CHAPTER V
CLOSING

5.1 Conclusion

Based on the comprehensive analytical results, empirical evaluations, and detailed discussions presented in Chapter IV, the following primary conclusions can be drawn to directly address each problem formulation and research objective:

1. Implementation of the XGBoost Algorithm Enhanced with a Cost-Sensitive Learning Approach to Handle Class Imbalance on the NF-UNSW-NB15 Dataset. An intrusion detection model based on the XGBoost architecture was successfully constructed and demonstrated high robustness in mitigating the extremely severe class imbalance within the NF-UNSW-NB15-v3 dataset, where the Normal class overwhelmingly dominated 94.60% of the training data, with the highest imbalance ratio reaching 122.28:1 against the Probe class. This mitigation was achieved through a Cost-Sensitive Learning mechanism utilizing a hybrid class weighting strategy, which mathematically combines standard balanced class weights with a square root transformation ($w_i^{\text{hybrid}} = \sqrt{w_i^{\text{balanced}}} / \text{mean}\left(\sqrt{w^{\text{balanced}}}\right)$). This hybrid weighting scheme successfully restructured the model's weighted attention space, dampening the Normal class weight to 0.76× while significantly amplifying the minority classes (DoS by 7.10×, Probe by 8.40×, and Malware by 3.93×). The operational effectiveness of this approach was substantiated by a substantial increase in the detection sensitivity (recall) of DoS attacks on unseen testing data, rising from 64.71% (default baseline configuration) to 76.20% (optimized TPE model), alongside an increase in the Malware class recall from 91.18% to 92.72%. Furthermore, the resulting Cohen's Kappa reliability index of 0.9291 (categorized under the Almost Perfect Agreement qualification) confirms that the constructed system possesses exceptionally high classification reliability rather than being an artifact of statistical coincidence.

2. Comparative Performance of Multi-Objective Optimization Among NSGA-II, TPE, and the Default XGBoost Baseline. Comparative evaluations of the three methods demonstrate that the NSGA-II and TPE algorithms consistently outperform the Baseline XGBoost model (configured with default parameters) in navigating the hyperparameter search space to identify the optimal trade-off between the Macro F_1-Score and inference execution time. On the unseen testing data involving 473,085 samples, the TPE model achieved a Macro F_1-Score of 0.8632 (representing a +3.84% relative performance gain over the default baseline score of 0.8313) with an inference latency of 2.1681 µs/sample. In comparison, the NSGA-II model achieved a Macro F_1-Score of 0.8615 (a +3.63% relative gain) with an inference latency of 4.7720 µs/sample. Formal non-parametric Kruskal-Wallis statistical testing confirmed that these performance discrepancies were highly significant, yielding a Macro F_1-Score p-value of 0.0090 and a latency p-value of 0.0019 (both falling well below the significance threshold of $\alpha = 0.05$). Based on Stratified 5-Fold Cross-Validation, the TPE and NSGA-II frameworks proved to be equivalent in terms of classification quality (yielding a marginal and statistically non-significant numerical delta of 0.8424 vs. 0.8427 in mean Macro F_1-Scores); however, TPE exhibited a clear and definitive superiority in computational speed, executing its inference phase 47.5% faster than NSGA-II (0.1492 s versus 0.2840 s). In terms of optimization efficiency, TPE completed the search space traversal 25.4% faster than NSGA-II (24.30 minutes versus 32.59 minutes). Consequently, TPE is recommended as the optimization method providing the most balanced operational trade-off for high-speed NIDS deployment scenarios.
3. Pareto Front Characteristics and Optimal Hyperparameter Configurations for NIDS Scenarios. The multi-objective optimization process generated Pareto Front curves with distinct geometric and structural characteristics for each metaheuristic: TPE produced 5 Pareto solutions forming a compact efficiency frontier (bounded within a tight latency range of 1.41–2.15 µs/sample), while NSGA-II yielded 7 Pareto solutions with a more spread frontier (latency range

