Machine Learning-Based Network Intrusion Detection Using NetFlow Traffic Analysis with Linear Kernel PCA

Dr Suriya Prakash J[1]
Postdoc Researcher
Lincoln University College,
Malaysia.
dr.suriya.prakash.j.s.v@gmail.com
ORCID: 0000-0002-2041-3166

Mashael M. Khayyat[2]
[1]Lincoln University College, Malaysia
[1]pdf.mashael@lincoln.edu.my
[2] Department of IST, College of CSE,
[2]University of Jeddah, Jeddah 23218, Saudi Arabia
[2]mkhayyat@uj.edu.sa
ORCID: 0000-0003-3770-432X

Abstract— The growth of computer network has led to complex and voluminous network traffic. this increased exposure to cyber threats. old intrusion detection systems, which rely on signatures struggle to detect attacks. this study proposes a system using machine learning and netflow data to improve detection. the system uses 1,048,575 labeled network flow records with 53 features. these features describe communication sessions, the data has dimensions and potential redundancies. to simplify linear kernel principle component analysis(KPCA) was used. KPCA reduced complexity while keeping traffic patterns. this helped remove correlated attributes and improve model stability. several classification algorithms were tested. these included decision tree, k-nearest neighbors, gradient boosting, catboost, lightGBM, linear discriminant analysis and logistic regression. different train- test splits were used. The results show that the decision tree classifier works best. it achieved an accuracy of 95.18% with the 0.2 split the models performed well across data distributions. This confirms that the proposed framework is effective. the findings suggest that combining flow-based analysis with linear kernel PCA and machine learning can create an intrusion detection system. such a system is suitable, for large scale network environments. the system is efficient and scalable. it can handle network traffic, the approach can detect evolving attacks. it overcomes the limitations of intrusion detection systems.
Keywords : Intrusion Detection, Machine Learning, Algorithms, Attacks. Network

Introduction

Computer networks are a part of our daily lives now, we use them for banking and storing data in the cloud. even smart devices and big company systems rely on networks. for all these things to work smoothly we need to make sure network communication is secure and reliable. as more people use networks there is more traffic and it gets more complicated. this makes it easier for people to hack into systems. we see cyber attacks like blocking services stealing data and getting into systems without permission. traditional security methods like firewalls and intrusion detection systems are not very good at catching attacks. they mostly look for patterns they already know, this is why we need security systems that can learn and find unusual activity on their own machine learning is a way to make network security better. it looks at network traffic to see what is normal and what is not. it does not just rely on what it knows. one way to do this is by looking at traffic statistics and how devices talk to each other, this is called flow-based analysis. it uses something called netflow data. it is good because it can handle a lot of traffic and keep information safe. netflow data has a lot of information and some of it is not necessary, this can make it hard for computers to process. it can make the system not work as well to fix this we can use something called kernel principle component analysis. it helps make the data simpler while still keeping the information. in this study we made a network security system using machine learning and netflow data we used kernel principal component analysis to make the data smaller. we tried different classifiers to see how well they work. the goal is to make a system that can catch hackers and is fast and accurate. we wanted to be able to handle real world network traffic. we are using a dataset to train our system, we are trying ways to split the data to train and test the system. this helps us see how well it works and if it is strong. the idea is to make a system that's good at catching hackers and can handle a lot of network traffic. network security is very important, we need to make sure our systems are good at catching hackers machine learning and flow based analysis are ways to do this. we can use them to make our networks more secure network security systems like this can help keep our data safe.

Major Contributions

The main goal of this study is to develop a flow-based intrusion detection system, this study creates a network intrusion detection system that uses machine learning and flow level netflow data. the framework we propose looks at the behavior of traffic than the content of packets. this makes the framework work well for companies with large networks.
we use linear kernel PCA to optimize features in this study, to handle features that are complex and related we use linear kernel principle component analysis before classifying the network intrusion detection system. this step makes the feature space less complex, removes redundant information. it also makes the process more efficient without losing information about traffic.
we test the proposed approach using a dataset, the dataset has over one million labeled flow records with 53 time-based features of the network intrusion detection system. theproposed framework supports both complex classification scenarios. this shows that the network intrusion detection system framework is flexible.
we compare machine learning algorithms in this study, we compare several classifiers to evaluate the strength and generalization of the network intrusion detection system. this comparison gives us insights into the stability and performance of the model with data distributions of the network intrusion detection system.
The final goal of this study is to find a model, our results show that the decision tree classifier is very accurate with an accuracy of 95.18% after applying dimensionality reduction to the network intrusion detection system our findings confirm the combining linear kernel PCA with classifiers improves the detection performance of the network intrusion detection system while keeping it simple.
overall this work provides an approach to improve the network intrusion detection system through efficient feature reduction and systematic model evaluation of the network intrusion detection system.

