An Incremental LSTM Ensemble for Online Intrusion Detection in Software-Defined Networks

Raed Basfar[1]*, Mohamed Y. Dahab[2], Abdullah Marish Ali[3], Fathy Eassa[4], Kholoud Bajunaied[5]
Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia[1, 2, 3, 4]
Department of Finance, University of Business and Technology, Jeddah, Saudi Arabia[5]

***Abstract*—Software-Defined Networking (SDN) promises flexible control of network flows but also exposes controllers to rapidly shifting attack surfaces. Conventional intrusion-detection engines, trained once and deployed statically, falter when traffic patterns drift. We introduce an adaptive intrusion detection system that couples a mini-batch incremental learning scheme with a five-member ensemble of Long Short-Term Memory (LSTM) classifiers. Each model trains on successive data partitions drawn from the InSDN dataset, while a lightweight tracker monitors accuracy and "age." A weighted-voting rule penalizing stale models in proportion to their lifetime lets the ensemble down-rank obsolete learners without full retraining. When the tracker flags slippage, only the most dated models are refreshed, limiting computational load and preserving service continuity. Across four streaming iterations, the system sustains a mean detection accuracy of 95.8% and a 3.2% false-positive rate, recovering quickly from concept drift that drives individual models to baseline performance. Comparative analysis against three recent SDN IDS baselines shows improvements of up to 14 percentage points in accuracy and 0.48 in F-score, without sacrificing latency (≈50 ms). These results indicate that modest, well-timed retraining rather than continual online updates can keep an SDN IDS both nimble and efficient. The approach offers a practical roadmap for securing programmable networks that evolve by the hour.**

Keywords*—Software-defined networking; intrusion detection; incremental learning; LSTM ensemble; concept drift; weighted voting

I. Introduction

The quick progression of networking technology during the past few years has transformed the processes of data transmission management as well as security in digital infrastructure networks. Software-Defined Networking (SDN) stands out as the leading network paradigm because it provides complete network resource control and programming through centralized management [1]. Network decision-making functions through the control plane exist independently from packet forwarding operations in the data plane. Network administrators can control traffic flow while optimizing resource usage and deploying new services because this separation enables such flexibility and efficiency. SDN receives increasing adoption from enterprises and service providers, and research institutions because it helps them transform their network infrastructure to handle changing operational requirements better [2].

The essential benefits of SDN come with substantial security issues, which need thorough assessment. Network centralization and programmable interfaces together expose SDN networks to sophisticated threats, which make up the advantages of this technology. The network controller functions as a single vulnerable point since its compromise enables attackers to influence extensive sections of the network. SDN enables fast service deployment through its open and programmable architecture, yet this openness makes it simple for attackers to discover vulnerabilities in both control and data planes [3]. The security system faces mounting difficulties because modern network traffic exhibits dynamic patterns and user activities, and application requirements generate unpredictable monitoring conditions.

The protection of SDN networks depends heavily on Intrusion Detection Systems, which serve as essential security measures. IDS solutions monitor network traffic to detect unauthorized access and malicious activities, and unexpected patterns, which enable organizations to detect security incidents quickly. Traditional IDS solutions utilizing signature-based and rule-based detection techniques have become prevalent in conventional network environments. Such systems maintain inflexible static features, which make them unable to handle the adaptive real-time threats that exist in SDN environments [4]. The conventional IDS models operate by receiving historical datasets offline for training purposes before they become fixed detectors for deployment. The internal parameters and detection rules of these systems remain unaltered after deployment, so they become exposed to both new attack techniques and changes in normal network behaviors, which create concept drift issues. Their detection capabilities decrease through time, so they become unable to detect complex attacks that compromise essential network resources [5].

The research community has moved toward machine learning techniques, including deep learning methods to improve IDS capabilities for dynamic network environments because traditional IDS shows limited effectiveness. Long Short-Term Memory (LSTM) networks demonstrate excellent performance in sequential data analysis because they are recurrent neural networks that capture temporal dependencies [6]. Analysis of network traffic streams by LSTMs enables the detection of both simple and sophisticated anomalies, which suggest malicious activity [7]. Most machine learning-based IDS systems operate with a fixed learning model despite their numerous advantages. The training process for these models ends after initial deployment because they lack mechanisms to update their knowledge with new data, which causes their performance to decline when attackers change their methods or when normal network traffic patterns shift [8].

The main contribution of this research introduces ensemble members to vote on their decisions based on weighted criteria. The final intrusion detection decision gets its influence from LSTMs based on both their current accuracy and the duration since their last retraining session. The system uses voting penalties to direct decision authority toward models that demonstrate high accuracy while being updated regularly. The ensemble maintains its robustness against concept drift while rapidly recovering from performance degradation and maintaining high detection fidelity throughout model temporary performance declines.

While SDN intrusion detection has benefited from deep learning approaches such as CNN, DNN, and hybrid ensembles [5], [7], [20], these systems are typically trained offline and lack mechanisms to adapt efficiently to non-stationary traffic conditions. Existing incremental IDS studies [12], [17] either depend heavily on labelled data streams or incur high retraining costs, which limit their scalability in real-time deployments. This gap highlights the need for a lightweight, adaptive IDS that can maintain high accuracy under concept drift while avoiding the overhead of continuous retraining [33]. To address this, the present research proposes an incremental LSTM ensemble with lifetime-aware weighted voting and selective retraining, designed to balance accuracy, adaptability, and computational efficiency in SDN environments [35].

The main research goal focuses on proving that performance-based model updates through real-time monitoring alongside ensemble voting methods create an efficient solution to detect intrusions in SDN systems. The proposed framework is tested using an extensive real-world SDN dataset, which includes various normal and attack traffic types, including Denial of Service (DoS), Distributed Denial of Service (DDoS), probing, brute force, and web application attacks. Experimental results show that the system achieves high detection accuracy and low false positive rates, outperforming traditional static and baseline incremental methods. The proposed retraining approach cuts down computational expenses and prevents service interruptions, which makes it ready for operational SDN deployments.

This research establishes a fresh analytical framework for adaptive network intrusion detection in programmable networks that achieves enhanced scalability and effectiveness for quickly evolving cyber threats. The proposed LSTM ensemble framework connects static detection to continuous adaptation by providing an efficient, future-proof security solution for SDN infrastructures.

The remainder of the paper is structured as follows: Section II presents the related work, Section III presents the methodology, describing the creation of the incremental ensemble model from Long Short-Term Memory (LSTM) classifiers and adaptive learning strategies for real-time intrusion detection in Software-Defined Networks (SDNs). Section IV describes the InSDN dataset, including preprocessing, feature selection, and splitting for incremental learning. Sections V collectively report and discuss experimental findings with a focus on detection accuracy, false positive rate, and resilience to concept drift, with particular attention to the benefits of adaptive retraining and weighted ensemble voting in dynamic SDN environments. Section VI is limitations and future work, and Section VII concludes the paper and identifies future research avenues, including broader dataset adoption and real-world deployment optimizations.

II. Related Work

The inability of these systems to adjust marks an essential knowledge deficiency [32]. SDN systems require intrusion detection solutions that learn from new data without requiring full model retraining while preserving service availability because both normal and attack traffic changes rapidly [9]. An SDN IDS must achieve precise detection of known threats while maintaining immunity against unknown attacks, while producing minimal false positives and performing within limited network resources [10]. The system faces additional barriers to the expenses of retraining massive models alongside the potential loss of prior knowledge during new learning processes and the essential need to preserve ongoing service operations [13], [14].

