A SEQUENTIAL MODEL SELECTION AND OPTIMIZATION FRAMEWORK FOR SECURITY IN ZERO-TOUCH NETWORK ENVIRONMENTS

Alharith Tawfig Elghasri[1] and *Abdulbaset Mustafa Hamed*[2]

[1]The Libyan International Telecom Company, Tripoli, Libya
[2]University of Tripoli, Tripoli, Libya

ABSTRACT

Zero-touch network and Service Management (ZSM) introduces a principle for next generation 5G and 6G networks, aiming to enable fully-automated, self-healing, and self-securing infrastructures with minimal human intervention. As networks evolve toward these autonomous structures, traditional static security mechanisms may become insufficient due to the scale and variability of network traffic. This paper proposes an adaptive Intrusion Detection System (IDS) framework designed for Zero-touch networks (ZTN). We present the Sequential Model Selection and Optimization (SMSO) framework, which is designed to integrate online learning with Automated Machine Learning (AutoML) principles. The system continuously screens a diverse pool of candidate learners, including ensembles, trees, and lazy learners, while dynamically optimizing hyperparameters using the Tree structured Parzen Estimator (TPE). We evaluate the mechanism using subsets of the CIC-IDS-2017 and 5G-NIDD datasets. The results indicate that SMSO achieves near-instant convergence and maintains sub-5ms latency, aligning with the real-time requirements of 3GPP-compliant ZTN architectures. Furthermore, we discuss the integration of the framework with the Network Data Analytics Function (NWDAF) to support standardized autonomous security operations in 5G core environments.

Keywords: 6G Security, Automated Machine Learning, Intrusion Detection, Online Learning, zero-touch networks.

1. INTRODUCTION

The transition to 5G and 6G networks drives a shift toward fully-autonomous infrastructures. To address the resulting complexity, ETSI introduced the Zero-touch Network and Service Management (ZSM) framework [1], which leverages Machine Learning (ML) to minimize human intervention. However, operators face challenges managing these architectures with traditional, static management techniques [3]. While the ZSM vision targets "Self-X" operations and automated orchestration to ensure Quality of Experience (QoE) [1], practical ML deployment remains difficult due to tasks like feature engineering and hyperparameter tuning. Automated ML (AutoML) offers a solution by automating these pipelines, thereby reducing manual effort [2]. In the security domain, where manual configuration is untenable, AutoML facilitates real-time model selection and updates [4]. Consequently, this paper proposes the Sequential Model Selection and Optimization (SMSO) framework. By integrating online learning with Bayesian optimization on continuous data streams, SMSO enables dynamic adaptation to evolving attack patterns, overcoming the limitations of static batch learning.

Our contributions can be summarized as follows:

1. **ZTN compliant architecture:** An adaptive security framework that aims to reduce the need for manual retraining, aligning with Zero-touch principles of automation and self-management [4].
2. **Real-time screening:** A dynamic candidate screening process that identifies suitable algorithms for the current network state from a diverse pool of learners.
3. **Bayesian optimization:** A Bayesian optimization engine (TPE) that tunes hyperparameters to support autonomous adaptation.
4. **Weighted ensemble voting:** A weighted ensemble mechanism that combines predictions from the top optimized models to reduce variance and improve robustness.
5. **Empirical validation:** An evaluation using a subset of the CIC IDS 2017 dataset, indicating effectiveness in differentiating benign traffic from malicious activities.

2. BACKGROUND AND RELATED WORK

2.1 Zero-touch Network and Service Management

The ZSM framework represents a paradigm shift toward fully-automated network management. It leverages AI to enhance operations, decision-making, and resource allocation. The core principle of ZSM is the closed-loop control mechanism, which continuously monitors network status, analyzes data, decides on actions, and executes them without human intervention. This closed loop is essential for maintaining Service Level Agreements (SLAs) in 6G slices where manual intervention would likely introduce unacceptable latency [4]. In the security domain, ZSM requires self-protecting capabilities. This implies that the network should automatically detect threats, isolate

compromised segments, and update its defense posture [2]. The traditional Security Operation Center workflow may be too slow for the latency requirements of 6G networks. An effective ZTN security module should be integrated directly into the management plane [4].