LITERATURE SURVEY

TABLE I. COMPARATIVE ANALYSIS WITH EXISTING METHODS.

SI. No	Authors	Title	Algorithm	Accuracy (in %)
1	Suriya Prakash Jambunathan, Suguna Ramadass, Palanivel Rajan Selva Kumaran	Analyzing the Behavior of Multiple Dimensionality Reduction Algorithms To Obtain Better Accuracy using Benchmark KDD CUP Dataset	PCA + Logistic Regression (with & without K-Fold)	98% (without K-Fold), 99% (with K-Fold)
2	Suriya Prakash J,	Advancing Intrusion	KNN	97.68%

		Snehitha Narasani, Thangadurai N, U. Prakash, Kiran S	Detection Precision Through Analysis of Diverse Classification Algorithms		
	3	Suriya Prakash J, Chandra Haasitha Guntupalli, Snehitha Narasani, Srinidhi N N, Kiran S	Boosting Accuracy in Intrusion Detection Systems: A Comprehensive Examination of Dimensionality Reduction and Classification Methods	Logistic Regression (without LDA)	99.97%
	4	Suriya Prakash J, Srinidhi N, Latha A, Chaithra, Kiran S	Enhance Intrusion Detection by Analyzing the Behavior of Labeled and Unlabeled Classification to Obtain Better Accuracy	Labeled Statistical Classification (Mean & Minimum Variance Method)	93%
	5	Suriya Prakash J, Talluri Rashmika, Thangadurai N, U. Prakash, Kiran S	Elevating Intrusion Detection Precision with Multi-Classification Algorithm Analysis	CatBoost Classifier	99.73% (99.738% – 80:20 split)

“Enhancing Network Security Through Advanced Intrusion Detection: A Fusion of Dimensionality Reduction and Machine Learning Classification for Improved Accuracy” by Suriya Prakash J., Deeksha Gandhi P., Dilip Kumar, Vishwas M H, and Yash Shah focused on improving intrusion detection accuracy using dimensionality reduction (PCA) combined with multiple machine learning classifiers onthe Friday-WorkingHours-Afternoon-DDos.pcap_ISCX11 dataset. The authors

evaluated algorithms such as XGBoost, CatBoost, Random Forest, AdaBoost, Gradient Boosting, SVM variants, KNN, Logistic Regression, LDA, QDA, MLP, and LightGBM with different PCA components (5, 15, 25,
35, 45) and test sizes (0.2, 0.5, 0.7). Their results showed that CatBoost achieved the highest accuracy of 0.9998 (99.98%) with test size 0.2 and PCA 15, while XGBoost reached 0.99962 (99.962%) without PCA at test size 0.5, and overall the proposed work reported a maximum accuracy of 0.99974 (99.974%), outperforming several previous studies on the same domain.

“Enhancing the Security by Analyzing the Behaviour of Multiple Classification Algorithms with Dimensionality Reduction to Obtain Better Accuracy” by Suriya Prakash J., Chandra Haasitha Guntupalli, Siri Nandan Chilamkurthy, Gamidi Kowshik, and Abburi Alekhya investigated the performance of multiple machine learning classification algorithms combined with Principal Component Analysis (PCA) for intrusion detection using the CICIDS2017 dataset. The authors applied preprocessing techniques such as label encoding, feature scaling, and train–test splits (80:20, 60:40, 50:50, and 40:60), and evaluated Logistic Regression, SVM, Kernel SVM, KNN, Decision Tree, and Random Forest models with and without dimensionality reduction. Their experimental results showed that without PCA, the Decision Tree algorithm achieved the highest accuracy of 0.99984 (99.984%), while with PCA, Random Forest achieved 0.999743 (99.9743%) with 15 principal components, demonstrating that although dimensionality reduction improves efficiency, certain classifiers performed slightly better without PCA on the CICIDS2017 dataset.