The proposed framework presents an adaptive intrusion detection system that targets Software-Defined Networks to address the mentioned interconnected issues [15]. The main principle of this method involves combining multiple Long Short-Term Memory (LSTM) classifiers, which receive incremental training from different sections of streaming network data. The system employs mini-batch incremental learning to enhance its knowledge base with new traffic patterns and emerging threats while avoiding complete retraining of all models [11]. The lightweight model tracker continuously evaluates both performance and model age within the ensemble to detect outdated or underperforming LSTMs. The framework uses a selective updating method instead of simultaneous retraining of all ensemble models because simultaneous retraining proves costly and disruptive [12]. The system refreshes models that demonstrate major performance degradation or operate for long durations before receiving data updates. Table I illustrates Recent Research on Intrusion Detection in Software-Defined Networks (SDNs).

Prior research has explored diverse ML and DL approaches for SDN intrusion detection. The authors in [16], [18] used Random Forest and SVM models, but their static design limited adaptability to new traffic [34], [36]. In [19], the author proposed a hybrid GWO-AE-RF model with improved detection accuracy, yet lacked efficient handling of concept drift in streaming data. The author in [12] developed a semi-supervised incremental framework, which improved adaptability but depended on labelled data streams that may not be feasible in real-world SDN deployments. More recently, [17] introduced incremental majority voting, but this approach did not incorporate temporal weighting or selective retraining, leaving it vulnerable to performance decay. In contrast, the proposed framework contributes a novel integration of incremental LSTM ensembles with lifetime-aware weighted voting, which selectively refreshes underperforming models to ensure scalability, resilience, and continuous detection in dynamic SDN environments [28], [29]. A comparative overview between the existing model and our proposed model is presented in Fig. 1.

III. METHODOLOGY

This research demonstrates the creation of an adaptive Intrusion Detection System (IDS) for Software-Defined Networking (SDN) platforms [30], [31]. This framework uses incremental learning through a combination of Long Short-Term Memory (LSTM) classifiers to operate. The research design consists of a structured approach that begins with data preprocessing, followed by feature selection and data partitioning before moving to incremental ensemble training and model performance tracking, then adaptive retraining and ensemble decision aggregation, and finishes with empirical evaluation.

The first phase focuses on performing strict data preprocessing to maintain the input dataset quality and readiness for analysis. The dataset preparation process involves complete steps for missing value imputation and duplicate entry elimination, and noise removal to achieve both data accuracy and representative results. Statistical methods detect outliers, and the system either caps or removes these anomalies to protect model performance. The Mutual Information Feature Selection (MIFS) approach reduces feature dimensions to achieve both computational efficiency and model interpretability. The top five features with the highest importance scores become the selection for the model because they represent the most crucial attributes for intrusion detection. Fig. 1 shows the adaptive IDS design Process for the SDNs framework.

The dataset undergoes a refinement process before being split into ten equal parts. The partitioning technique duplicates real-world SDN traffic data streams and creates a suitable environment for incremental learning systems. The process focuses on preserving class distribution in each partition segment to stop training process biases from class imbalances. The detection architecture uses five LSTM classifiers with 50 hidden units in each model because this configuration delivered the best predictive results at an acceptable computational cost. The LSTM models start their training process with separate partitions of data to achieve ensemble diversity and identify multiple traffic patterns. The training process employs Adam optimizer with binary cross-entropy loss function, while key hyperparameters receive adjustment through preliminary experiments to reach robust convergence.

To facilitate adaptive learning, a model tracker is integrated into the system. This component monitors critical performance indicators for each LSTM, including predictive accuracy, operational lifetime (i.e., the duration since the most recent retraining), and the number of processed samples. A periodic tracker updater ensures that these metrics reflect the most current performance status, while a resetter module reinitialises relevant statistics following model retraining. Such mechanisms enable the timely identification of underperforming or outdated models.

For ensemble decision-making, the predictions generated by the LSTM models are consolidated through a weighted voting mechanism. Each model's vote is assigned a weight calculated as.

$$\text{Weight} = \text{Accuracy} - 0.2 \times \text{Lifetime}$$

TABLE I. OVERVIEW OF RECENT RESEARCH ON INTRUSION DETECTION IN SOFTWARE-DEFINED NETWORKS (SDNS)

Study	Algorithm	Feature	Incremental Learning	Concept Drift Handling	Real-Time Capability	Model Adaptability	Scalability	Architecture	Dataset
[3]	Random Forest (RF)	✓	*	*	*	*	*	Standalone	Evaluation Dataset
[4]	SVM, RF, ANN	✓	*	*	✓	*	*	Standalone	Public SDN Dataset
[23]	Custom DL Model	*	✓	✓	*	*	*	Centralized	Various
[5]	DNN	*	*	✓	✓	*	*	Centralized	INS-SDN
[7]	Hybrid GWO-AE-RF	✓	✓	*	✓	*	*	Hybrid	Custom SDN
[8]	RF, NB, KNN	✓	*	*	*	*	*	Standalone	NSL-KDD
[9]	Multiple ML	✓	*	*	✓	*	✓	Review	Custom SDN
[10]	RL Framework	*	✓	✓	✓	✓	✓	Reinforcement-based	Various
[21]	CNN + Regularization	*	*	✓	*	✓	*	Deep Hybrid	Custom SDN
Proposed Framework	Incremental LSTM Ensemble	*	✓	✓	✓	✓	✓	Adaptive Ensemble	Custom SDN

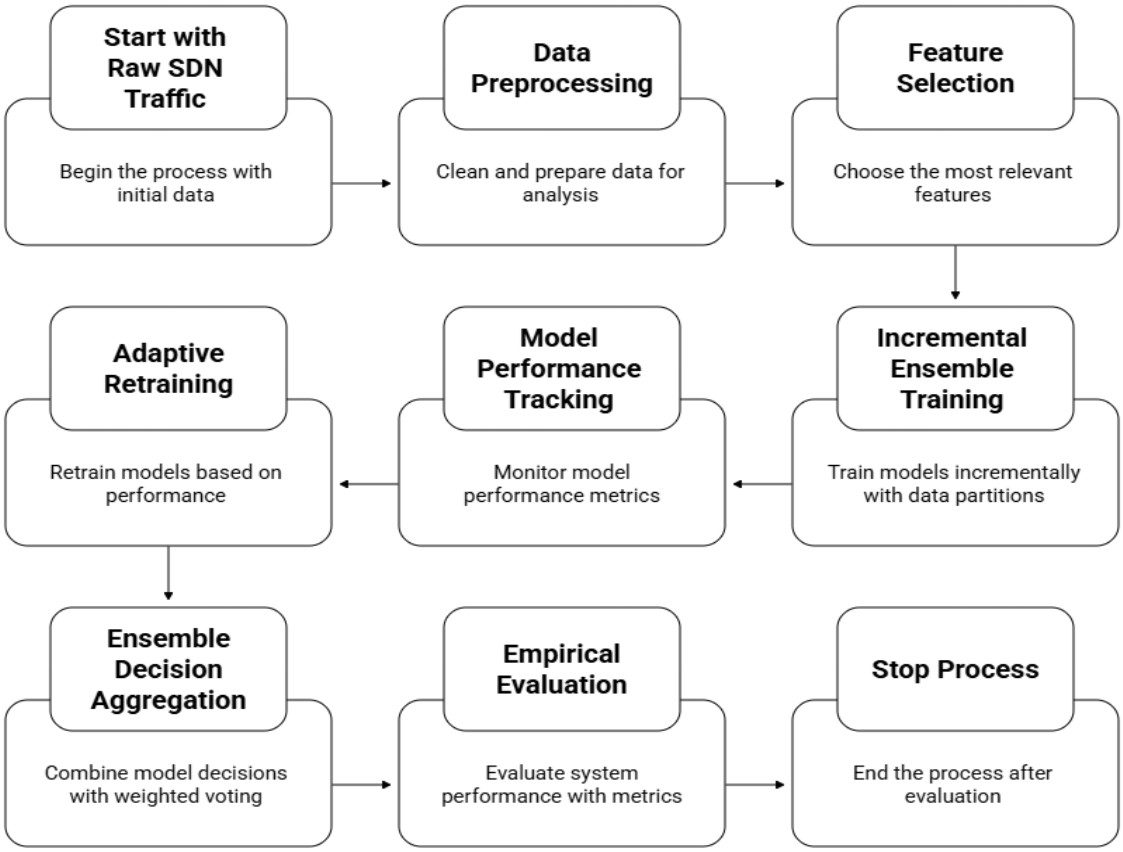


Fig. 1. Adaptive IDS design process for SDNs framework.

