International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V72I1P107

Volume 72 Issue 1, 40-47, January 2024

Original Article

Transmission Deviation based Windowed Training for Intrusion Detection on Streaming Data

A. Sagaya Priya[1], S. Britto Ramesh Kumar[2]

[1,2]*Department of Computer Science, St. Joseph's College(Autonomous), Affiliated to Bharathidasan University, Tamil Nadu, India.*

[1]*Corresponding Author : rschlrsagayapriya@outlook.com*

Received: 24 November 2023 Revised: 30 December 2023 Accepted: 17 January 2024 Published: 30 January 2024

Abstract *- New and increased cyber-attacks have been launched frequently on network systems due to the large number of highly sensitive data transmitted in these systems. Hence, it becomes mandatory to improve the intrusion detection systems' capability and handle the high variations in data distributions that are common on systems experiencing concept drift. The proposed Transmission Deviation based Windowed Model (TDWM) for intrusion detection on streaming network data is a novel approach that addresses the need for improved intrusion detection systems in the face of high variations in data distributions. The TDWM model considers imbalance levels and is designed to handle varied imbalance levels effectively, ensuring unbiased training. Two training models have been designed, each level capable of handling varied imbalance levels. Retraining of models is triggered based on the drift levels, ensuring that the model never becomes obsolete. Experiments were performed on three different intrusion detection datasets containing varied imbalances and varied drift levels. Experimental results and comparisons indicate the model exhibits high accuracy levels of >97% over all three datasets. Such high performance on varied datasets indicates the model's capability to handle data with varied distributions and its ability to be deployed in real time.*

Keywords *- Network intrusion detection, Ensemble modeling, Boosting, Stacking, Time window, Online training.*

1. Introduction

The current society is increasingly becoming technology-reliant due to the high usage levels of internetwork devices on a daily basis [1]. The current big data era has witnessed the emergence of several networking devices, which are handheld and low-powered IoT devices [2]. The inclusion of these devices into people's daily lives has improved their quality of life to a large extent. However, these devices have also increased the amount of data flowing through the network. Further, a considerable amount of this data is composed of sensitive information that cannot be compromised [3, 4]. Transmission of such highly sensitive information has drawn a huge number of fraudsters into this domain, resulting in a potential increase in network attacks. Protecting the data being transmitted in the network has become highly significant by effectively securing the network. The very high volume and velocity levels of the information flowing through the network present huge challenges in providing a secure network. This has motivated the usage of machine learning models for network intrusion detection. Such models are called Network Intrusion Detection Systems (NIDS) [5].

Network intrusion detection models have become highly popular in the current decade. However, several systems use standard machine learning models, train them statically, and deploy them in the network environment for intrusion detection [6]. Providing a standalone model does not effectively suit the domain of network intrusion detection. The major reason for this is the presence of concept drift [7]. Data generated from the network tends to vary with changes in network features and user requirements. Network equipment is evolving fast and is becoming increasingly effective. Reliance on users towards networking technologies is also increasing due to the increased number of handhelds, standalone, and IoT devices [8]. These fast changes tend to affect the machine learning models, as they largely contribute towards variations in the network traffic [9, 10].

Network intrusion detection models are expected to be highly dynamic to ensure that the model does not become obsolete due to variations in the network traffic [11]. The network intrusion detection model is required to be trained online periodically based on the currently generated data. This process ensures that the model does not become outdated [12, 13]. This work presents a network intrusion detection model that performs periodical training based on a time window factor and operates on streaming data. The model has been designed in two levels, which are disjoint in nature, each level containing a different machine learning model. The level to be used for prediction is determined based on the imbalance

levels contained in the current data. These factors ensure highly effective detection and maintain the model current. Experimental results and comparisons indicate high performance, validating the usage of this model in real-time.

The increasing frequency of cyber-attacks on network systems and the transmission of highly sensitive data have highlighted the need to improve intrusion detection systems. These systems must also be able to handle high variations in data distributions, which are common in systems experiencing concept drift. The research paper aims to address this problem by proposing a Transmission Deviation based Windowed Model (TDWM) for intrusion detection on streaming network data. The model considers imbalance levels and is designed to handle varied imbalance levels effectively. Retraining of models is triggered based on drift levels to ensure that the model remains up-to-date and effective.

