CNN-BiLSTM Hybrid Model for Network Anomaly Detection in Internet of Things

Bauyrzhan Omarov[1], Omirlan Auelbekov[2], Azizah Suliman[3], Ainur Zhaxanova[4]
Al-Farabi Kazakh National University, Almaty, Kazakhstan[1]
Institute Information and Computational Technologies CS MES RK, Almaty, Kazakhstan[2]
INTI International University, Kuala Lumpur, Malaysia[3]
M. Auezov South Kazakhstan University, Shymkent, Kazakhstan[4]

***Abstract*—Anomaly detection in internet of things network traffic is a critical aspect of intrusion and attack detection, in which a deviation from typical behavior signals the existence of malicious or inadvertent assaults, faults, flaws, and other issues. The necessity to examine a large number of security events to identify anomalous behavior of smart devices adds to the urgency of addressing the challenge of picking machine-learning and deep learning models for identifying anomalies in network traffic. For the challenge of binary data categorization, a software implementation of an intrusion detection system based on supervised-learning algorithms has been completed. The UNSW-NB15 open dataset, which contains 2,540,044 records - vectors of TCP/IP network connection signals and their associated class labels are used to train and test the system. This research compares different machine-learning models and proposes CNN-BiLSTM hybrid model for IoT network intrusion detection. The metrics for measuring the quality of classification and the running duration of algorithms for different ratios of train and test samples are the result of the built framework testing.**

***Keywords*—IoT; internet of things; network anomalies; network security; anomaly attack; machine learning; supervised learning; UNSW-NB15**

I. Introduction

The Internet of Things (IoT) is a network of electronic devices with built-in technologies that allow them to connect with one another and with the outside world. The Internet of Things (IoT) idea has been ingrained in our daily lives, presenting consumers with new options ranging from home automation to medical equipment [1]. IoT devices can effectively gather, analyze, and send massive volumes of data thanks to ultra-high-speed wireless networks and a sophisticated electronic database. Microelectronic improvements combined with low power consumption have made it increasingly easier to operate IoT devices in remote places with minimum physical oversight and maintenance [2]. Although IoT devices appear to be innocent, they are not without security and privacy concerns, since the present IoT framework contains several risks and vulnerabilities.

According to analysts, the Internet of Things will soon become a part of everyday life. According to IDC, the worldwide market for relevant solutions was valued at $ 646 billion in 2018, and it will surpass the trillion-dollar level by 2022. All of this pushes us to learn more about the security of IoT systems [3].

Automated methods, for managing and interpreting the data are required due to the complexity and diversity of data created by heterogeneous devices. Therefore, machine learning technologies that enable the development of profiles of device behavior in the network, anomalies detection and prediction of abnormal scenarios, claim the role of technology in automatically detecting dependencies and connecting devices [1].

Peripherals, sensors, gateways based on industrial communication protocols, centralized data storage; and end devices users interact with the four major pieces of an Internet of Things system. The addition of big data tools and systems based on machine learning technologies to this setup results in the creation of a new block (Fig. 1) that is responsible for the quality of data and, as a result, the quality of the system's choices and alerts. Furthermore, centralized or cloud data storage expenses are decreased due to adaptive prioritizing and filtering of the information [2-3].

The difficulty with the advancement of attacks is that it is getting more difficult to detect and distinguish between legal and malicious network data. Intrusion detection systems (IDS) [4] do a good job of identifying malicious traffic, but they must be regularly updated with rule sets and upgrades in order to remain relevant when it comes to detecting changing threat vectors. Even if the major corporations disclose fresh sets of regulations on a regular basis, this may not be enough. As a result, the question of employing different methods for identifying irregular incursions becomes significant. The use of machine learning algorithms [5] is one of these ways. Machine learning is used because it can help automate threat processing and keep the system up to date by studying and detecting threats. That is, the software is taught to detect different types of communications in order to classify them and reject or skip them [6].

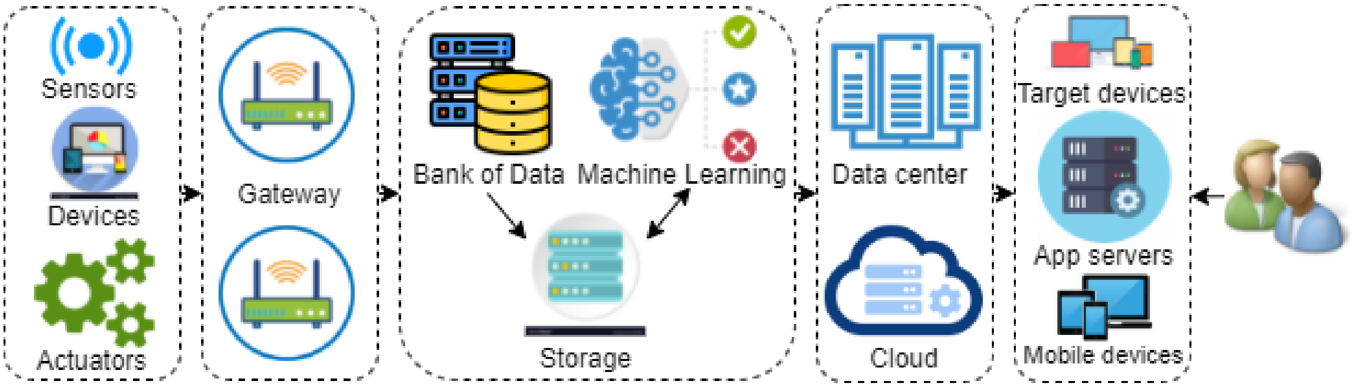

Fig. 1. Comparison of machine learning methods in network anomaly detection.

The following is a reminder of the paper. The next section discusses relevant work on detecting Internet of Things network anomalies using various machine learning algorithms. The third section discusses the problem statement. Section IV depicts the materials and procedures employed in the current study, as well as the research flowchart, dataset, and assessment criteria. In Section V, we provide the outcomes of the experiments and compare machine-learning approaches based on various factors. The results are discussed in Section VI by mentioning obstacles, open questions, and future views. The paper comes to a close in the Section VII.

