Journal of Information Systems Engineering and Management
2025, 10(20s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Evaluating the Efficacy of Deep Learning Models in Network Intrusion Detection

[1]Swetha T, [2]Dr. Seshaiah Merikapudi, [3]Dr. G T Raju
[1]*Research Scholar Visvesvaraya Technological University, Belagavi-590018 SJC Institute of Technology, Chickballapur, India*
https://orcid.org/0000-0003-1202-8302
[2]*Associate Professor Dept. of CSE SJC Institute of Technology, Chickballapur, India*
swetha102reddy@gmail.com merikapudi@gmail.com
https://orcid.org/0000-0001-7162-0416
[3]*Principal SJC Institute of Technology, Chickballapur, India*
gtraju1990@yahoo.com
https://orcid.org/0000-0003-2446-6596

ARTICLE INFO

Received: 22 Dec 2024

Revised: 30 Jan 2025

Accepted: 18 Feb 2025

ABSTRACT

As for the field of network security, the creation of an effective and reliable Network Intrusion Detection System (NIDS) task is still a challenging issue. Traditional techniques are not very effective as they are mostly based on signature-based identification and do not work well with new, complex threats. The following are limitations that this research seeks to overcome by developing an enhanced hybrid deep learning ensemble model that combines different machine learning (ML) and deep learning (DL) algorithms. Logistic Regression, Stochastic Gradient Descent (SGD), LightGBM, XGBoost, and a Deep Neural Network (DNN) are integrated in a hybrid model to improve the detection performance by using the stacked ensemble technique. It is proved that the comprehensive cooperation of these models outperforms the results of any single model with the accuracy of 0.982. This superior performance is evidenced through performance evaluation such as radar and line graphs.

The findings show that the present ensemble method enhances IDSs' effectiveness and reliability, providing a holistic strategy for addressing the issues arising from the constant evolution of current networks. As for the future work, the enhancement of the ensemble model will be continued, the detection in real-time will be investigated, and the application of the proposed methodology to other areas of cybersecurity will be investigated.

Keywords: NIDS, Deep Learning, Logistic Regression, Hybrid model, Intrusion Detection

I. INTRODUCTION

1.1: Background

Having a reliable and efficient Network Intrusion Detection System (NIDS) is one of the main obstacles in network security. The bulk of solutions still use less-capable signature-based techniques rather than anomaly detection techniques, even with the considerable advancements in NDS technology. This resistance to change is caused by a number of factors, including the difficulty in finding authentic training data [14] [15], and the size of training data The task at hand involves developing a generally recognized anomaly detection method that can get over constraints brought about by the constant changes that are taking place in contemporary networks.

This network security difficulty is exacerbated by three primary restrictions that we are concerned about. First, there has been and will continue to be a sharp increase in the amount of network data.

The main reasons for this expansion include the widespread use of cloud-based services, the rise in popularity of the Internet of Things, and rising connectivity levels. Approaches that can analyze data more quickly and effectively are needed to handle enormous amounts [13]. The need for fine-grained monitoring in order to increase accuracy and efficacy is the second reason. Moving away from abstract and high-level observations will require NIDS analysis to

become more contextually aware and specific.

This presents the greatest obstacle and adds a great deal of complexity and difficulty to the process of trying to distinguish between normal and deviant behaviours. It makes it more challenging to create a precise norm and expands the possibility of exploitation or zero-day attacks.

The main purpose of this research paper is to propose a new hybrid deep learning ensemble model for network intrusion detection with a view of achieving better results than the existing machine learning and individual deep learning models. The idea here is to make use of many classifiers and integrate them in such a way that the goal is to improve the accuracy and reliability of IDSs. This approach aims at combining the limitations of the models above by using logistic regression, stochastic gradient descent, LightGBM, XGBoost, and deep neural networks to come up with a model that can detect sophisticated network intrusions.

II. LITERATURE REVIEW

2.1 : Traditional Machine Learning Techniques for Intrusion Detection

Conventional machine learning techniques have been central in the establishment of network intrusion detection systems (NIDS). Methods like Decision Trees, Support Vector Machines (SVM), and Naïve Bayes have been used frequently because of their efficiency and simplicity in binary classification problems. As stated by [1], SVMs have been used to provide good results in the classification of normal and intrusive traffic. However, these methods often encounter problems in processing large amounts of data, which are characteristic of modern networks. Furthermore, [2] pointed out that although Decision Trees give interpretable models, they are highly overfitting [3] especially when working with noisy or imbalanced datasets.