2. Related Works

Intrusion detection on data containing concept drift is one of the currently analyzed topics in the domain of network intrusion detection. Intrusion Detection Systems (IDS) are crucial for protecting network systems from cyber-attacks and unauthorized access to sensitive data. IDS monitors network traffic and identifies any suspicious or malicious activities that may indicate an intrusion. Traditional IDS approaches often rely on static models and predefined rules, which may not be effective in detecting new and evolving attack patterns. Streaming network data poses additional challenges for IDS, as it involves high variations in data distributions and concept drift, where the underlying data patterns change over time. Handling data imbalance is another important consideration in intrusion detection, as the number of instances in the majority and minority classes can vary significantly. Imbalance handling mechanisms, such as oversampling or boosting, are used to ensure unbiased training and improve detection performance. This section discusses the existing works dealing with intrusion detection on streaming data.

An ensemble-based machine learning model for intrusion detection has been designed by Martindale et al. [14]. This work considers multiple heterogeneous and homogeneous ensembles and proposes three heterogeneous ensemble models that can be used for intrusion detection over streaming data. The model also uses adaptive learning techniques to ensure that the periodically trained model is highly effective. The efficiency of using multiple homogeneous or heterogeneous models to enhance the process of intrusion detection has been constructed by Bian et al. [15]. This work shows the efficiency of an ensemble model and the prediction enhancement it imparts in the general machine-learning process. A comparison of standalone models with ensemble-based modeling techniques has been performed by Verma et al. [16] for intrusion detection over IoT systems. Ensemble models for intrusion detection on streaming data have been designed by Rettig et al. [17]. The model also concentrates on performing online detection over streaming data. However, a supervised learning model for intrusion detection has been constructed by Yang et al. [18]. This work is based on using an autoencoder with generalized adversarial networks combined with regularization to improve the network intrusion detection process.

A model for network intrusion detection over streaming data has been created by Desale et al. [19]. This work presents an ensemble of multiple classification techniques to handle data imbalance in a streaming context effectively. An intrusion detection model based on centralized anomaly detection was constructed by Faisal et al. [20]. This model has been designed to perform intrusion detection over smart grids. An intrusion detection model specifically designed to operate on wireless mesh networks has been created by Anzi [21]. This model has been specifically designed to perform cross-layer intrusion detection systems common in mesh networks.

A network that has been specifically designed to perform intrusion detection in non-line-of-sight environments has been designed by Gui et al. [22]. The model is highly effective in small-scale indoor environments. A binary optimization-based model to perform intrusion detection has been constructed by Hassan et al. [23]. This work uses a binary manta ray forging optimization algorithm to perform feature selection and uses random forest for the process of intrusion detection.

A deep learning-based model to perform intrusion detection in a wireless environment has been designed by Simon et al. [24]. The model considers energy limitation as the major constraint in designing the intrusion detection models. This work uses a deep learning model for feature selection, and a decision tree algorithm is used to detect attacks in the network. A feature elimination-based model for intrusion detection was created by Kannari et al. [25]. This work mainly concentrates on identifying significant features to improve the process of intrusion detection. Recursive elimination of features is performed to improve the data quality, and a random forest algorithm is used for intrusion detection. A deep learning-based model that uses spark architecture to perform intrusion detection on big data has been created by Ramkumar et al. [26].

This work integrates a sea lion optimization algorithm with the deep residual network to enhance the intrusion detection process. A meta-classifier-based approach for intrusion detection has been designed by Ravi et al. [27]. This work performs feature extraction from the hidden layers of recurrent models to optimize the intrusion detection model's decision-making performance. A deep learning-based model using a deep autoencoder for intrusion detection has been designed by Quazi et al. [28]. This work relies on a nonsymmetric deep autoencoder for the prediction process.

