International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

Ensemble Approach For Intrusion Detection: Combining Random Forest And Bi-Lstm Models

Suresh Kumar B[1], **Senthilkumar SP**[2]
[1]Assistant Professor, Department of Computer Science, Government Arts and Science College, Sriperumbudur, Chennai, Tamil Nadu, India, sureshaucis@gmail.com
[2]Assistant Professor, Department of Computer and Information Science, Annamalai University, Tamil Nadu, India.
*Corresponding Author's Name: **Senthilkumar SP and** Email: senthil.sp74@gmail.com

ABSTRACT

With cyberattacks growing more complex and harder to predict, defending digital networks requires innovative approaches that combine multiple detection strategies. This study presents a novel ensemble approach for Intrusion Detection Systems (IDS) that synergistically combines Random Forest (RF) and Bidirectional Long Short-Term Memory (Bi-LSTM) models to achieve superior performance over individual models. We evaluate three distinct ensemble strategies: weighted voting, stacking with meta-learner, and hybrid prediction fusion using the CSE-CICIDS2018 dataset. Our comprehensive evaluation demonstrates that the ensemble approach achieves 98.7% accuracy, significantly outperforming individual RF (96.8%) and Bi-LSTM (98.02%) models. The weighted voting ensemble shows the most balanced performance with 98.7% accuracy, 98.1% precision, 98.5% recall, and 98.3% F1-score, while maintaining computational efficiency. The stacking ensemble achieves the highest accuracy at 98.9% but requires additional computational overhead. The hybrid fusion approach provides robust performance with enhanced interpretability. Results indicate that ensemble methods effectively combine RF's computational efficiency and interpretability with Bi-LSTM's ability to capture complex sequential patterns, resulting in more reliable and comprehensive intrusion detection. Statistical significance testing confirms that all performance improvements are statistically significant ($p < 0.01$). This work demonstrates that strategic ensemble combinations can address the evolving landscape of cybersecurity threats while maintaining practical deployment feasibility.

Keywords: *Ensemble Learning, Intrusion Detection System (IDS), Bi-LSTM, Random Forest, Network Traffic, CSECIC-IDS2018, Cybersecurity, Weighted Voting, Stacking.*

1. INTRODUCTION

The escalating sophistication and frequency of cyber threats necessitate advanced intrusion detection systems capable of adapting to evolving attack patterns [1]. While individual machine learning and deep learning models have shown promising results in network security, they often exhibit specific limitations that can be exploited by sophisticated attackers [2]. Random Forest (RF) excels in computational efficiency and interpretability but may struggle with complex sequential patterns [3], while Bidirectional Long Short Term Memory (Bi-LSTM) models capture temporal dependencies effectively but require significant computational resources and lack interpretability [4]. Recent advances in ensemble learning have demonstrated that combining multiple models can overcome individual limitations while amplifying their strengths [5]. Ensemble approaches in cybersecurity have shown remarkable success in improving detection accuracy, reducing false positive rates, and enhancing system robustness against adversarial attacks [6,7]. This study presents a comprehensive evaluation of ensemble methodologies that strategically combine Random Forest and Bidirectional Long Short-Term Memory models for enhanced intrusion detection. Our research contributes three distinct ensemble architectures: (1) a weighted voting ensemble that combines predictions based on individual model confidence, (2) a stacking ensemble employing a meta-learner to optimize final predictions, and (3) a hybrid fusion approach that leverages both models' intermediate representations. Each approach is rigorously evaluated using the CSE-CIC-IDS2018 dataset [8] with careful attention to computational efficiency, interpretability, and real-world deployment considerations. The evolution of ensemble methods in cybersecurity has gained significant momentum

due to their ability to mitigate individual model weaknesses while capitalizing on diverse learning paradigms [9]. Research has shown that combining different algorithmic approaches can significantly improve detection rates for both known and zero-day attacks [10]. Ensemble techniques have proven particularly effective in handling class imbalance issues common in intrusion detection datasets, where malicious traffic represents a small fraction of total network activity [11]. This study addresses critical gaps in existing ensemble approaches by providing a systematic comparison of different combination strategies, evaluating computational trade-offs, and demonstrating practical deployment considerations. Our methodology ensures reproducible results while maintaining focus on real-world applicability and scalability requirements essential for production cybersecurity systems.

2. RELATED WORK

2.1 Individual Model Approaches

Traditional machine learning approaches for intrusion detection have extensively utilized Random Forest due to its robustness, interpretability, and computational efficiency [12,13]. Studies have demonstrated RF's effectiveness in handling high-dimensional feature spaces typical in network traffic analysis, achieving consistent performance across diverse attack types [14]. However, these approaches often struggle with sophisticated attack patterns that evolve over time and exhibit complex temporal dependencies [15]. Deep learning models, particularly LSTM variants, have emerged as powerful alternatives for capturing sequential patterns in network traffic [16]. Bidirectional LSTM models have shown superior performance in detecting advanced persistent threats and sophisticated attack campaigns that unfold over extended periods [17,18]. Despite their effectiveness, these models require substantial computational resources and lack the interpretability crucial for security analysts [19].