Suriya Prakash J, Trishlaa S, Soniya R, V. Lakshmanan, and Kiran S examined ways to improve network intrusion detection accuracy using machine learning techniques. their study used the Kyoto 2015 december 31 dataset and involved preprocessing steps such as data cleaning, selecting 14 key features from 24, and normalizing the data. they tested several classifiers, including AdaBoost, KNN, XGBoost, CatBoost, Random Forest, Gradient Boosting, Decision Tree, Naïve Bayes, LSVM, LGBM, MLPC, LDA, and QDA,
across different train–test splits (80:20, 70:30, and 60:40) to compare performance. their results showed that AdaBoost, LGBM, XGBoost, Decision Tree, Gradient Boosting, and Random Forest achieved 100% accuracy on multiple splits, while KNN achieved 99.96% accuracy, Naïve Bayes achieved 99.99%, and other classifiers also reported above 95% accuracy, demonstrating the strong effectiveness of selected machine learning models for intrusion detection on the Kyoto dataset.

“Improving Intrusion Detection Precision via Multi Classification Algorithm Examination” by Suriya Prakash J., Varun V, S. Jagannathan, Vasanthakumar C, and Kiran S examined the performance of fifteen machine learning classification algorithms on the Kyoto 2015 May Day1 dataset for network intrusion detection. The authors applied preprocessing techniques such as label encoding, handling missing values, feature scaling, and different train–test splits (80:20, 70:30, and 60:40), and evaluated models including AdaBoost, CatBoost, KNN, LDA, Logistic Regression, LGBM, LSVM, MLPC, Naïve Bayes, QDA, XGBoost, Decision Tree, Gradient Boost, KSVM, and Random Forest. Their experimental results showed that LGBM, Decision Tree, Gradient Boost, and Random Forest achieved 100% accuracy across multiple train–test splits, while XGBoost achieved 99.9944%, KNN achieved 99.9153%, and Logistic Regression achieved 99.7634%, demonstrating that ensemble-based models provided the highest precision for intrusion detection on the Kyoto dataset.

Chandini Lekkalapudi, Niranjana Holla V P, S. Jagannathan, Vasanthakumar C, and Suriya Prakash J explored ways to strengthen intrusion detection systems by applying multiple machine learning algorithms to the Kyoto 20151208 dataset. in their study, several models—including AdaBoost, CatBoost, KNN, LDA, Logistic Regression, LGBM, LSVM, MLPC, Naïve Bayes, QDA, XGBoost, Decision Tree, Gradient Boosting, KSVM, and Random Forest—were tested without using dimensionality reduction. the experiments were carried out with different train–test splits (80:20, 70:30, and 60:40). among all the models evaluated, XGBoost delivered the best performance, achieving an accuracy of 99.85% with the 70:30 split. the authors concluded that XGBoost was the most effective approach for detecting intrusions in the Kyoto 20151208 dataset.

PROPOSED METHODOLOGY

The proposed method is used to build a network Intrusion detection system, this system uses data that shows how information flows and machine learning techniques. the process has steps: getting the data ready making the features better training the model and seeing how well it works.

Data analysis: this study uses a set of data with 1,048,575 labeled records of network intrusion detection system communication sessions. each record has 53 features like the number of packets the amount of bytes and protocol information. this data helps the network intrusion detection system tell the difference between malicious activities and it also helps with classifying different types of attacks for the network intrusion detection system before building the model the data is checked carefully for things like how the features ray spread out if there are too many or too few of some kinds of data and if there are any inconsistencies in the network intrusion detection system.

Data preprocessing: to make sure the network intrusion detection system model works well several things are done. these include handling missing values encoding attributes and scaling the features so they are all similar for the network intrusion detection system, scaling the features is important because the machine learning algorithms can be sensitive to how big or small the features for the network intrusion detection system.

Dimensionality reduction using Linear Kernel PCA: the data for the network intrusion detection system has a lot of features, which can make it take longer to process and might make the network intrusion detection system model not work as well, to fix this linear kernel principle component analysis is used to reduce the number of dimensions for the network intrusion detection system. this helped simplify the feature space for the network intrusion detection system and gets rid of any attributes for the network intrusion detection system.

Model classification: after the dimensions are reduced several machine learning classifiers are trained on the data set for the network intrusion detection system. the classifiers are evaluated using a method where the data is split into training and testing sets for the network intrusion detection system each classifier tries to learn the difference between malicious traffic patterns for the network intrusion detection system. then the classifiers are compared to see which one works best for the network intrusion detection system.

Performance Evaluation: the trained models for the network intrusion detection system are tested using metrics, how accurate they are how precise they are, how well they remember things and their F1-score. these metrics help figure out how well the network intrusion detection system can detect activities. the impact of using linear kernel PCA on the network intrusion detection system is also looked at the stability of the results, across different data splits is examined for the network intrusion detection system.