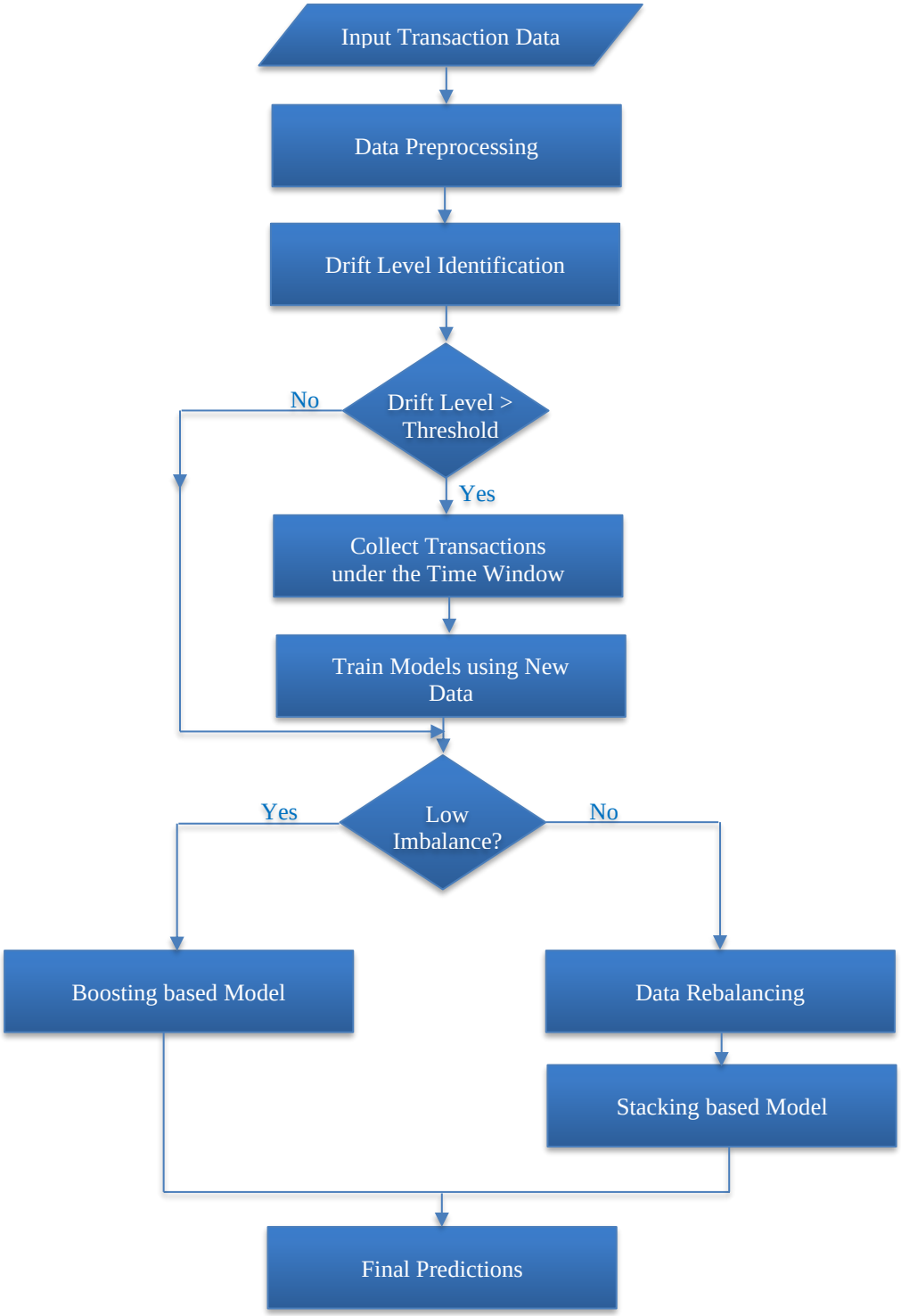

Fig. 1 Flow of TDWM Model

3. Transmission Deviation based Windowed Model (TDWM)

Identifying intrusions in networks has become a major requirement due to the increased use of networking technologies. The occurrence of deviation in the transmissions creates a huge challenge in designing machine learning models for intrusion detection. This work presents a Transmission Deviation-based Windowed Model (TDWM) for intrusion detection in network transmissions. The proposed model is composed of the data preprocessing phase, transmission deviation-based window selection phase,

imbalance level-based model selection process, and the creation of models to handle the varied imbalance levels. The algorithm for the TDWM model is provided below.

Algorithm-1 TDWM
Input: Imbalanced data (KDD CUP 99, NSL-KDD, UNSW-NB15)
Output: Predictions on imbalanced data

1. Input network transmission data
2. Data preprocessing to perform encoding and remove inconsistencies
3. Identify drift level
4. If drift level > threshold
    a. Collect transactions falling within the window as training data
    b. Identify the imbalance level of training data
    c. If the imbalance level is low
        i. Train boosting model
        ii. Set it as the current model
    d. If the imbalance level is high
        i. For each additional majority record contained in the training data
        1. Random select two minority instances
        2. Generate new instances based on the mean value of the selected instances
        ii. Merge the generated instances with the training data to form the updated data
        iii. Train the stacking model with the training data
        iv. Set it as the current model
5. For each instance i, in test data
    a. Pass i to the current model
    b. Obtain final predictions

The Flow diagram of the TDWM model is provided in Figure 6.1.