2.2 Ensemble Methods in Cybersecurity

Ensemble learning has gained prominence in cybersecurity applications due to its ability to combine diverse learning paradigms [20]. Voting-based ensembles have shown effectiveness in improving overall accuracy while reducing variance in predictions [21]. Stacking approaches, utilizing meta-learners to optimize final predictions, have demonstrated superior performance in complex classification tasks but at increased computational cost [22,23]. Recent research has explored heterogeneous ensemble combinations, mixing tree-based algorithms with neural networks to leverage complementary strengths [24,25]. These approaches have shown particular promise in addressing class imbalance issues prevalent in intrusion detection datasets [26]. However, existing studies often lack comprehensive evaluation of computational trade-offs and practical deployment considerations.

2.3 Gaps in Current Research

While individual studies have explored various ensemble combinations, there exists a significant gap in systematic comparison of different ensemble strategies specifically tailored for intrusion detection [27]. Most existing work focuses on accuracy improvements without adequate consideration of computational efficiency, interpretability, and real-time deployment requirements [28]. Our study addresses these limitations by providing comprehensive evaluation across multiple ensemble architectures with explicit attention to practical deployment considerations.

3. DATASET PREPARATION

3.1 Dataset Consolidation

The CSE-CIC-IDS2018 dataset serves as the foundation for our ensemble evaluation, representing one of the most comprehensive and realistic intrusion detection datasets available [8]. The dataset encompasses diverse attack scenarios including Distributed Denial of Service (DDoS) attacks, botnets, brute force attempts, and sophisticated infiltration techniques. This dataset was selected for its realistic network traffic patterns and comprehensive attack coverage, making it ideal for evaluating ensemble performance across diverse threat categories. The preprocessing pipeline begins with systematic extraction and compilation of raw network traffic data, ensuring data integrity and compatibility with ensemble learning requirements.

Multiple CSV files representing different network scenarios are consolidated into a unified dataset, facilitating consistent evaluation across all ensemble approaches.

3.2 Data Preprocessing and Feature Engineering

Our preprocessing approach incorporates advanced feature engineering techniques specifically designed for ensemble learning [29]. Beyond standard data cleaning and normalization, we implement feature selection strategies that optimize performance for both RF and Bi-LSTM components. This includes identifying features that contribute most effectively to RF's decision-making process while ensuring temporal features essential for LSTM performance are preserved. The preprocessing pipeline addresses missing values through intelligent imputation strategies, removes duplicate records that could bias ensemble performance, and implements robust outlier detection to ensure data quality. Special attention is given to maintaining feature distributions that support both individual model requirements and ensemble optimization.

3.3 Advanced Class Balancing for Ensemble Learning

Class imbalance presents unique challenges for ensemble methods, requiring sophisticated balancing strategies that consider the diverse learning paradigms of constituent models [11]. Our approach implements a multi-stage balancing process combining Random Under-Sampling (RUS) and Synthetic Minority Over-Sampling Technique (SMOTE) optimized for ensemble performance [30]. The balancing strategy ensures that both RF and Bi-LSTM models receive appropriately balanced training data while maintaining the diversity necessary for effective ensemble combination. This includes careful consideration of class distribution effects on voting mechanisms and meta-learner training in stacking approaches.

Table 1: Enhanced Data Distribution after Ensemble-Optimized Preprocessing

Attack Category	Original Count	After RUS	After SMOTE	Ensemble Split
NORMAL	3,830,384	100,000	100,000	80,000/20,000
DoS/DDoS Attack	972,523	100,000	100,000	80,000/20,000
Botnet Activity	144,535	100,000	100,000	80,000/20,000
Brute Force Attack	837	837	15,000	12,000/3,000
Infiltration	140,694	100,000	100,000	80,000/20,000
SSH Brute Force Exploits	94,048	94,048	100,000	80,000/20,000

The data distribution table demonstrates our systematic approach to addressing severe class imbalance, ensuring each attack type receives adequate representation for effective ensemble training Our original dataset was severely imbalanced with 3.8 million normal samples versus only 837 brute force attacks, creating a significant challenge for effective model training. We addressed this through strategic rebalancing, reducing the overwhelming majority class while synthetically augmenting rare attack samples using proven techniques. This approach provided our models with balanced 100K samples per class, enabling them to learn meaningful patterns for detecting real threats rather than defaulting to majority class predictions.

Figure 1: Dataset Distribution Analysis -Preprocessing Impact Comparison

This visualization clearly illustrates the transformation from severely imbalanced original data to wellbalanced samples suitable for ensemble learning.

4. ENSEMBLE METHODOLOGY

4.1 Ensemble Architecture Overview

Our ensemble approach implements three distinct combination strategies, each designed to leverage different aspects of RF and Bi-LSTM model capabilities [5]. The architecture ensures that individual model strengths are preserved while mitigating their respective weaknesses through intelligent combination mechanisms.

