# Detecting anomalies in network traffic using machine learning techniques

Tuleubay Safiullin
*Department of Mathematical Modelling and Data Analysis*
*Belarusian State University*
Minsk, Belarus
tuleubay.safiullin@mail.ru

Abramovich Michael
*Department of Mathematical Modelling and Data Analysis*
*Belarusian State University*
Minsk, Belarus
AbramovichMS@bsu.by

***Abstract*—The problem of anomaly detection in network traffic using machine learning and neural network methods is considered. Logistic regression, support vector method, random forest, gradient boosting, fully connected neural network and recurrent LSTM neural network were used as classification models for anomaly detection. A grid search for optimal parameters on cross-validation of these models was carried out. The architectures of the fully connected and recurrent LSTM neural network were developed. One-Class SVM, isolation Forest, Local Outlier Factor, Elliptic Envelope methods of one-class classification were also applied. The application of ensembles of classifiers for detection of anomalous traffic, in particular, built using the stacking procedure, is considered. The efficiency of all algorithms is analysed.**

***Keywords—anomaly detection; machine learning; neural networks***

## I. Description of the test data set

The NSL-KDD dataset [1] was used for software testing. This dataset stands as an industry standard, renowned for its effectiveness in assessing the performance of adaptive algorithms across a spectrum of network protocols, including TCP, UDP, and ICMP.

To provide a comprehensive understanding of the dataset's composition and attributes, we relied on the detailed information presented in [2]. This valuable resource furnished us with a comprehensive catalogue of attributes, complete with their corresponding variable types and the range of possible values. There are 43 features in total, including the target variable.

The training dataset contains 125,973 observations. The test dataset contains 22,544 observations. A noteworthy aspect of the NSL-KDD dataset that bolstered the robustness of our testing was its balanced nature. This equilibrium was reflected in the training dataset, which harboured 67,343 representatives of legitimate traffic and 58,630 representatives of anomalous traffic. This balanced distribution ensured that our software was rigorously tested against both normal and anomalous network activities, enhancing its adaptability and effectiveness.

As we delved into the realm of feature selection, a meticulous approach was employed to identify the most informative attributes among the dataset's numerical features. Leveraging the power of L1-regularized logistic regression, a cutting-edge technique in feature selection, we carefully curated a subset of 15 features. This judicious selection process was undertaken to optimize our software's performance and ensure it focused on the most relevant aspects of the data.

To ensure the robustness and consistency of our data preprocessing pipeline, an additional step was taken to further normalize the traits. This meticulous process involved the application of the *MinMaxScaler*, an essential component from the vast arsenal of tools offered by the scikit-learn library. *MinMaxScaler* transforms the numerical feature X by the formula:

$$(x - X_{min}) / ( X_{max} - X_{min}) \quad (1)$$

where $X_{min}$, $X_{max}$ – the highest and the lowest value of the trait, respectively.

Thus, the numerical feature will take values on the interval [0, 1].

From the categorical features, 30 informative features were selected using the mutual information criterion [3]. Thus, the total number of informative features was 45.

## II. Anomalous traffic detection using classification methods and neural networks

Logistic regression, support vector method, random forest, gradient boosting, fully connected neural network and recurrent LSTM neural network were used as classification models. A cross-validation grid search for optimal parameters for classification models was carried out.

In order to save time, a subset of the dataset consisting of 20000 observations was taken for training. The proportion of data with respect to the target variable was kept.

A fully connected neural network was trained on the entire training sample. The neural network is 4 fully connected layers (with 30, 60, 90 and 1 neurons in each layer, respectively). The last layer is the output layer, so it has only one neuron, since we are solving a binary classification problem. Each inner layer is followed by a ReLU activation function. The output layer is followed by a sigmoid activation function. After the first and third fully connected layers are dropout layers with a $p$ parameter equal to 0.5 (to prevent overtraining). The training lasted for 20 epochs. The size of the batches is 32 and the optimiser is ADAM. Early stopping with patience parameter equal to 5 is also used as a regularisation technique. The loss function is binary cross entropy. Computations were performed on CPU.

The architecture of the LSTM recurrent neural network is two LSTM layers with 60 and 30 neurons, respectively, each followed by a dropout layer with the parameter $p$ equal to 0.5. The activation function of the output layer is sigmoid, the optimiser is ADAM, the batch size is 64, and the number of epochs is 20. Early stopping with the parameter patience equal to 5 is used. The loss function is binary cross-entropy. Computations were performed on CPU.