Furthermore, to ensure 3GPP compliance, the SMSO framework must integrate with the Network Data Analytics Function (NWDAF). This integration allows the framework to utilize standardized 5G core telemetry to effectively execute "Self-X" autonomous operations.

2.2 Challenges in ZTN security

Securing Zero-touch Networks presents unique challenges [4]:

- **Scale and heterogeneity:** ZTNs are expected to connect billions of diverse devices. A single static security model may not protect such a mixed landscape.
- **Dynamic topologies:** In 6G, network slices may be created and destroyed on demand. Security policies must be flexible and scalable.
- **Data privacy:** With intelligence distributed at the edge, transmitting raw traffic to a central cloud raises privacy concerns and consumes bandwidth.

Our SMSO framework addresses heterogeneity by maintaining a diverse learner pool and addresses dynamic behavior by using online learning that adapts to new flows incrementally.

2.3 Adaptive learning and flexibility

Conventional machine learning approaches in security typically employ a static learning paradigm. This rigidity makes them susceptible to changes in the environment, where traffic distributions evolve over time. Such shifts may happen suddenly, such as a new attack appearing, or gradually, such as users changing their habits [5]. In this work, we utilize the natural variations found in the CIC-IDS-2017 and 5G-NIDD datasets. Because these datasets capture traffic over multiple days and various 5G-specific attack profiles, they provide a realistic environment to assess whether the model can adapt. Because the dataset captures traffic over multiple days with different attack profiles, it provides a realistic environment to assess whether the model can adapt. While we rely on natural changes rather than simulated stress tests in this study, this offers an initial indication of system behavior under evolving traffic. Online learning algorithms, such as Hoeffding Trees, update the model incrementally. However, they often rely on static settings. If the environment changes significantly, these fixed settings may become suboptimal, which motivates dynamic optimization [6].

2.4 AutoML in network security

AutoML aims to automate the end-to-end process of applying machine learning. Recent work has explored AutoML for static IDS, for example Auto WEKA [7]. However, most existing solutions are designed for offline batch processing. Prior studies [2] indicate that ZSM still faces ML challenges such as the need for skilled personnel and complex model selection. They argue that AutoML can reduce manual intervention by automating selection and tuning. Our SMSO framework operationalizes this direction by creating an online streaming-compatible AutoML pipeline that aligns with the automated model updating phase described in the literature [2].

3. PROPOSED METHODOLOGY

The proposed SMSO framework is designed to operate within the intelligence plane of a ZTN architecture. It consists of three sequential stages: (A) Online Candidate Screening, (B) Bayesian Hyper-parameter Optimization, and (C) Weighted Ensemble Aggregation. The logic is summarized in Algorithm 1.

Algorithm 1 SMSO Framework Execution Logic

Input: Stream $\mathcal{S}$, Pool $\mathcal{M}$, Search Space Λ
Output: Ensemble prediction $\hat{y}_{ens}$
Initialize running F1 trackers for all $h \in \mathcal{M}$
$\mathcal{M} \leftarrow \{HT, LB, KNN, ARF, SRP\}$
for each (x_t, y_t) in screening window **do**
  for each $h_k \in \mathcal{M}$ **do**
    Predict $\hat{y}_{t,k} = h_k(x_t)$
    Update h_k with (x_t, y_t)
    Update running $F1_k$
  end for
end for
Select top three models: $\mathcal{M}_{top} \leftarrow$ Top3$(\mathcal{M})$
for each $h \in \mathcal{M}_{top}$ **do**
  $\lambda^* \leftarrow$ TPE_Optimize$(h, \Lambda, \mathcal{S}_{val})$
  Update h hyper-parameters using λ^*
end for
Compute weights w_k from validation F1 scores
while stream is active **do**
  Receive new sample x_{new}
  for each class $c \in \{0, 1\}$ **do**
    $score(c) \leftarrow \sum_{k=1}^{3} w_k \cdot \mathbb{I}\big(h_k(x_{new}) = c\big)$
  end for
  $\hat{y}_{ens} \leftarrow \text{argmax}_{c \in \{0,1\}}\, score(c)$
end while

3.1 Stage A: Online candidate screening

Let $\mathcal{S} = \{(x_t, y_t) \mid t = 1, 2, \dots\}$ denote a potentially unbounded data stream where $x_t \in \mathbb{R}^d$ is the feature vector and $y_t \in \{0, 1\}$ is the class label (benign or attack) at time t.