PROPOSED WORK

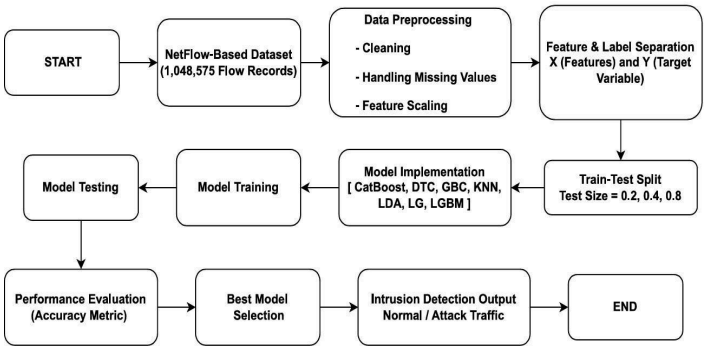

Fig-1: Flow diagram of proposed work

The work we are talking about is creating a network intrusion detection system, this system uses information about network flows and machine learning to do its job. we start with a set of data that has 1,048,575 network flow records. each record shows what happens when two things talk to each other on the network. these records also have information about when things happen. to get results from our network intrusion detection system we need to make sure the data is good, so we clean up the data set we fix any information and make sure all the data is on the same scale. our dataset has a lot of features, some of these features might not be might be similar to each other that is why we use a

technique called linear kernel principle component analysis for short on our network intrusion detection system. this helps us get rid of features we do not need it makes the data easier to understand, it helps our network intrusion detection system run faster and be more reliable it also keeps the information, about network traffic that our network intrusion detection system needs.

The transformed data is then divided into training and testing sets using different train–test splits (0.2, 0.4, and 0.8). Several machine learning classifiers are trained and evaluated to identify the most effective model for distinguishing between normal and malicious traffic. Based on performance comparison, the best-performing model is selected to generate the final intrusion detection output. Overall, the proposed work integrates dimensionality reduction and machine learning to create a scalable and reliable intrusion detection framework suitable for large- scale network environments.

RESULTS AND DISCUSSION

Seven machine learning models were evaluated using a netflow dataset consisting of 1,048,575 flow records. for a balanced assessment, the dataset was split into training and testing sets with test ratios of 0.2, 0.4, and 0.8. accuracy was used to measure how well each model classified network traffic is normal or attack.

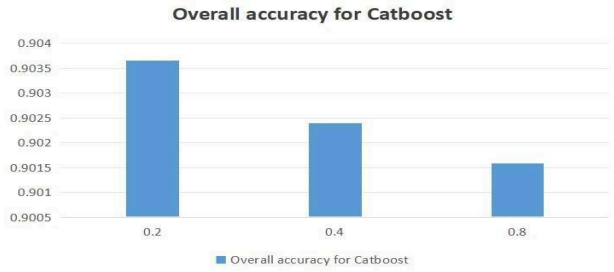

Fig-2: overall accuracy for Catboost

Fig-2 shows the overall accuracy of the CatBoost classifier for three train–test splits (0.2, 0.4, and 0.8). The model achieved 90.37% accuracy at the 0.2 split, which slightly decreased to 90.24% at 0.4 and 90.16% at 0.8.

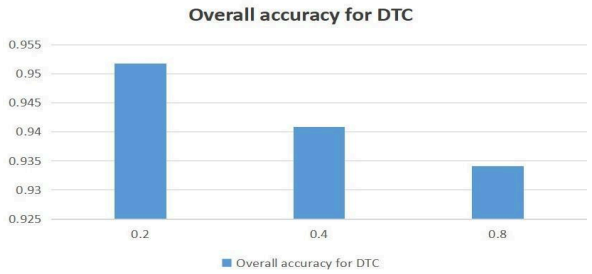

Fig-3: overall accuracy for DTC

Fig-3 shows how well the decision tree classifier works when we try out train and test sets. the decision tree classifier does its job when we use 0.2 of the data to test it and it gets 95.18% accuracy. the decision tree classify against 94.09% accuracy when we use 0.4 of the data to test it a 93.42% accuracy when we use 0.8 of the data to test it Even though the decision tree classifier does a little worse when we use data to test in the decision tree classifier always does a good job. this means the decision tree classifier is really good at finding intrusions and it works well every time we use it. the decision tree classifier is very stable.