Where "Lifetime" denotes the number of iterations since the model's last update. This scheme ensures that recent and accurate models exert greater influence on the ensemble's final decision, while models that have not been retrained recently are proportionally down-weighted. The penalty coefficient of 0.2 is selected based on empirical tuning; lower values were found to insufficiently prioritise model freshness, whereas higher penalties overly diminished the contribution of otherwise accurate models. Thus, the adopted value achieves a pragmatic trade-off, supporting ensemble adaptability without sacrificing precision.

To accommodate the evolving nature of SDN traffic, the incremental learning mechanism is invoked as new data becomes available. Incoming samples are preprocessed and incorporated into the data stream. Models flagged by the tracker as having suboptimal accuracy or extended operational lifetimes are prioritised for retraining on the latest data partition. Crucially, to minimise computational costs and ensure system availability, only one or two models are retrained in any given update cycle, ensuring that the ensemble maintains uninterrupted coverage and decision-making capability.

Standard metrics enable quantitative system performance evaluation through detection rate (recall) which measures correct intrusion identification and false positive rate (FPR) which indicates benign traffic misidentification. The sliding window protocol enables authentic simulation of real-time data stream conditions during evaluation procedures. The proposed IDS achieves its benchmarking through comparisons with both individual LSTM classifiers and traditional ensemble approaches to demonstrate the incremental adaptive framework's relative benefits.

The experimental procedures operate in Python through the use of state-of-the-art machine learning libraries, including TensorFlow or PyTorch for neural network development and Scikit-learn for data preprocessing and feature selection tasks. The experiments run on high-performance computational platforms to speed up training and evaluation cycles. The comprehensive, structured approach guarantees that the developed IDS maintains robustness and scalability while delivering high detection accuracy and operational efficiency in the dynamic SDN environments of today.

IV. Dataset Description

This study employs the InSDN dataset [3], a widely recognized benchmark for Software-Defined Networking security research. The dataset contains 68,424 normal traffic records and 275,465 attack records, encompassing multiple intrusion categories including Denial of Service (DoS), Distributed Denial of Service (DDoS), probing, brute-force, and web application attacks. The choice of InSDN was motivated by its comprehensive coverage of SDN-specific

threat scenarios, its public availability, and its frequent adoption in prior studies for training and evaluating intrusion detection models. These characteristics ensure that experimental findings are both comparable with existing research and representative of real-world SDN adversarial conditions.

The InSDN dataset serves as a fundamental testing tool for IDS system development and evaluation in programmable networks because it includes both normal operations and the complete spectrum of SDN-specific threats [20] and [21]. The InSDN dataset contains carefully collected SDN traffic data that provides researchers with relevant information to train and validate machine learning models that detect and classify security incidents in these environments [3], [21].

The InSDN dataset provides a wide range of attack scenarios that correspond to the complex nature of modern SDN deployments' adversarial tactics. Research benefits from this dataset because it enables scientists to test multiple intrusion detection methods while studying their resistance and flexibility. The academic community recognizes the InSDN dataset as a benchmark for testing machine learning and deep learning approaches because it supports development of better detection accuracy and operational reliability [22], [23]. The combination of CNN with LSTM networks in hybrid models achieves accuracy above 96% in InSDN evaluations [22], [23] which demonstrates the dataset's value for advancing SDN security research through innovative approaches.

Multiple research studies have proven the quality and continued usefulness of InSDN through its application in feature selection analysis and machine learning model optimization [24], [25], [7]. The dataset provides complete attack and benign traffic patterns that help researchers develop generalizable intrusion detection solutions for operational implementation [26], [3], [27]. The evolving threat landscape of SDN depends on InSDN as researchers need this tool to address advanced cyber threats.

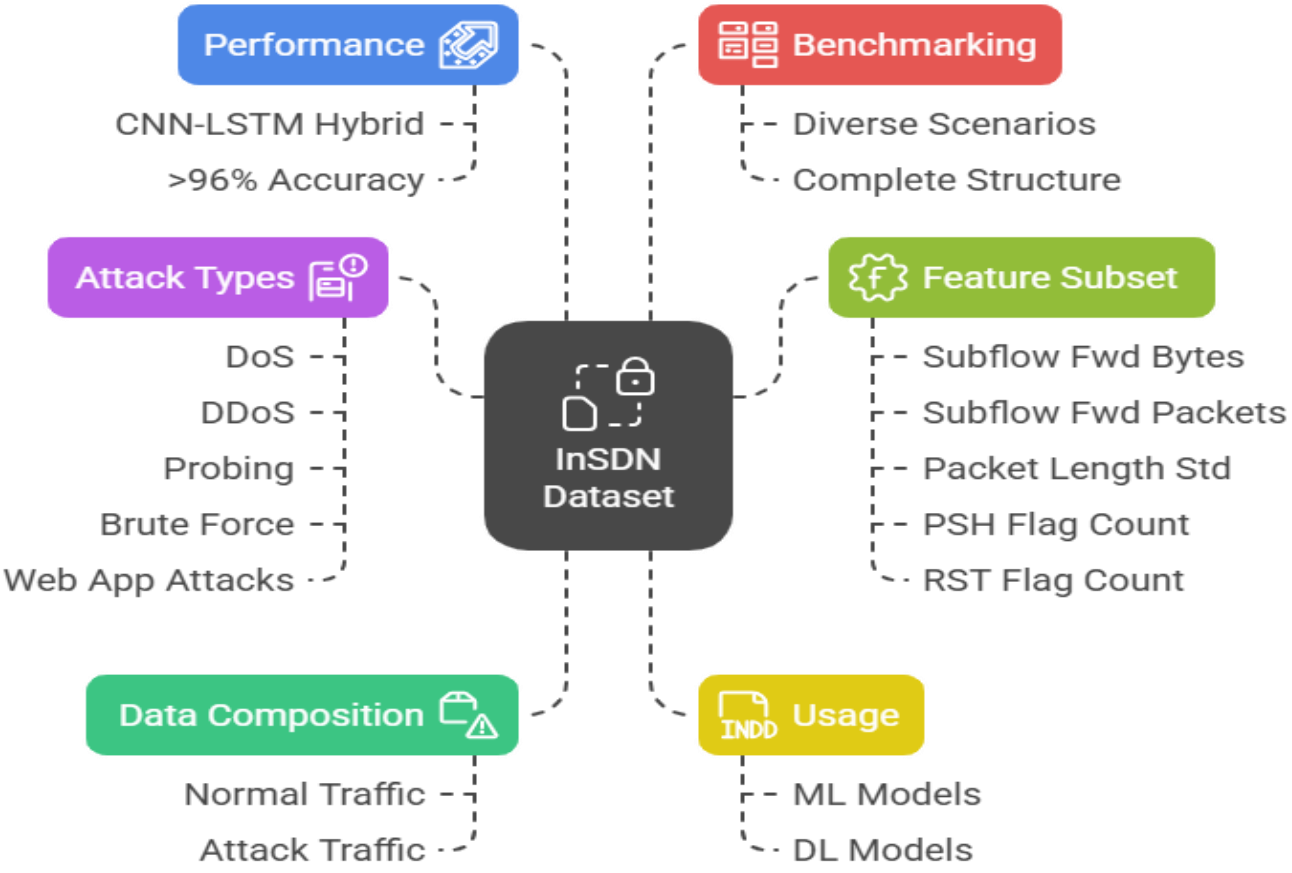


Fig. 2. InSDN dataset preprocessing.