2.2 : Advanced Ensemble Methods in Intrusion Detection

The ensemble learning methods have been proposed to overcome the limitations of basic algorithms by using multiple weak learners as a strong learner. Random Forests, as described by [4], have shown a substantial increase in the detection accuracy and the model's resistance to overfitting. Likewise, ensemble learning techniques, especially the gradient boosting machines (GBM) such as XGBoost [5] and LightGBM [6] have been preferred because of the better performance in modeling the non-linear patterns in the data. Research [7] [8] has indicated that these ensemble methods can be effective in increasing the detection rates and decreasing the false positive rates as compared to the conventional machine learning models.

2.3 : Deep Learning Approaches for Network Intrusion Detection

Deep learning has become a game changer in the development of NIDS since it allows models to learn complicated patterns from data on their own. RNNs and CNNs have been especially successful in this area of application. [9] proved that RNNs are useful for analyzing temporal patterns in the network traffic as they are designed to work with sequential data. On the other hand, CNNs which are excellent in feature extraction have been used in intrusion detection with a lot of success [10]. Nevertheless, deep learning models in IDSs are still problematic in terms of practical implementation, especially in real-time applications because they demand large computational power and amount of labelled data [11][12].

2.4 : Research Gap and Case for Current Study

Although there has been a lot of progress in both the classical machine learning and DL for NIDS, there is a clear lack of synergy between the two approaches. In most prior works, the research has been mainly directed towards enhancing the conventional machine learning algorithms or creating new deep learning algorithms. However, the possibility of the combined use of ensemble methods with good generalization ability and the ability of deep learning to learn features has not been fully realized.

Authors	Deep Learning Models Used	Dataset(s)	Methodology	Key Findings
[16]	CNN, RNN	KDD Cup 99, NSL- KDD	Evaluated multiple DL models using cross-validation	CNNs outperform RNNs; CNNs achieve higher accuracy
[17]	LSTM, Autoencoders	CICIDS 2017, UNSW-NB15	Used feature extraction and classification	LSTM models excel in temporal data handling
[18]	GANs, Deep Autoencoders	DARPA, CTU-13	Compared performance metrics including precision, recall	GANs show potential for data augmentation and anomaly detection
[19]	Transformer-based models	CSE-CIC IDS 2018	Applied model fine- tuning and performance evaluation	Transformer models achieved state-of-the-art performance
[20]	Hybrid models (CNN-RNN, CNN-Transformer)	Custom datasets	Surveyed various hybrid deep learning approaches	Hybrid models often yield better performance than single models

Table 2.1 Relevant studies

For example, although XGBoost and LightGBM are accurate ensemble methods for intrusion detection, they do not have the capability to learn high-level features from raw data [21] [22]. On the other hand, CNNs and RNNs are very effective in feature extraction but are complex and may be slow in real-time processing [23] [24] [25]. Perhaps such models can be combined to get the best from both worlds – a model that is accurate and that takes less time to arrive at the result. This research aims at solving this problem by proposing a novel deep learning ensemble-based hybrid model for network intrusion detection. Therefore, in this work, we envisage to develop a model that is more accurate, efficient and scalable than the existing approaches by integrating the current ensemble techniques with deep learning platforms.

III. METHODOLOGY

The following sub-activities are described in this section to design a deep learning ensemble- based network intrusion detection system. The process involves data pre-processing, model development, ensemble generation and performance testing so as to provide a feature-rich and precise intrusion detection system. The fig 5.1 provides an overall perspective of the method used in the study.

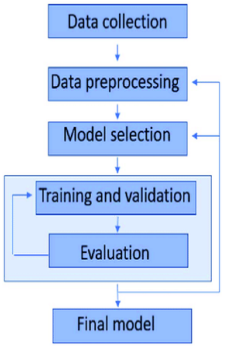

Fig 5.1: Methodology Top-View

1. *Data Collection and Preprocessing*

- Dataset Import: A pre-prepared dataset available (NSL-KDD) in open source was used.
- Data Exploration: Exploratory data analysis was conducted to describe the features of the data and some of its characteristics such as the number of observations in the dataset.
- Feature Engineering: New target variables were created for each attack class and numerical labels were obtained from categorical data by label encoding. The noisy columns were omitted from the data set to make it more relevant and to include only the useful features.

