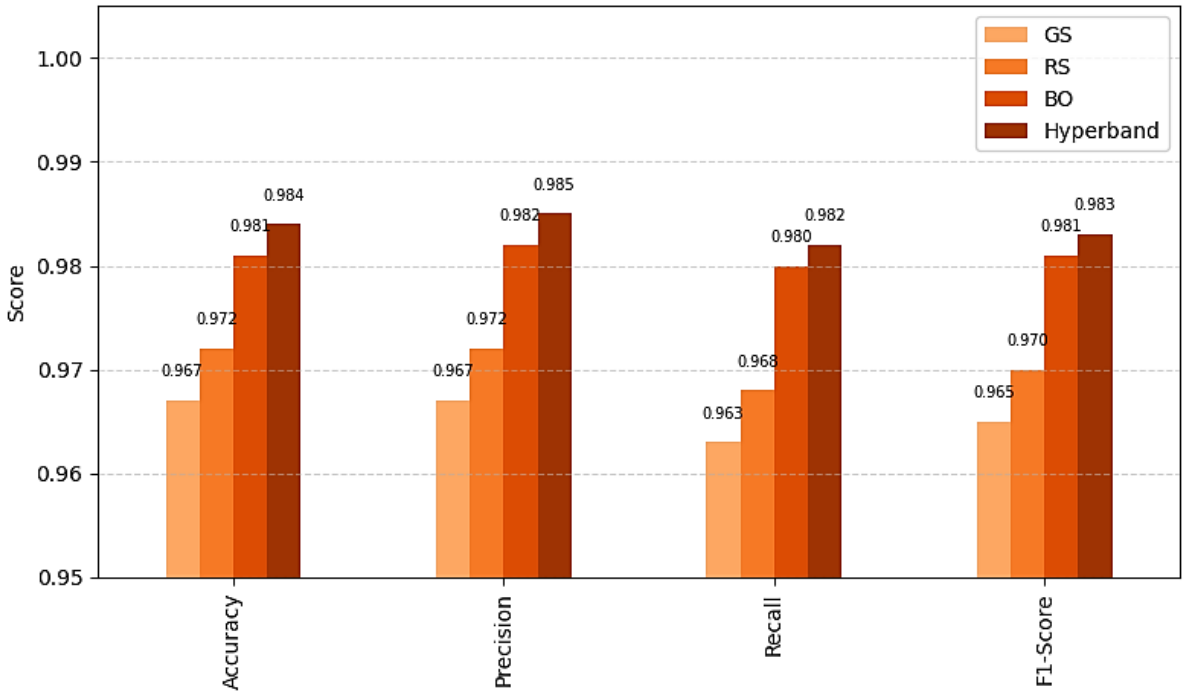

Fig.3. DNN after Hyperparameter Tuning

The above figure shows a clear performance escalation from GS to Hyperband and BO, reflecting the deep model's sensitivity to tuning depth, dropout, and batch size. All four metrics rise notably in the tuned configuration, with BO producing the most pronounced improvement. The results confirm that deeper architectures extract richer flow-pattern representations after tuning.