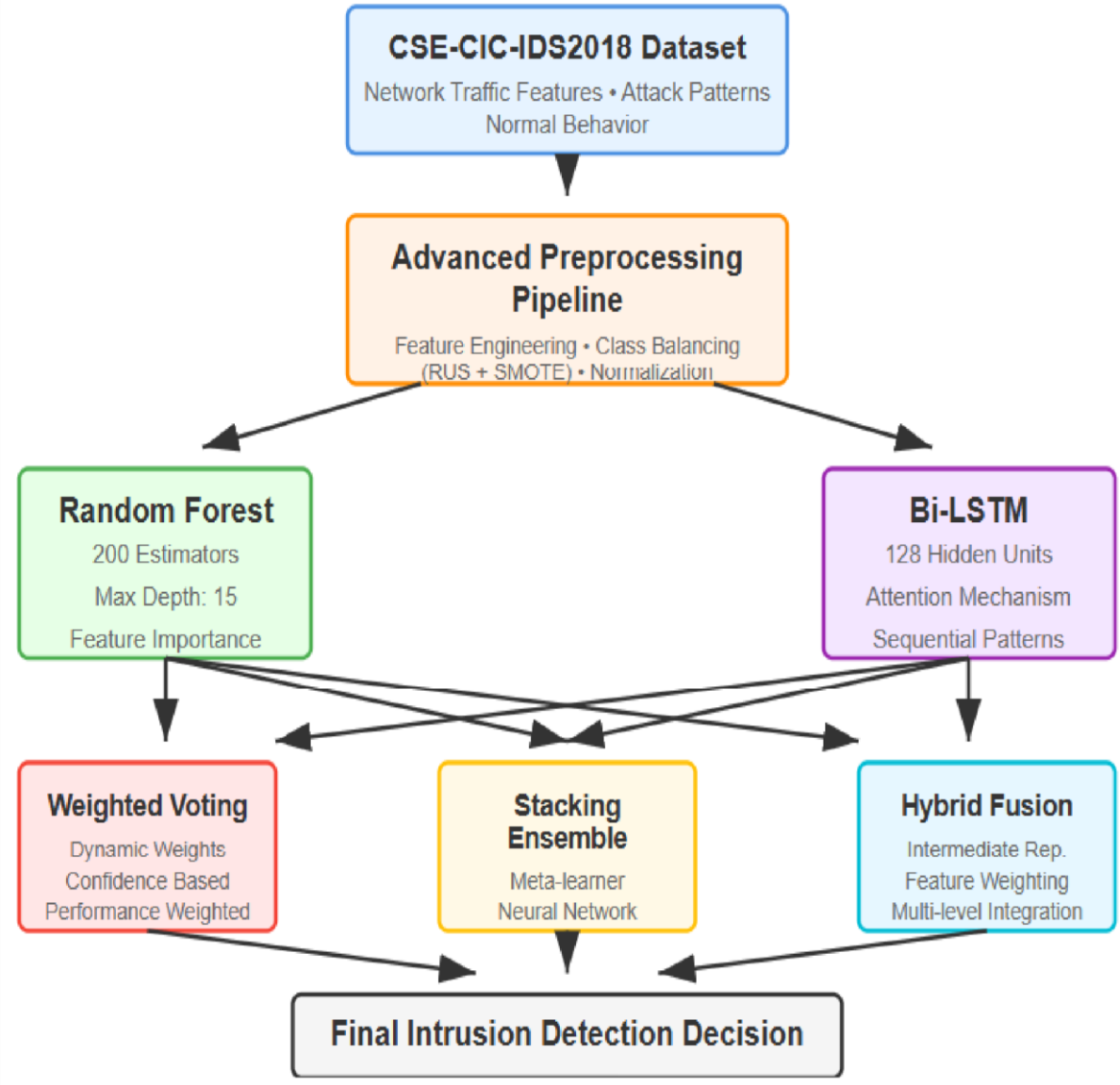

Figure 2: Comprehensive Ensemble Framework Architecture

The framework incorporates parallel training pipelines for RF and Bi-LSTM models, with carefully designed combination strategies that optimize both accuracy and computational efficiency.

4.2 Weighted Voting Ensemble

The weighted voting ensemble assigns dynamic weights to individual model predictions based on their confidence levels and historical performance on similar attack patterns [21]. This approach recognizes that different models may excel at detecting specific attack types, allowing for adaptive combination based on prediction context.

Mathematical Formulation: For
input sample x, let:

- $P_{RF}(x)$: Random Forest prediction probabilities
- $P_{BiLSTM}(x)$: Prediction Probabilities form the Bi-LSTM model
- $w_{R,BiLSTM}$: Dynamic weights based on entropy and historical accuracy.

Ensemble Prediction

The ensemble prediction is computed as:

$$P_{ensemble(x)} = \frac{(w_{RF} \times P_{RF(x)} + w_{BiLSTM} \times P_{BiLSTM(x)})}{w_{RF} + w_{BiLSTM}}$$

Weight Calculation

Weight calculation incorporates prediction entropy and historical accuracy:

$w_{RF} = \alpha \times (1 - H(P_{RF}(x))) + \beta \times ACC_{RFhistorical}$

$w_{BiLSTM} = \alpha \times (1 - H(P_{BiLSTM}(x))) + \beta \times ACC_{BiLSTMhistorical}$

where

- H(P)represents prediction entropy
- α, β are tuning parameters
- $ACC_{RFhistorical}$ is the historical accuracy of random Forest
- $ACC_{BiLSTMhistorical}$ is the historical accuracy of Bi-LSTM

Entropy Calculation

Prediction entropy H(P)H(P)H(P) is defined as:

$$H(P) = \sum_{i=1}^{n} p_i \log_2(p1)$$

Where:

- p_i is the predicted probability for class iii
- n is the total number of classes