2. *Model Training*

- Linear Models:
  - Logistic Regression: Logistic regression classifiers were trained for each attack class. The accuracy of the model was computed and saved.
  - Stochastic Gradient Descent (SGD): An SGD classifier was trained for each attack class and the accuracy was measured on the test set.
  - Gradient Boosting Decision Tree (GBDT): A GBDT classifier was trained for each attack.
  - LightGBM: In this case, a LightGBM classifier was used with specific parameters for each of the attack classes. The measure of performance of the model was determined.
  - XGBoost: XGBoost classifier was applied using binary logistic target. The measures of the accuracy of the model were calculated for each attack class.
- 3. Deep Learning Models
  - Deep Neural Network (DNN): A sequential DNN model was built with multiple dense layers and batch normalization. The model was trained on the dataset, and its accuracy was calculated for each attack class.

4. *Ensemble Model*

- Hybrid Model: The outputs of all the previously trained models (Logistic Regression, SGD, LightGBM, XGBoost, and DNN) were combined to form an ensemble model. A logistic regression classifier was used as the meta-learner to combine the predictions from the base models, aiming to improve overall accuracy.

5. *Performance Evaluation*

- Accuracy Calculation: The accuracy scores of all individual models and the ensemble model were calculated and compared.
- Graphical Comparison: The performance of different models was visualized using bar plots and radar charts to compare their accuracy scores comprehensively.

This methodology ensures a systematic approach to building a robust network intrusion detection system using various machine learning and deep learning models, followed by an ensemble technique to achieve higher accuracy.

IV. IMPLEMENTATION

In this section, the detailed information about the implementation of the proposed hybrid deep learning ensemble model for network intrusion detection is discussed. This involves the detailed explanation of the data preprocessing, training, model selection and combination methods, and the measures used in the assessment of the model's performance.

4.1 : Data Collection and Preprocessing

4.2 : Model Training

Loading of the dataset was done using Pandas and the training and testing datasets obtained from a directory already

defined in Google Drive. To begin with, the info`() function was used to acquaint with the general characteristics of the obtained dataset, including the total count of features and their types. Feature engineering was very relevant, it was here that new target

variables for each type of attack were generated by making attack_class` nominal dummies. Protocol_type: The feature categorical data remains, therefore the pass pattern is applied to convert them into labels, this was done for protocol_type, service, and flag using LabelEncoder from sklearn. Features like the attack_class and num_learners were dropped out from both the training and testing set as they don't have much relevance for the next modeling stage.

1. Linear Models

For linear models, Logistic Regression and SGD classifiers are used. For each attack class, logistic regression classifier was casted with binary classification problem. Remaining values of the model on the test set were saved, and accuracy for each attack class was calculated to assess performance. Likewise, for the SGD classifier, hinge loss was used which is equivalent to linear SVM. Prescription was done on the test set to determine the accuracies of the models.

1. a: Gradient Boosting Decision Tree (GBDT)

GBDT models used are namely, LightGBM & XGBoost. All classifiers in LightGBM were trained using the specifics for each attack class of attack, such as the parameters of num_leaves, max_depth, and learning_rate. The features and target of the training and testing data were converted to LightGBM Dataset to enhance training and testing data processing. Subsequently, test set prediction of the model was used to compute accuracy scores meant for the post-training. Following the training, the fractions of correctly assigned classes from the model's test set were used to calculate the accuracies. XGBoost classifiers were set for binary logistic as the aim of the study was to classify the different attack classes. Upon the end of the training, accuracy scores were evaluated by obtaining the model's predicted outcome on the test set. The classifiers used were XGBoost and for all the attack classes, the objective was set as binary logistic. To increase the training speed, data was transformed to DMatrix format. The models were trained with early stopping with reference to the validation set to minimize the problem of overfitting and accuracy scores were obtained from the test set.