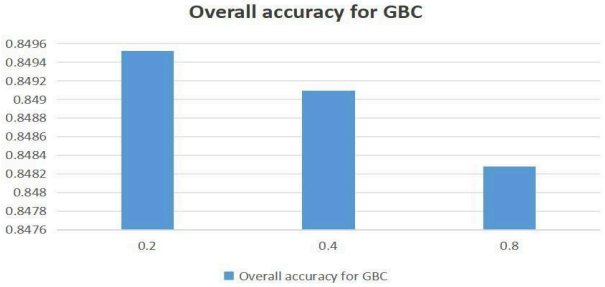

Fig-4: overall accuracy for GBC

Fig-4 shows how well the gradient boosting classifier did for the three train and test splits. the gradient based in classifier got it 84.95% accuracy of the time when the split was 0.2 it was accuracy 84.91% of the time when the split was 0.4 and 84.83% accuracy of the time when the split was

0.8. this tells us that the gradient boosting classifier works the same for different splits of the data. the gradient boosting classifier only gets a little worse when the test size gets bigger.

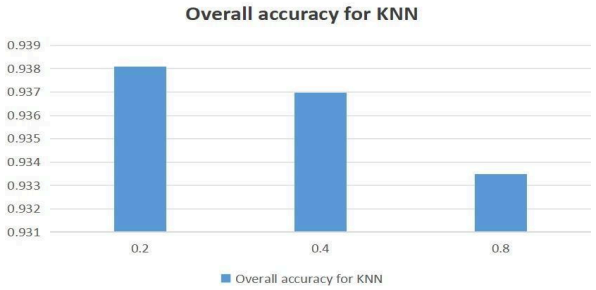

Fig-5: overall accuracy for KNN

Fig-5 shows how well the k-nearest neighbors model works it got 93.81% accuracy when using a test set 93.70% accuracy when using a medium test set in 93.35% accuracy when using a large test set the k-nearest neighbors model still does a job at finding network traffic patterns even when the test set gets bigger k-nearest neighbors model is really good, at this the accuracy does go down a bit as the test size increases.

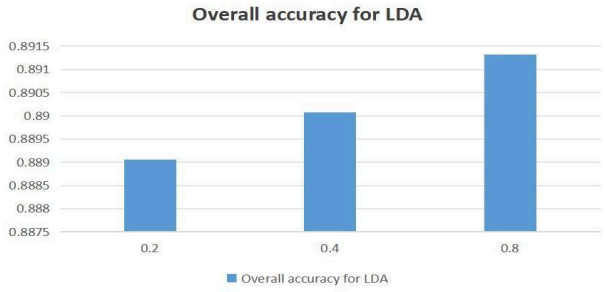

Fig-6: overall accuracy for LDA

Fig 6 shows how well the linear discriminant analysis model works for three train-test splits. the LDA model got 88.91% accuracy with a 0.2 split, 89.01% with a 0.4 split, and 89.13% with a 0.8 split. the LDA model's accuracy goes up a bit as the test size gets bigger.

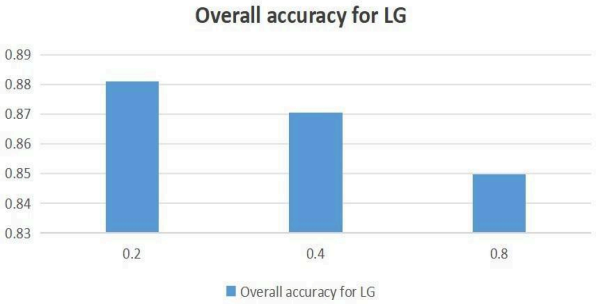

Fig-7: overall accuracy for LG

Fig-7 shows how well the logistic regression model does overall. the logistic regression model gets it right 88.10% of the time when the split 0.2. when the split is 0.4 the logistic regression model is right 87.04% of the time. when the split is 0.8 the logistic regression model got 84.99% of the time. this means the logistic regression model does not do well when the test size gets bigger.

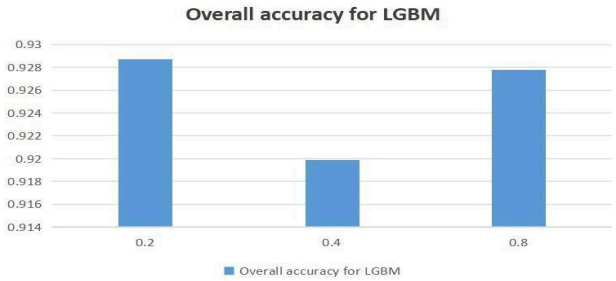

Fig-8: overall accuracy for LGBM

Fig-8 shows how well the lightGBM model works, it got 92.87% accuracy with the 0.2 split, 91.99% with a 0.4 split and finally 92.78% with the 0.8 split. the lightGBM model doesn't its performance does not change much across different data splits. it has performance with 92.87% 91.99% and 92.78% accuracy for lightGBM model. these results tell us that lightGBM model works well and is stable.