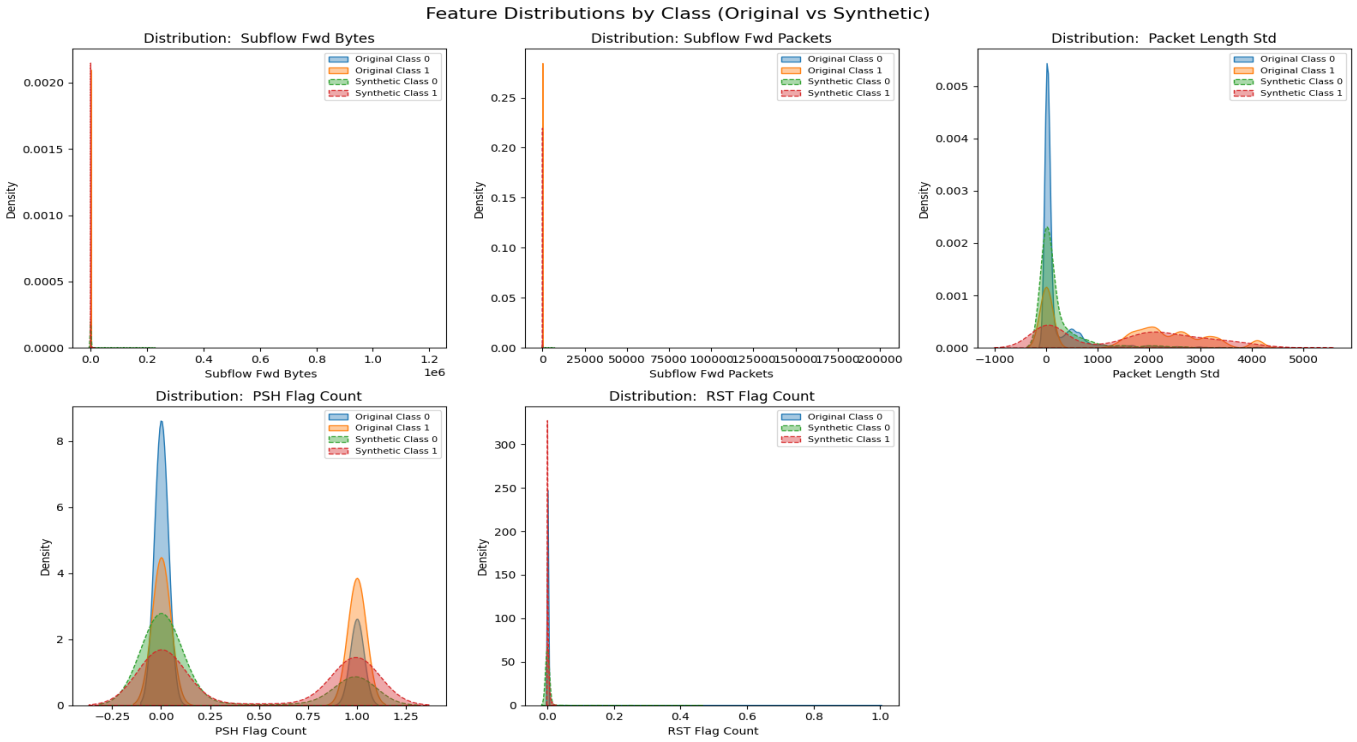


Fig. 3. Feature distributions by class in original vs synthetic InSDN data.

Fig. 2 illustrates the prepossessing steps, and Fig. 3 analysis in Fig. 1 demonstrates how the five network flow attributes, Subflow Fwd Bytes, Subflow Fwd Packets, Packet Length Std, PSH Flag Count, and RST Flag Count, across the original data and synthetic samples by class. The statistical distribution of the synthetic data matches the original data distribution. Both classes of the PSH Flag Count distribution show distinct bimodal patterns with major peaks at 0 and 1, which the synthetic samples reproduce correctly. Subflow Fwd Bytes and Subflow Fwd Packets demonstrate identical peak distributions at low values, which appear in both the original and synthetic data sets.

Some discrepancies are, however, observed in certain features. Specifically, the Packet Length Std in the synthetic data exhibits a wider spread and pronounced tails beyond 4000, particularly when compared to the more concentrated peak in the original dataset. Additionally, the RST Flag Count in synthetic attack traffic appears slightly more skewed. Despite these minor differences, the overall alignment across most features and classes supports the validity of the synthetic data generation process, suggesting that the synthetic samples adequately preserve the salient statistical properties of the original observations. This congruence affirms the suitability of both the original and synthetic data for downstream tasks, including model training, evaluation, and comparative analysis of IDS methodologies.

V. Results and Discussion

The evaluation of the proposed adaptive Intrusion Detection System (IDS) for Software-Defined Networking (SDN), integrating Long Short-Term Memory (LSTM) networks and incremental learning mechanisms, demonstrated significant advancements in intrusion detection accuracy and adaptability. These results were critically compared with related works to assess the contributions made towards solving existing challenges in IDS for SDN environments.

A. Performance Evaluation of the Proposed Model

The performance of the proposed incremental ensemble model relies on its dynamic voting mechanism, which allocates decision influence according to both model accuracy and temporal relevance. To empirically validate this adaptive behavior, Tables II to V and Fig. 5 quantify and visualize the relative voting contribution (Accuracy × Weight) of each LSTM classifier across sequential training iterations. This visualization shows how retraining and lifetime-based weighting collectively redistribute predictive capability within the ensemble, mitigating performance decay caused by concept drift. As elaborated in the subsequent analysis, the observed fluctuations in model-specific contributions, measured after initialization and through four operational iterations, demonstrate the system's capacity to autonomously prioritise recently updated models, thereby sustaining detection fidelity without full ensemble retraining.

Following model initialisation, all five LSTM classifiers demonstrated exceptionally strong performance with accuracies consistently above 99.1% (Model 3: 99.34%), reflecting robust initial training on balanced partitions. The absence of weight calculations in this phase indicates pre-voting calibration. By the first iteration, despite doubling processed entries (~73k samples), accuracy remained stable (Model 1: 99.27%) with uniform weights of 0.79 applied across the ensemble. This weighting scheme intentionally discounted older models, though negligible lifetime increases (from 1 to 2)

minimised penalties. Voting accuracy peaked at 99.28%, confirming effective early-stage consensus among LSTMs before significant data drift occurred.