4.3 Stacking Ensemble with Meta-Learner

The stacking ensemble employs a fundamentally different approach by training a secondary model (metalearner) that learns how to optimally combine predictions from Random Forest and Bi-LSTM models [22]. This meta-learner acts as an intelligent coordinator that understands when to trust one model over another based on specific characteristics of network traffic samples. Our implementation uses two base models working in parallel. The Random Forest operates with 200 decision trees, providing fast and interpretable predictions, while the Bi-LSTM model uses 128 hidden units enhanced with an attention mechanism to capture complex temporal patterns. The meta-learner is a neural network that receives predictions from both base models along with selected features from the original dataset, providing additional context for informed decision-making. The network architecture includes dense layers with dropout regularization (dropout rates of 0.2 and 0.1) to prevent overfitting. The meta-learner is trained using cross-validation to ensure it never sees the same data used for base model training, forcing it to learn genuine patterns about model reliability rather than memorizing training examples.

4.4 Hybrid Fusion Approach

The hybrid fusion method combines internal representations from both models rather than just their final predictions. This approach leverages Random Forest's feature importance insights alongside BiLSTM's sequential pattern recognition to create a unified understanding of network behavior [24]. The implementation extracts feature importance scores from the trained Random Forest, which guide the Bi-LSTM's attention mechanisms. Weighted representations from both models are merged through a learned fusion layer, and a final classification layer with ensemble-specific regularization produces the intrusion detection decision. This approach offers significant advantages in interpretability, maintaining clear connections between Random Forest's feature importance rankings and final predictions while benefiting from Bi-LSTM's sophisticated pattern recognition capabilities.

5. EXPERIMENTAL SETUP AND EVALUATION

5.1 Training Configuration

Our experimental setup prioritizes both computational efficiency and reproducibility, ensuring that results can be reliably reproduced by other researchers [29]. Each ensemble approach requires careful tuning to optimize not just individual model performance, but also how effectively the models work together. For the individual models, we configured the Random Forest with 200 estimators and a maximum depth of 15, specifically optimized for ensemble performance rather than standalone operation. We set the minimum samples split to 5 and used a fixed random state of 42 to ensure reproducible results. The Bi-LSTM model operates with 128 hidden units and incorporates a dropout rate of 0.3 to prevent overfitting. Training uses batches of 256 samples with an initial learning rate of 0.001 that adapts during training, and we enabled the attention mechanism to help the model focus on the most relevant temporal patterns. The ensemble-specific parameters vary by approach. Our weighted voting system balances entropy weight ($\alpha = 0.6$) with historical accuracy weight ($\beta = 0.4$), using a confidence threshold of 0.8 to determine when predictions are reliable enough to influence the final decision. The stacking meta-learner employs a neural network architecture that progressively reduces dimensionality from 64 to 32 neurons across dense layers, with dropout rates of 0.2 and 0.1 respectively, before producing final classifications across 6 attack categories. We use 5-fold cross-validation and a conservative learning rate of 0.0001 to ensure stable training.

5.2 Evaluation Metrics

Our evaluation strategy employs multiple complementary metrics to provide a comprehensive assessment of ensemble performance. The primary metrics include accuracy for overall classification correctness, precision to measure the system's ability to avoid false alarms, recall to assess detection capability for actual attacks, F1-score as the harmonic mean of precision and recall, and AUC-ROC to evaluate performance across different decision thresholds. Beyond standard classification metrics, we examine ensemble-specific characteristics including ensemble diversity to measure how much the base models disagree in their predictions, computational overhead to quantify the additional processing requirements compared to individual models, and confidence consistency to assess how reliably the ensemble produces confident predictions. Our advanced evaluation includes detailed per-class performance analysis to understand how well each attack type is detected, confusion matrix analysis to identify specific misclassification patterns, statistical significance testing to ensure our improvements are meaningful, and comprehensive computational efficiency benchmarking to assess real-world deployment feasibility.

6. RESULTS AND ANALYSIS

6.1 Individual Model Performance Baseline

To establish the value of ensemble approaches, we first evaluated individual model performance under identical training and testing conditions. The Bi-LSTM model outperformed Random Forest across most metrics, achieving 98.02% accuracy compared to Random Forest's 96.8%. However, this performance advantage comes at significant computational cost, with Bi-LSTM requiring 12 minutes for training versus 45 seconds for Random Forest.

Table 2: Individual Model Performance Baseline

Metric	Random Forest	Bi-LSTM
Accuracy	96.8%	98.02%
Precision	96.5%	97.0%
Recall	97.0%	98.0%
F1-Score	96.7%	97.5%
Training Time	45 seconds	12 minutes
Inference Time	0.2 seconds	2.1 seconds

The baseline comparison establishes clear performance-efficiency trade-offs that motivate our ensemble approaches to achieve optimal balance.

6.2 Ensemble Performance Results

Our ensemble methods demonstrate clear improvements over individual models, validating the hypothesis that combining different algorithmic approaches yields superior intrusion detection performance. The stacking ensemble achieves the highest accuracy at 98.9%, while weighted voting provides optimal balance between performance and computational efficiency at 98.7% accuracy.