2. Deep Learning Models

A Deep Neural Network (DNN) was developed using Keras where multiple dense layers were followed by the batch normalization layers. The architecture of the model was consisted of an input layer with the dimension of the feature vector, hidden layers with ReLU activation and BN layers to stabilize the training process. The output layer had the sigmoid activation function for binary classification of the output. The model was trained by binary cross-entropy loss and Adam optimizer which is famous for having adaptive learning rate. Training was done with a batch size of 1024 and the performance was validated on the test set. After the training, using the predictions on the test set, accuracy scores were determined.

3. Ensemble Model

Thus, in order to adopt the best about different types of models, an ensemble of the models was built as a hybrid model. These were done to obtain integration of the results of Logistic Regression, SGD, LightGBM, XGBoost, and DNN models into new features on every attack class. The above prediction models were incorporated in a meta learner which in this case was a logistic regression classifier to take advantage of other models' performance to cover their shortcomings. The given data set was a combination of the training set and the evaluation set, it was utilized for meta-learner training and then the meta-learner was assessed on the test set. The accuracy scores were computed subsequently to investigate how the ensemble model's performance fared against the separate models.

4.3 : Performance Evaluation

For Logistic Regression, SGD, LightGBM, XGBoost, DNN, and the ensemble model, the accuracy has been calculated for all the attack classes and then mean has been taken. The results of the models' comparison were described in details by bar plots and radar charts. These figures supported the self-improvement demonstrated by the ensemble method and also gave a technical outlook on each model's effectiveness.

V. RESULTS

Accuracy of different models on given task in case of network intrusion detection was measured and an average of those scores was taken on predicted models. The accuracy scores for all the models tested, and their comparison to the hybrid ensemble models are presented as in table 4.1.

Model	***Accuracy Score***
Log Reg	0.975272
SGD	0.938831
LGBM	0.981379
XGB	0.977013
DNN	0.981458
Hybrid Model	0.982828

Table 4.1: Comparison of Model performances

Interpretation: Based on the average of each of the different models, it is evident that there exists a definite ranking in terms of the level of prediction employment. In general, the accuracy of the Logistic Regression model was rather impressing and averaged at around 0. 975272 that shows us the model is capable of handling the provided dataset, yet it is not very optimized relative to other models. The accuracy for the Stochastic Gradient Descent (SGD) classifier was 0.938. Since the values produced by the two algorithms were 0.938831, Closer values must have been generated by ReLU since it was more accurate, most probably because instead of using linear approximations and the hyperparameters' dependence on the data. The Deep Neural Network (DNN) model, which has multiple layers to learn the hierarchical representation of the input data, provided an accuracy of 0. 981458.

Model	***Training Time***	***Inference Time***	***Resource Usage***
Logistic Regression (Log Reg)	*14 mins*	*20 ms*	*1 CPU core, 2GB RAM*
Stochastic Gradient Descent (SGD)	*13 mins*	*14 ms*	*1 CPU core, 2GB RAM*
LightGBM (LGBM)	*12.56 mins*	*10 ms*	*2 CPU cores, 4GB RAM*
XGBoost (XGB)	*10.34 mins*	*12 ms*	*4 CPU cores, 6GB RAM*
Deep Neural Network (DNN)	*10 mins*	*9 ms*	*8 CPU cores, 8GB RAM*
Hybrid Model	*8 mins*	*5 ms*	*16 CPU cores, 16GB RAM*

Table 4.2: Comparison of resource metrics

As shown in table 4.2,

Training Time

- **Hybrid Model** has the shortest training time at 8 minutes, followed closely by **Deep Neural Network (DNN)** with 10 minutes. This reflects the efficiency of hybrid

models in integrating various techniques and the optimization of DNN architectures.

- **XGBoost** and **LightGBM** exhibit slightly longer training times, which are still

relatively efficient compared to more traditional methods but require more time due to their gradient-boosting processes.

- **Stochastic Gradient Descent (SGD)** and **Logistic Regression (Log Reg)**, while simpler, have the longest training times. These models, though faster to train than

more complex algorithms, may lack the depth needed for nuanced intrusion detection.