The results of classification models and neural networks are presented in tables I and II.

TABLE I. PERFORMANCE RESULTS OF CLASSIFICATION MODELS AND NEURAL NETWORKS ON A TRAINING DATASET

| Algorithm | Metric | | | | | |
|---|---|---|---|---|---|---|
| | *Accuracy* | *Precision* | *Recall* | *F1* | *Training time* | *Prediction time* |
| Log Reg | 0.96 | 0.97 | 0.95 | 0.96 | 341 ms | 15.3 ms |
| SVM | 0.94 | 0.90 | 0.98 | 0.94 | 26.2 sec | 24.3 sec |
| Random Forest | 1 | 1 | 1 | 1 | 2.26 sec | 214 ms |
| Light GBM | 1 | 1 | 1 | 1 | 442 ms | 68.9 ms |
| Neural Network | 0.98 | 1 | 0.96 | 0.98 | 1 min, 53 sec. | 6, 09 sec |
| LSTM | 0.99 | 0.99 | 0.99 | 0.99 | 3 min, 25 sec | 10.3 sec |

TABLE II. PERFORMANCE RESULTS OF CLASSIFICATION MODELS AND NEURAL NETWORKS ON THE TEST DATASET

| Algorithm | Metric | | | | | |
|---|---|---|---|---|---|---|
| | *Accuracy* | *Precision* | *Recall* | *F1* | *Prediction time* | *Optimal parameters* |
| Log Reg | 0.74 | 0.91 | 0.60 | 0.72 | 19.4 ms | C = 0.05<br>L1_ratio = 0.3 |
| SVM | 0.86 | 0.90 | 0.85 | 0.88 | 28.4 sec | C = 0.01<br>Gamma = 10 |
| KNN | 0.78 | 0.95 | 0.65 | 0.77 | 33.5 sec | N_neighbors = 1 |
| Random Forest | 0.78 | 0.97 | 0.63 | 0.76 | 326 ms | Max_depth = 11<br>Max_features = 0.5<br>N_estimators = 100 |
| Light GBM | 0.77 | 0.97 | 0.62 | 0.75 | 76 sec | Learning_rate = 0.5<br>Max_depth = 7<br>N_estimators = 50<br>Num_leaves = 71 |
| Neural Network | 0.74 | 0.97 | 0.57 | 0.71 | 1.34 sec | |
| LSTM | 0.76 | 0.92 | 0.64 | 0.75 | 2.61 sec | |

Drawing insights from the comprehensive tables I and II, a salient observation emerges: the support vector method emerged as the standout performer in terms of classification effectiveness. This method's prowess in discerning patterns and making accurate predictions cannot be denied, but it does come with a trade-off that deserves attention - it happens to be the most time-consuming among the algorithms evaluated. While its effectiveness is commendable, the time factor poses a challenge in real-world applications where efficiency is paramount. It should be noted that there is a prospect of optimising its performance by selecting a more appropriate architecture of a fully connected neural network.

Based on the results of classification of the training sample, we can assume that there is an overtraining effect, but it is also possible that the distribution of data in the test sample differs from the distribution of data in the training sample (splitting the training sample into a training and validation sample showed good results in the validation sample).

## III. DETECTING ANOMALOUS TRAFFIC USING ONE-CLASS CLASSIFICATION ALGORITHMS

In the pursuit of identifying and mitigating anomalous traffic patterns effectively, a comprehensive array of One-Class classification algorithms was thoughtfully employed. This strategic selection of algorithms not only diversified the approach but also ensured a holistic evaluation of their performance in the task at hand. The algorithms chosen for this pivotal task included the One-Class support vector method [4], the isolation forest [5], the local outlier factor [6], and the multivariate normal distribution (specifically, the Elliptic Envelope class of the scikit-learn library)[7]. Each of these algorithms brought its unique strengths and capabilities to the table, contributing to the overall robustness of the anomaly detection framework.

The One-Class support vector method, a well-established technique, was given a special role in this ensemble of classifiers. It was entrusted with the responsibility of training on legitimate observations spanning the entire dataset. This approach ensured that this algorithm had access to a comprehensive view of normal traffic patterns, allowing it to establish a strong baseline for anomaly detection.

The other algorithms were trained on a subset of 20000 observations, which contains only 5% of the anomalous class objects. The performance results of the one-class classification models for the training and test samples are shown in tables III and IV, respectively.

