



Deep Comparison Analysis : Statistical Methods and Deep Learning for Network Anomaly Detection

Amit Kumar Jaiswal ^{#1}, Alireza Nik Aein Koupaei ^{*2}

[#] *Moscow Institute of Physics and Technology (MIPT)*
Department of Radio Engineering and Cybernetics
141701, Institutsky Lane, 9, Moscow, Russian Federation
 <https://orcid.org/0000-0003-2036-0989>,
¹ dzhaisval.a@phystech.edu

^{*} *Moscow Institute of Physics and Technology (MIPT)*
Department of Radio Engineering and Cybernetics
141701, Institutsky Lane, 9, Moscow, Russian Federation
 <https://orcid.org/0000-0001-8848-2654>,
² anikaeinkoupaei@phystech.edu

Abstract—The detection of attacks and anomalies in supervised datasets is critical for maintaining the security and integrity of systems and networks. Traditional methods for attack detection often rely on supervised learning techniques, which may struggle to accurately identify novel or complex threats. In this study, we propose a novel approach to enhancing attack detection models by leveraging deep learning techniques for anomaly detection. Specifically, we introduce a deep learning framework that effectively captures intricate patterns and relationships within supervised datasets to identify anomalous behavior indicative of potential attacks. Through extensive experimentation and evaluation on diverse datasets, we demonstrate the superior performance of our proposed approach compared to traditional methods. Our results highlight the effectiveness of deep learning in enhancing the accuracy and efficiency of attack detection in supervised datasets, paving the way for more robust and adaptive security systems. Overall, this research contributes to the advancement of anomaly detection methodologies and underscores the importance of integrating deep learning techniques into the realm of cybersecurity for improved threat mitigation and defense.

Keywords: Attack Detection , Supervised Datasets , Deep Learning Approach , Anomaly Detection , Cybersecurity.

I. INTRODUCTION

The escalating dependence on digital technologies has rendered contemporary networks and systems susceptible to a diverse array of cyber threats. Anomaly detection, which encompasses the identification of atypical patterns or behaviors, constitutes an essential element of cybersecurity methodologies. Nevertheless, conventional approaches frequently exhibit limitations in their capacity to identify intricate and evolving threats. While machine learning and deep learning techniques have shown potential in enhancing anomaly identification, their functionality in real-world datasets is regularly weakened by qualities for example disproportionate class representation, noise, and moving concepts. This article investigates applying guided machine learning and deep learning to boost attack discovery models in real-world datasets. We introduce an

innovative structure and assess its effectiveness using real-world datasets, illustrating the aptitude of directed anomaly identification to better cyber security dangers detection in complicated environments.

II. LITERATURE REVIEW

The digital landscape is experiencing an exponential expansion presenting unmatched opportunities while giving rise to intricate cyber threats that constantly put conventional cybersecurity measures to the test. In this dynamic environment machine learning (ML) approaches especially anomaly detection techniques have attracted considerable attention as invaluable assets for bolstering cybersecurity defenses. In this current age of infrastructure and tech landscapes cybersecurity emerges as a critical concern. The persistent attempts to infiltrate the network systems of organizations and nations have spotlighted the deficiencies of traditional firewall and antivirus defenses against complex digital cyberattacks. This flaw exposes systems to various risks as attackers utilize an array of techniques to manipulate network traffic for unauthorized access or data theft [1], [4]. Individual attackers along with big organizations or companies might plan attacks to sneakily get hold of specific information. These attacks are not confined to any particular network either internal or external and could be initiated by actors from within or outside. Due to the distinct features inherent in each attack, distinguishing between them poses a significant challenge, making it arduous to comprehend their nature. Moreover, without the assistance of diverse machine learning (ML) algorithms developed by computers utilizing various programming languages, humans encounter considerable difficulty in independently detecting numerous types of information security breaches [5],[10]. In our rapidly evolving digital world, cyber-attacks are constantly changing, posing significant challenges. To stay ahead, proactive defense measures are essential. Organizations must invest in robust cybersecurity strategies and foster a culture of resilience to

effectively mitigate evolving threats, protecting their digital assets and the broader global ecosystem [11], [15]. Addressing significant cybersecurity challenges through machine learning solutions is crucial. The provision of new datasets supports academic research in understanding these challenges and devising effective methods. Additionally, the introduction of a label generation method using pivoting addresses the issue of insufficient labels in cybersecurity [16], [20].