II. Related Works

In this part, we look at studies that employ machine learning-based techniques to solve the challenge of detecting network abnormalities. Recent research suggests that machine learning (ML) techniques might be ideal for detecting anomalies in network data [7]. For example, Abou Daya et al. [8] used machine learning to leverage correlations between packet and flow-level data. On many anomaly detection tasks, Gaddam et al. [9] offered a solution that combined K-means clustering with an ID3 decision tree. For DDOS detection in self-defined networks, Alamri and Thayananthan [10] used XGBoost [11]. Shone et al. [12] developed a deep autoencoder (NDAE) for unsupervised feature learning and intrusion detection utilizing stacked NDAEs. To learn from anomalous traffic, Zhange et al. [13] created a semi-supervised learning system. For intrusion detection, Ullah et al. [14] developed an LSTM-based model using autoencoders. An XGBoost-DNN model was presented by Devan et al. [15] to identify cyber assaults. To solve the unbalanced class problem, Du et al. [16] integrated reinforcement learning with the SMOTE method. For the network anomaly detection problem, we now look at each machine learning approach independently.

K nearest neighbour (K-NN). The KNN technique is one of the most basic and widely used nonparametric methods. It estimates the approximate distances between the input vectors' different points, then assigns the unlabeled point to the class of its K-nearest neighbor. When building a KNN classifier, the parameter (K) is crucial, and various values (K) might have varied outcomes. If K is big, the neighbors utilized for prediction will take a long time to classify and have an impact on accuracy [17].

Zhu et al. offer a Grid-based Approximate Average Outlier Detection (GAAOD) framework to maintain KNN-based anomaly recognition in network traffic streaming data [18]. In the first stage, the proposed framework presents a grid-based coefficient to control resulting data. It can self-adaptively configure the resolution of units, and reach the target of effectively filtering items that cannot become outliers. In the second stage, GAAOD framework utilizes a min-heap-based method to calculate the upper-/lower-bound distance between items and their k-th nearest neighbors. In the third stage, the author applies a k-skyband based method to support anomaly items and possible anomaly items. Technical outcomes prove the effectiveness and high correctness of the proposed approach.

Bayesian networks. A Bayesian network (BN) is a mathematical model for encoding probabilistic correlations between variables. This strategy is typically used in conjunction with statistical schemes for intrusion detection. It has several benefits, including the capacity to encode interdependencies between variables and predict occurrences, as well as the ability to incorporate existing knowledge and data [19].

The BN system, according to Lotfollahi et al. [20], provides the necessary mathematical foundation for making an apparently complex operation simple. They expected that by comparing the measurements of each network traffic sample, BN-based IDS would be able to identify assaults from regular network activity. Mohammed et al. [21] employed a controlled Naive Bayesian classifier and 248 function streams to distinguish between several sorts of information, including packet length and delivery time, as well as a variety of TCP headers. To find strong functions, feature selection correlation was performed, and it revealed that just a small subset of fewer than 20 features is required for accurate classification.

Neural networks (NNs). The behavior of numerous users and daemons in a system is predicted by NNS. If correctly planned and executed, NNS can alleviate many of the issues that rule-based systems have. The key benefit of NNS is their tolerance for erroneous data and information, as well as their capacity to generate solutions without prior understanding of data patterns [22].

This, paired with their capacity to generalize the facts under investigation, qualified them for IDS. Data representing attacks and non-attacks must be fed into the machine learning model for automated modification of network coefficients during the training stage in order to use this technique to IDS. The most prevalent types of regulated neural networks are multilayer perceptron (MLP) and radial basis function (RBF) [23].

Only linearly separable instances of sets may be systematized using MLPs. The perceptron will be able to discover a solution if a straight line or a plane can be drawn to partition input examples into permissible categories, and the input instances are linearly separable. Learning will never reach the point where all examples are adequately systematized if the instances are not linearly separable. To address this issue, multilayer perceptrons (artificial neural networks) were developed.

There have been studies that have used multilayer perceptions to develop intrusion detection system for network traffics, which has the capacity to identify both legitimate and malicious connections, such as [24]. MLPs of three and four layers of a neural network were used to implement them.

Another prominent form of neural network is the Radial Basis Function (RBF). RBF networks are significantly quicker than back propagation because they accomplish classification by measuring the distance between inputs and RBF centers of hidden neurons. They are best suited for problems with a high sample size.

Decision tree (DT). Quinlan [25], for example, characterized decision trees as "a useful and widely used categorization and forecasting method. A decision tree is a tree made up of three primary parts: nodes, arcs, and leaves. Each node has a unique characteristic that is the most informative of the features not yet examined on the path from the root. Each sheet is allocated to a category or class, and each arc from the node identifies the values of the node attribute. Starting at the root of the tree and working down until a node leaf is reached, a decision tree may be used to categorize a data point. The data point is classified using the node sheet. Quinlan's ID3 and C4.5 are the most widely used decision tree implementation alternatives."

As an intrusion detection model, Davahli et al. [26] recommended employing decision trees (DT) and the support vector machine (SVM). They also created a hybrid DTSVM technique that employs both SAM and DT as fundamental classifiers. Decision trees were adapted by Ghanem et al. [27] for DDoS attacks, R2 as well as U2R assaults, and scanning attacks. The ID3 method is used as a learning algorithm to generate a decision tree automatically.

Support Vector Machine (SVM). Cortes and Vapnik [28] proposed the support vector machine (SVM) technique. The input vector is transformed into a multidimensional feature space by SVM, which then finds the best separating hyperplane in a high-dimensional feature space. Furthermore, because the boundary solution, i.e. the separating hyperplane, determines the reference vector rather than the whole training sample, it is impervious to significantly deviating values. SVM is especially well-suited to binary classification. That is, to distinguish between two sets of training vectors with distinct class labels. The penalty function, which is a user-defined parameter in SVM, is also available. This helps users to strike a balance between the amount of samples and the erroneous solution border width categorization.

Mukkamala et al. [29] used SVM "core classifiers and classifier design approaches to apply to the network with the task of identifying abnormalities." They looked at the impact of core type values and parameters on the Support Vector Machine's (SVM) intrusion classification accuracy. The PSA-SVM model was suggested by Gauthama Raman et al. [30], where the PSO standard is used to establish the free parameters of the support vectors and the binary PSO is utilized to produce the optimal subset function in the intrusion detection system. Eskandari et al. [31] provided a model of an intrusion detection system based on network traffic behavior and message analysis and categorization. Anomalies are detected using two artificial intelligence methods: the Kohonen neural network (KSN) and support vectors (SVM).