We initialize a pool of candidate learners $\mathcal{M} = \{h_1, h_2, \dots, h_K\}$. In our implementation:

$$\mathcal{M} = \{\text{HT, LB, KNN, ARF, SRP}\} \tag{1}$$

These candidates are continuously evaluated using a test-then-train procedure:

1. **Prediction:** The model predicts the class of the incoming sample $\hat{y}_{t,k} = h_{k,t-1}(x_t)$.
2. **Update:** The model parameters θ_k are updated using the true label y_t:

$$\theta_{k,t} = \text{Learn}(\theta_{k,t-1}, x_t, y_t) \tag{2}$$

3. **Metric tracking:** We maintain a running F1 score $F1_k^{(t)}$ for each model.

After processing an initial screening window, we rank the models and select the top subset $\mathcal{M}_{top} \subset \mathcal{M}$.

3.2 Candidate Learner Architectures

To support robustness, the screening pool $\mathcal{M}$ consists of algorithms with different learning strategies.

3.2.1 Hoeffding Tree (HT)

HT builds the tree incrementally, relying on the Hoeffding Bound:

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}} \tag{3}$$

This allows the HT to construct a model asymptotically identical to a batch learner with constant time per sample [8].

3.2.2 Leveraging Bagging (LB)

LB improves upon standard OzaBagging by increasing diversity. It trains each base learner with weights drawn from a Poisson distribution ($\lambda = 6$):

$$w \sim \text{Poisson}(\lambda = 6) \tag{4}$$

LB employs a randomization filter to discard poor performers [9].

3.2.3 k Nearest Neighbors (KNN)

Streaming KNN maintains a fixed size sliding window W. For a new instance x_{new}, it calculates the Euclidean distance:

$$d(x_{new}, x_i) = \sqrt{\sum_{j=1}^{d} (x_{new}^{(j)} - x_i^{(j)})^2} \tag{5}$$

The sliding window ensures decisions are based on the most current network state [10].

3.2.4 Adaptive Random Forest (ARF)

ARF uses online bagging and incorporates the ADWIN drift detection operator for each tree. If a tree performance degrades, it is reset. A common form of the ADWIN cut threshold is:

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} \ln \frac{4}{\delta'}} \tag{6}$$

where m is related to the effective window size (often expressed via a harmonic mean term) and δ' is the confidence value. This allows ARF to evolve in response to changes in data [11].

3.2.5 Streaming Random Patches (SRP)

SRP combines random subspaces with bagging. For each base learner h_j in the ensemble, it trains on a random subset of features $x'_t \in \mathbb{R}^{d'}$ where $d' < d$. The final prediction is a majority vote from the ensemble $\mathcal{E}$:

$$\hat{y}_{SRP} = \underset{c \in \{0,1\}}{\text{argmax}} \sum_{h_j \in \mathcal{E}} \mathbb{I}\big(h_j(x'_t) = c\big) \tag{7}$$

This diversity makes SRP resilient when certain features become noisy or less relevant [12].

3.3 Stage B: Bayesian Optimization

We employ the Tree structured Parzen Estimator (TPE). For each model $h \in \mathcal{M}_{top}$, we define a hyperparameter space Λ. The goal is to find the optimal configuration λ^* that minimizes the loss function $\mathcal{L}$ (negative F1 score) over a validation stream $\mathcal{S}_{val}$:

$$\lambda^* = \underset{\lambda \in \Lambda}{\text{argmin}}\, \mathcal{L}(h_\lambda, \mathcal{S}_{val}) \tag{8}$$

TPE samples hyperparameters that maximize the likelihood ratio of good to poor performance [13].

3.4 Stage C: Weighted ensemble

The final prediction is generated by a weighted majority vote of the three optimized models. We assign a weight w_k to each model proportional to its validation performance:

$$w_k = \frac{F1_k}{\sum_{j=1}^{3} F1_j} \tag{9}$$

The ensemble prediction $\hat{y}_{ens}$ is:

$$\hat{y}_{ens} = \underset{c \in \{0,1\}}{\text{argmax}} \sum_{k=1}^{3} w_k \cdot \mathbb{I}\big(h_k(x_{new}) = c\big) \tag{10}$$

3.5 Explainability of the proposed algorithm

A key requirement for autonomous systems in ZTN is explainability so that decisions made by AI agents remain transparent and verifiable by network operators. The SMSO framework supports this through its weighting mechanism. The contribution of each model in the ensemble is explicitly quantified by its weight w_k. If the system flags a packet as malicious, an operator can inspect the weights to understand which learner drove the decision. This provides a practical interpretation layer without relying on opaque deep learning models. In addition, the modular nature of the candidate pool helps isolate performance drops to specific algorithmic limitations, supporting targeted refinement without disrupting the entire system.