TABLE III. PERFORMANCE RESULTS OF ONE-CLASS CLASSIFICATION MODELS ON TRAINING SAMPLE

| Algorithm | Metric | | | | | |
|---|---|---|---|---|---|---|
| | *Accuracy* | *Precision* | *Recall* | *F1* | *Training time* | *Prediction time* |
| One-Class SVM | 0.96 | 0.94 | 0.97 | 0.96 | 1 min, 36 sec | 8.23 sec |
| Isolation Forest | 0.97 | 0.72 | 0.72 | 0.72 | 5.23 sec | 1.39 sec |
| Local Outlier Factor | 0.93 | 0.23 | 0.19 | 0.21 | 27.8 sec | 29.5 sec |
| Elliptic Envelope | 0.08 | 0.04 | 0.82 | 0.08 | 6.73 sec | 91.2 ms |

TABLE IV. PERFORMANCE RESULTS OF ONE-CLASS CLASSIFICATION MODELS ON THE TEST SAMPLE

| Algorithm | Metric | | | | | |
|---|---|---|---|---|---|---|
| | *Accuracy* | *Precision* | *Recall* | *F1* | *Training time* | *Prediction time* |
| One-Class SVM | 0.89 | 0.90 | 0.90 | 0.90 | 9.22 sec | Nu = 0.05 Gamma = 5 |
| Isolation Forest | 0.72 | 0.93 | 0.56 | 0.70 | 1.58 sec | Contamination = 0.05 |
| Local Outlier Factor | 0.53 | 0.74 | 0.27 | 0.39 | 31.3 sec | Contamination = 0.05 |
| Elliptic Envelope | 0.43 | 0.50 | 0.74 | 0.60 | 133 ms | Contamination = 0.05 |

As shown in tables III and IV, the One-Class support vector method had the highest values of machine learning metrics.

## IV. ANOMALOUS TRAFFIC DETECTION USING ENSEMBLES OF CLASSIFIERS

An ensemble of classification algorithms, including the support vector method, k-nearest neighbours method, and One-Class support vector method, was thoughtfully applied to the task of detecting anomalous traffic patterns. These algorithms were chosen after an exhaustive evaluation process, and their individual performances indicated that they were well-suited for the task at hand. However, in order to harness the collective power of these classifiers, they were ingeniously combined into a comprehensive ensemble model. The decision-making mechanism for this ensemble was implemented using a simple voting method, ensuring that no single classifier would dominate the final outcome.

As the experimentation continued, a keen eye was kept on the performance metrics, particularly the recall metric. It was during this meticulous analysis that a noteworthy observation was made: the k-nearest neighbours method was exhibiting a lower recall value compared to the other classifiers. Such a revelation could not be overlooked, prompting further investigation and adjustments to the established rule. Consequently, the rule was strategically modified to ensure that if at least one classifier within the ensemble voted in favour of label 1, it would be the prevailing choice.

Continuing on the journey of refining the ensemble model, a novel approach was explored - the concept of stacking classifiers. Stacking, a powerful technique in machine learning, involves training multiple classifiers on the same dataset and combining their predictions to boost overall performance. However, an intriguing challenge arose when implementing this approach. The One-Class support vector method, due to its unique characteristics, necessitated a minimum number of anomalous observations in the training sample. As a result, it could not be included in the stacking ensemble as originally planned. To address this, an innovative solution was devised: the inclusion of a random forest classifier in place of the One-Class support vector method. Remarkably, the random forest exhibited exceptional performance and complemented the existing ensemble methods seamlessly, reaffirming its suitability for the task at hand.

It's important to note that throughout this meticulous process, consistency in parameter selection was maintained. The optimal parameters discovered in the earlier stages of experimentation were diligently employed across all models, ensuring fairness and accuracy in the comparative analysis.

To further validate the models and assess their generalizability, a well-considered subset of the data comprising 20,000 observations was judiciously utilized for training. This approach allowed for efficient model training while preserving the integrity of the broader dataset. The results of the above approaches are presented in tables V and VI.