Deep learning. Recurrent neural networks paired with long short-term memory are investigated in this research [32] for their ability to identify Internet of Things malware. Models constructed using more traditional machine learning techniques are compared to the results of the experiment. These techniques include the Support Vector Machine, the Naive Bayes classifier, the random Forest, adaptive Boosting, and the k-nearest neighbors algorithm. According to the findings of the inquiry, the technique based on deep learning gives the greatest outcomes. Other deep learning models were not compared since there was none.

As described in the study [33], a variety of deep learning methods for recognizing DDoS attacks are being researched, including multilayer perceptron, convolutional neural network, RNN-LSTM, CNN+LST ensemble, and RNN-LSTM and CNN. Their performance is compared to that of standard machine learning algorithms such as the support vector machine, Bayesian classifier, and random forest, among others. They reach the conclusion that deep learning approaches, particularly recurrent networks, are more successful than standard methods.

It is proposed in the research [34] that an auto-encoder and a deep neural network with direct communication be utilized to develop their own anomaly detection solution for industrial Internet of Things systems that they feel will be effective. When the properties of the newly constructed model are compared to those of many previously developed anomaly detection approaches, such as the deep trust network [35], the recurrent network [36], the DNN [37], and the Ensemble-DNN [38], the results show that the newly constructed model outperforms them all. Meanwhile, these models were evaluated on multiple subsets of the source data as well as on a range of different hardware and software platforms at various points in time, according to the research.

III. Problem Statement

It is required to define the mathematical and software techniques in order to analyze abnormalities in network traffics. Anomaly detection, according to our findings, leads to a data categorization issue. We divide the traffic into two categories: regular traffic and abnormal traffic. As a result, the issue is a binary classification problem. We will utilize basic mathematical methods to identify severe fluctuations in the graph, such as:

$$S = \int_{t_1}^{t_2} |x'(t)| dt \tag{1}$$

This is the total of all potential variations from time t1 to time t2. The formula will look like this since the function is discrete:

$$S = \sum_{t=t_1}^{t_2-1} |x(t+1) - x(t)| \tag{2}$$

In the next part, we utilize machine learning approaches to discover IoT network abnormalities and assess them using various measurement parameters for the supplied dataset.

IV. Materials and Methods

In this part, we describe the whole outline of the Machine Learning (ML)-based system that has been recommended for fault and attack differentiation. According to the results presented in Section III, it might be difficult to differentiate between assaults that behave similarly to node issues at the receiving ends due to the fact that their impact on the communication channel is identical. If we monitor the state of the channel, there is a chance that we will be able to record the state transition activities that the attackers execute in order to produce a number of attacks. We came to the conclusion that the best way to overcome the challenge of differentiating between assaults and difficulties on the receiving end was to directly monitor the channel data. Next, in order to differentiate between the two abnormality groups based on channel qualities, we used machine learning models to fit those measurements (and hence channel state).

A. *Methodology*

As can be seen in Fig. 2, the whole process may be broken down into three distinct stages. In the initial step of development, the system is modeled for the normal, faulty, and attack classes respectively. As a consequence of this, the second step entails conducting a number of execution scenarios with the purpose of constructing datasets that define the behavior of the system under normal, faulty, and attack settings. In the third phase, the gathered datasets are put to use in order to assess a number of supervised machine learning algorithms for classification purposes in relation to the differentiation issue.

As a result, the proposed framework is flexible in that it may be used to investigate multiple classes of defects and assaults in a variety of experimental setups, as well as to evaluate the datasets generated by different supervised machine learning algorithms. Furthermore, by concentrating solely on the features of the communication channel, this framework is insensitive to the characteristics of the devices employed in any cyber-physical system of any type.

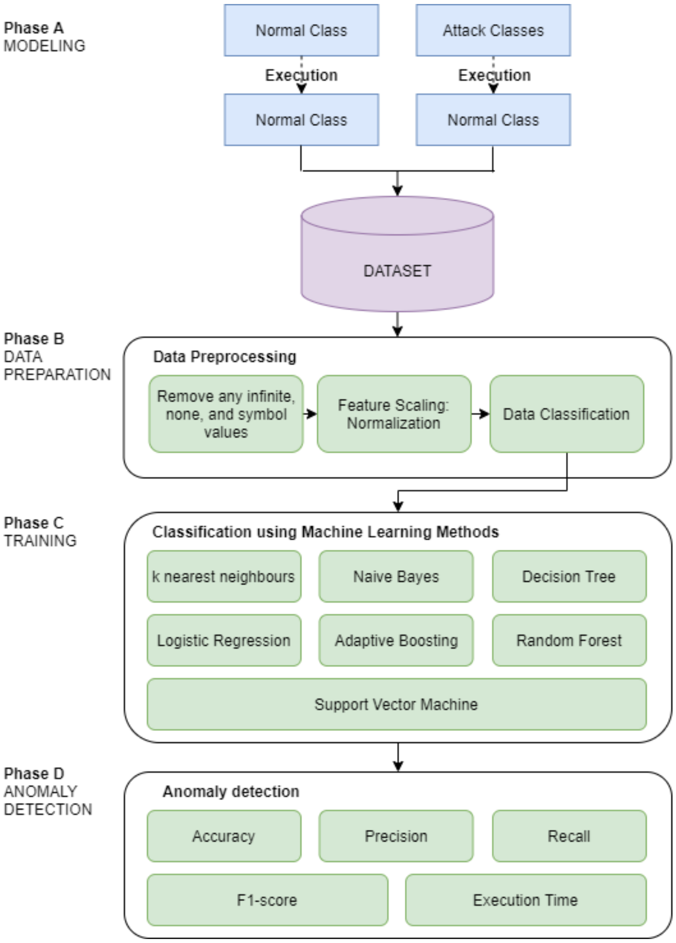

Fig. 2. Comparison of machine learning methods in network anomaly detection.