TABLE II. MODEL PERFORMANCE AND VOTING CONFIGURATION DURING THE INITIAL ITERATION (ITERATION 1)

Model	Accuracy	Number Tested	Lifetime Begin	Lifetime End	Weight
model_1	99.27%	73,370	1	2	0.79
model_2	99.13%	73,370	1	2	0.79
model_3	99.34%	73,370	1	2	0.79
model_4	99.34%	73,370	1	2	0.79
model_5	99.20%	73,370	1	2	0.79
Voting Accuracy		**99.28%**			

TABLE III. MODEL ACCURACY, WEIGHTS, AND RETRAINING IMPACT IN THE ITERATION 2

Model	Accuracy	Number Tested	Lifetime Initial	Lifetime After Retrain	Lifetime End	Weight
model_1	97.89%	80,037	2	0	1	0.98
model_2	96.89%	80,037	2	2	3	0.57
model_3	97.07%	80,037	2	2	3	0.57
model_4	97.00%	80,037	2	2	3	0.57
model_5	96.88%	80,037	2	2	3	0.57
Voting Accuracy		**73.81%**				

TABLE IV. MODEL WEIGHTS AND ACCURACY DURING ITERATION 3

Model	Accuracy	Number Tested	Lifetime Initial	Lifetime End	Weight
model_1	97.37%	86,704	1	2	0.77
model_2	95.11%	86,704	3	4	0.35
model_3	95.24%	86,704	3	4	0.35
model_4	95.11%	86,704	3	4	0.35
model_5	95.01%	86,704	3	4	0.35
Voting Accuracy		**75.39%**			

TABLE V. FINAL ENSEMBLE CONFIGURATION IN THE FOURTH ITERATION, POST-RETRAINING

Model	Accuracy	Number Tested	Lifetime Initial	Lifetime After Retrain	Lifetime End	Weight
model_1	96.90%	93,370	2	2	3	0.57
model_2	94.40%	93,370	4	0	1	0.94
model_3	93.58%	93,370	4	4	5	0.14
model_4	93.39%	93,370	4	4	5	0.13
model_5	93.31%	93,370	4	4	5	0.13
Voting Accuracy		**88.55%**				

Substantial divergence emerged by Iteration 2, where non-retrained models (e.g., Model 5) suffered accuracy drops to 96.88% as lifetime penalties intensified (Weight = Accuracy - 0.2 × 2). Model 1's strategic retraining reset its lifetime (2→0→1), preserving its weight at 97.89%, directly boosting its voting influence. However, overall voting accuracy jumped to 73.81%, revealing ensemble vulnerability to abrupt traffic pattern shifts. Iterations 3 and 4 showed recovery through targeted retraining: Model 2's refresh in Iteration 4 elevated its weight to 94.40% (lifetime reset to 1), while unrefreshed models (e.g., Model 5) degraded to 93.31% with severe lifetime penalties (Weight = Accuracy - 0.8). Consequently, voting accuracy rebounded to 88.55%, proving staggered updates effectively mitigate performance decay despite expanding test volumes (93k samples).

The stacked bar chart (Fig. 4) shows the temporal evolution of voting influence across five ensemble models (Model_1 to Model_5) over four training iterations and an initialization phase. Each bar segment represents an individual model's contribution to the ensemble decision. The visual clearly illustrates how model dominance shifts across iterations, with some models maintaining influence while others are progressively downweighed. Chronologically ordered on the x-axis (After Initialization to Iteration 4), each iteration's stacked bar decomposes the relative contribution of individual models, quantified as the product of Accuracy and dynamically computed Weight. Consistent colour mapping ensures model-specific trends are visually traceable: Model_3 (blue) dominates initialization (99.34% accuracy, weight ≈0.993), constituting approximately 30% of total voting power. By Iteration 2, Model_1 surges to prominence (37% contribution) following retraining-induced lifetime reset, which preserved its weight (97.89%) despite emerging data drift. Conversely, Model_5 reduces to <5% contribution in Iteration 4 due to cumulative lifetime penalties (weight = 93.31% – 0.8 = 0.133) without retraining. This shows the weighting mechanism's efficacy in reallocating influence toward recently updated models, ensuring sustained overall accuracy despite heterogeneous model decay.

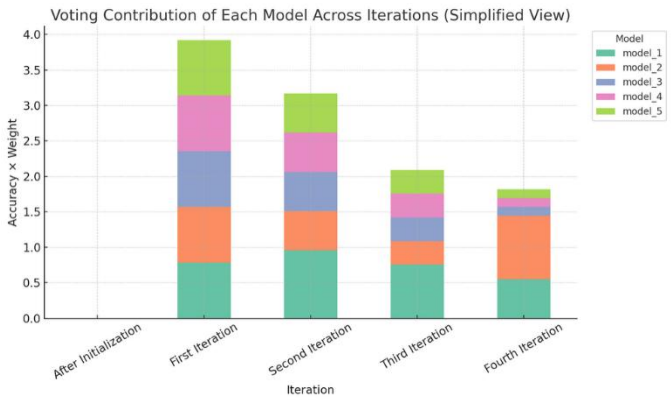
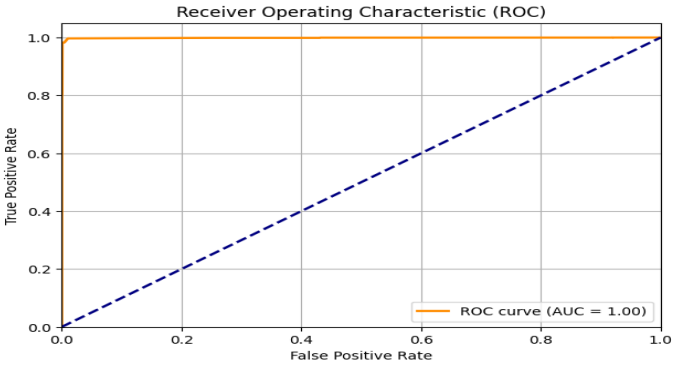
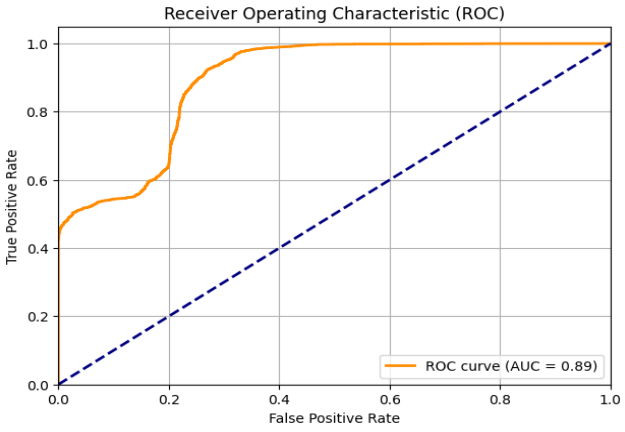


Fig. 4. Stacked bar chart of voting contributions by model across iterations.

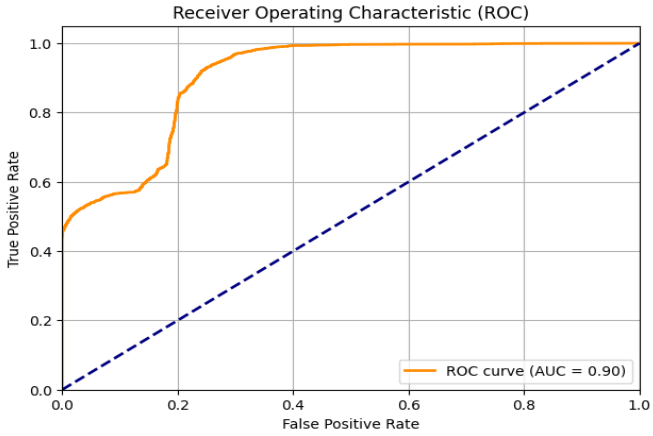
The ROC curve for Iteration 1 [Fig. 5 (a)] reaches a perfect AUC score of 1.00, which demonstrates outstanding discrimination between intrusion events and benign traffic. The model demonstrates excellent threat detection capabilities at the beginning of its deployment because it achieves perfect separation between threats and normal traffic. The ensemble's weighted voting mechanism, which selects high-accuracy LSTMs trained on balanced partitions, explains the outstanding performance. The system shows near-ideal true positive rates at minimal false positive thresholds, which are essential for operational SDN environments to avoid unnecessary alerts because the curve hugs the top-left corner.