III. ML AND CLASSIFICATION FOR ANOMALY DETECTION TECHNIQUES

The classification of anomaly detection techniques has a wide variety of applications, notably in cybersecurity defensive reinforcement. They try to detect attacks sooner, which allows the business to react fast and minimizes the company's security breach. Furthermore, one of its advantages is that it detects real-world difficulties by reporting a set of usual behaviors within systems, allowing the professional team to respond to a potential security danger efficiently. Furthermore, the algorithms in use may assist to reduce the amount of false positives, allowing resources for incident response to be used more efficiently. The algorithms employ machine learning to adapt and learn more about traffic behavior, eventually increasing their expertise over time. This would not only improve the stability of cybersecurity equipment, but it would also make it more resistant to new threats that emerge often. If these technologies are properly applied, they are likely to provide organizations with a complete picture of diverse data sources, allowing the company to improve security measures and secure the enormous assets in our volatile digital environment.

A. Machine Learning Models for Anomaly Detection

1) *Decision Tree Classifier*: A decision tree method divides the feature space into subsets, then splits each subset again to produce other subsets, and so on, until we are left with additional splits that do not need to be examined. The method splits the feature space recursively and then classifies the data based on the feature values. It's simple to perceive and understand, making it ideal for investigating feature-target correlations.

2) *Random Forest Classifier*: An ensemble learning technique that uses numerous decision trees to increase classification accuracy and resilience. It is well-known for its superior performance, scalability, and resistance to overfitting, and resistance to overfitting.

3) *GaussianNB Classifier*: The GaussianNB Classifier is a fundamental probabilistic classifier that operates under the premise of Bayes' theorem, assuming feature independence. It is particularly well-suited for datasets with continuous characteristics that follow a Gaussian distribution.

4) *LGBM Classifier*: A complex model that stood out not just for its efficiency, speed, and accuracy, but also for its LGBM Classifier. The LGBM Classifier excels at creating decision trees that maximize the number of nodes. Furthermore, it performs admirably when dealing with massive volumes of data.

5) *Catboost Classifier*: The Catboost Classifier is the outcome of a hierarchical boosting library that was created with considerable care to accommodate the requirements of categorical feature handling. It not only excels at handling categorical data, but also exceeds all of its competitors by employing sophisticated challenges such as ordered boosting and target encoding. The unique method of ordered boosting and ease of coding provides strength for the analysis of structured categorical data. That is the foundation for the strength and competitiveness of this remarkable instrument.

B. Feature Selection and Dimensionality Reduction

It is claimed that feature selection is used to create the best model possible by selecting the most important features in any given sequence that maximize the model's performance, whereas dimensionality reduction is used to obtain a new dataset with the same amount of information but a smaller size.

IV. PROPOSED METHODOLOGY

This theory presents a way for identifying numerous network attacks that have the potential to affect any network. It entails describing the dataset, performing preprocessing processes, and employing feature extraction methods. Furthermore, it describes the creation of eight machine learning approaches for this aim. The fig.1 the chosen model is trained

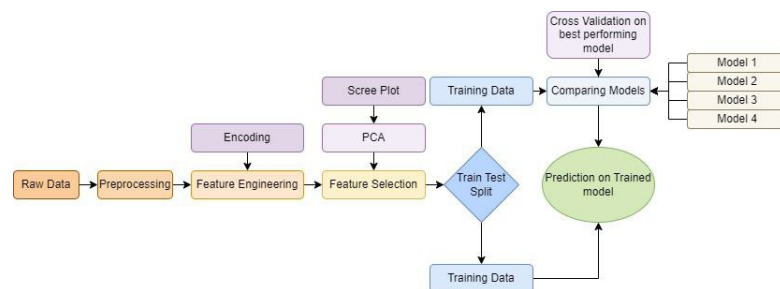


Fig. 1. Proposed Methodology

using a conventional machine learning approach for developing and assessing a predictive model. The training data. Here's a breakdown:

A. Data Acquisition & Preparation:

- 1) *Raw Data*: : The process begins with raw data.
- 2) *Preprocessing*: : This stage comprises cleaning, processing, and preparing raw data for analysis, as well as dealing with missing values, outliers, and so on.
- 3) *Encoding*: : Converting category characteristics into numerical data, often by one-hot encoding.
- 4) *Feature Engineering*: : Creating new features from current data, which might improve the model's prediction capacity.

B. Feature Selection:

- 1) *PCA (Principal Component Analysis)*: : A dimensionality reduction strategy that finds the primary components (features) that account for the majority of the data's variation.

2) *Scree Plot*: Visualizes the variation explained by each major component, assisting in determining the ideal number of components to keep.

C. Model Training & Evaluation:

1) *Training Data*: The cleaned and prepared data is separated into training and testing sets.

2) *Train Test Split*: Divide the dataset into training and test sets to train and evaluate the model, accordingly.

3) *Model Training*: The specified model is trained using the training data.

4) *Comparing Models*: Multiple models are trained and tested to choose the most effective one.

5) *Cross-Validation on Best Performing Model*: Cross-validates the model's performance to confirm its ability to generalize to previously encountered data.

D. Prediction:

1) *Prediction on Trained Model*: The best-performing model is used to make predictions on unseen data.

E. Outcome:

The overall goal of this workflow is to build a reliable predictive model by iteratively improving the data preparation, feature selection, and model training processes.

F. Data Description

The statistical summary of network traffic data in the dataset provides a multifaceted insight into various metrics like flow duration packet counts and window sizes. Comprising 84 columns and 555278 rows the dataset includes a timestamp indicating the recording time of the data. These columns offer a wide array of network traffic measurements encompassing packet exchanges data packet dimensions and flow durations. Moreover they present details regarding initial and final window sizes along with diverse evaluations of idle time. Each column in the dataset contains statistical measurements such as the count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum. These measurements offer a summary of the distribution of data in each column, allowing for simple comparison and study of various metrics. The dataset does not appear to have any missing or null values, since each column has all of the statistical measures. However, it is important to note the presence of a great number of columns, where most of the elements are zero – a fact that might be indicative of under-collection or the decision by the network traffic person in charge not to use them. It is important to mention the fact that the dataset contains a report of network traffic and the statistics consisting of packet counts, window sizes, and the periods of inactivity. The dataset may be used for various purposes including the analysis of network performance, detection of anomalies, and strategic capacity planning.

1) *Preparing Dataset*: The preparation of the dataset is a critical step to provide a good foundation for effective attack detection model creation using deep learning models. This procedure includes installing various necessary packages thus setting up the data so that it would be suitable for the production of robust and accurate models.

2) *Data Collection*: Obtaining the raw data from a variety of sources, such as logs of network connections, intrusion detection systems, and other relevant data repositories is the first phase. To this end, we used a well-designed supervised dataset that contains labeled instances indicating normal and malicious activities.

3) *Data Cleaning*: Raw data very frequently contains noises, missing values, and inconsistencies leading to the diminished performance of models. We did data cleaning to remove any irrelevant information, handle missing values, and correct inconsistencies. This phase is applied to getting rid of incorrect and misleading data and hence will make the data set ready for learning.

4) *Data Reprocessing*: Preprocessing is essential to transform raw data into a format that can be effectively used by deep learning algorithms. This includes:

- **Normalization**: Scaling numerical features to a standard range must be done in a right way to make all the features the same size and thus ensure the strength of the learning and improvement process granted by this method if the system is unable to
- **Encoding**: To change categorical data into numerical values one can use techniques like one-hot encoding.
- **Feature Engineering**: Concept of how to choose those that are useful, and creation of the relevant features by the data that are too important in them, which makes the model work well.