3.1. Data Preprocessing

The network transmission data is composed of a huge number of instances depicting network transmissions and is composed of several network-based attributes. The initial process is to identify the existence of null values in the data. This is followed by eliminating the instances that contain null values. The data has also been observed to contain several textual attributes. Textual attributes are generally categorized as categorical and string attributes. String attributes are eliminated, and categorical attributes are converted to numerical attributes using one hot encoding technique. Some network transmission data record the class label as a categorical attribute. This feature is label encoded, and the numerical feature replaces the categorical class feature. This marks the end of the preprocessing phase.

3.2. Transmission Deviation based Window Selection

Network transmissions generally exhibit high volume and velocity levels. They also exhibit variations with time. After a period of time, generated network transmissions exhibit variations in data distributions. This property is known as concept drift. Concept drift has been a common property that could be observed in several domains that are based on customer behavior. Improvements in networking capabilities, high-speed Internet, and the possibility of widespread network usage have been major components contributing to concept drift.

Intrusion detection models are generally supervised learning models entirely based on the training data initially used to build the model. Although such models perform effectively in the initial stages, the presence of concept drift in the transmission data gradually makes them ineffective and obsolete. The speed of depreciation depends on the level of drift the domain exhibits. Concept drift can be measured based on parameters like trend and seasonality. Trend refers to long-term variations in the data distribution, and seasonality refers to short-term variations. Since this work involves windowed operations, seasonality levels are considered for analysis. Further, correlation is also considered a secondary analysis metric for measuring the variations in data distribution.

Seasonality and correlation levels are identified based on the current time window and the time window used for training the current model. If the analysis exhibits high seasonality levels, then it is an indicator of drift. Correlation indicates the level of variation that has occurred during the two-time windows. Correlation levels of less than 75% indicate that the data has exhibited considerable drift, and the model has to be retrained to make it up to date. This triggers the process of selecting a new training set for training the model. Transactions since the last time window marks the beginning of the training data. Data between the last time window and the current transaction with labels is selected for model retraining. The trained model is deployed to predict the current transactions.

Every predicted transaction is recorded in the buffer and is held in the buffer until the waiting time elapses. After the end of the waiting time, changes in the predictions, if any, are made, and the transactions are moved from the buffer to the time window for usage in future training processes.

3.3. Imbalance Level-based Model Selection

The retraining process begins with the training data selection. However, a single model cannot be applied to varied scenarios due to the presence of data imbalance in the network transmission data. The imbalance level can considerably vary between time windows. Hence, the imbalance levels are identified to determine the type of model that will suit the current data effectively. Imbalance level is identified by determining the ratio between the number of instances in the majority and the minority classes. The imbalance ratio is given by,

$$IR = \frac{\#\ of\ Majority\ Class\ Instances}{\#\ of\ Minority\ Class\ Instances}$$

Low to moderate imbalance levels can be handled by the machine learning models. Additional imbalance handling mechanisms are unnecessary, and including them would lead to unnecessary complexities in the model. Hence, if the data within the time window exhibits low to moderate imbalances, it is passed to the boosting-based model. This model has been specifically designed to reduce the variance levels in the prediction process.

High data imbalance levels require an architecture that exhibits higher imbalance handling capabilities. Hence, data exhibiting high imbalance levels are passed to the stacking architecture incorporated with the auto-rebalancing technique. The additional data balancing mechanism and the stacking nature of the model have been identified to handle imbalance effectively.

3.4. Boosting Incorporated Intrusion Detection (Model 1)

Intrusion detection is performed using the boosting model on data exhibiting low to moderate imbalance levels. A tree-based machine learning model, identified to handle low data imbalance intrinsically, is used to create the boosting model. The decision tree model is used as the base learner model, and the boosting process is performed by iterative retraining. During every retraining phase, the significance of the instances exhibiting errors is increased to improve the training process and reduce the error levels. The ability of decision trees to perform dynamic model building ensures that the model can be trained effectively and at a faster rate.

3.5. Stacking and Auto-Rebalancing based Intrusion Detection (Model 2)

Data exhibiting high imbalance requires additional rebalancing mechanisms to ensure that the model is not biased due to the presence of imbalance. An oversampling-based rebalancing technique is used to generate balanced data. The major reason for using oversampling over the under-sampling technique is that the transmission records are significant and cannot be eliminated from the training data. Every new instance is generated using three varied instances belonging to the minority class.