of 1.47–4.81 µs/sample), indicating a broader exploration of the underlying solution space. All Pareto solutions from both automated frameworks strictly dominated the baseline configuration across the Macro F_1-Score dimension, although the baseline maintained its advantage regarding absolute raw inference velocity (0.67 µs/sample). The specific hyperparameter configuration delivering the superior non-dominated solution was identified in Trial #10 of the TPE study, comprising the following parameters: n_estimators=800, learning_rate=0.0500, max_depth=8, min_child_weight=5, gamma=0.3164, subsample=0.9359, colsample_bytree=0.6301, reg_alpha=0.4806, and reg_lambda=0.0269. Sensitivity analysis executed via a Random Forest surrogate model revealed that learning_rate was the most dominant hyperparameter influencing the Macro F_1-Score (importance scores of 0.3261 for TPE and 0.7310 for NSGA-II), whereas n_estimators acted as the primary driver of inference latency overhead (importance scores of 0.6164 for TPE and 0.6445 for NSGA-II). These empirical insights provide a practical architectural guideline demonstrating that for NIDS scenarios requiring high detection accuracy alongside rapid response times, a moderate configuration consisting of a low learning rate (0.027–0.065), intermediate tree depth (6–8), and an ensemble size within the 500–800 range serves as the optimal sweet spot.

4. Design and Implementation of a Trace-Driven Intrusion Detection Prototype Simulation. A trace-driven intrusion detection simulation prototype was successfully designed and implemented utilizing the Python Streamlit framework to serve as an interactive security monitoring dashboard. The data processing architecture is structured modularly with three primary functional components:
    a. Data Streamer: Sequentially reads historical NetFlow records row by row to simulate a pseudo-real-time traffic flow.
    b. Inference Engine: Dynamically loads the optimized XGBoost model (JSON format) alongside its corresponding StandardScaler artifact to execute stateless predictions on each incoming data flow.

c. Live Dashboard: Visualizes real-time detection outcomes via a dynamic security status indicator (green for Normal conditions, red for attack indications), an inference latency gauge in milliseconds, a chronological detection log table, and historical threat trend charts.

The simulation was rigorously evaluated under two testing scenarios: Scenario 1 (100% Normal traffic) to measure the structural baseline latency and validate the complete absence of false positives, and Scenario 2 (mixed traffic with injected DoS, Probe, and Malware attacks) to evaluate status transition responsiveness and latency stability under varying attack loads. The developed prototype successfully visualized the operational mechanics of the XGBoost model in detecting threats in a pseudo-real-time manner and displayed inference latency metrics as a robust proof of concept (PoC) for system efficiency, where the TPE model's inference latency of 2.1681 µs/sample mathematically translates into a sequential processing capacity of approximately 461,200 NetFlow flows per second.

5.2 Future Recommendations

For future research development in the field of Network Intrusion Detection Systems (NIDS), the following improvements and avenues of exploration are recommended:

1. Utilization of a Higher Number of Cross-Validation Folds and Formal Post-Hoc Testing: Future studies should implement a cross-validation scheme with an increased number of folds (such as $N = 10$ or $N = 20$) to expand the evaluation sample size. This modification is critical to enhancing statistical power, thereby enabling valid pairwise post-hoc tests (such as Dunn's test with Bonferroni or Holm significance corrections) to be executed. This will allow researchers to establish performance discrepancies among different optimization frameworks with greater precision.
2. Exploration of Alternative Tabular Classification Algorithms: Beyond XGBoost, the multi-objective optimization frameworks (TPE and NSGA-II) and the hybrid class weighting scheme should be integrated with other modern

tabular classification algorithms, such as LightGBM, CatBoost, or specialized tabular deep learning architectures like TabNet. Conducting cross-algorithm comparative analyses may uncover configurations capable of achieving even lower inference latencies (below 1 μs) while sustaining or improving overall detection quality.

3. Dynamic Evaluation in Active Network Environments (Real-Time Deployment): It is recommended to dynamically deploy the optimal TPE model derived from this study into an active network testbed. Feasible environments include Software-Defined Networking (SDN) architectures utilizing Ryu or ONOS controllers, or Network Functions Virtualization (NFV) infrastructures. This dynamic evaluation is essential to measure real-world packet throughput performance and assess model stability under instantly fluctuating network traffic loads.
4. Development of Continuous Temporal Feature Engineering: To detect stealthy or low-frequency cyber threats more sensitively such as slow-rate DDoS or Advanced Persistent Threats (APTs) future work should extend the feature engineering pipeline by incorporating continuous temporal relational features. For example, packet inter-arrival times calculated via a sliding window framework would capture behavioral correlations across consecutive network connections more effectively.

This page is intentionally left blank