4. DATA AVAILABILITY AND FEATURE ENGINEERING

4.1 Dataset description and subset utilization

We selected the CIC-IDS-2017 dataset as our primary benchmark because of its realistic variety of modern attack behaviors. In addition, we introduce the **5G-NIDD dataset** alongside the original CIC-IDS-2017 to provide a comprehensive evaluation.

For the experimental evaluation, we utilized an explicitly defined 12 000-instance sequential subset for 5G-NIDD set and 28 303 for CIC-IDS-2017. This specific volume was selected for two distinct reasons:

1. To enable a 1-to-1 benchmark against the reference paper that utilized the same specific subset.

2. To accommodate hardware resource constraints (specifically, laptop performance limits) during the computationally intensive Bayesian optimization (TPE) phase.

By using a sequential segment of the original traffic, the data naturally retains its chronological order. This approach provides a suitable environment for simulating online learning scenarios, where the system processes events in the order they occurred, without requiring extensive manual re-balancing.

4.2 Attack profiles

DoS and DDoS attacks are typically identifiable by high packet frequency and low variance in packet size. Port scans often manifest as a single source connecting to multiple destination ports with high SYN flag counts. Botnet traffic is often distinguished by regularities in inter-arrival time.

4.3 Preprocessing

We adopted a pipeline approach inspired by AutoML principles [4]:

1. **Cleaning:** Rows containing infinity or NaN values were removed.
2. **Outliers:** Extreme values in continuous features were capped at the 95th percentile.
3. **Encoding:** Label encoding mapped attack classes to binary format (0 for benign, 1 for malicious).
4. **Normalization:** Min-max normalization scaled numerical features to $[0, 1]$.
5. **Balancing:** SMOTE was employed to generate synthetic samples for the minority class.
6. **Feature selection:** RFECV reduced the initial set of 20 features to 13, retaining the most significant predictors.

It is important to state that the **exact same preprocessing pipeline** (cleaning, scaling, and encoding) was used for both the CIC-IDS-2017 and the 5G-NIDD datasets to guarantee scientific fairness in the evaluations.

5. EXPERIMENTAL RESULTS

5.1 Ensemble performance (CIC-IDS-2017)

The final weighted ensemble shows stable behavior compared to single models. The voting mechanism smooths variance, helping the system remain reliable when individual learners temporarily degrade.

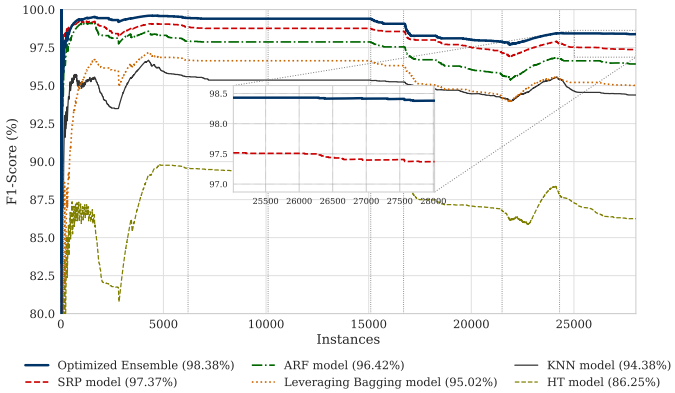

Figure 1: Average F1 score of the SMSO ensemble over the CIC-IDS-2017 data stream.