The generated data is combined with the actual data to form the final training data. The training data is passed to the level one training models. The level one training model comprises a mixture of supervised and unsupervised models to incorporate variety in the training process. This type of varied learning can effectively provide improvements in decision rules. The unsupervised model is created with the Gaussian mixture model, and the supervised model is created using decision trees. After the completion of the training process, the validation data is passed to the models, and prediction is performed. These predictions are grouped along with the final class label to formulate the training data for the level 2 model. This process creates a stacking mechanism which improves the prediction process. Logistic regression is used as the second level's base learning model of choice. The final data to be predicted is passed to the level 1 model, and the results obtained from level one models are grouped and passed to the level 2 model to obtain the final prediction.

4. Results and Discussion

The TDWM model has been implemented using Python and the scikit library. The performance of the TDWM model is analyzed over the KDD CUP 99 dataset, NSL- KDD dataset, and UNSW- NB15 dataset.

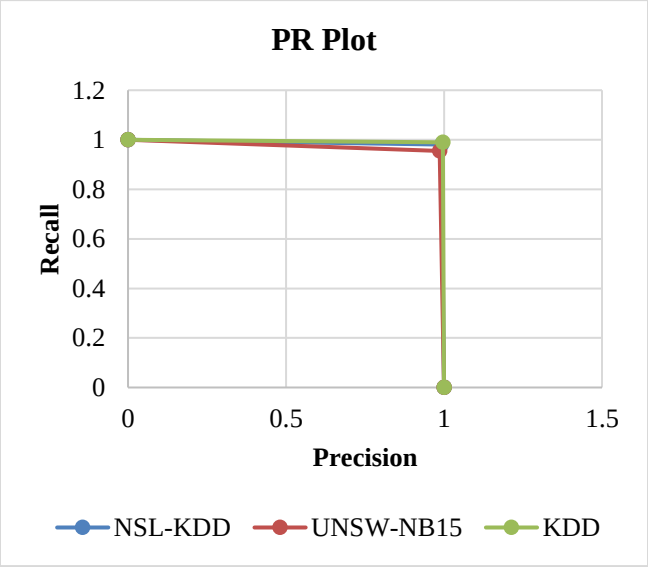

Fig. 2 PR plot for TDWM

The datasets are loaded into the model and are passed for prediction in batches. Results obtained are determined by calculating the overall performance by considering all the data batches. The PR plot representing the precision and recall values of the TDWM model is shown in Figure 2. High precision and recall values represent an effective intrusion detection process. The plot shows very high precision and recall levels, and the curve has been observed to be at the top right corner, depicting that the model is highly capable of identifying anomalous transmissions. This indicates that the model can effectively demarcate between normal and anomalous transactions, depicting a low influence of data imbalance over the prediction process.

A comparison of the aggregate measures, accuracy, F-Measure, and AUC is shown in Figure 3. High values of all three measures represent high performance. The chart shows that the TDWM model exhibits greater than 95% performance over all three measures on NSL-KDD, KDD, and UNSW-NB15 datasets. This performance shows that the model is highly capable of providing high performance over data with varied imbalances and varied distribution levels. Hence, this indicates a generic model that is highly capable of handling data composed of varied distributions. This shows the model's capability to handle data exhibiting concept drift effectively.

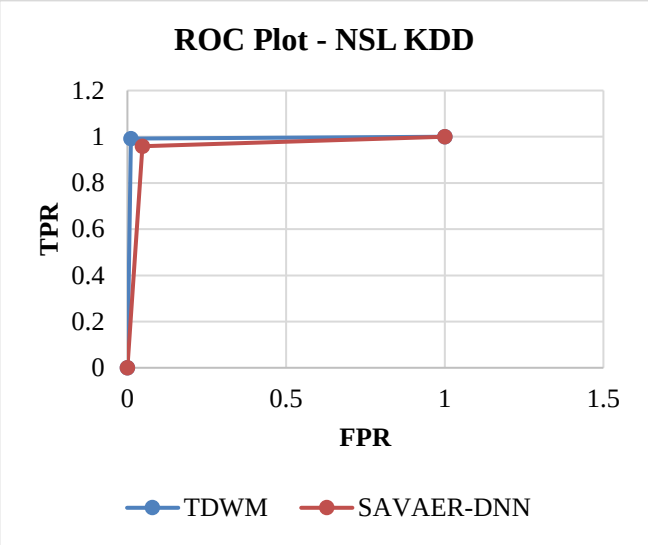

Fig. 3 Aggregate Measure Comparison for TDWM

Table 1. Performance of TDWM

Technique	NSL-KDD	UNSW-NB15	KDD
TPR	0.992	0.986	0.996
TNR	0.989	0.965	0.986
Recall	0.992	0.986	0.996
Precision	0.980	0.954	0996.989
F-Measure	0.986	0.970	0.993
FPR	0.011	0.035	0.014
FNR	0.008	0.014	0.004
Accuracy	0.990	0.974	0.992
AUC	0.990	0.976	0.991

A tabulated view of the performance of TDWM on NSL-KDD, KDD, and UNSW-NB15 datasets is shown in Table 1. The performance in TPR, Recall, Precision, TNR, Accuracy, F-Measure, and AUC shows greater than 95%, depicting the capability of the model to differentiate between normal and anomalous transactions effectively. Similarly, the false error levels FPR and FNR exhibit values ranging between 0.4% and 3%, showing that the model exhibits very low false error levels.