This part focuses on this general framework and goes through the anomaly classes, various ML classification techniques we are looking at, and the evaluation metrics we are using to evaluate the algorithms.

B. *Data*

An open data collection UNSW-NB15 [39, 40] was chosen as experimental data for the examination of DNN models in the tasks of detecting network abnormalities in the Internet of Things. It contains 2,540,044 records - vectors of TCP/IP network connection attributes and their related class labels. Network packets in this collection of data provide information about typical network activity as well as nine different forms of attacks: fuzzers, analyzers, backdoors, denial of service (DOS), exploits, generic, Reconnaissance, shellcode, and worms. UNSW-NB15 data contains 47 characteristics, such as IP addresses, port numbers, transaction bytes, and so on [41], as well as two class labels — the attack category and the connection abnormality label — for training and testing intrusion detection systems. The first 35 characteristics are for integrating data packet information, while the remainder is for connection circumstances.

The process of detecting deviations from the system's typical profile is known as anomaly detection. To detect anomalies in UNSW-NB15 network data, a binary classification is utilized, with the connection anomaly criteria serving as a class label, with 0 corresponding to the normal profile and 1 corresponding to anomalies.

C. The Proposed CNN-BiLSTM Hybrid Model

This study uses BiLSTM as the model's foundation since it can successfully extract data characteristics. It can perform high-level abstraction and nonlinear transformation of intrusion data, evaluate two-way data information, and give more fine-grained computation. BiLSTM is an upgraded variant of LSTM. Fig. 3 displays the CNN-BiLSTM structure that has been suggested.

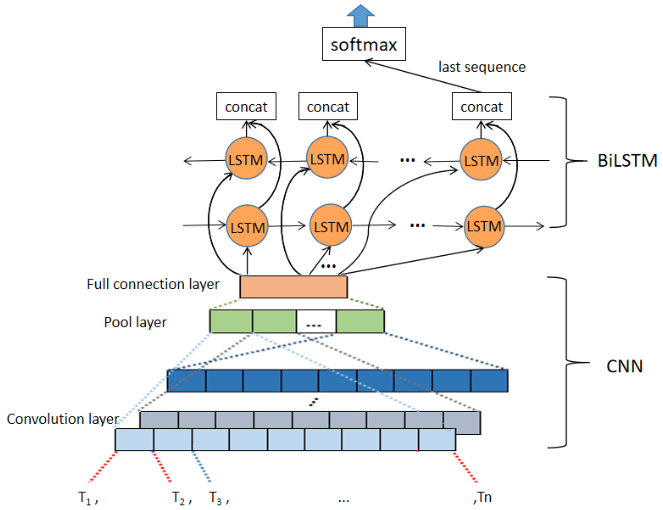

Fig. 3. Proposed CNN-BiLSTM architecture.

The distribution of the data in the neural network may alter after BiLSTM analysis of the data. Batch Normalization process is used to address the inconsistent data distribution problem while deep neural networks are being trained. Deep neural network training may be sped up by batch normalization. After the nonlinear transformation of the activation function, it normalizes the input data of the preceding layer, ensuring the network's trainability and enabling the neural network to continuously maintain the consistency of the input data distribution, thereby minimizing significant changes in the network's node distribution. The network's convergence rate may be accelerated while maintaining the neural network's capacity for representation.

In the IoT, information flow often exhibit significant local correlations, and some of this information even directly correlates with information across a long span. The Bidirectional LSTM network can handle this time-sequential data successfully by using an algorithm to filter out the important and irrelevant information from the data. Hence, in order to enhance the detection capabilities of the detection system, this study incorporates the BiLSTM network based on CNN. The suggested CNN-BiLSTM IoT intrusion detection model is shown in Fig. 4.

The first thing that has to be done in the detection model is to do some kind of preprocessing on the original data set. The process begins by converting all of the data into numerical data, which is followed by the standardization and normalization steps. The data that has been processed will now go into the record representation layer. When the data has been preprocessed, the record presentation layer will use an embedded representation for each individual item of data. The output feature is generated once the features of all the data have been twisted using the convolution check.

While obtaining the feature sequence, all of the features acquired by convolution are layered on one another. The pooling layer receives the feature map from the convolution layer after it has been processed by the convolution layer to produce the feature map. The feature sequences are then pooled together by the pooling layer. The eigenvector may be obtained by first dividing the input data into M blocks, then taking the maximum value for each block, and then splicing all of the results together. This process is known as maximum pooling.

After the pooling of the data in the layer for pooling the data, the acquired feature sequence is then fed into the layer for the BiLSTM. The long-term memory layer is made up of two LSTM modules that are facing in opposite directions, and various weights that are shared between them. The BiLSTM module will choose and then delete each individual piece of data in sequence.

Upon the completion of the data processing, the CNN-BiLSTM network acquires the data features. In order to integrate these feature sequences, a full connection layer is used, and the results that are acquired from the utilization of the full connection layer are then entered into the softmax classifier. In the last step, the results of classifying each piece of information are acquired.

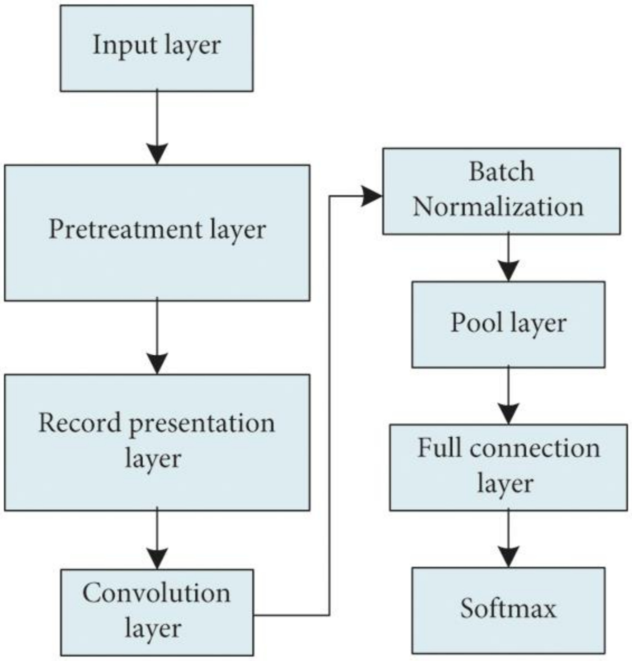

Fig. 4. Intrusion detection model of industrial Internet of Things based on CNN-BiLSTM.