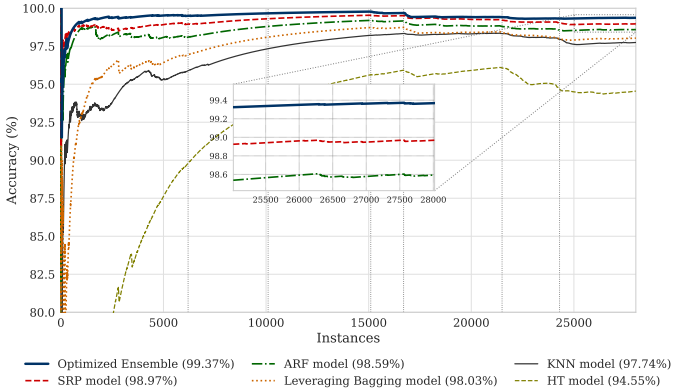

Figure 2: Average accuracy of the SMSO ensemble over the CIC-IDS-2017 data stream.

Figures 1 and 2 illustrate the performance of the SMSO framework. The framework converged quickly and stabilized within the first 1 000 instances. When a natural drift event appears around the 17 500 instance mark, the ensemble remains comparatively stable and recovers faster than individual learners, supporting the use of ensemble architectures for evolving traffic.

Table 1 highlights the performance comparison across all candidate models and the final ensemble. The optimized ensemble achieved an impressive 99.50% accuracy and 98.72% F1 score, outperforming all individual baseline algorithms.

Table 1: Performance comparison of individual learners and the SMSO ensemble (CIC-IDS-2017)

Model	Accuracy (%)	F1-Score (%)
Hoeffding Tree	94.55	86.26
k-Nearest Neighbors	97.61	94.00
Leveraging Bagging	98.03	95.02
Adaptive Random Forest	98.58	96.37
Streaming Random Patches	98.97	97.37
SMSO Ensemble	**99.37**	**98.38**

To assess the operational viability of the proposed framework, we measured the average per-packet inference latency (Table 2).

Table 2: Latency and operational overhead (CIC-IDS-2017)

Model	Avg Latency (ms)
Hoeffding Tree	0.09
k-Nearest Neighbors	3.27
Leveraging Bagging	0.30
Adaptive Random Forest	0.26
Streaming Random Patches	0.32
SMSO Ensemble	**1.47**

In evaluating the framework, total cumulative runtime was disregarded as a primary performance metric. Unlike traditional batch learning, where total training time is a fixed constraint on a static dataset, the proposed SMSO framework operates within a continuous **online learning** paradigm. In such environments, "total runtime" is merely a function of the data stream's duration and does not accurately reflect the system's efficiency or its ability to handle live traffic. The critical performance bottleneck for real-time 5G/6G security is instead the per-packet inference latency. As shown in Table 2, the SMSO ensemble achieves an average latency of 1.47 ms. This ensures the framework satisfies the rigorous 3GPP sub-5ms requirement for Ultra-Reliable Low-Latency Communications (URLLC) [14], confirming that the complexity of sequential optimization does not compromise the real-time responsiveness of the network.

5.2 Telecom-Specific validation and overhead (5G-NIDD)

To ensure our framework generalizes to next-generation architectures, the same pipeline was evaluated on the 5G-NIDD dataset.

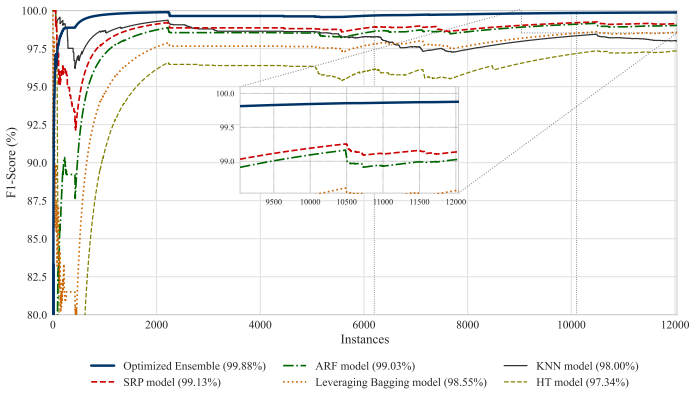

Figure 3: Average F1 score of the SMSO ensemble over the 5G-NIDD data stream.