4.1. Comparative Analysis

A comparative analysis of the TDWM model with the SAVAER-DNN [17] model is provided below. The comparison is presented regarding the ROC plot on the NSL-KDD and UNSW-NB15 datasets. ROC Plot representing the performance of TDWM and SAVAER-DNN models is shown in Figure 4. Curves exhibiting a higher area and occupying the top left corner of the plot depict better performance. The plot represented by the TDWM model exhibits higher TPR levels and lower FPR levels compared to the plot represented by the SAVAER-DNN model. This shows that TDWM exhibits lower false alarms and better identification of anomalous records.

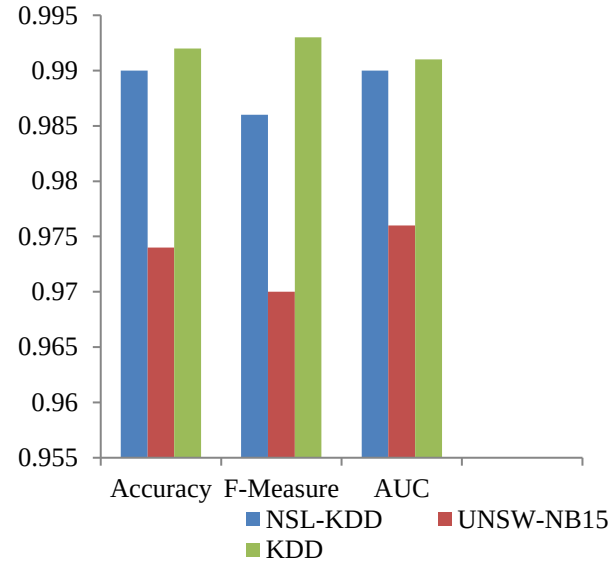

Fig. 4 ROC Comparison on NSL-KDD for TDWM

The ROC plot representing comparisons on the UNSW-NB15 data set is shown in Figure 5. The plot represented by TDWM exhibits higher TPR levels and lower FPR levels compared to the SAVAER-DNN model. This signifies the generic and high-performing nature of the TDWM model.

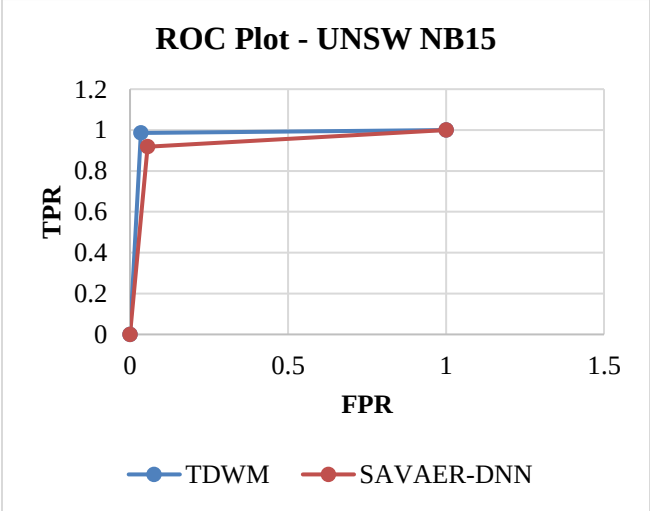

Fig. 5 ROC Comparison on UNSW-NB15 for TDWM

A tabulated view of the comparison is shown in Tables 2 and 3. The best performances are highlighted in bold. Comparisons indicate that the TDWM model demonstrates 7% improved TPR levels, 4% improved accuracy levels, and 4% improved F-Measure levels over the UNSW-NB15 data set. Similarly, 3% improved TPR levels, 10% improved accuracy levels, and 8% improved F-Measure levels were observed in NSL-KDD data. A reduction of false alarms at 2%

was observed in UNSW-NB15 data, and a reduction of false alarms at 3% was observed in NSL-KDD data. These performances show that the TDWM model is highly capable of identifying anomalous transactions with low false alarm levels.