5) *Data Splitting*: The data would need to be preprocessed and cleaned before splitting it into the training, validation, and test protocols. The training dataset is the base on which the deep learning models are trained, the validation dataset is the used to tune up hyperparameters, normalize and preclude overfitting, and the test dataset is used for evaluating the performance of the final model.

G. Model Training and Hyperparameter Tuning

1) *Decision Tree Classifier*: The model is created by arranging branches representing decision rules on an inverted tree structure, with leaves providing instructions for making the choice. The technique is non-parametric, which means it does not assume a predefined distribution for the data, making it very successful at processing complex datasets and thus not limited to strictly parametric restrictions. The main parameters that affect the model's accuracy are max depth, which limits the tree's height to avoid overfitting, and min samples split, which defines the minimal amount of samples necessary to divide an internal node.

2) *Random Forest Classifier*: It secures high performance and scalability by randomizing both feature selection and data sampling, which makes it able to handle huge data sets

TABLE I
MATRIX EVALUATION VALUES

	Greenberg(Nonlinear)	Underwood(Nonlinear)	CSUF Model(Linear)
r	0.998	0.978	0.95

successfully. This type of classifier is especially appreciated for its noise resistance and its ability to learn from a seemingly complex dataset from a different domain. The tuning of specifics like the number of trees in the forest and their maximal depth can be done to make it work even better.

3) *GaussianNB Classifier*: This classifier is especially useful for datasets where the continuous variables have a Gaussian distribution. Moreover, a GNB model estimates the likelihood for every class of the input features, and then uses these probabilities to create the predictions. It is very efficient programmatically and shows good results, even with a small number of records, which makes it a great tool for the preliminary research in classification tasks.

4) *LGBM Classifier*: Traditional boosting methods, which grow trees level-wise, are not at all like LGBM, which constructs trees in a leaf-wise approach first, so it can achieve better accuracy with fewer iterations. The LGBM classifier is an excellent choice for large-scale datasets because it uses its memory efficiently and can perform more than one computation at a time. Users can change the learning rate, number of leaves, or maximum depth of the model with the help of hyperparameter tuning options. By this, they can individually adapt the model to the characteristic features of the data without losing its predictive value.

5) *Catboost Classifier*: A gradient boosting library designed for categorical feature handling. It utilizes techniques like ordered boosting and target encoding to effectively handle categorical data, offering robustness and competitive performance.

H. Evaluation Matrix

The Evaluation Matrix primarily compares the efficacy of the three models: Greenberg (Nonlinear), Underwood (Nonlinear), and the CSUF Model (Linear) using the relevant correlation coefficients (r). The Greenberg model correlates strongly with the relevant correlation coefficient (0.998), indicating a statistically excellent match and accurate predicting of anomalies. The Underwood model has a good correlation coefficient (0.978), implying that it may successfully foresee prospective concerns, despite being a nonlinear model. The CSUF Model has the potential to be an effective detector, despite its relatively high correlation value of 0.95. This correlation is smaller than that of the nonlinear models, clearly highlighting the probable benefits of the latter class of algorithms in detecting anomalies.

I. Experiments and Findings

1) *Data Splitting*: The cleaned and preprocessed dataset is further divided into three sets, as shown below. These are the training, validation, and testing subsets. The training component teaches the deep learning models, followed by

the validation subset, which is used to learn and mitigate hyperparameter overfitting, and lastly, the testing subset is used to assess the final model's performance.

2) *Confusion Matrix*: The confusion matrix is an important assessment tool for creating attack detection models utilizing machine and deep learning techniques. It gives a thorough analysis of the model's performance by comparing anticipated and actual classifications.