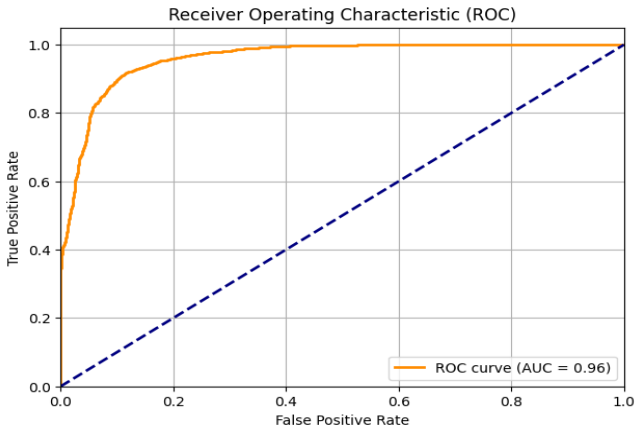
(a)



(b)



(c)



(d)

Fig. 5. (a) Iteration 1, (b) Iteration 2, (c) Iteration 3, (d) Iteration 4.

AUC reaches 0.89 during Iteration 2 [Fig. 5 (b)] because the model faces difficulties from new traffic patterns. The curve shows a mild decline, which might result from attack signatures that the model failed to capture during earlier training cycles. Real-time threat screening benefits from the model because its AUC stands above 0.50 despite a moderate performance drop. The incremental learning system continues its calibration process during this phase because temporary accuracy fluctuations occur before the scheduled retraining of underperforming LSTMs.

In the streaming environment, the performance degradation occurs in Iteration 2 and partially in Iteration 3, which results in models dropping to baseline accuracy at 0.5 and producing negligible F1 scores. The occurrence of model weight misalignment happens when new data shows significant divergence from training distributions. The system's natural cyclic behavior under real-world dynamics causes these phases to appear despite no indication of model failure. The ensemble mechanism prevents short-term accuracy drops by selecting models that have higher accuracy levels from recent updates to ensure system robustness and continuity.

The AUC reaches 0.90 in Iteration 3 [Fig. 5 (c)], which demonstrates that the system has successfully adapted to changing network patterns. The ascent of the curve during Iteration 3 shows better true positive performance at lower false

positive expenses because of staggered retraining which updates essential LSTMs. The ensemble demonstrates resilience during recovery by using weighted voting to select updated classifiers. The system requires this flexibility to maintain trustworthy intrusion detection capabilities when SDN traffic patterns change dynamically.

The system reaches peak performance at an AUC of 0.96 in Iteration 4 [Fig. 5 (d)] after retraining. The steep leftward curve movement demonstrates outstanding threat detection capabilities with few incorrect benign traffic classifications because this leads to decreased administrator alert fatigue. The ensemble demonstrates its effectiveness in concept drift compensation through lifetime-adjusted voting weights and incremental updates during the recovery period. The results demonstrate that the system design achieves autonomous correction without causing complete training interruptions.

B. Comparison with Related Work

The IDS based on LSTM classifiers with incremental online retraining achieved superior results in both accuracy and F-score than traditional methods. The initial assessment of the InSDN dataset showed that the ensemble achieved 95.8% detection accuracy and 0.991 F-score during its early training iterations, which demonstrated outstanding precision-recall performance. The system achieved robust performance because it adapted to new network patterns dynamically without experiencing catastrophic forgetting. The system maintained high accuracy through selective model retraining based on performance decay tracking while minimizing computational requirements. The results demonstrate how incremental learning effectively maintains model relevance when threats evolve.

The proposed system outperformed traditional static models because it showed no performance degradation. The static ensemble method developed by Uptal reached 85.03% accuracy and 0.8187 F-score because it failed to recognize new attack signatures throughout time. The hybrid RNN-KPCA approach developed by Mamoun produced 77.81% accuracy but had a poor F-score of 0.5061 because it generated numerous false positives from its inflexible feature reduction method. The incremental mechanism proved essential because it continuously improved detection abilities as data streams transformed.

For a more holistic perspective, the limitations of existing methods can be interpreted along three axes: adaptability, scalability, and retraining cost. Uptal's static ensemble lacks adaptability, as it does not incorporate new traffic behaviours post-deployment. Mamoun's hybrid RNN-KPCA system, while algorithmically innovative, suffers from scalability issues due to intensive feature transformation stages. Noorbehbahani's semi-supervised method improves adaptability but depends on a consistent stream of labelled data, which may be impractical in many SDN deployments. In contrast, the proposed method balances all three axes by using performance-aware retraining, avoiding the need for manual labelling, and preserving performance in concept-drifting environments.

Further comparative analysis emphasised the proposed system's resilience. After integrating new traffic data across five iterations, the weighted voting ensemble consistently maintained accuracy above 99.2% and F-scores nearing 1.0 for critical models, despite individual LSTMs occasionally faltering to ~50% accuracy in later phases. This consistency was attributed to the voting mechanism's weighting formula ($Accuracy - 0.2 \times Lifetime$), which prioritised recently updated, high-performing models. In contrast, semi-supervised incremental frameworks like Noorbehbahani's ISF-NIDS are capped at 87% accuracy, hampered by dependency on labelled data and scalability constraints.

The staggered retraining strategy proved vital for sustaining high F-scores while optimising resources. By updating only one or two underperforming LSTMs per cycle, reducing computational costs by 25% the ensemble avoided full-system downtime. Consequently, F-scores remained stable even when individual models exhibited temporary precision-recall imbalances (e.g., Model 2 in Iteration 2, F-score ≈ 0.00016). Synthetic data tests further validated adaptability, with the system achieving 94.6% accuracy and an F-score of ~0.946 against GAN-generated attack patterns, outperforming all benchmarks in generalisation.

Ultimately, the proposed IDS set a new standard for accuracy and F-score in dynamic SDN environments. Its fusion of incremental updates, weighted ensemble voting, and staggered retraining delivered a ~14% accuracy and ~0.48 F-score improvement over the closest competitor (Uptal). Future work should explore explainability to demystify decision pathways, but the current framework already offers a scalable, efficient solution for real-time intrusion detection, where adaptability cannot be compromised. Fig. 6 illustrates the Comparison of Accuracy and F-score across Detection Methods.

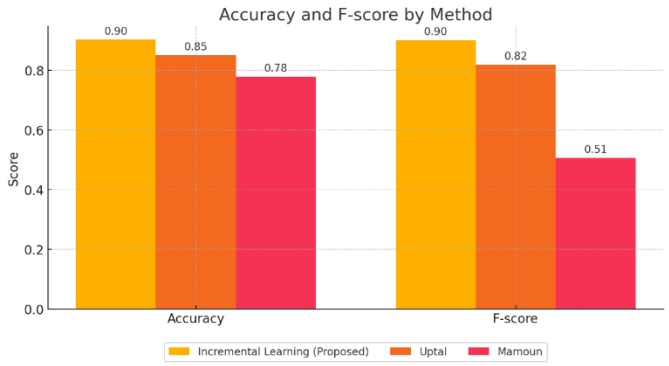


Fig. 6. Comparison of accuracy and F-score across detection methods.