Table 2. Performance Comparison of TDWM on UNSW-NB15

UNSW-NB15	SAVAER-DNN [17]	TDWM
FPR	0.056	0.035
TPR	0.919	0.986
Accuracy	0.930	0.974
F-Measure		

Table 3. Performance Comparison of TDWM on NSL-KDD

NSL-KDD	SAVAER-DNN [17]	TDWM
FPR	0.047	0.011
TPR	0.959	0.992
Accuracy	0.89	0.990
F- Measure	0.9	0.986

5. Conclusion

The continuously escalating threats and attacks warrant high levels of security while transmitting sensitive information. Networked systems' volume and velocity levels make intrusion detection over communication networks challenging. This work presents a window-based model to perform intrusion detection over streaming data. The window level is determined by the level of drift experienced in the domain. This is a multi-model-based architecture comprising models for data that exhibit varied levels of imbalances. The major advantage of this model is that it is adaptive, and the time window size is varied based on the drift levels. Further, imbalance levels of transactions in every window are identified, and the models are assigned appropriately. Experimental results and comparisons indicate High performance with greater than 97% accuracy levels and greater than 98% anomaly detection levels, indicating the model is highly capable of handling imbalance and concept drift. Future enhancements of the model will be based on Creating a model capable of updating on the fly to avoid separate retraining after encountering considerable drift levels.

References

[1] Dylan Chou, and Meng Jiang, "A Survey on Data-Driven Network Intrusion Detection," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–36, 2021. [Cross Ref] [Google Scholar] [Publisher Link]

[2] Kelton A.P. da Cost et al., "Internet of Things: A Survey on Machine Learning-Based Intrusion Detection Approaches," *Computer Networks*, vol. 151, pp. 147–157, 2019. [Cross Ref] [Google Scholar] [Publisher Link]

[3] Javier Martínez Torres, Carla Iglesias Comesaña, and Paulino J. García-Nieto, "Review: Machine Learning Techniques Applied to Cybersecurity," *International Journal of Machine Learning and Cybernetics*, vol. 10, pp. 2823–2836, 2019. [Cross Ref] [Google Scholar] [Publisher Link]

[4] Neha Srivastava, and Umesh Chandra Jaiswal, "Big Data Analytics Technique in Cyber Security: A Review," *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, pp. 579-585, 2019. [Cross Ref] [Google Scholar][[Publisher Link]

[5] T. Ryan Hoens, Robi Polikar, and Nitesh V. Chawla, "Learning From Streaming Data with Concept Drift and Imbalance: An Overview," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 89–101, 2012. [Cross Ref] [Google Scholar] [Publisher Link]

[6] Bartosz Krawczyk, and Alberto Cano, "Online Ensemble Learning with Abstaining Classifiers for Drifting and Noisy Data Streams", *Applied Soft Computing*, vol. 68, pp. 677–692, 2018. [Cross Ref] [Google Scholar] [Publisher Link]

[7] B. Mukherjee, L.T. Heberlein, and K.N. Levitt, "Network Intrusion Detection," *IEEE Network*, vol. 8, no. 3, pp. 26-41, 1994. [Cross Ref] [Google Scholar] [Publisher Link]

[8] Robert C. Newman, *Computer Security: Protecting Digital Resources*, 1st ed., Jones and Barret Learning, 2010. [Google Scholar] [Publisher Link]

[9] J.B.D. Caberera, B. Ravichandran, and R.K. Mehra, "Statistical Traffic Modeling for Network Intrusion Detection," *Proceedings 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (Cat. No.PR00728)*, San Francisco, CA, USA, pp. 466-473, 2000. [Cross Ref] [Google Scholar] [Publisher Link]

[10] Pedro Casas, Johan Mazel, and Philippe Owezarski, "Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge," *Computer Communications*, vol. 35, no. 7, pp. 772–783, 2012. [Cross Ref] [Google Scholar] [Publisher Link]

[11] Ali H. Sayed, "Adaptation, Learning, and Optimization Over Networks," *University of California*, Los Angeles, USA, vol. 7, no. 4-5, pp. 311-801, 2014. [Cross Ref] [Google Scholar] [Publisher Link]

[12] András A. Benczúr, Levente Kocsis, and Róbert Pálovics, "Online Machine Learning Algorithms Over Data Streams," *Encyclopedia of Big Data Technologies*, pp. 1199–1207, 2019. [Cross Ref] [Google Scholar] [Publisher Link]