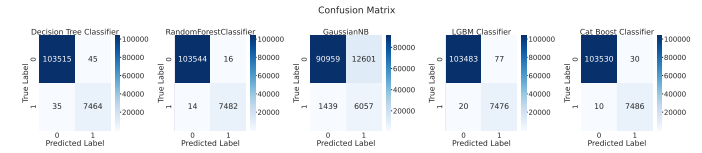


Fig. 2. Confusion Matrices Models Analytical Result

In this study, it was observed that the CatBoost Classifier was the key model to detect threats, functioning with nearly perfect accuracy. A complete comparison examination of various machine learning algorithms based on attack results makes it easier to evaluate the system's efficacy through anomaly detection.

In terms of attack detection strategies, the confusion matrix is an important tool for determining the accuracy and reliability of ML and DL systems for detecting anomalies in supervised datasets. As a result, this method produces safer and more efficient systems.

When constructing attack detection models, the unit acts as a sort of confusion matrix, ensuring that the machine learning or deep learning technique is not only accurate but also dependable and resilient in finding and reducing abnormalities in human-assisted data sets. Indeed, this is how more secure and efficient systems emerge.

J. CatBoost Classifier Math Model for Attack Detection

1) Decision Trees and Boosting:

- CatBoost builds a collection of decision trees.
- Every single tree grows in a step-by-step process, fixing mistakes from the trees that came before it.
- The final forecast is a combination of predictions from all individual trees, each with its own weight.

2) *Objective Function*: In a classification task, the goal is to minimize the loss function L , with y as the true label and \hat{y} as the model's prediction.

- CatBoost uses the cross-entropy loss function specifically for binary classification tasks.
- y denotes the actual label for the given instance.
- \hat{y} stands for the expected chance for a specific occurrence, where $\hat{y} = \sigma(z)$ defined as the sigmoid function is.

The cross-entropy loss function L can be expressed as:

$$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

3) Gradient Boosting:

- The model is built step by step, where each tree aims to reduce the loss function.
- \hat{y}_0 refers to the forecast generated from the original trees.
- η represents the learning rate, which controls the impact of every tree.

4) *Handling Categorical Features:* CatBoost effectively deals with categorical features by transforming them into numerical representations using methods like target encoding or one-hot encoding, which helps prevent overfitting.

5) Final Prediction:

- For a novel instance x , the ultimate prediction \hat{y} the process of achieving this entails utilizing the sigmoid function to merge the weighted results of trees.
- $\sigma(z)$ the sigmoid function is recognized as.

K. Steps in Training the CatBoost Classifier

1) Initialization:

- The designated parameters (such as number of trees, learning rate, etc.) are used to initialize the model.

2) Sequential Training:

- For $t = 1$ to T (number of trees):
 - Compute the negative gradient (pseudo-residuals) based on the current model's predictions.
 - Fit a decision tree h_t to the pseudo-residuals.
 - Adjust the model by incorporating the predictions of the latest tree multiplied by the learning rate. η .

3) *Model Output:* The final result of the model comes from combining all tree predictions and then running them through the sigmoid function to obtain probabilities.

V. EXPERIMENTAL RESULTS

Table 2 compares five supervised learning models: Decision Tree Classifier, Random Forest Classifier, Gaussian NB, LGBM Classifier, and CatBoost Classifier, with experimental outcomes. An methodology was developed for each model by combining the best encoding methods, PCA (Principal Component Analysis) for dimension reduction, optimal feature selection, and different hyperparameter tuning strategies. The Decision Tree Classifier, Random Forest Classifier, and CatBoost Classifier have been identified as the models with the highest accuracy and performance; they are the top three out of five, and their accuracy is fairly similar. Nonetheless, the Random Forest Classifier achieved the highest accuracy score of 99.97%. These three top-performing models underwent cross-validation using the k-fold approach ($k = 5$). Random Forest Classifier was found to be underperforming, as it was on the left in the first, fourth, and fifth folds when compared to the other two. However, during the five folds, the Decision Tree Classifier and CatBoost Classifier consistently produced the same results.