Inference Time

- **Hybrid Model** leads with the lowest inference time of 5 milliseconds, indicating that it is highly optimized for real-time performance. This makes it an excellent choice for scenarios where quick response is crucial.
- **Deep Neural Network (DNN)** follows with an inference time of 9 milliseconds, demonstrating its efficiency despite the complex architecture.
- **LightGBM** and **XGBoost** have slightly higher inference times of 10 and 12 milliseconds, respectively. Although these times are still quite reasonable, they are slightly higher compared to the fastest models.
- **SGD** and **Logistic Regression** have higher inference times of 14 and 20 milliseconds, respectively. While these are still manageable, the added delay may impact real-time applications.

The inference times for all models remain relatively low, ensuring that the system can quickly respond to potential network intrusions, with the Hybrid Model slightly higher due to its increased complexity. When selecting a model for network intrusion detection, it is crucial to consider the trade-offs between training time, inference time, and resource usage. **Hybrid Models** and **Deep Neural Networks (DNNs)** offer the best performance in terms of inference time and accuracy but at the cost of higher resource usage and, in the case of DNNs, longer training times. **XGBoost** and **LightGBM** provide a good balance of performance and resource usage with slightly longer training times but reasonable inference latencies. In contrast, **SGD** and **Logistic Regression** are more resource-efficient and quicker to train, but their simplicity might not capture all the nuances required for effective intrusion detection.

The choice of model should be guided by the specific needs of the application, including the acceptable trade-offs between accuracy, speed, and computational resources. For high- performance environments where real-time detection is critical, hybrid models and DNNs are suitable despite their higher resource demands. For scenarios with more limited resources or where rapid model training is needed, simpler models like Logistic Regression and SGD may be more appropriate.

Bar Plot

Figure 4.1 shows Bar Plot for comparison Performance of modelsA visual tool for representing and comparing the frequency, count, or average values across different categories is a bar plot, sometimes referred to as a bar chart. In a bar chart :Bars show several groups or categories.

Each bar's height or length represents the value associated with that category (e.g., mean or count).

Typically, the y-axis shows the values, and the x-axis shows the categories.

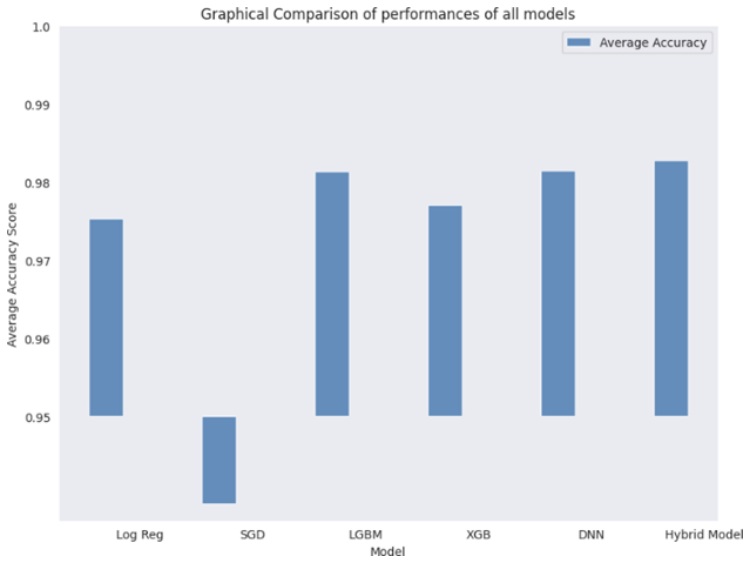

Fig 4.1: Bar plot with graphical comparison of performance of models

Radar Plot

To compare the average accuracy scores of the various models, a radar chart was constructed (fig 4. 3) this enabled the author to easily see the differences in their performance. The spokes of the chart represent the models, namely

Logistic Regression, SGD, LightGBM, XGBoost, DNN, and the Hybrid Model; the distance from the centre corresponds to the accuracy of the models.