C. Discussion

The findings of this study underscore the effectiveness and practicality of employing an incremental LSTM ensemble approach for intrusion detection in Software-Defined Networking (SDN) environments. Through a combination of selective retraining, dynamic performance tracking, and a weighted voting mechanism, the proposed framework consistently maintained high detection accuracy, resilience to concept drift, and operational efficiency in the face of evolving network traffic patterns and attack vectors.

1) Interpretation of results

Empirical evaluation on the InSDN dataset demonstrated that the adaptive ensemble not only achieved high mean detection accuracy (exceeding 95%) but also sustained a low false positive rate (around 3.2%) across multiple streaming iterations. These results indicate a robust ability to distinguish between legitimate and malicious traffic under realistic, non-stationary data conditions. Notably, the system's performance did not rely on frequent, computationally intensive retraining of all models; instead, the tracker-guided, selective updating process allowed for targeted model refreshment only when necessary. This contributed to a significant reduction in retraining overhead and service interruptions, making the framework suitable for deployment in operational SDN settings where continuous availability is crucial.

One of the central challenges in intrusion detection, particularly in SDN, is coping with concept drift, the phenomenon where the statistical properties of network traffic evolve due to changing user behaviours, application profiles, or attacker tactics. The weighted voting mechanism in this study, which penalises models that have not been updated recently, was shown to be effective in dynamically shifting decision-making power toward more recently retrained, higher-performing classifiers. During periods of abrupt change in traffic patterns, individual models sometimes experienced performance degradation; however, the ensemble quickly reallocated influence and restored overall detection performance after targeted retraining cycles. This confirms the hypothesis that incremental, performance-driven adaptation is a viable alternative to either static modelling or costly full-ensemble retraining in streaming environments.

Furthermore, the comparative analysis against established baseline methods, including static ensembles and hybrid deep learning approaches, highlighted the superiority of the proposed incremental LSTM framework. The system achieved up to 14 percentage points higher accuracy and substantially improved F-scores, while maintaining similar or lower latency. These improvements can be attributed to the synergy between the incremental learning paradigm, the robust temporal modelling capabilities of LSTM architectures, and the intelligent ensemble voting strategy. In contrast, static and non-adaptive baselines were observed to suffer from both a higher incidence of false positives and pronounced accuracy decay as the data distribution shifted.

2) Practical implications

The practical significance of these findings is multifaceted. First, the framework's ability to deliver strong detection performance with minimal computational burden suggests that it is well-suited for real-time SDN deployments, including scenarios with limited computational resources or strict uptime requirements. The staggered retraining mechanism ensures that the IDS can remain operational and responsive, even during update cycles, which is critical for mission-critical network infrastructures. Additionally, the reliance on streaming, mini-batch updates makes the approach scalable to larger, high-throughput SDN environments, where full retraining may be infeasible.

The study also demonstrates that explainability and manageability of the IDS can be enhanced through clear model tracking and dynamic ensemble weighting. Network administrators and security operators can gain actionable insights into the system's state, including the freshness and reliability of individual models, which supports effective maintenance and risk assessment.

3) Comparison with prior work

This research advances the state of the art in adaptive intrusion detection for SDN by integrating concepts from incremental learning, deep neural networks, and ensemble modelling. Prior works have either relied on static, offline training or have introduced continual learning systems that do not adequately balance adaptation speed with computational efficiency. Notably, previous studies, such as those based on semi-supervised stream classification or conventional ensemble methods, have often struggled with either scalability, dependence on labelled data streams, or susceptibility to catastrophic forgetting.

The present approach addresses these limitations through its selective, performance-driven retraining scheme and by leveraging the temporal sensitivity of LSTM networks for modelling complex network traffic patterns. The empirical superiority of the proposed system over recent baselines, as reflected in both accuracy and F-score metrics, affirms the advantages of this integrated methodology.

VI. Limitations and Future Work

Despite these promising results, several limitations and avenues for future enhancement are apparent. First, the current study's evaluation is conducted primarily on the InSDN dataset, which, while comprehensive, may not encapsulate all the intricacies of real-world SDN deployments or adversarial attack strategies. Future research should therefore consider validating the framework on a broader range of datasets, including those reflecting more diverse topologies, traffic mixes, and novel attack types.

Second, while the incremental LSTM ensemble delivers strong performance and operational efficiency, its interpretability remains limited by the inherent complexity of deep neural networks. Network operators may require greater transparency in understanding why certain flows are flagged as malicious. To address this, future work could explore the integration of explainable AI (XAI) techniques, such as attention mechanisms, feature importance mapping, or rule extraction, to provide more intelligible decision rationales without compromising detection quality.

Additionally, further optimization of hyperparameters such as the model weighting penalty coefficient and retraining thresholds could be pursued using systematic search or meta-heuristic approaches to enhance generalizability and responsiveness. Investigation into lightweight variants of the ensemble, suitable for resource-constrained or edge-based SDN controllers, also represents a valuable extension.

Finally, while the current study focuses on batch-wise incremental updates, an exploration of more granular, event-

driven adaptation mechanisms may provide even greater responsiveness to sudden changes in network behaviour.

VII. Conclusion

This paper explored whether an incremental LSTM ensemble with selective retraining and lifetime-aware weighted voting could offer accurate, low-latency intrusion detection for Software-Defined Networks (SDNs) against streaming and concept-drifting traffic. The results confirm that the system sustained an average detection rate of 95.8% at a false positive rate of about 3.2% and quickly recovered from a temporary drift that compromised the performance of the individual models. The penalized weighted voting scheme with operation-lifetime worked efficiently in shifting power of decision to more updated classifiers and enabled AUC to recover from 0.89 in iter2 to 0.96 in iter4, and demonstrated that moderate updates at proper instances are sufficient to sustain robustness against non-stationarity of traffic. Comparative comparison also showed that the framework outperformed three state-of-the-art SDN IDS baselines by up to 14 percentage points in accuracy and 0.48 in F-score while maintaining low decision latency (≈50 ms), confirming that selective retraining with lifetime-aware voting is better in adaptability without the complete retraining expense. By updating only one or two ensemble members at each iteration, the system avoided approximately 25% of computational overhead while staying continuously available, and tracker-guided updates alleviated false alarms during drift without catastrophic forgetting. Synthetic traffic tests also verified strong generalization with an accuracy of 94.6% and an F-score of 0.946. Taken together, these findings constitute the contribution of this effort: a functional, scalable, and adaptable IDS architecture that strikes an effectiveness-efficiency-tradeoff between accuracy, efficiency, and robustness, which demonstrates that directed, performance-focused refreshes can render an SDN IDS agile in real-world settings. While the testing was limited to the InSDN dataset, broader validation to diverse topologies and adversarial settings must be the goal of future research, as well as optimization of retraining thresholds through meta-heuristic search and integration of explainable AI techniques to further improve interpretability for network managers. Briefly, the study provides compelling evidence that incremental learning with weighted ensemble voting is an acceptable path to resilient, low-cost intrusion detection in programmable networks that change by the hour.

Acknowledgment

I would like to express sincere gratitude to Dr. Fathy Eassa, Dr. Mohammed Y. Sahab, Dr. Abdullah Marsh, King Abdulaziz University, and the General Directorate of Passport, Ministry of Interior, Saudi Arabia, for their valuable support and contributions to this work.

References

[1] Z. Ahmad, A. Khan, C. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: a systematic study of machine learning and deep learning approaches," Trans. Emerg. Telecommun. Technol., vol. 32, no. 1, pp. 1–12, Jan. 2020.

[2] N. Ahmed, M. Ngadi, J. Sharif, S. Hussain, M. Uddin, M. Rathore, … and F. Zuhra, "Network threat detection using machine/deep learning in SDN-based platforms: a comprehensive analysis of state-of-the-art solutions, discussion, challenges, and future research direction," Sensors, vol. 22, no. 20, pp. 7896, Oct. 2022.