Table 3 compares, an exploration was also undertaken, deep learning models like ANN, CNN and RNN. In addition, the accuracy of RNN was higher from rest two with 99.95% while model processing time was higher in respect to RNN

TABLE II
ACCURACY REPORT: SUPERVISED MACHINE LEARNING MODELS

Models	Decision Tree Classifier	Random Forest Classifier	GaussianNB	LGBM Classifier	Catboost Classifier
Accuracy Score	99.93	99.97	87.35	99.91	99.96

whose accuracy was 99.95% which was Execution Time: 453.9617456730002 seconds in respect to ANN which was Execution Time: 273.9491973869999 seconds with accuracy was 99.94%.

TABLE III
ACCURACY REPORT: DEEP LEARNING MODELS

Models	CNN	ANN	RNN
Accuracy Score	95.66	99.94	99.95

The Table.4 provided represents a comparison of classifiers based on K-Fold cross validation. K-Fold cross validation is a machine learning approach that includes dividing the data into K s subsets, with each test used to train the learner and the remainder to train. In K-Fold, the average score across all folds provides a more accurate evaluation of model performance.

TABLE IV
CROSS VALIDATION SCORE-ACCURACY SCORE MODEL

Cross Validation Score for highest accuracy score model						
Models	KFold 1	KFold 2	Kfold 3	KFold 4	KFold 5	Average KFold Score
Decision Tree Classifier	0.94	1.00	1.00	1.00	1.00	0.98
Random Forest Classifier	0.90	1.00	1.00	1.00	0.91	0.96
Catboost Classifier	0.94	1.00	1.00	1.00	1.00	0.98

Further, the top-performing models were executed in Table 3. with the help of k-fold cross-validation. The Decision Tree Classifier showed K-Fold scores of 0.94, 1.0, 1.0, 1.0, and 0.98, so the mean k-fold value, in this case, was 0.98. The Random Forest Classifier which achieved a minimum score of 0.90, a maximum score of 1.0, and a median score of 0.91 in the five iterations received an average k-fold of 0.96. The Catboost Classifier also reached a score of 0.94, 1.0, 1.0, 1.0, and 0.98, through the folds, resulting in an average k-fold score of 0.98. (Ko2)

Fig.3, as depicted, is a bar chart, which visually communicates the classification metrics (Precision, Recall, and F1-Score) for five different models: Decision Tree Classifier, RandomForestClassifier, GaussianNB, LGBM Classifier, and Cat Boost Classifier. One of these is the Catboost Classifier.

Based on the chart provided, the key findings are:

- 1) *Model Performance Comparison:* The differences in the classification abilities of the five models Decision Tree Classifier, RandomForestClassifier, GaussianNB, LGBM Classifier, and Cat Boost Classifier are presented in the chart. The Precision, Recall, and F1-Score metrics for each model are presented in the chart.