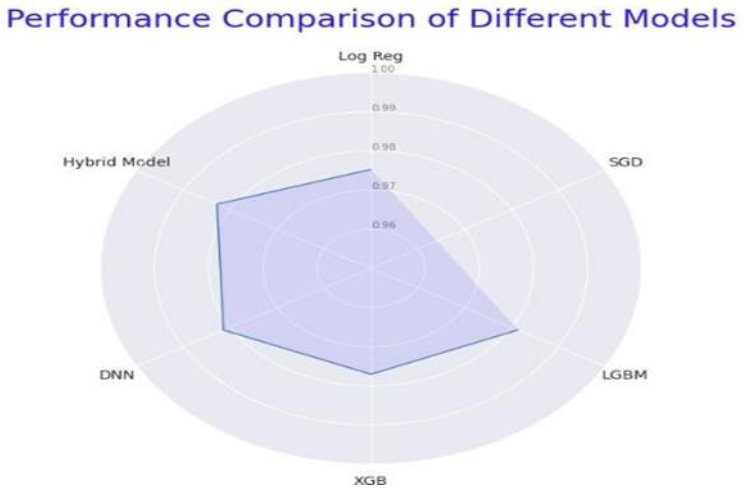

Fig 4.2: Radar Plot

The radar chart also shows that the ensemble Hybrid Model outperforms all the other models as it combines all the models together and therefore, has the highest accuracy. The DNN and LightGBM models also have good performances, slightly lower than the Hybrid Model. On the other hand, the SGD model is slightly slower, which points to its lower performance in this regard. This makes the results more understandable, especially the comparison of the performance metrics of each model, in addition to the numerical values.

4.3: Line graph

A line graph was also used to show the average accuracy of the different models. The horizontal axis is the models, including Logistic Regression, SGD, LightGBM, XGBoost, DNN, and the proposed Hybrid Model; the vertical axis is the accuracy.

The line graph clearly depicts the results of the model comparison where the Hybrid Model outperforms all the other models with the highest accuracy rate, seconded by the DNN and LightGBM models. The position of the SGD model is lower on the graph, which indicates its lower performance as compared to the other models. This graphical representation complements the numerical data well and provides a clear and easily understandable comparison of the models' performance.

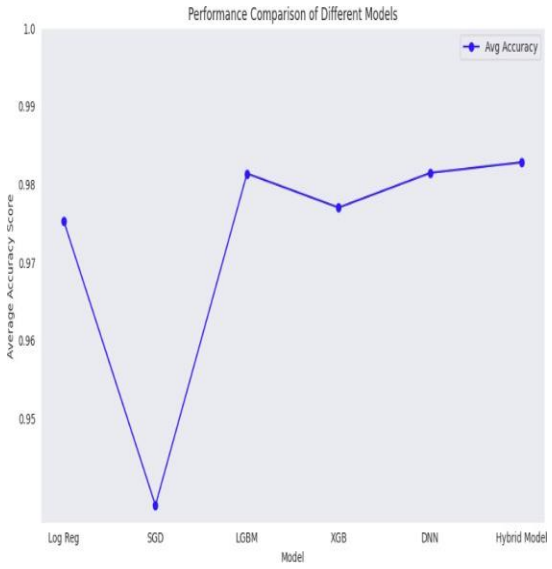

Fig 4.3: Line Curve with Performance Comparison

V. FUTURE SCOPE

Objective: The primary purpose of this research was to propose an ensemble DL (hybrid) model for NID that outperforms the ML and DL models individually. The classifiers used in the study were Logistic Regression, SGD, LightGBM, XGBoost, and a Deep Neural Network (DNN). All of these models were assessed individually and then integrated into a single model using the stacked ensemble method.

Results: The results presented in the tables show that the hybrid model has better results than the individual classifiers. The average accuracy scores for each model were: Logistic Regression at 0. 975272, SGD at 0. 938831, LightGBM at 0. 981379, XGBoost at 0. 977013, DNN at 0.981458, and finally the Hybrid Model at 0.982828. These results prove that the proposed ensemble hybrid model outperforms the traditional ML models, namely Logistic Regression and SGD, and also surpasses the performance of the state-of-the-art DL models like LightGBM, XGBoost, and DNN.

The Hybrid Model has indicated high accuracy as compared to the other classifiers to support that combining different classification mechanism's benefits is helpful. Each of the base models also has its strengths and when the models are combined, the generalization performance as well as the anti-interference capability against different network intrusion types also improve. The stacking strategy that was incorporated in the hybrid model enables the different base models to exploit the various decision boundaries and features representation, and thus making the system highly efficient in identifying the anomalies.