D. Evaluation Metrics

In machine learning tasks, the following metrics are most often used to evaluate the effectiveness of constructed models [42]: accuracy (precision), completeness (recall), F-measure (F-score), ROC-Curve (from the English Receiver Operating Characteristic curve - error curve), AUC-ROC and AUC-PR (from the English Area Under Curve - the area under the error curve and the area under the precision-recall curve) [43].

After classification, to obtain four types of results is possible. Table I demonstrates different classification parameters, where yy' is the algorithm response on the object, and yy is the true class label on this object.

TABLE I. DATASET DESCRIPTION

Dataset	Y=1	Y=0
Y'=1	True Positive (TP)	False Positive (FP)
Y'=0	False Negative (TN)	True Negative (TN)

Overall accuracy or accuracy is an indicator that evaluates the correctness of anomaly detection. The overall accuracy determines what percentage of the data the system or algorithm can classify correctly. Calculated by the formula:

$$Accuracy = \frac{TP + TN}{Pos + Neg} \tag{3}$$

The precision of a classification system may be measured by the percentage of items that are labeled positive by the classifier and are, in fact, positive:

$$precision = \frac{TP}{TP + FP} \tag{4}$$

Completeness (recall) shows the proportion of correctly labeled positive objects among all objects of a positive class:

$$recall = \frac{TP}{TP + FN} \tag{5}$$

The completeness of the data is not affected by the distribution of the data, in contrast to accuracy. Completeness does not represent the number of things that are wrongly identified as positive, and accuracy does not provide any information about the number of positive objects that are incorrectly identified [44].

The (F-score, Fß) combines the above two metrics into one measurement parameter:

$$F_\beta = \left(1 + \beta^2\right) \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{6}$$

Where β - takes values in the range 0 < β < 1 if accuracy is given priority, and β > 1 if completeness is given priority.

The F-measure reaches a maximum with completeness and accuracy equal to one, and is close to zero if one of the arguments is close to zero.

The ROC curve, also known as the error curve, is a graph that shows the relationship between the algorithm's sensitivity (TPR, True Positive Rate) and the proportion of objects in a negative class that the algorithm predicted incorrectly (FPR, False Positive Rate) when the threshold of the decisive rule is changed [45]:

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

In addition to these evaluation parameters, we used area under the curve receiver operating characteristics (AUC-ROC) parameters.

V. EXPERIMENTAL RESULTS

Data preparation (1) entails preparing an input data set, which includes 47 indicators of network connections and class labels, in a manner that can be fed into the studied models. To nominal-type information like IP addresses, protocol names, and data transfer services, one-hot encoding, a method of describing categorical variables in the form of binary vectors, is used. After that, all sign values are normalized to the range [0...1]. Because an imbalance between the values of features can create model instability, degrade learning results, and slow down the modeling process, data normalization is done. A total of 80% of the original data set (1,547,081 records) is chosen for model training, while 20% (386,771 records) is chosen for model testing.

Fig. 5 and Fig. 6 demonstrate the model accuracy and model loss for the proposed CNN-BiLSTM. Fig. 5 demonstrates accuracy of the proposed CNN-BiLSTM model. The results show that the proposed model show high accuracy during the tested 12 epochs of training. The results show that, the proposed model is applicable for practical cases to detect IoT network intrusions or anomalies.

Fig. 6 demonstrates the model loss of the proposed deep CNN-BiLSTM model for intrusion detection problem in internet of things network. There, we show the results for 12 epochs of learning. The results show that the model loss is low from the 4th epochs of training. The result of 12th epochs demonstrates little loss and high accuracy, respectively.

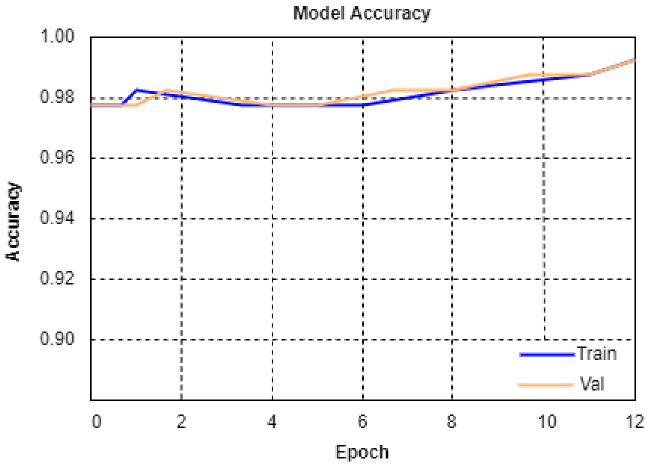

Fig. 5. CNN-BiLSTM Model accuracy.

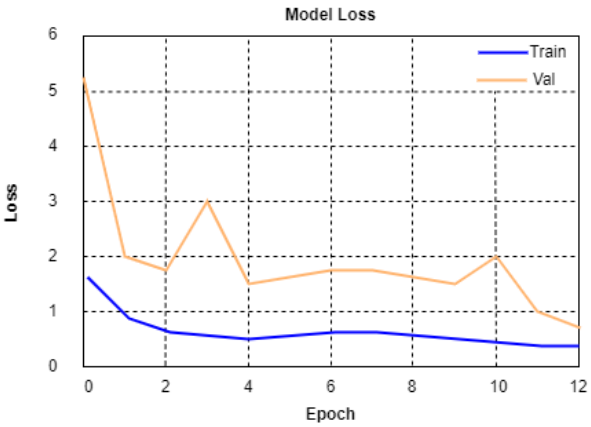

Fig. 6. CNN-BiLSTM Model loss.

In the IoT network anomalies detection challenge, Table II shows a comparison of the investigated machine learning algorithms and training time values. As shown in the table, support vector machine (SVM) has a high level of accuracy in detecting network anomalies, but it takes a long time to train. As a result, it is unfit for real-time anomaly identification. In comparison, for the provided dataset, logistic regression is the best approach for detecting network abnormalities in internet of things.