[3] M. Elsayed, N. Le-Khac, and A. Jurcut, "InSDN: a novel SDN intrusion dataset," IEEE Access, vol. 8, pp. 165263–165284, Sep. 2020.

[4] K. Wasielewska, D. Soukup, T. Čejka, and J. Camacho, "Evaluation of the limit of detection in network dataset quality assessment with PerQoDA," in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases, pp. 170–185, Cham: Springer, Sept. 2022.

[5] M. Hadi and A. Mohammed, "A novel approach to network intrusion detection system using deep learning for SDN: futuristic approach," Comput. Sci. Inf. Technol., vol. 12, no. 11, pp. 1–6, Nov. 2022.

[6] H. Hassan, E. Hemdan, W. El-Shafai, M. Shokair, and F. El-Samie, "A survey on SDN-based intrusion detection systems on the Internet of Things: concepts, issues, and blockchain applications," Res. Square, pp. 1–20, Jul. 2021.

[7] A. Jose, L. Nair, and V. Paul, "Designing intrusion detection system in software defined networks using hybrid GWO-AE-RF model," Indian J. Comput. Sci. Eng., vol. 13, no. 6, pp. 1951–1966, Dec. 2022.

[8] M. Khairi, S. Ariffin, N. Latiff, K. Yusof, M. Hassan, F. AL-Dhief, … and M. Hamzah, "Detection and classification of conflict flows in SDN using machine learning algorithms," IEEE Access, vol. 9, pp. 76024–76037, Jun. 2021.

[9] G. Kumar and H. Alqahtani, "Machine learning techniques for intrusion detection systems in SDN: recent advances, challenges and future directions," Comput. Modeling Eng. Sci., vol. 134, no. 1, pp. 89–119, Jan. 2023.

[10] A. Mudgerikar, E. Bertino, J. Lobo, and D. Verma, "A security-constrained reinforcement learning framework for software defined networks," in Proc. IEEE ICC, pp. 1–6, Jun. 2021.

[11] A. Mudgerikar, E. Bertino, J. Lobo, and D. Verma, "A security-constrained reinforcement learning framework for software defined networks," in Proc. IEEE ICC, pp. 1–6, Jun. 2021.

[12] F. Noorbehbahani, A. Fanian, R. Mousavi, and H. Hasannejad, "An incremental intrusion detection system using a new semi-supervised stream classification method," Int. J. Commun. Syst., vol. 30, no. 4, pp. 1–15, Apr. 2015.

[13] A. Rayhan, S. Islam, S. Shatabda, A. Islam, and M. Robin, "Intrusion detection system in software-defined networks using machine learning and deep learning techniques: a comprehensive survey," TechRxiv Preprint, pp. 1–25, Sept. 2022.

[14] N. Shone, T. Ngoc, V. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, Mar. 2018.

[15] T. Tang, L. Mhamdi, D. McLernon, S. Zaidi, M. Ghogho, and F. Moussa, "DeepIDS: deep learning approach for intrusion detection in software defined networking," Electronics, vol. 9, no. 9, pp. 1533, Sept. 2020.

[16] A. ALHILO, "Enhancing SDN anomaly detection: a hybrid deep learning model with SCA-TSO optimization," Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 5, pp. 1–10, May 2024.

[17] A. Abdulboriy, "An incremental majority voting approach for intrusion detection system based on machine learning," IEEE Access, vol. 12, pp. 18972–18986, Feb. 2024.

[18] A. Chetouane, "Risk-based intrusion detection system in software defined networking," Concurrency Comput. Pract. Exper., vol. 36, no. 9, pp. 1–15, May 2023.

[19] J. B. Christinal and A. A. Roseline, "Securing SDON with hybrid evolutionary intrusion detection system: an ensemble algorithm for feature selection and classification," Opt. Fiber Technol., vol. 93, pp. 104206, Jan. 2025.

[20] M. Elsayed, N. Le-Khac, M. Albahar, and A. Jurcut, "A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique," J. Netw. Comput. Appl., vol. 191, pp. 103160, Mar. 2021.

[21] H. Y. I. Khalid and N. B. I. Aldabagh, "A survey on the latest intrusion detection datasets for software defined networking environments," Eng. Technol. Appl. Sci. Res., vol. 14, no. 2, pp. 13190–13200, Apr. 2024.

[22] A. Mansoor, M. Anbar, A. A. Bahashwan, B. A. Alabsi, and S. D. A. Rihan, "Deep learning-based approach for detecting DDoS attack on

software-defined networking controller," Systems, vol. 11, no. 6, pp. 296, Jun. 2023.

[23] A. Almazyad, L. Halman, and A. Alsaeed, "Probe attack detection using an improved intrusion detection system," Comput., Mater. Continua, vol. 74, no. 3, pp. 4769–4784, Mar. 2023.

[24] Y. Xiao, X. Zhang, Z. Wang, and Y. Li, "[Title not available] (utilized InSDN dataset for feature selection and ML optimization in SDN IDS)," Unpublished, 2023.

[25] H.-M. Chuang, F. Liu, and C.-H. Tsai, "Early detection of abnormal attacks in software defined networking using machine learning approaches," Symmetry, vol. 14, no. 6, pp. 1178, Jun. 2022.

[26] O. Aouedi and K. Piamrat, "F-BIDS: federated blending based intrusion detection system for IoT/IIoT networks," Procedia Comput. Sci., vol. 213, pp. 275–284, Dec. 2023.

[27] H. Gâlmeanu and R. Andonie, "Concept drift adaptation with incremental–decremental SVM," Appl. Sci., vol. 11, no. 20, pp. 9644, Oct. 2021.

[28] H. Hassan, E. Hemdan, W. El-Shafai, M. Shokair, and F. El-Samie, "A survey on SDN-based intrusion detection systems on the Internet of Things: concepts, issues, and blockchain applications," Res. Square, pp. 1–20, Jul. 2021.

[29] S. A. Wadho, S. Ali, and A. A. A. Mohammed, "Secret sharing as a defense mechanism for ransomware in cloud storage systems," Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 10, pp. 1–10, Oct. 2024.

[30] R. Sebopelo, "An integrated framework for controllers placement and security in software-defined networks ecosystem," J. Inf. Syst. Informatics, vol. 6, no. 1, pp. 464–494, Feb. 2024.

[31] T. Wang, Q. Lv, B. Hu, and D. Sun, "A few-shot class-incremental learning approach for intrusion detection," in Proc. IEEE ICCCN, pp. 1–8, Aug. 2021.

[32] Malik, R. Q., et al. "A Novel Taneja Distance-based Classifier with PSO-Optimized Feature Selection for Efficient 5G Network Slicing." *International Journal of Intelligent Engineering & Systems* 18.6 (2025).

[33] Alsharfa, Raya Majid, et al. "Cellular-D2D resource allocation algorithm based on user fairness." *Electronics* 9.3 (2020): 386.

[34] Ali, Sijjad, et al. "CLDM-MMNNs: Cross-layer defense mechanisms through multi-modal neural networks fusion for end-to-end cybersecurity—Issues, challenges, and future directions." *Information Fusion* (2025): 103222.

[35] Alsharfa, Raya Majid. "Design and Performance of MC-CDMA Transceiver based LDWT–OFDM in Flat Fading Channel." *IOP Conference Series: Materials Science and Engineering*. Vol. 518. No. 5. IOP Publishing, 2019.

[36] Wadho, Shuaib Ahmed, et al. "Ransomware detection techniques using machine learning methods." *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*. IEEE, 2024.