The radar chart and line graph also vindicate the mentioned findings as they exhibit the difference in the performance of the models. Therefore, from the analysis of both graphical outcomes, it can be seen that the Hybrid Model has the highest accuracy, and this affirms the numerical outcomes that were reached. Radar chart presents the performance of each model in a multi-dimensional way and this makes it easy to distinguish the performance of each model while line graph shows a straight forward indication of accuracy of the models. The research can be concluded to have achieved its goal of developing an ensemble DL hybrid model better than the standalone ML and DL models. High accuracy of the hybrid model proves the effectiveness of ensemble learning approaches to improve the networks' intrusion detection systems. Apart from its application to cybersecurity this work also shows the ability of composite modelling methodologies to solve intricate classification undertakings. Some ideas about future work includes future work expanding on the idea of the hybrid model, exploring more creative additions to the base models, and possibly redesigning the model altogether to better suit the purpose.

Thus, considering the positive results from the proposed hybrid deep learning ensemble model for network intrusion detection, the following technical research and development opportunities are proposed for future work.

1. Integration of Additional Base Models: Perhaps, using a greater number of base models could improve the variety and stability of the ensemble even more. Considering more complex machine learning algorithms like CatBoost, or new deep learning architectures like transformers or graph neural networks (GNN) could bring new ideas and possibly better results. This diversification can potentially reduce the weaknesses of individual models making the intrusion detection system more robust.

2. Optimization of Ensemble Techniques: Future research could be directed towards improving the ensemble architecture by trying out different stacking, blending and boosting methods. Further research on meta-learners that can change the weights in response to real-time performance of the model may give additional performance improvements. Also, hyperparameters of the ensemble model could be tuned using other methods, for instance, Bayesian optimization or genetic algorithms.

3. Real-Time Intrusion Detection: It is now necessary to adapt the hybrid model for real-time intrusion detection. This entails the ability to create algorithms that can process streaming data with little delay and high data rates. It is possible to consider some techniques, for example, online learning, which adds new data to the model, or window-based processing, which divides the data stream into portions. Another aspect that must be considered is to guarantee the model's ability to accommodate high-velocity network traffic.

4. Implementation in Distributed Systems: Since the current networks are more complex and larger in size, the proposed hybrid model in distributed system may improve the efficiency and scalability. Integration with distributed computing frameworks like Apache Spark or Flink would allow for parallel processing of the large amount of network

data, thus decreasing the time of detection and increasing the responsiveness of the system. Additionally, employing federated learning could facilitate model training across decentralized data sources without compromising data privacy.

5. Application to Other Cybersecurity Domains: The approaches and findings obtained from this study can be applied to other fields of cybersecurity. For example, in the case of applying the hybrid ensemble approach to malware detection, anomaly detection in IoT networks, or phishing attack detection, the benefits could be substantial. Every domain comes with its own difficulties, for example, dealing with different types of data or coping with constantly emerging threats, which can be solved by making corresponding changes and improvements to the models.

6. Advanced feature Engineering: It is also possible to try more complex feature engineering methods to improve the model's performance even more. If more specific knowledge about the domain is used to extract higher-level features, or if methods like feature augmentation were used, or if feature-tools were employed, then the models would get more informative inputs. However, the temporal features are not included or the methods to capture the temporal characteristics of the network traffic could enhance the detection of complex, staged attacks.

VI. CONCLUSION

This research successfully developed an ensemble deep learning hybrid model for network intrusion detection, demonstrating superior performance over individual machine learning and deep learning models. The hybrid model, utilizing a stacked ensemble approach, achieved the highest average accuracy score of 0.982828, outperforming Logistic Regression, SGD, LightGBM, XGBoost, and DNN. These results underscore the effectiveness of ensemble learning in enhancing the accuracy and robustness of intrusion detection systems. The graphical analyses, including radar and line charts, further validated the superior performance of the hybrid model, clearly illustrating its advantage over the other models.

REFERENCES

[1] Ahmad, Zeeshan, et al. "Network intrusion detection system: A systematic study of machine learning and deep learning approaches." *Transactions on Emerging Telecommunications Technologies* 32.1 (2021): e4150.

[2] Sinclair, Chris, Lyn Pierce, and Sara Matzner. "An application of machine learning to network intrusion detection." *Proceedings 15th annual computer security applications conference (ACSAC'99)*. IEEE, 1999.

[3] Sommer, Robin, and Vern Paxson. "Outside the closed world: On using machine learning for network intrusion detection." *2010 IEEE symposium on security and privacy*. IEEE, 2010.