TABLE II.comparative accuracy results of implemented algorithms

Algorithms	0.2	0.4	0.8
Catboost	0.90366	0.902391	0.901584
DTC	0.951822	0.940851	0.93415
GBC	0.84952	0.849093	0.848279
KNN	0.938097	0.93696	0.933473
LDA	0.889059	0.890072	0.891312
LG	0.881031	0.870404	0.849872
LGBM	0.928724	0.919902	0.927757

Table 2 summarises the overall comparison of algorithms across the three train–test splits (0.2, 0.4, and 0.8) highlights clear performance differences among the models.

The decision tree achieve the accuracy in all splits it reached 95.18% at 0.2, the decision tree classifier reached 94.09% of 0.4 in the decision tree classifier reached 93.42% at 0.8. this makes the decision tree classify are the best performing model overall. the KNN model also did well it had strong and consistent results the KNN model maintained accuracy above 93% across all splits. the light GBM model followed it performed around 92 to 93% for lightGBM model showed behaviour even with some minor variation. the catboost model maintained accuracy above 90% for all splits this shows the catboost model had steady and reliable performance.

The linear discriminant analysis model achieved close to 89% it got a little better as the test size increased. the logistic regression model did not do well it had a noticeable decline in accuracy as the test size grew the logistic regression model dropped from 88.10% at 0.2 to 84.99% at

0.8. the gradient boosting classifier model had the performance it maintained accuracy around 84.8%b - 85%. overall the tree-based models, the decision tree classifier did better and were more stable than the linear and boosting approaches, in this research of detecting intrusions.