Fig. 7 and Fig. 8 demonstrate performance evaluation and training time comparison in graphical form. In Fig. 7, we compare six machine learning methods by four evaluation parameters as accuracy, precision, recall, and F1 score. As it is illustrated in the figure, random forest, Adaptive Boosting (AdaBoost), and k nearest neighbours (KNN) show higher performance in the measured evaluation parameters than the other machine learning methods. Nevertheless, we can also consider training and testing time of each algorithm to understand how fast the applied method copes with the given problem.

Fig. 8 demonstrates training times of each algorithm in network anomalies detection. For convenience, the figure is illustrated in logarithmic scale. If we compare the three methods that shown high performance, KNN has the longest training time, Adaboost and Random Forest gives shorter training time, that makes the methods suitable for practical use.

TABLE II. COMPARISON OF THE PROPOSED MODEL WITH MACHINE LEARNING METHODS

Classification method	Accuracy	Precision	Recall	F-measure	Execution Time, s
Proposed CNN-BiLSTM	**96.28**	**96.17**	**95.14**	**95.09**	**47.16**
KNN	85.02	85.12	85.02	85.02	12698.27
Naïve Bayes	82.05	83.29	82	82.56	175.48
DT	81.17	81.17	81.06	81.23	846.24
SVM	88.26	88.32	88.26	88.26	10624.85
Logistic Regression	85.83	85.89	85.83	85.85	178.56
AdaBoost	87.34	87.34	87.34	87.34	965.45
Random Forest	87.62	87.66	87.66	87.66	574.20

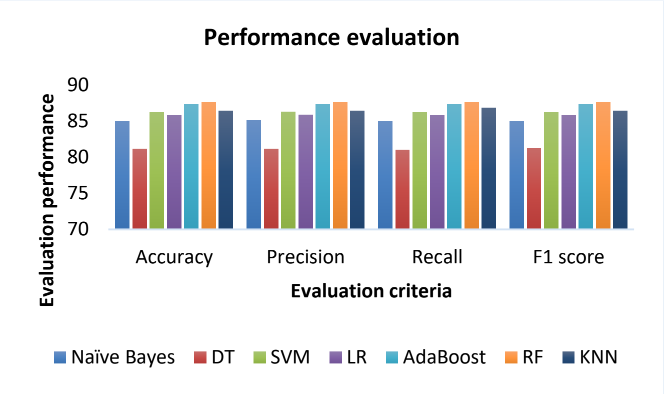

Fig. 7. Comparison of machine learning methods in network anomaly detection.

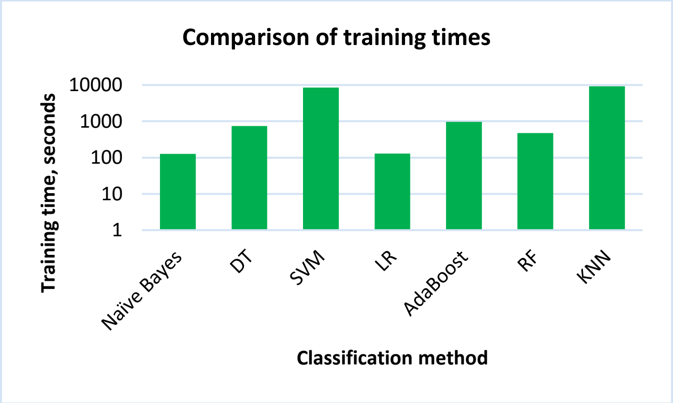

Fig. 8. Comparison of training times (in logarithmic scale).

Fig. 9 indicates the ROC curves of each method. The applied methods show high results in the given problem. The results show, that machine learning techniques can be successful in internet of things network traffic anomaly detection.

Thus, we compared different machine learning methods for network anomalies detection problem in two types of performance parameters. The results show that Logistic Regression is more suitable for practical use than the other methods in intersection of two indicators. It has comparatively short training time and high accuracy in network anomalies detection.

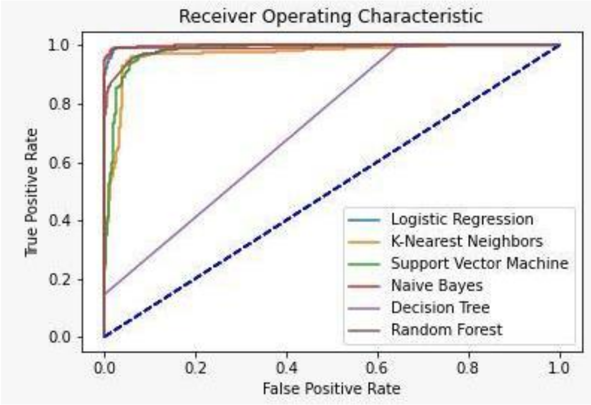

Fig. 9. The ROC curve of the applied methods for IoT anomalies detection.

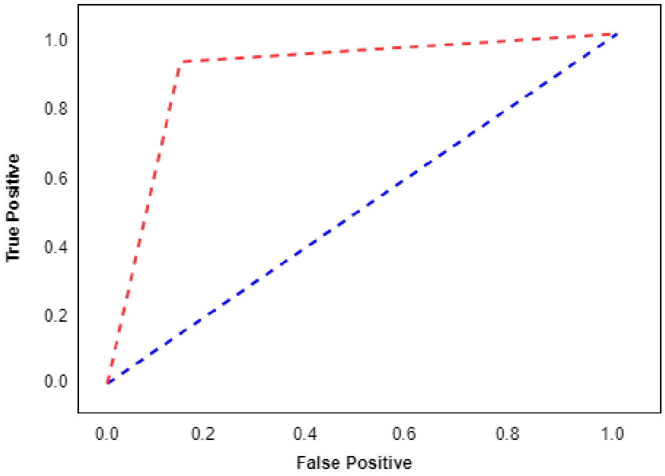

Fig. 10. The ROC curve of the applied method for IoT anomalies detection.

Fig. 10 demonstrates the ROC curve of the proposed CNN-BiLSTM for anomalies detection in internet of things. The obtained results show that, the AUC-ROC curve show high value. The obtained results of all evaluation parameters demonstrate that the proposed CNN-BiLSTM model is applicable for practical cases.