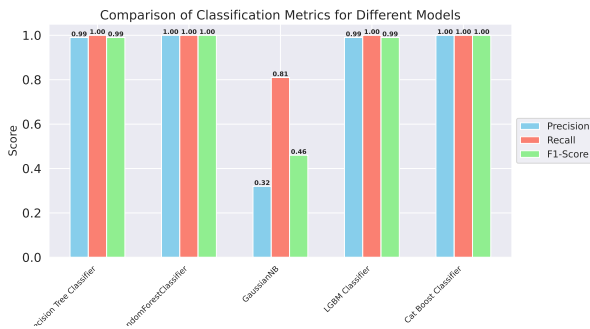


Fig. 3. Comparison of Classification Metrics - Different Models

- 2) Precision Scores: All models have very high precision rates, namely 1.00 and 0.99, 1.00, 0.99, and 1.00 for Decision Tree Classifier, RandomForest Classifier, LGBM Classifier, and Cat Boost Classifier. However, the GaussianNB model is less accurate as its recall for class 0 is only 0.98.
- 3) Recall Scores: The Decision Tree Classifier, RandomForestClassifier, LGBM Classifier, and Cat Boost Classifier also have very good Recall (1.00) for both groups of Class 0 and Class 1. The GaussianNB model achieved a higher level of recall for Class 1 and a lower level for Class 0 which are 0.88 and 0.81 respectively.
- 4) F1-Score Comparison: The F1-Score, which integrates Precision and Recall, exhibits a nearly identical trend. The Decision Tree Classifier, RandomForestClassifier, LGBM Classifier, and Cat Boost Classifier have F1-Scores of 1.00, 1.00, 1.00, and 1.00 for Class 0 respectively and 0.99, 1.00, 0.99, and 1.00 for Class 1. Nevertheless, the GaussianNB model has a somewhat lower F1-Score of 0.93 for Class 0 and 0.46 for Class 1.
- 5) Overall Performance: According to the Precision, Recall, and F1-Score metrics, the Decision Tree Classifier, the RandomForestClassifier, the Light Gradient Boosting Machine (LGBM) Classifier, and the Cat Boost Classifier seem to have almost the same and pretty good classification abilities, while the GaussianNB model seems to have the least efficient performance, especially for Class 1.

VI. CONCLUSION

This research looked into improving attack detection systems in supervised datasets using deep learning for detecting anomalies. Through the use of complex machine learning frameworks, the suggested method showed significant improvements in detecting and addressing anomalies in complicated data environments. The deep learning approach outperformed traditional methods in terms of accuracy and operational efficiency, as confirmed by a comprehensive evaluation matrix. This matrix provided a comprehensive evaluation of the performance measures, highlighting the durability and reliability of the suggested structures. Together, the results reveal

the possibilities of using deep learning techniques to advance anomaly detection. By enhancing the ability to detect attacks, this research helps in the development of increasingly secure and resilient systems, setting the stage for future advancements in cybersecurity and machine learning technologies.

ACKNOWLEDGMENT

Alexy Nazarov, our wonderful and dedicated supervisor for his professional guidance which really directed us throughout this at times rocky terrain with a steady hand. His incredible experience and vision shaped our research, as did his careful eye for elevating its rigor. We also thank Alireza Nik Aein Koupaei, a dedicated second author who played an important part in the realization of this article because his impressive jewett & co-pro max power experience and devotion. Finally, we would like to give thanks for the helpful feedback and recommendations from faculty within Computer Science at Moscow Institute of Physics and Technology (MIPT) Department of Radio Engineering and Cybernetics 141701, Institutsky Lane, 9, Moscow, Russian Federation that helped shape an intellectually diverse conversation around our work. Likewise, we appreciate the faculty members for their diverse insights in class. None of this work would have been possible without the input and advice from these fantastic people.

REFERENCES

- [1] G. Breda and M. Kiss.: Overview of Information Security Standards in the Field of Special Protected Industry 4.0 Areas & Industrial Security. *Procedia Manufacturing*, vol. 46, pp. 580-590 (2020)
- [2] R. Shree and K. Sandhu.: A Multi-Objective Decision Assistance System for Selecting Security Controls Based On Simulation. 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2023, pp. 1754-1756 (2023)
- [3] A. Amit, I. Matherly, J. Hewlett, W. Xu, Z. Meshi, Y., & Weinberger, Y.: Machine learning in cyber-security-problems, challenges and data sets. *arXiv*, 1812. 07858. (2018)
- [4] Maxime Labonne.: Anomaly-based network intrusion detection using machine learning. *Cryptography and Security [cs.CR] Institut Polytechnique de Paris*, (2020)
- [5] Lidong Wang, Reed L. Mosher, Patti Duett, Terril C. Falls.: Data Analytics of Network Intrusion Based on Deep Neural Networks with Weights Initialized by Stacked Autoencoders and Deep Belief Networks. *IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, pp.511-515, (2023)
- [6] K. Alrawashdeh and C. Purdy.: Toward an online anomaly intrusion detection system based on deep learning. *Proc. IEEE 15th Int. Conf. Mach. Learn. Appl. (ICMLA)*, pp. 195-200, (2016)
- [7] Y. Imamverdiyev and F. Abdullayeva.: Deep learning method for denial of service attack detection based on restricted Boltzmann machine. *Big Data*, vol. 6, no. 2, pp. 159-169, (2018)
- [8] M. H. Haghighat and J. Li.: Intrusion detection system using votingbased neural network. *Tsinghua Sci. Technol.*, vol. 26, no. 4, pp. 484-495, (2021)
- [9] Y. Yang et al.: ASTREAM: Data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment. *IEEE Trans. Netw. Sci. Eng.*, (2022)
- [10] C. Yin, Y. Zhu, J. Fei and X. He.: A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, vol. 5, pp. 21954-21961, (2017)
- [11] C. Park, J. Lee, Y. Kim, J. -G. Park, H. Kim and D. Hong.: An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks. in *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2330-2345.