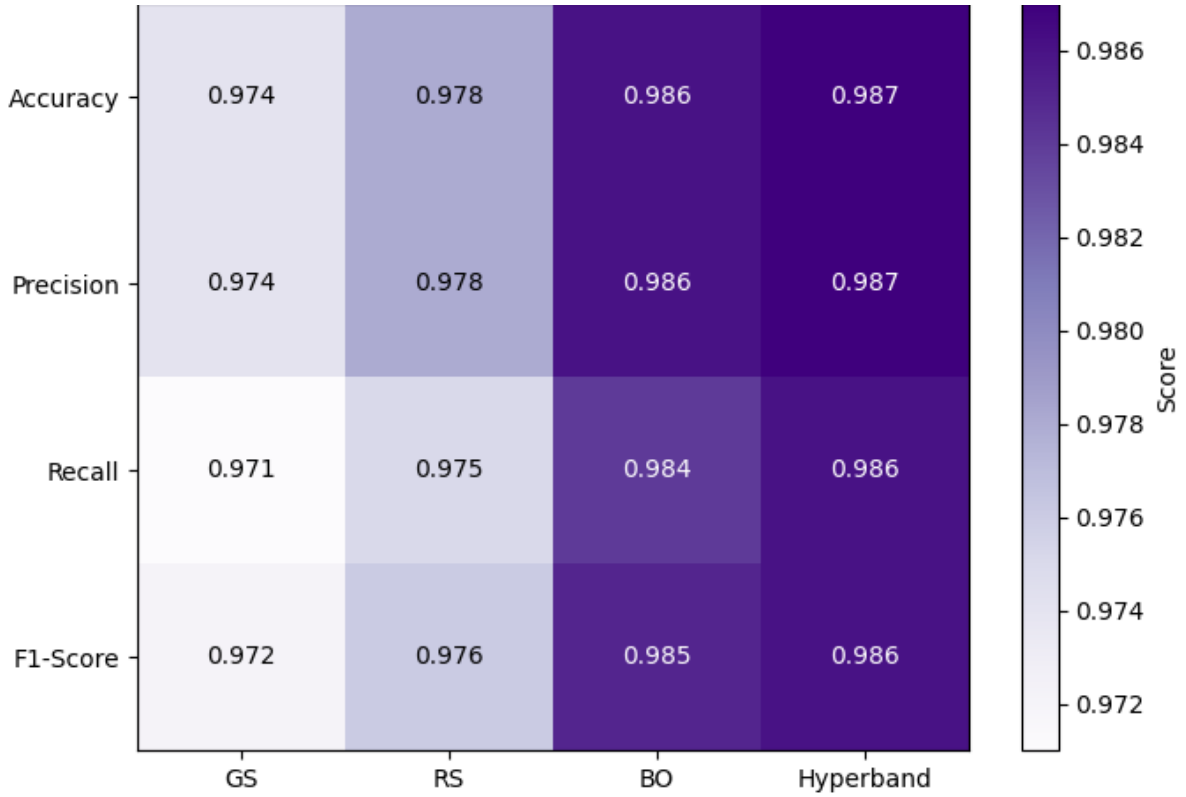

Fig.4. DBN after Hyperparameter Tuning

This displays high-intensity values across all tuning methods, with Hyperband and BO forming the brightest regions. This pattern indicates that the DBN benefits strongly from layer-wise pretraining and optimized learning rates. The consistently high values across precision, recall, and F1-score demonstrate improved abstraction of MSCAD's hierarchical traffic patterns after tuning.

5. Conclusion

This study demonstrates that effective model optimisation is essential for improving intrusion detection performance on the Multi Step Cyber Attack Dataset. The evaluation shows that Grid Search, Random Search, Bayesian Optimisation, and Hyperband each contribute to measurable gains, with the deeper models benefiting most from refined parameter settings. Overall, the findings highlight the importance of combining strong learning architectures with efficient tuning methods to enhance the detection of complex cyber attack behaviour in modern network environments.

References

1. Zou, D.; Ma, C.; Wang, P.; Geng, Y. Hyperparameter Optimization EM Algorithm via Bayesian Optimization and Relative Entropy. Entropy 2025, 27, 678.
2. C. M. Liyew, E. Di Nardo, S. Ferraris, and R. Meo, "Hyperparameter optimization of machine learning models for predicting actual evapotranspiration," Machine Learning with Applications, vol. 20, 100661, 2025.
3. Bohang, L., Li, N., Yang, J. et al. Image steganalysis using active learning and hyperparameter optimization. Sci Rep 15, 7340 (2025).
4. Madhurika, B., Malleswari, D.N. Deep learning based SentiNet architecture with hyperparameter optimization for sentiment analysis of customer reviews. Sci Rep 15, 35525 (2025).

5. M. Ogunsanya, J. Isichei, and S. Desai, "Grid search hyperparameter tuning in additive manufacturing processes," Manufacturing Letters, vol. 35, pp. 1031–1042, 2023.
6. Ali, Y.A.; Awwad, E.M.; Al-Razgan, M.; Maarouf, A. Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity. Processes 2023, 11, 349.
7. Li, Q., Kamaruddin, N., Zhang, J. et al. A novel method of bayesian genetic optimization on automated hyperparameter tuning. Sci Rep 15, 43181 (2025).
8. A. R. Manga, M. A. F. Latief, A. W. M. Gaffar, H. Azis, R. Satra and Y. Salim, "Hyperparameter Tuning of Identity Block Uses an Imbalance Dataset with Hyperband Method," 2024 18th International Conference on Ubiquitous Information Management and Communication(IMCOM), Kuala Lumpur, Malaysia, 2024.