VI. Discussion

The goal of this research is to explore machine learning and deep learning models for identifying abnormalities in Internet of Things network data and develop a new deep learning model for the given problem. Deep learning models were evaluated utilizing a single set of hardware and software, as well as equal sections of the UNSW-NB15 dataset for training and testing. Test models include logistic regression, random forest, KNN, decision tree, Naive Bayes, SVM, and Adaptive Boosting. The built models have high rates of IoT network anomaly detection accuracy, ranging from 80% to 88%. The article proposes CNN-BiLSTM hybrid model for detection of anomalies in internet of things network. The proposed deep model shown about 96% accuracy. In addition, the paper chooses the best machine learning model based on the amount of time it takes to train the model and the importance of identifying abnormalities in internet of things network traffic.

It is intended to continue examining the properties of models employed in cybersecurity jobs in the future. One of the upcoming research objectives is to look at the effect of internet of things network traffic topology on the performance metrics of deep learning models [46]. Based on the findings, a deep CNN-BiLSTM strategy is proposed for recognizing and linking security incidents.

VII. Conclusion

The goal of this research is to look at machine learning models for identifying abnormalities in Internet of Things network data. Deep learning models were evaluated utilizing a single set of hardware and software, as well as equal sections of the UNSW-NB15 dataset for training and testing. Test models include logistic regression, random forest, KNN, decision tree, Naive Bayes, SVM, and Adaptive Boosting. The built models have high rates of network anomaly detection accuracy, ranging from 80% to 88%. The article offers suggestions for selecting the best deep learning model based on the amount of time it takes to train the model and the importance of identifying abnormalities in network traffic.

It is intended to continue examining the properties of models employed in cybersecurity jobs in the future. One of the upcoming research objectives is to look at the effect of network traffic topology on the performance metrics of deep learning models. Based on the findings, it is proposed to build a deep learning-based strategy to recognizing and linking security incidents.

References

[1] A. Nauman, Y. Qadri, M. Amjad, Y. Zikria, M. Afzal et. al. “Multimedia Internet of Things: A comprehensive survey,” IEEE Access, vol. 8, no. 1, pp. 8202-8250, 2020.

[2] Z. Lv, L. Qiao, J. Li, J. and H. Song. “Deep-Learning-Enabled Security Issues in the Internet of Things,” IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9531-9538, 2020.

[3] Sultan, D., Toktarova, A., Zhumadillayeva, A., Aldeshov, S., Mussiraliyeva, S., Beissenova, G., ... & Imanbayeva, A. (2023). Cyberbullying-related hate speech detection using shallow-to-deep learning. CMC-COMPUTERS MATERIALS & CONTINUA, 74(1), 2115-2131.

[4] S. Park, G. Li and J. Hong. “A study on smart factory-based ambient intelligence context-aware intrusion detection system using machine learning,” Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 4, pp. 1405-1412, 2020.

[5] Murzamadieva, M., Ivashov, A., Omarov, B., Omarov, B., Kendzhayeva, B., & Abdrakhmanov, R. (2021, January). Development of a system for ensuring humidity in sport complexes. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 530-535). IEEE.

[6] Omarov, B., Altayeva, A., Turganbayeva, A., Abdulkarimova, G., Gusmanova, F., Sarbasova, A., ... & Omarov, N. (2019). Agent based modeling of smart grids in smart cities. In Electronic Governance and Open Society: Challenges in Eurasia: 5th International Conference, EGOSE 2018, St. Petersburg, Russia, November 14-16, 2018, Revised Selected Papers 5 (pp. 3-13). Springer International Publishing.

[7] F. Cauteruccio, L. Cinelli, E. Corradini, G. Terracina, D. Ursino et. al. “A framework for anomaly detection and classification in Multiple IoT scenarios,” Future Generation Computer Systems, vol. 114, pp. 322-335, 2021.

[8] A. Abou Daya, M. Salahuddin, N. Limam and R. Boutaba. “Botchase: Graph-based bot detection using machine learning,” IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 15-29, 2020.

[9] S. Gaddam, V. Phoha and K. Balagani, “K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods,” IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, pp. 345-354, 2007.

[10] H. Alamri and V. Thayananthan. “Bandwidth Control Mechanism and Extreme Gradient Boosting Algorithm for Protecting Software-Defined Networks Against DDoS Attacks,” IEEE Access, vol. 8, no. 1, pp. 194269-194288, 2020.

[11] H. Jiang, Z. He, G. Ye and H. Zhang. “Network intrusion detection based on PSO-XGBoost model,” IEEE Access, vol. 8, no. 1, pp. 58392-58401, 2020.

[12] N. Shone, T. Ngoc, V. Phai and Q. Shi, “A deep learning approach to network intrusion detection”, IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, 2018.

[13] Y. Zhang, M. Li, Z. Ji, W. Fan, S. Yuan et. al. “Twin self-supervision based semi-supervised learning (TS-SSL): Retinal anomaly classification in SD-OCT images,” Neurocomputing, vol. 462, no. 1, pp. 491-505, 2021.

[14] W. Ullah, A. Ullah, I. Haq, K. Muhammad, M. Sajjad et. al. “CNN features with bi-directional LSTM for real-time anomaly detection in

surveillance networks". Multimedia Tools and Applications, vol. 80, no. 11, pp. 16979-16995, 2021.

[15] P. Devan and N. Khare, "An efficient xgboost–dnn-based classification model for network intrusion detection system," Neural Computing and Applications, vol. 32, no. 1, pp. 1-16, 2020.

[16] X. Du, W. Susilo, M. Guizani and Z. Tian, Z. "Introduction to the Special Section on Artificial Intelligence Security: Adversarial Attack and Defense," IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 905-907, 2021.

[17] Z. Geler, V. Kurbalija, M. Ivanović, and M. Radovanović, M. "Weighted kNN and constrained elastic distances for time-series classification," Expert Systems with Applications, vol. 162, no. 1, pp. 113829, 2020.

[18] R. Zhu, X. Ji, D. Yu, Z. Tan, L. Zhao et. al. "KNN-based approximate outlier detection algorithm over IoT streaming data," IEEE Access, vol. 8, pp. 42749-42759, 2020.