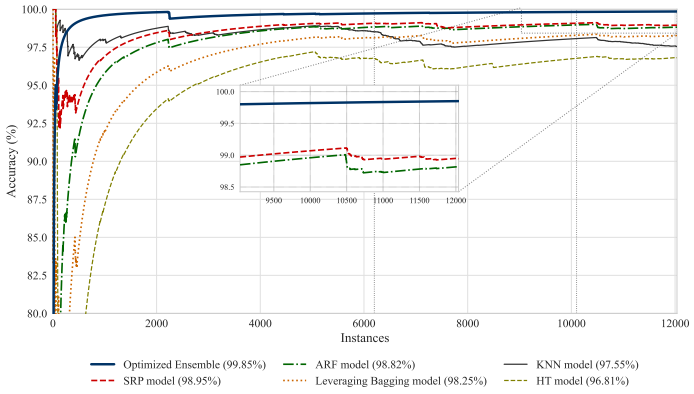

Figure 4: Average accuracy of the SMSO ensemble over the 5G-NIDD data stream.

As shown in Table 3, the SMSO ensemble scaled exceptionally well to 5G-specific telemetry, achieving a near-perfect 99.85% accuracy and 99.88% F1-Score.

Table 3: Performance comparison of individual learners and the SMSO ensemble (5G-NIDD)

Model	Accuracy (%)	F1-Score (%)
Hoeffding Tree	96.81	97.34
k-Nearest Neighbors	97.55	98.00
Leveraging Bagging	98.25	98.55
Adaptive Random Forest	98.82	99.03
Streaming Random Patches	98.95	99.13
SMSO Ensemble	**99.85**	**99.88**

Evaluation on the 5G-NIDD dataset further confirms the framework's real-time viability under 5G-specific traffic patterns (Table 4).

Table 4: Latency and operational overhead (5G-NIDD)

Model	Avg Latency (ms)
Hoeffding Tree	0.09
k-Nearest Neighbors	3.62
Leveraging Bagging	0.33
Adaptive Random Forest	0.22
Streaming Random Patches	0.33
SMSO Ensemble	**2.02**

Consistent with the CIC-IDS-2017 results, cumulative computational runtime is rejected as a valid performance metric for this environment. In traditional batch learning, runtime is a critical constraint for offline processing; however, in a 5G core environment relying on an **online learning** paradigm, the system processes an essentially infinite, non-stationary data stream. In this context, total runtime is a function of the stream's duration rather than the model's efficiency.

The definitive measure of operational worth for integrations like the NWDAF is per-packet inference latency. The SMSO ensemble maintains an average latency of 2.02 ms on 5G telemetry, which is well below the 3GPP-specified 5ms threshold [14]. This successfully demonstrates that the framework natively meets the rigorous requirements for real-time, critical 5G operations without creating processing bottlenecks or disrupting traffic flow.

6. RELEVANCE TO STANDARDIZATION

This work aligns with the objectives of **ITU T Study Group 13 (SG13)** for the 2025 to 2028 study period, particularly in future networks and emerging network technologies where AI and ML-enabled functions support automation [15].

6.1 Alignment with Q20/13 (IMT 2020 and AI ML)

The main contribution relates to **Question 20/13** on IMT networks and AI and ML requirements and architecture [15]. The proposed SMSO framework can be interpreted as a practical realization of an ML-enabled closed-loop workflow consistent with architectural principles described in **Recommendation ITU T Y.3172** [16]. In particular:

- **ML pipeline realization:** The online screening and ensemble stages provide an example of maintaining multiple candidate models within an operational ML pipeline, including selection and decision-formation functions.
- **Closed-loop adaptation:** The Bayesian optimization stage supports adaptive reconfiguration of hyperparameters based on observed performance over the traffic stream, consistent with the intent of ML orchestration for future networks.

6.2 Potential contribution to related SG13 work items

This study may serve as a technical input to SG13 efforts on autonomous networking, particularly in the following areas:

1. **Y.3063 (ex Y.IMT2020 MEVE req frame):** The interleaved test-then-train evaluation procedure provides a reproducible way to track effectiveness over time, including stability and recovery behavior under traffic changes, which can complement effectiveness measurement discussions for autonomous security functions [17].
2. **Y.3064 (ex Y.KM AN):** The learner pool and its artifacts, including model configurations, weights, and performance traces, can be viewed as a lightweight knowledge repository that supports retention and reuse over time, relevant to knowledge management considerations in autonomous networks [18].

6.3 Relevance to Q16/13 (Trustworthy Networking)

The explainability aspect of SMSO is also relevant to **Question 16/13** on trustworthy networking [15]. By using an interpretable weighting mechanism in the ensemble, SMSO supports auditability and operational transparency considerations consistent with trustworthy networking frameworks such as **Recommendation ITU T Y.3053** [19].