[13] Nathan Martindale, Muhammad Ismail, and Douglas A. Talbert, "Ensemble-Based Online Machine Learning Algorithms for Network Intrusion Detection Systems Using Streaming Data," *Information*, vol. 11, no. 6, p. 315, 2020. [Cross Ref] [Google Scholar] [Publisher Link]

[14] Bian Shun, and Wenjia Wang, "On Diversity and Accuracy of Homogeneous and Heterogeneous Ensembles," *International Journal of Hybrid Intelligent Systems*, vol. 4, no. 2, pp. 103–128, 2007. [Cross Ref] [Google Scholar] [Publisher Link]

[15] Abhishek Verma, and Virender Ranga, "Machine Learning Based Intrusion Detection Systems for IoT Applications," *Wireless Personal Communications*, vol. 111, pp. 289–312, 2019. [Cross Ref] [Google Scholar] [Publisher Link]

[16] Laura Rettig et al., "Online Anomaly Detection Over Big Data Streams," *Applied Data Science*, pp. 289–312, 2019. [Cross Ref] [Google Scholar] [Publisher Link]

[17] Yanqing Yang et al., "Network Intrusion Detection Based on Supervised Adversarial Variational Auto-Encoder with Regularization," *IEEE Access*, vol. 8, pp. 42169–42184, 2020. [Cross Ref] [Google Scholar] [Publisher Link]

[18] Ketan Sanjay Desale, Chandrakant Namdev Kumathekar, and Arjun Pramod Chavan, "Efficient Intrusion Detection System Using Stream Data Mining Classification Technique," *2015 International Conference on Computing Communication Control and Automation*, pp. 469–473, 2015. [Cross Ref] [Google Scholar] [Publisher Link]

[19] Mustafa Amir Faisal et al., "Securing Advanced Metering Infrastructure Using Intrusion Detection System with Data Stream Mining," *Pacific-Asia Workshop on Intelligence and Security Informatics (PAISI) 2012: Intelligence and Security Informatics*, pp. 96–111, 2012. [Cross Ref] [Google Scholar] [Publisher Link]

[20] Fawaz S. Al-Anzi, "Design and Analysis of Intrusion Detection Systems for Wireless Mesh Networks," *Digital Communications and Networks*, vol. 8, no. 6, pp. 1068–1076, 2022. [Cross Ref] [Google Scholar] [Publisher Link]

[21] Linqing Gui, Wenyang Yuan, and Fu Xiao, "CSI-Based Passive Intrusion Detection Bound Estimation in Indoor NLos Scenario," *Fundamental Research*, vol. 3, no. 6, pp. 988–996, 2023. [Cross Ref] [Google Scholar] [Publisher Link]

[22] Ibrahim Hayatu Hassan et al., "An Improved Binary Manta Ray Foraging Optimization Algorithm Based Feature Selection and Random Forest Classifier for Network Intrusion Detection," *Intelligent Systems with Applications*, vol. 16, 2022. [Cross Ref] [Google Scholar] [Publisher Link]

[23] Judy Simon et al., "Hybrid Intrusion Detection System for Wireless IoT Networks Using Deep Learning Algorithm," *Computers and Electrical Engineering*, vol. 102, 2022. [Cross Ref] [Google Scholar] [Publisher Link]

[24] Phanindra Reddy Kannari, Noorullah Shariff Chowdary, and Rajkumar Laxmikanth Biradar, "An Anomaly-Based Intrusion Detection System Using Recursive Feature Elimination Technique for Improved Attack Detection," *Theoretical Computer Science*, vol. 931, pp. 56–64, 2022. [Cross Ref] [Google Scholar] [Publisher Link]

[25] M.P. Ramkumar et al., "Intrusion Detection in Big Data Using Hybrid Feature Fusion and Optimization Enabled Deep Learning Based on Spark Architecture," *Computers and Security*, vol. 116, 2022. [Cross Ref] [Google Scholar] [Publisher Link]

[26] Vinayakumar Ravi, Rajasekhar Chaganti, and Mamoun Alazab, "Recurrent Deep Learning-Based Feature Fusion Ensemble Meta-Classifier Approach for Intelligent Network Intrusion Detection System," *Computers and Electrical Engineering*, vol. 102, 2022. [Cross Ref] [Google Scholar] [Publisher Link]

[27] Emad-ul-Haq-Qazi et al., "An Intelligent and Efficient Network Intrusion Detection System Using Deep Learning," *Computers and Electrical Engineering*, vol. 99, 2022. [Cross Ref] [Google Scholar] [Publisher Link]