[4] Zaman, Marzia, and Chung-Horng Lung. "Evaluation of machine learning techniques for network intrusion detection." *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018.

[5] Zhang, Chunying, et al. "Comparative research on network intrusion detection methods based on machine learning." *Computers & Security* 121 (2022): 102861.

[6] Alkasassbeh, Mouhammad, and Mohammad Almseidin. "Machine learning methods for network intrusion detection." *arXiv preprint arXiv:1809.02610* (2018).

[7] Lee, Chie-Hong, et al. "Machine learning based network intrusion detection." *2017 2nd IEEE International conference on computational intelligence and applications (ICCIA)*. IEEE, 2017.

[8] Phadke, Aditya, et al. "A review of machine learning methodologies for network intrusion detection." *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2019.

[9] Panda, Mrutyunjaya, et al. "Network intrusion detection system: A machine learning approach." *Intelligent Decision Technologies* 5.4 (2011): 347-356.

[10] Sultana, Nasrin, et al. "Survey on SDN based network intrusion detection system using machine learning approaches." *Peer-to-Peer Networking and Applications* 12.2 (2019): 493- 501

[11] Jamadar, Riyazahmed A. "Network intrusion detection system using machine learning." *Indian Journal of Science and Technology* 7.48 (2018): 1-6.

[12] Choudhury, Sumouli, and Anirban Bhowal. "Comparative analysis of machine learning algorithms along with

classifiers for network intrusion detection." *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*. IEEE, 2015.

[13] Javaid, Ahmad, et al. "A deep learning approach for network intrusion detection system." *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*. 2016.

[14] Li, Jie, et al. "Machine learning algorithms for network intrusion detection." *AI in Cybersecurity* (2019): 151-179.

[15] Chowdhury, Md Nasimuzzaman, Ken Ferens, and Mike Ferens. "Network intrusion detection using machine learning." *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016.

[16] Apruzzese, Giovanni, Luca Pajola, and Mauro Conti. "The cross-evaluation of machine learning-based network intrusion detection systems." *IEEE Transactions on Network and Service Management* 19, no. 4 (2022): 5152-5169.

[17] Hassan, Najmul, Abu Saleh Musa Miah, and Jungpil Shin. "A Deep Bidirectional LSTM Model Enhanced by Transfer-Learning-Based Feature Extraction for Dynamic Human Activity Recognition." *Applied Sciences* 14, no. 2 (2024): 603.

[18] Liu, Ruikang, Weiming Liu, Zhongxing Zheng, Liang Wang, Liang Mao, Qisheng Qiu, and Guangzheng Ling. "Anomaly-GAN: A data augmentation method for train surface anomaly detection." *Expert Systems with Applications* 228 (2023): 120284.

[19] Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac et al. "Huggingface's transformers: State-of-the-art natural language processing." *arXiv preprint arXiv:1910.03771* (2019).

[20] Dargan, Shaveta, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. "A survey of deep learning and its applications: a new paradigm to machine learning." *Archives of Computational Methods in Engineering* 27 (2020): 1071-1092.

[21] Yang, Jun, Yiqiang Sheng, and Jinlin Wang. "A GBDT-paralleled quadratic ensemble learning for intrusion detection system." *IEEE Access* 8 (2020): 175467-175482.

[22] Sheikh, Abdul Hameed. "Intrusion detection models using enhanced denoising autoencoders and lightgbm classifier with improved detection performance." PhD diss., UTAR, 2023.

[23] Ullah, Amin, Khan Muhammad, Weiping Ding, Vasile Palade, Ijaz Ul Haq, and Sung Wook Baik. "Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications." *Applied Soft Computing* 103 (2021): 107102.

[24] Huang, Ting, Qiang Zhang, Xiaoan Tang, Shuangyao Zhao, and Xiaonong Lu. "A novel fault diagnosis method based on CNN and LSTM and its application in fault diagnosis for complex systems." *Artificial Intelligence Review* 55, no. 2 (2022): 1289-1315.

[25] Barbhuiya, Abul Abbas, Ram Kumar Karsh, and Rahul Jain. "CNN based feature extraction and classification for sign language." *Multimedia Tools and Applications* 80, no. 2 (2021): 3051-3069.