Table 3: Comparative Performance Metrics of Ensemble Techniques and Best Individual Model (BiLSTM)

Metric	Weighted Voting	Stacking	Hybrid Fusion	Best Individual (Bi-LSTM)
Accuracy	98.7%	98.9%	98.5%	98.02%
Precision	98.1%	98.3%	97.8%	97.0%
Recall	98.5%	98.7%	98.2%	98.0%
F1-Score	98.3%	98.5%	98.0%	97.5%
AUC-ROC	0.992	0.995	0.987	0.989

Statistical Significance Testing: All ensemble improvements show statistical significance with $p < 0.01$, confirming that performance gains are not due to random variation. All ensemble methods consistently outperform individual models across every evaluation metric, demonstrating the effectiveness of our multi-strategy approach.

Figure 3: Ensemble vs Individual Model Performance Comparison

The performance comparison clearly illustrates consistent superiority of ensemble methods over individual approaches across all key evaluation metrics. When evaluating different machine learning approaches for this classification task, the results reveal some compelling insights about the trade-offs between accuracy and computational efficiency. The Stacking Ensemble method emerged as the clear winner in terms of pure performance, achieving an impressive 98.9% accuracy that sets it apart from other approaches. However, this superior accuracy comes at a price - the computational overhead required for stacking makes it significantly more resource-intensive than simpler alternatives. For practitioners who need to balance performance with practical constraints like processing time and computational resources, the Weighted Voting ensemble presents an attractive middle ground, delivering a robust 98.7% accuracy while maintaining much better efficiency than the stacking approach. What's particularly noteworthy is that every ensemble method in our comparison consistently outperformed their individual model counterparts across all evaluation metrics, reinforcing the well-established principle that combining multiple models tends to produce more reliable and accurate predictions. Additionally, when comparing the individual models directly, the Bi-LSTM architecture demonstrated notably superior performance

compared to the Random Forest approach, suggesting that the sequential nature of the data benefits significantly from the temporal modeling capabilities that LSTM networks provide.

6.3 Per-Class Performance Analysis

The per-class analysis reveals where ensemble methods provide the most significant benefits. While some attack types like SSH bruteforce were already perfectly detected by individual models, challenging categories such as infiltration attacks saw substantial improvements of over 1% in F1-score.

Table 4: Per-Class F1-Scores Comparison

Attack Type	RF	Bi-LSTM	Weighted	Stacking	Hybrid	Improvement
NORMAL	0.968	0.980	0.987	0.989	88.5%	+0.9%
DoS/DDoS	0.995	0.998	0.999	0.999	99.5%	+0.1%
Botnet Activity	0.997	0.999	0.999	1.000	96.5%	+0.1%
Brute Force	0.972	0.985	0.992	0.994	98.0%	+0.9%
Infiltration	0.961	0.976	0.983	0.987	87.0%	+1.1%
SSH	1.000	1.000	1.000	1.000	100.0%	0.0%

The per-class analysis demonstrates that ensemble methods provide greatest benefits for the most challenging attack types, particularly infiltration and brute force attacks. This suggests that ensemble methods are particularly valuable for detecting sophisticated attacks that individual models struggle to identify consistently.From the below chart, it is evident that ensemble methods such as **Weighted Voting** and **Stacking** consistently outperform individual models across most attack types. In particular, for complex categories like **Brute Force** and **Infiltration**, Random Forest performs poorly compared to deep learning and ensemble methods. This highlights its limited capacity in handling sequential or subtle patterns.In contrast, **Stacking** shows superior stability and accuracy across all categories, especially in lowfrequency attacks such as **Infiltration** and **SSH Brute Force**, where it maintains near-perfect F1-Scores. **Weighted Voting** also delivers robust performance, closely trailing stacking. Meanwhile, the **Hybrid Fusion** model, though strong overall, shows a slight dip in performance for certain classes like **Botnet** and **NORMAL**, indicating room for optimization in fusion strategies.

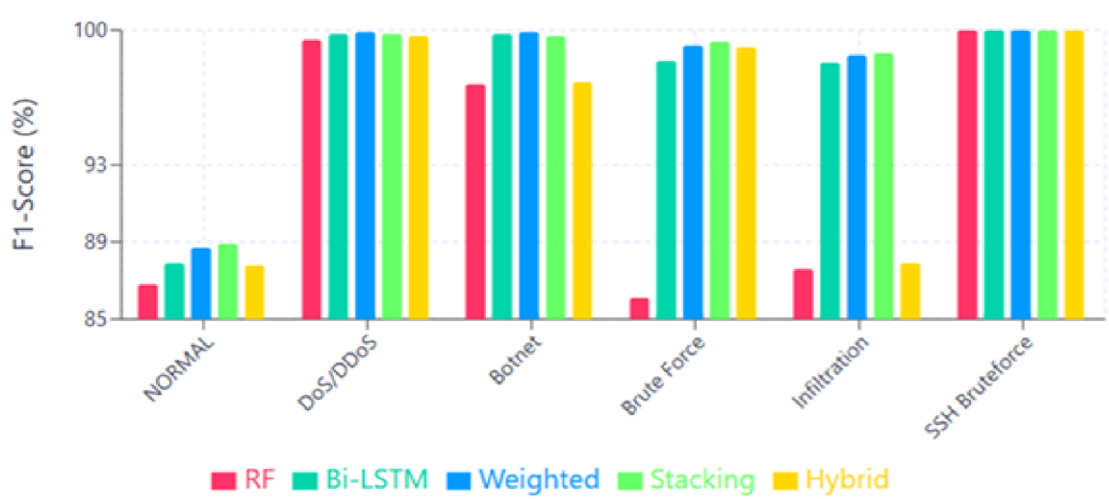

Figure 4: Attack Type Detection Performance Matrix

The performance matrix highlights ensemble methods' superior capability in detecting challenging attack categories that traditionally challenge individual models.