- [12] B. Dong and X. Wang.: Comparison deep learning method to traditional methods using for network intrusion detection. 8th IEEE International Conference on Communication Software and Networks (ICCSN), pp. 581-585, (2016)
- [13] Y. Xue.: Research on Time Series Anomaly Detection Based on Graph Neural Network. IEEE International Conference on Electrical, Automation and Computer Engineering (ICEACE), Changchun, China, pp. 1670-1674, (2023)
- [14] M. Munir, S. Siddiqui, A. Dengel et al.: Deepan T: A deep learning approach for unsupervised anomaly detection in time series[J]. IEEE Access, vol. 7, pp. 1991-2005, (2018)
- [15] A. Deng and B. Hooi.: Graph neural network-based anomaly detection in multivariate time series[C]. Proceedings of the AAAI Conference on Artificial Intelligence, pp. 4027-4035, (2021)
- [16] S. Naseer et al.: Enhanced Network Anomaly Detection Based on Deep Neural Networks. in IEEE Access, vol. 6, pp. 48231-48246.
- [17] M. Luo, L. Wang, H. Zhang and J. Chen.: A research on intrusion detection based on unsupervised clustering and support vector machine. Proc. 5th Int. Conf. Inf. Commun. Secur. (ICICS), pp. 325-336.
- [18] R. C. Aygun and A. G. Yavuz.: Network anomaly detection with stochastically improved autoencoder based models. Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud), pp. 193-198.
- [19] S. Mukkamala, G. Janoski and A. Sung.: Intrusion detection using neural networks and support vector machines. Proc. Int. Joint Conf. Neural Netw. (IJCNN), pp. 1702-1707, (2002)
- Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
- [20] Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
- [21] LNCS Homepage, <http://www.springer.com/lncs>, last accessed 2023/10/25

Call for Papers: International Journal of Computer Science and Information Security (IJCSIS)

Scope and Topics: The International Journal of Computer Science and Information Security (IJCSIS) invites researchers, practitioners, and academicians to submit original, unpublished contributions covering a wide range of topics in the field of computer science and information security. We welcome submissions that include but are not limited to:

- Computer and Network Security
- Cryptography and Data Security
- Information Assurance
- Artificial Intelligence and Machine Learning in Security
- Cybersecurity Policies and Standards
- Security in Cloud Computing
- Internet of Things (IoT) Security
- Blockchain and Distributed Ledger Technologies
- Secure Software Development
- Privacy-Enhancing Technologies

Submission Guidelines:

- Manuscripts must be original and not currently under consideration for publication elsewhere. All papers should be submitted in English.
- The manuscript should follow the IJCSIS formatting guidelines, available on our website: <https://sites.google.com/site/ijcsis/ijcsis>
- Submissions must include the title of the paper, abstract, keywords, and full contact information for all authors.
- Papers should be submitted via our submission system here: <https://sites.google.com/site/ijcsis/submit-paper>

Important Dates:

- **Paper Submission Deadline:** Monthly, 2025
- **Notification of Acceptance:** Within TWO weeks, 2025
- **Final Manuscript Due:** Monthly, 2025
- **Publication Date:** Monthly Issue

Review Process: All submitted papers will undergo a rigorous peer-review process by the IJCSIS editorial board and selected external reviewers. Authors will be notified of the review results by the notification of acceptance date.

Contact Information: For further inquiries, please contact the editorial office at:

Email: ijcsiseditor@gmail.com

We look forward to receiving your submissions!

Website: <https://sites.google.com/site/ijcsis/ijcsis>