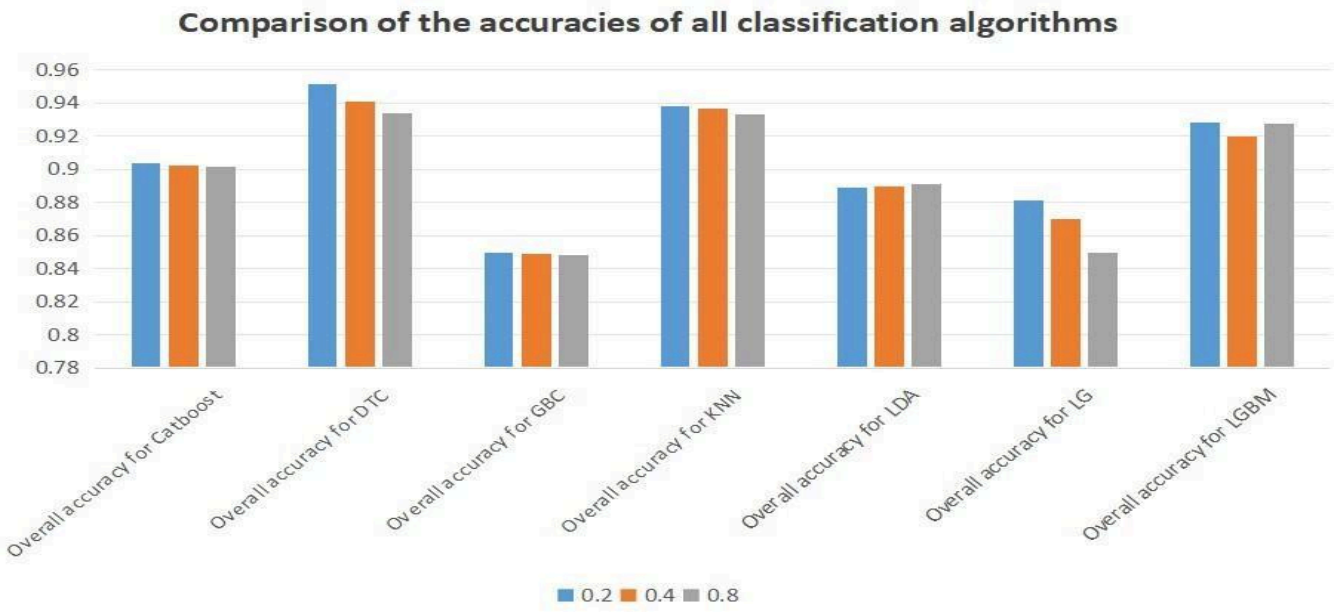

Fig-9: comparison of the accuracies of all classification algorithms

Fig-9 shows how all the classification algorithms do when we use Kernel PCA with a kernel to reduce the number of features. this step helps get rid of features in the netflow dataset and lets the classifiers learn from a smaller and more organised set of features. the decision tree classifier does the best getting 95.18% accuracy when we split the data 0.2 and doing well with 0.4 and 0.8 splits too. k-nearest neighbors and lightGBM also do well getting more than 92% accuracy most of the time which means they work well even after we change the features. catboost and LDA do good getting around 89 to 90% accuracy. gradient boosting and logistic regression do not do as well When we use data to test the accuracy goes down a little but overall using linear kernel PCA helps the classification algorithms work better and more efficiently especially for tree based and instance-based models, like decision tree classifier and k-nearest neighbors

Conclusion

This study is about a network intrusion detection system that uses machine learning. it is built on flow-level netflow traffic analysis and Linear Kernel principal component analysis to reduce features. the system is designed to deal with the challenges of network data and the limitations of detection systems. by using linear kernel principal component analysis the system makes the feature space simpler. this helps get rid of redundancy and makes the system more efficient. the dataset was checked using machine learning algorithms. these algorithms were tested with train and test splits to see how well they work and how well they can be used in different situations. the results show that the decision tree classifier is very accurate. It got 95.18% accuracy with a 0.2 split, did well with other splits KNN and LightGBM also did well Catboost and LDA were good. logistic regression and gradient boosting did not do well. the study found that tree based models work better they handle the feature space made by linear kernel principal component analysis well. the study shows that using flow- based analysis with dimensionality reduction in machine learning models can create a network intrusion detection system. this approach is good for network environments. in these environments accuracy and speed are important. in the future we might look at types of kernels for deep learning models, this could help detect attack patterns even better using linear kernel principal component analysis and machine learning models like Decision Tree Classifier KNN and LightBM can make network intrusion detection systems better.

References

1. Suriya Prakash Jambunathan,Suguna Ramadass,Palanivel Rajan Selva kumaran, "Analyzing the Behavior of Multiple Dimensionality Reduction Algorithms to Obtain Better Accuracy using Benchmark KDD CUP Dataset", The International Arab Journal of Information Technology (IAJIT) ,Volume 19, Number 01, pp. 121 - 131, January2022, doi: 10.34028/iajit/19/1/14.
2. J. Suriya Prakash, S. Narasani, N. Thangadurai, U. Prakash and S. Kiran, "Advancing Intrusion Detection Precision Through Analysis of Diverse Classification Algorithms," 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2024, pp. 1-6, doi: 10.1109/NMITCON62075.2024.10698858.