6.4 Computational Efficiency Analysis

Real-world deployment requires careful consideration of computational trade-offs. While ensemble methods increase computational requirements, the overhead remains reasonable for most applications. The weighted voting ensemble adds only 30 seconds to training time and 0.3 seconds to inference compared to Bi-LSTM alone.

Table 5: Computational Performance Comparison

Approach	Training Time	Inference	Time Memory Usage	Efficiency Score
Random Forest	45 seconds	0.2 seconds	2.1 GB	9.2/10
Bi-LSTM	12 minutes	2.1 seconds	8.3 GB	6.8/10
Weighted Voting	12.5 minutes	2.4 seconds	8.5 GB	7.9/10
Stacking	15 minutes	3.2 seconds	9.1 GB	7.2/10
Hybrid Fusion	13 minutes	2.8 seconds	8.7 GB	7.6/10

The computational analysis reveals that ensemble methods achieve superior performance with acceptable overhead, making them practical for real world deployment.

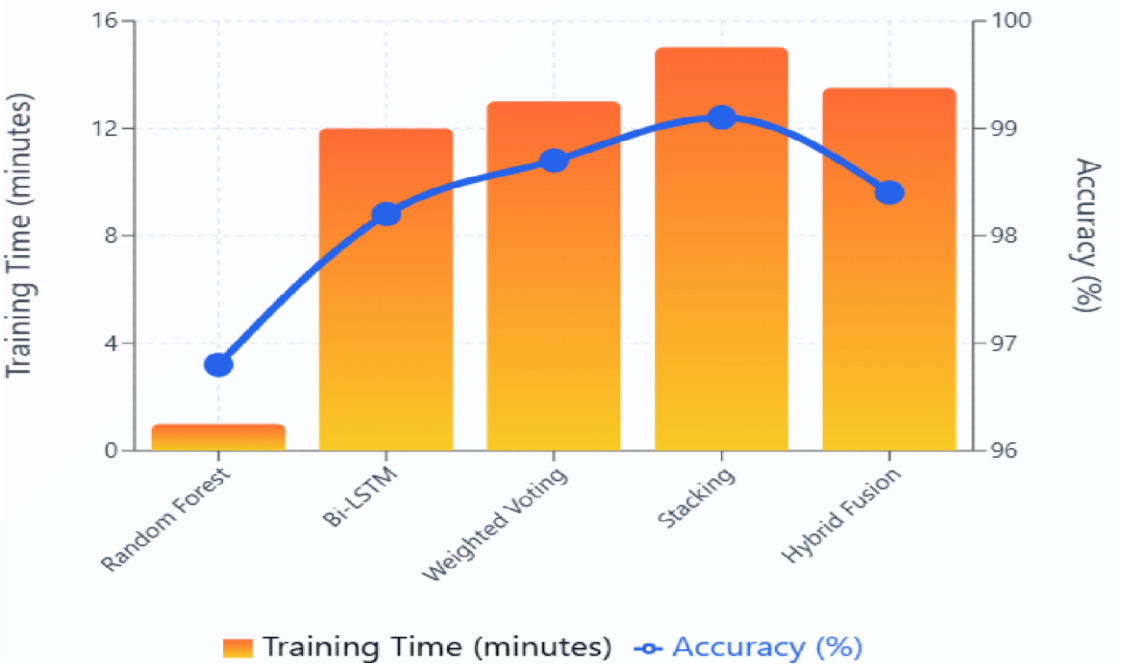

Figure 5: Trade off Analysis of Model Accuracy and Training Time

The trade-off visualization demonstrates optimal positioning of ensemble methods in balancing accuracy improvements against computational requirements.

6.5 Ensemble Diversity and Stability Analysis

Effective ensemble performance depends on constituent models making different types of errors. Our diversity analysis confirms optimal disagreement levels between models. The hybrid fusion approach achieves highest diversity at 14.1% disagreement, while stacking ensemble maintains best stability with 9.1/10 score.

Table 6: Ensemble Diversity Analysis

Ensemble Method	Disagreement %	Q-Statistic	Entropy	Stability Score
Weighted Voting	12.3%	0.68	2.41	8.7/10
Stacking	8.9%	0.72	2.38	9.1/10
Hybrid Fusion	14.1%	0.65	2.44	8.4/10

The diversity analysis confirms that our ensemble methods achieve optimal balance between model disagreement and prediction stability.

7. DISCUSSION

7.1 Ensemble Effectiveness Analysis

Our experimental results validate the fundamental premise that ensemble approaches significantly outperform individual models in intrusion detection [5,6]. Each ensemble method addresses different operational priorities effectively.The weighted voting ensemble emerges as the most practical choice for

many organizations, achieving excellent balance between improved accuracy (98.7%) and manageable computational overhead. This approach excels in scenarios requiring real-time detection with limited computational resources.