[19] Omarov, B., Orazbaev, E., Baimukhanbetov, B., Abusseitov, B., Khudiyarov, G., & Anarbayev, A. (2017). Test battery for comprehensive control in the training system of highly Skilled Wrestlers of Kazakhstan on National wrestling "Kazaksha Kuresi". Man In India, 97(11), 453-462.

[20] M. Lotfollahi, M. Siavoshani, R. Zade and M. Saberian, M. "Deep packet: A novel approach for encrypted traffic classification using deep learning," Soft Computing, vol. 24, no. 3, pp. 1999-2012, 2020.

[21] B. Mohammed, M. Hamdan, J. Bassi, H. Jamil, S. Khan et. al. "Edge Computing Intelligence Using Robust Feature Selection for Network Traffic Classification in Internet-of-Things," IEEE Access, vol. 8, no. 1, pp. 224059-224070, 2020.

[22] M. Xibilia, M. Latino, Z. Marinković, A. Atanasković and N. Donato. "Soft sensors based on deep neural networks for applications in security and safety," IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 10, pp. 7869-7876, 2020.

[23] Omarov, B., Suliman, A., & Kushibar, K. (2016). Face recognition using artificial neural networks in parallel architecture. Journal of Theoretical and Applied Information Technology. Vol.91., No.2, pp. 238-248.

[24] Onalbek, Z. K., Omarov, B. S., Berkimbayev, K. M., Mukhamedzhanov, B. K., Usenbek, R. R., Kendzhaeva, B. B., & Mukhamedzhanova, M. Z. (2013). Forming of professional competence of future tyeacher-trainers as a factor of increasing the quality. Middle East Journal of Scientific Research, 15(9), 1272-1276.

[25] S. Peddabachigari, A. Abraham, C. Grosan and J. Thomas. "Modeling Intrusion Detection System using Hybrid Intelligent Systems," Journal of Network and Computer Applications, Vol. 30, no1, pp. 114–132, 2007.

[26] A. Davahli, M. Shamsi and Abaei. "A lightweight Anomaly detection model using SVM for WSNs in IoT through a hybrid feature selection algorithm based on GA and GWO," Journal of Computing and Security, vol. 7, no. 1, pp. 63-79, 2020.

[27] W. Ghanem and A. Jantan. "Training a neural network for cyberattack classification applications using hybridization of an artificial bee colony and monarch butterfly optimization," Neural Processing Letters, vol. 51, no. 1, pp. 905-946, 2020.

[28] C. Cortes and V. Vapnik. "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273-297, 1995.

[29] S. Dwivedi, M. Vardhan, S. Tripathi and A. Shukla. "Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection," Evolutionary Intelligence, vol. 13, no. 1, pp. 103-117, 2020.

[30] M. Gauthama Raman, N. Somu, S. Jagarapu, T. Manghnani, T. Selvam et. al. "An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm," Artificial Intelligence Review, vol. 53, no. 5, pp. 3255-3286, 2020.

[31] M. Eskandari, Z. Janjua, M. Vecchio and F. Antonelli. "Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices," IEEE Internet of Things Journal, vol. 7, no. 8, pp. 6882-6897, 2020.

[32] H. HaddadPajouh, A. Dehghantanha, R. Khayami and K. Choo. "A Deep recurrent neural network based approach for internet of things malware threat hunting," Future Generation Computer Systems, vol. 85, pp. 88–96, 2018.

[33] S. Rathore and J. Park. "A blockchain-based deep learning approach for cyber security in next generation industrial cyber-physical systems," IEEE Transactions on Industrial Informatics, vol. 17, no. 8, pp. 5522-5532, 2020.

[34] A. Muna, N. Moustafa, E. Sitnikova. "Identification of malicious activities in industrial internet of things based on deep learning models," Journal of Information Security and Applications, vol. 41, pp. 1–11, 2018.

[35] S. Huda S. Miah, J. Yearwood, S. Alyahya, H. Al-Dossari et. al. "A malicious threat detection model for cloud assisted internet of things (CoT) based industrial control system (ICS) networks using deep belief network," Journal of Parallel and Distributed Computing, vol. 120, no. 1, pp. 23-31, 2018.

[36] O. Alkadi, N. Moustafa, B. Turnbull and K. Choo. "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9463-9472, 2020.

[37] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad. "Survey on SDN based network intrusion detection system using machine learning approaches," Peer-to-Peer Networking and Applications, vol. 12, no. 2, pp. 493-501, 2019.

[38] R. Abdulhammed, M. Faezipour, A. Abuzneid and A. AbuMallouh. "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," IEEE sensors letters, vol. 3, no. 1, pp. 1-4, 2018.

[39] UNSW-NB15 Network Dataset, 2021. [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFANB15- Datasets/.

[40] N. Moustafa and J. Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," In 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 2015, pp. 1–6.

[41] N. Moustafa, B. Turnbull and K. Choo. "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," IEEE Internet of Things Journal, 2019, vol. 6, no. 3, pp. 4815–4830.

[42] Murugesan, G., Ahmed, T. I., Shabaz, M., Bhola, J., Omarov, B., Swaminathan, R., ... & Sumi, S. A. (2022). Assessment of mental workload by visual motor activity among control group and patient suffering from depressive disorder. Computational Intelligence and Neuroscience, 2022.

[43] Anand, M., Sahay, K. B., Ahmed, M. A., Sultan, D., Chandan, R. R., & Singh, B. (2023). Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques. Theoretical Computer Science, 943, 203-218.

[44] R. Soleymani, E. Granger and G. Fumera, G. "F-measure curves: A tool to visualize classifier performance under imbalance,". Pattern Recognition, vol. 100, pp. 107146.

[45] M. Abu-Alhaija and N. Turab. "Automated Learning of ECG Streaming Data Through Machine Learning Internet of Things," Intelligent Automation & Soft Computing, vol. 32, no. 1, pp. 45–53, 2022.

[46] T. Heinis, C. Loy and Meboldt, M. "Improving Usage Metrics for Pay-per-Use Pricing with IoT Technology and Machine Learning: IoT technology and machine learning can identify and capture advanced metrics that make pay-per-use servitization models viable for a wider range of applications," Research-Technology Management, vol. 61, no. 5, pp. 32-40, 2018.