TABLE V. RESULTS OF THE ALGORITHMS ON THE TRAINING SAMPLE

| Algorithm | Metric | | | |
|---|---|---|---|---|
| | *Accuracy* | *Precision* | *Recall* | *F1* |
| Simple voting | 0.97 | 0.96 | 0.98 | 0.97 |
| Voting taking into account KNN's low recall value | 0.95 | 0.91 | 1 | 0.95 |
| Stacking | 0.99 | 0.99 | 0.99 | 0.99 |
| Simple voting | 0.97 | 0.96 | 0.98 | 0.97 |

TABLE VI. RESULTS OF ALGORITHMS WORK ON THE TEST SAMPLE

| Algorithm | Metric | | | |
|---|---|---|---|---|
| | *Accuracy* | *Precision* | *Recall* | *F1* |
| Simple voting | 0.87 | 0.92 | 0.84 | 0.88 |
| Voting taking into account KNN's low recall value | 0.89 | 0.89 | 0.92 | 0.91 |
| Stacking | 0.78 | 0.95 | 0.65 | 0.77 |
| Simple voting | 0.87 | 0.92 | 0.84 | 0.88 |

When examining the somewhat subpar performance of the stacking-based approach, it becomes evident that the decision to opt for a random forest instead of the One-Class SVM might have played a pivotal role. This strategic choice, while

motivated by the random forest's superior performance on the NSL-KDD dataset, may have inadvertently contributed to the lacklustre outcome. It's worth noting that the One-Class SVM, despite its relative underperformance on this specific dataset, might have offered a unique perspective that could have complemented the other classifiers within the ensemble.

An attempt has also been made to bring five classification methods instead of three into the above two voting methods. Random forest and gradient boosting were added to the above three methods.

The inclusion of random forest, renowned for its robustness and versatility, added a layer of stability to the ensemble. Its ability to handle complex data patterns and inherent noise made it a valuable addition to our arsenal. Meanwhile, gradient boosting, a powerful ensemble learning technique, injected a dose of boosting, which is particularly effective in refining the performance of individual classifiers.

The performance results of the ensembles based on the 5 classifiers are presented in tables VII and VIII respectively.

TABLE VII. ENSEMBLE RESULTS ON THE TRAINING SAMPLE (5 CLASSIFIERS)

| **Algorithm** | **Metric** | | | |
|---|---|---|---|---|
| | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1*** |
| Simple voting | 0.97 | 0.96 | 0.98 | 0.97 |
| Voting taking into account KNN's low recall value | 0.95 | 0.91 | 1 | 0.95 |
| Stacking | 0.99 | 0.99 | 0.99 | 0.99 |
| Simple voting (5 methods) | 1 | 1 | 1 | 1 |
| Low recall voting with KNN (5 methods) | 0.93 | 0.87 | 1 | 0.93 |

TABLE VIII. ENSEMBLE RESULTS ON THE TEST SAMPLE (5 CLASSIFIERS)

| **Algorithm** | **Metric** | | | |
|---|---|---|---|---|
| | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1*** |
| Simple voting | 0.87 | 0.92 | 0.84 | 0.88 |
| Voting taking into account KNN's low recall value | 0.89 | 0.89 | 0.92 | 0.91 |
| Stacking | 0.78 | 0.95 | 0.65 | 0.77 |
| Simple voting (5 methods) | 0.80 | 0.95 | 0.68 | 0.79 |
| Low recall voting with KNN (5 methods) | 0.90 | 0.89 | 0.95 | 0.92 |

As can be seen from the results in Tables VII and VIII, increasing the number of classifiers in the ensemble to 5 improved the classification performance of legitimate and anomalous network traffic.

## REFERENCES


[1] UNB. (2023, Dec. 14). NSL-KDD dataset [Online]. Available: https://unb.ca/cic/datasets/nsl.html

[2] G. Saporito, “NSL-KDD Features” [Online], Sep 16 2019. Available: https://docs.google.com/spreadsheets/d/1oAx320Vo9Z6HrBrL6BcfLH6sh2zIk9EKCv2OlaMGmwY/edit#gid=0

[3] Asir, D. Literature Review on Feature Selection Methods for High-Dimensional Data / D. Asir, S. Appavu, E. Jebamalar // International Journal of Computer Applications, V. 136, No 1. –2016. – P. 9–17.

[4] Lee G., Scott C. D. The one class support vector machine solution path //2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07. – IEEE, 2007. – T. 2. – C. II-521-II-524.

[5] Liu F.T., Ting K.M., Zhou Z.H. Isolation-based anomaly detection //ACM Transactions on Knowledge Discovery from Data (TKDD). – 2012. – V. 6. – №1 – P. 1-39.

[6] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. 2000. LOF: identifying density-based local outliers. In ACM sigmod record. – 2016. – V 6. – P. 8-12.

[7] Rousseeuw P.J., Driessen K.V. A fast algorithm for the minimum covariance determinant estimator // Technometrics. – 1999. – V. 41. – № 3. – P. 212-223.