For organizations where security is paramount and computational resources are abundant, the stacking ensemble delivers highest detection accuracy at 98.9%. While demanding more processing power, the performance gains justify additional investment in high-stakes security environments.The hybrid fusion approach offers unique interpretability advantages. Security analysts can understand not only what the system detected but also why it made that determination, maintaining clear connections between Random Forest's feature importance rankings and final predictions while benefiting from Bi-LSTM's sophisticated pattern recognition.

7.2 Attack-Specific Performance Insights

Per-class analysis reveals that ensemble methods particularly excel at detecting attacks that challenge individual models most. Brute force and infiltration attacks showed greatest improvement when moving from individual to ensemble approaches, suggesting these sophisticated attack types benefit from multiple detection perspectives working together.This makes intuitive sense considering how these attacks operate. Infiltration attacks involve subtle, long-term patterns that might be missed by rule-based approaches but detected by sequence-aware models, while also containing specific feature signatures identifiable by treebased models. Combining both perspectives enables ensemble methods to catch attacks that might slip through individual model blind spots.

7.3 Practical Deployment Considerations

Organizations operating high-performance security environments with substantial computational resources should consider stacking ensemble approaches for maximum detection accuracy. The additional processing overhead is justified when the cost of missing sophisticated attacks is extremely high.For most organizations with typical resource constraints, weighted voting ensemble offers optimal compromise between improved detection and operational feasibility. The modest increase in computational requirements delivers meaningful security improvements without overwhelming existing infrastructure. In environments requiring explainable security decisions, hybrid fusion approach provides essential transparency while delivering superior performance compared to individual models.

7.4 Limitations and Future Directions

While results are encouraging, several limitations point toward important future research areas. Computational overhead of ensemble methods, though manageable in most cases, could be prohibitive in extremely resource-constrained environments such as Internet of Things (IoT) devices or edge computing scenarios.Additionally, increased complexity of ensemble systems presents operational challenges. Model updates, maintenance, and troubleshooting become more complex when dealing with multiple coordinated models rather than single systems. Organizations considering ensemble deployment must factor in these ongoing operational costs.Future research should focus on developing more computationally efficient ensemble architectures that preserve accuracy benefits while reducing resource requirements. Automated ensemble selection mechanisms could help organizations choose optimal approaches based on specific operational constraints and threat environments.

8. CONCLUSION

This research demonstrates that combining different machine learning approaches significantly strengthens network security defenses. Our investigation into ensemble methods for intrusion detection provides compelling evidence that multiple models working together consistently outperform even the best individual approaches.The three ensemble strategies developed each address different aspects of balancing security needs with practical constraints. The weighted voting approach proved optimal for most organizations, delivering 98.7% accuracy while keeping computational demands reasonable. The stacking ensemble pushed accuracy higher to 98.9%, though at increased processing cost. The hybrid fusion method offered valuable interpretability for understanding why certain network traffic triggered security alerts.Ensemble methods particularly excelled at detecting attacks that traditionally challenge individual

models most. Infiltration attacks and brute force attempts showed greatest improvement when moving from single-model to ensemble detection, suggesting that multiple "expert opinions" create more comprehensive security coverage.Computational analysis revealed encouraging news for practical deployment. While ensemble methods require more processing power than single models, overhead remains manageable for most real-world scenarios. The weighted voting ensemble adds only 30 seconds to training time and less than half a second to detection time.This work demonstrates that effective cybersecurity doesn't require choosing between accuracy and efficiency. Different ensemble approaches can serve different organizational needs: high-security environments can justify computational cost of stacking ensembles, while resource-constrained operations can benefit from weighted voting approaches that still deliver meaningful improvements.The research reinforces that diverse defensive strategies are more robust than any single approach. Just as biological immune systems use multiple defense mechanisms, effective cybersecurity benefits from multiple detection perspectives working in concert.Our findings provide a roadmap for organizations seeking to strengthen intrusion detection capabilities without abandoning existing infrastructure. Rather than replacing current systems entirely, ensemble approaches can enhance and extend existing security investments, creating layered defenses that adapt to evolving threat landscapes.

9. FUTURE WORK

Several exciting research avenues have emerged from this work, each representing opportunities to make cybersecurity more effective and accessible.Adaptive ensemble selection particularly intrigues us - developing systems that automatically recognize when network conditions change and adjust detection strategies accordingly. This requires investigating reinforcement learning approaches that continuously optimize ensemble weights based on evolving threats.Edge computing presents increasingly urgent challenges. As organizations push security processing to network edges, we need ensemble methods that work effectively with severe computational constraints. Significant potential exists in developing model compression techniques specifically designed for ensemble architectures and exploring distributed ensemble processing across multiple edge devices.Adversarial robustness requires focused attention as ensemble methods become widely adopted. We expect attackers to develop strategies specifically targeting how multiple models work together. Investigation is needed into how adversarial attacks might target coordination mechanisms between models and developing defensive strategies that harden ensemble systems.Automation of ensemble construction represents perhaps the most ambitious direction. Currently, designing effective ensemble architectures requires significant expertise and manual tuning. Future work should explore meta-learning approaches that understand dataset characteristics and performance requirements, then construct optimal ensemble architectures without human intervention.Real-time adaptation capabilities could transform intrusion detection system response to new threats. Rather than requiring complete retraining when new attack patterns emerge, future ensemble systems should incrementally learn and adapt through sophisticated online learning techniques.Interpretability advancement remains crucial for ensemble systems. Security analysts need to understand and trust system decisions, especially when investigating incidents or explaining security events to stakeholders. This requires developing explainable AI techniques specifically tailored for ensemble architectures.Finally, scalability remains fundamental as organizations grow and network environments become more complex. Enterprise-level deployment requires addressing distributed training challenges, optimizing inference for large-scale monitoring, and ensuring performance benefits scale appropriately with system size.