3. J. Suriya Prakash, C. H. Guntupalli, S. Narasani, N. N. Srinidhi and S. Kiran, "Boosting Accuracy in Intrusion Detection Systems: A Comprehensive Examination of Dimensionality Reduction and Classification Methods," 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2024, pp. 1-8, doi: 10.1109/NMITCON62075.2024.10698893.
4. S. P. J, S. N, L. A, Chaithra and K. S, "Enhance Intrusion Detection by Analyzing the Behavior of Labeled and Unlabeled Classification to Obtain Better Accuracy," 2024 Second International Conference on Advanced Computing & Communication Technologies (ICACCTech), Sonipat, India, 2024, pp. 791-797, doi: 10.1109/ICACCTech65084.2024.00131.
5. J. Suriya Prakash, T. Rashmika, N. Thangadurai, U. Prakash and S. Kiran, "Elevating Intrusion Detection Precision with Multi- Classification Algorithm Analysis," 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2024, pp. 1-8, doi: 10.1109/NMITCON62075.2024.10698932.
6. J. Suriya Prakash, P. Deeksha Gandhi, D. Kumar, M. H. Vishwas and Y. Shah, "Enhancing Network Security Through Advanced Intrusion Detection: A Fusion of Dimensionality Reduction and Machine Learning Classification for Improved Accuracy," 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2024, pp. 1-7, doi: 10.1109/NMITCON62075.2024.10699225.
7. S. P. J, C. H. Guntupalli, S. N. Chilamkurthy, G. Kowshik and A. Alekhya, "Enhancing the Security by Analyzing the Behaviour of Multiple Classification Algorithms with Dimensionality Reduction to Obtain Better Accuracy," 2023 IEEE 3rd Mysore Sub Section International Conference (MysuruCon), HASSAN, India, 2023, pp. 1- 7, doi: 10.1109/MysuruCon59703.2023.10396971.
8. S. P. J, T. S, S. R, V. Lakshmanan and K. S, "Improving Accuracy in Network Intrusion Detection via Machine Learning Algorithms: An In-Depth Examination," 2024 International Conference on Distributed Systems, Computer Networks and Cybersecurity (ICDSCNC), Bengaluru, India, 2024, pp. 1-7,doi: 10.1109/ICDSCNC62492.2024.10939819.
9. S. P. J, V. V, S. Jagannathan, V. C and K. S, "Improving Intrusion Detection Precision via Multi-Classification Algorithm Examination," 2024 International Conference on Distributed Systems, Computer Networks and Cybersecurity (ICDSCNC), Bengaluru, India, 2024, pp. 1-7, doi: 10.1109/ICDSCNC62492.2024.10939569.
10. C. Lekkalapudi, N. H. V P, S. Jagannathan, V. C and S. P. J, "Refining Intrusion Detection Capabilities Through Combined Algorithmic Classification Techniques," 2024 Second International Conference on Advanced Computing & Communication Technologies (ICACCTech), Sonipat, India, 2024, pp. 775-782, doi: 10.1109/ICACCTech65084.2024.00129.
11. S. P. J, H. S. Nambiar, G. V. Kumar, S. C, C. S. K and S. K. H,"Upgrade Better Accuracy by Analyzing the Behavior of Classifying Algorithm to Detect Intrusion in Network Traffic," 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE), Bangalore, India, 2024, pp. 735-742, doi: 10.1109/ICWITE59797.2024.10502430.
12. Suriya Prakash, J., Suguna, R., Neethu, P.S,"Unleashing the power of YOLOV5: Revolutionizing person detection and counting in restricted zones"2024 Taylor and Francis, CRC Press,PP.16,Edition:1st ,Ebook ISBN: 9781003502470.
13. S. Nandini, S. Murthy, P. P. K, P. U and S. P. J, "Enhancing the Security by Analyzing the Behavior of Multiple Classification Algorithms with Dimensionality Reduction to Obtain Better Accuracy," 2024 Second International Conference on Advanced Computing & Communication Technologies (ICACCTech), Sonipat, India, 2024, pp. 783-790, doi: 10.1109/ICACCTech65084.2024.00130.
14. H. S. Nambiar, L. Rangaiah, P. K. Praksha, C. Vasanthakumar and J. Suriya Prakash, "Optimizing Security Performance: Leveraging Multiclass Algorithms and Dimensionality Reduction for Enhanced Accuracy," 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2024, pp. 1-7, doi: 10.1109/NMITCON62075.2024.10699188.
15. R. Suguna, Y. Praveen Kumar, J. Suriya Prakash, P. S. Neethu and S. Kiran, "Utilizing Machine Learning for Sport Data Analytics in Cricket: Score Prediction and Player Categorization," 2023 IEEE 3rd Mysore Sub Section International Conference (MysuruCon), HASSAN, India, 2023, pp. 1-6,doi: 10.1109/MysuruCon59703.2023.10396955.
16. DS, M. Karunya, O. J and S. P. J, "Accuracy Prediction using Machine Learning Techniques for Indian Patient Liver Disease," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 614-618, doi: 10.1109/CISES58720.2023.10183617.
17. M. Desai and S. Prakash J, "An Exploration of the Effectiveness of Machine Learning Algorithms for Text Classification," 2023 2nd International Conference on Futuristic Technologies (INCOFT), Belagavi, Karnataka, India, 2023, pp. 1-6, doi: 10.1109/INCOFT60753.2023.10425568.
18. M. Desai and S. Prakash J, "Expression of Concern for: An Exploration of the Effectiveness of Machine Learning Algorithms for Text Classification," 2023 2nd International Conference on Futuristic Technologies (INCOFT), Belagavi, Karnataka, India, 2023, pp. 1-1, doi: 10.1109/INCOFT60753.2023.10703757.
19. S. J and K. S, "Obtain Better Accuracy Using Music Genre Classification Systemon GTZAN Dataset," 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Vijaypur, India, 2022, pp. 1-5, doi: 10.1109/NKCon56289.2022.10126991.