7. CONCLUSION

This paper presented *SMSO-IDS*, a sequential model selection and optimization framework designed to automate security in Zero-touch Network environments. By combining online candidate screening (test-then-train), Bayesian hyperparameter optimization (TPE), and a weighted ensemble, SMSO supports closed-loop security operations with significantly reduced manual tuning. Evaluation on a subset of the CIC-IDS-2017 dataset demonstrates that the framework converges quickly and achieves strong streaming performance, where the screening stage consistently identified ensemble-based learners as top candidates. The final optimized weighted ensemble reached 99.37% accuracy and a 98.38% F1-score, outperforming both the best single online learner (SRP) and the non-optimized ensemble. Ultimately, by autonomously securing 6G infrastructure, SMSO-IDS ensures the reliability of critical societal service, such as healthcare and emergency response, that will depend on these next-generation networks. Future work will additionally involve real-time production validation via live NWDAF interfaces to further align the model with deployment realities.

REFERENCES

[1] ETSI ISG ZSM, *Zero touch network and service management (ZSM); Requirements based on documented scenarios*, ETSI, Tech. Rep. ETSI GS ZSM 001 V1.1.1, 2019.

[2] M. El Rajab, L. Yang, and A. Shami, *Zero Touch Networks: Towards Next Generation Network Automation*, arXiv preprint arXiv:2312.04159, 2023.

[3] C. Benzaid and T. Taleb, *AI driven zero touch network and service management in 5G and beyond: Challenges and research directions*, IEEE Network, vol. 34, no. 2, pp. 186–194, 2020.

[4] L. Yang, S. Naser, A. Shami, S. Muhaidat, L. Ong, and M. Debbah, *Towards zero touch networks: Cross layer automated security solutions for 6G wireless networks*, arXiv preprint arXiv:2502.20627, 2025.

[5] E. Yu, Y. Song, G. Zhang, and J. Lu, *Learn to adapt: Concept drift adaptation for hybrid multiple streams*, Neurocomputing, vol. 496, pp. 121–136, 2022.

[6] B. Veloso, J. Gama, B. Malheiro, and J. Vinagre, *Hyperparameter self tuning for data streams*, Information Fusion, vol. 76, pp. 101–115, 2021.

[7] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton Brown, *Auto WEKA: Combined selection and hyperparameter optimization of classification algorithms*, Proc. KDD, pp. 847–855, 2013.

[8] P. Domingos and G. Hulten, *Mining high speed data streams*, Proc. ACM SIGKDD, pp. 71–80, 2000.

[9] A. Bifet, G. Holmes, and B. Pfahringer, *Leveraging bagging for evolving data streams*, ECML PKDD, pp. 135–150, 2010.

[10] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, *MOA: Massive Online Analysis*, Journal of Machine Learning Research, vol. 11, pp. 1601–1604, 2010.

[11] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfahringer, G. Holmes, and T. Abdessalem, *Adaptive random forests for evolving data stream classification*, Machine Learning, vol. 106, no. 9–10, pp. 1469–1495, 2017.

[12] H. M. Gomes, J. Read, and A. Bifet, *Streaming random patches for evolving data stream classification*, IEEE ICDM, pp. 240–249, 2019.

[13] J. Wu, X. Chen, H. Zhang, L. Xiong, H. Lei, and S. Deng, *Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization*, Journal of Electronic Science and Technology, vol. 17, no. 1, pp. 26–40, 2019.

[14] 3GPP, *Service requirements for the 5G system*, Technical Specification (TS) 22.261, V16.14.0, 2021.

[15] ITU T Study Group 13, *Work programme for Study Period 2025 to 2028 (Questions including Q20/13 and Q16/13)*, ITU T, 2025.

[16] ITU T, *Recommendation ITU T Y.3172: Architectural framework for machine learning in future networks including IMT 2020*, 2019.

[17] ITU T, *Recommendation ITU T Y.3063: Framework and requirements for measurement of effectiveness of autonomous networks*, 2025.

[18] ITU T, *Recommendation ITU T Y.3064: Autonomous Networks knowledge management*, 2025.

[19] ITU T, *Recommendation ITU T Y.3053: Framework of trustworthy networking*, 2020.