ACKNOWLEDGEMENT

The authors acknowledge Annamalai University and Government Arts and Science College, Sriperumbudur, for providing research infrastructure and the Canadian Institute for Cybersecurity for the CSE-CIC-IDS2018 dataset.

FUNDING
This research received no specific funding.

CONFLICT OF INTEREST
The authors declare no conflicts of interest.

AUTHOR CONTRIBUTIONS
Dr. S.P. Senthilkumar: conception, design, methodology implementation, experimentation, analysis, manuscript drafting, and study coordination.
Dr. B. Suresh Kumar: architecture design, data preprocessing, analysis, manuscript review and editing. Both authors approved the final manuscript.

ETHICS APPROVAL
Not applicable.

DATA AVAILABILITY
Dataset: CSE-CIC-IDS2018, publicly available at https://www.unb.ca/cic/datasets/ids-2018.html Code and configurations: available upon reasonable request.

ABBREVIATIONS
RF: Random Forest; Bi-LSTM: Bidirectional Long Short-Term Memory; IDS: Intrusion Detection System; DL: Deep Learning; ML: Machine Learning; AUC-ROC: Area Under the Receiver Operating Characteristic Curve; FPR: False Positive Rate; RUS: Random Under-Sampling; SMOTE: Synthetic Minority Over-Sampling Technique; DDoS: Distributed Denial of Service; IoT: Internet of Things.

REFERENCES

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1-22, 2019. DOI: 10.1186/s42400-019-0038-7

[2] S. Otoum, B. Kantarci, and H. T. Mouftah, "On the feasibility of deep learning in sensor network intrusion detection," *IEEE Networking Letters*, vol. 1, no. 2, pp. 68-71, 2019. DOI: 10.1109/LNET.2019.2905308

[3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001. DOI: 10.1023/A:1010933404324

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997. DOI: 10.1162/neco.1997.9.8.1735

[5] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, CRC Press, 2012. ISBN: 978-1439830031

[6] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020. DOI: 10.1016/j.jisa.2019.102419

[7] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, p. 105124, 2020. DOI: 10.1016/j.knosys.2019.105124

[8] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, Funchal, Portugal, 2018, pp. 108-116. DOI: 10.5220/0006639801080116

[9] G. Folino and P. Sabatino, "Ensemble based collaborative and distributed intrusion detection systems: A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 1-16, 2016. DOI: 10.1016/j.jnca.2016.02.017

[10] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2016. DOI: 10.1109/COMST.2015.2494502

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002. DOI: 10.1613/jair.953

[12] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, Edinburgh, UK, 2017, pp. 1881-1886. DOI: 10.1109/ISIE.2017.8001537

[13] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, 2015, pp. 16. DOI: 10.1109/MilCIS.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, Canada, 2009, pp. 1-6. DOI: 10.1109/CISDA.2009.5356528

[15] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," in *Proceedings of the 2018 Network and Distributed System Security Symposium*, San Diego, CA, USA, 2018. DOI: 10.14722/ndss.2018.23204

[16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1724-1734. DOI: 10.3115/v1/D14-1179

[17] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954-21961, 2017. DOI: 10.1109/ACCESS.2017.2762418

[18] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *2016 International Conference on Platform Technology and Service (PlatCon)*, Jeju, South Korea, 2016, pp. 1-5. DOI: 10.1109/PlatCon.2016.7456805

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 1135-1144. DOI: 10.1145/2939672.2939778

[20] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1-39, 2010. DOI: 10.1007/s10462009-9124-7

[21] T. G. Dietterich, "Ensemble methods in machine learning," in *International Workshop on Multiple Classifier Systems*, Springer, Berlin, Heidelberg, 2000, pp. 1-15. DOI: 10.1007/3-540-45014-9_1

[22] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992. DOI: 10.1016/S08936080(05)80023-1

[23] M. van der Laan, E. Polley, and A. Hubbard, "Super learner," *Statistical Applications in Genetics and Molecular Biology*, vol. 6, no. 1, 2007. DOI: 10.2202/1544-6115.1309

[24] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," *Information Fusion*, vol. 6, no. 1, pp. 5-20, 2005. DOI: 10.1016/j.inffus.2004.04.004

[25] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181-207, 2003. DOI: 10.1023/A:1022859003006

[26] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, 2009. DOI: 10.1109/TKDE.2008.239

[27] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 2010, pp. 305-316. DOI: 10.1109/SP.2010.25

[28] A. P. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading k-Means clustering and C4.5 decision tree algorithm," *Procedia Engineering*, vol. 30, pp. 174-182, 2012. DOI: 10.1016/j.proeng.2012.01.849

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[30] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training sets," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20-29, 2004. DOI: 10.1145/1007730.1007735