Anomaly detection for the internet-of-medical-things

REJI, Alan, PRANGGONO, Bernardi <http://orcid.org/0000-0002-2992-697X>, MARCHANG, Jims <http://orcid.org/0000-0002-3700-6671> and SHENFIELD, Alex <http://orcid.org/0000-0002-2931-8077>

Published version

REJI, Alan, PRANGGONO, Bernardi, MARCHANG, Jims and SHENFIELD, Alex (2023). Anomaly detection for the internet-of-medical-things. In: 2023 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 1944-1949.

Anomaly Detection for the Internet-of-Medical-Things

Alan Reji
Department of Engineering and Mathematics
Sheffield Hallam University
Sheffield, UK
alanvreji@gmail.com

Bernardi Pranggono
School of Computing and Information Science
Anglia Ruskin University
Cambridge, UK
bernardi.pranggono@aru.ac.uk

Jims Marchang
Department of Computing
Sheffield Hallam University
Sheffield, UK
jims.marchang@shu.ac.uk

Alex Shenfield
Department of Engineering and Mathematics
Sheffield Hallam University
Sheffield, UK
a.shenfield@shu.ac.uk

***Abstract*—Cybersecurity for the Internet of Medical Things (IoMT) is a very concerning issue because of emerging cyber threats and security incidents targeting IoMT devices all over the world. The healthcare system has near-zero tolerance for inexplicability. In this paper, we propose a machine learning-based anomaly detection for the IoMT and evaluate the performance using a realistic public dataset. We implement various machine learning algorithms: Random Forest, Decision Tree, Logistic Regression, Support Vector Machine, and K-Nearest Neighbor with TON_IoT dataset. Two types of classifications are implemented: binary and categorical. In the categorical classification, evaluation for nine attack scenarios (Scanning, DoS, password cracking attack, and Man-in-The-Middle (MITM)) are performed. The test results demonstrate that Support Vector Machine models produce better performance compared to the other models.**

***Index Terms*—Cybersecurity, Intrusion Detection, IoMT**

I. Introduction

As the adoption of the Internet of Things (IoT) is growing significantly in diverse fields including healthcare, in such scenarios achieving robust security in IoT is becoming increasingly challenging. Due to its benefits and advantages, the adoption of IoT devices in healthcare organizations has reached 70% with increasing reliance in such organizations on the Internet of Medical Things (IoMT). The COVID-19 pandemic increased the adoption of the IoMT to reduce the risks of getting infected while treating patients. It is expected that the global IoT in the healthcare market will reach USD 290 billion by 2028 from USD 128 billion in 2023. However, we also see increasing cyber-attacks during the pandemic where cyber criminals and Advanced Persistent Threat (APT) groups have taken advantage of targeting vulnerable people and systems [1]. The IoMT refers to the interconnected network of medical devices, sensors, and software that enable the exchange of data for various healthcare purposes. However, with the increase in connectivity also comes an increase in the risk of security breaches and cyber-attacks. Hackers may target IoMT devices to access sensitive patient data or disrupt critical medical processes. Therefore, it is essential to develop effective intrusion detection systems (IDSs) to protect the IoMT from cyber threats. One promising approach to intrusion detection in the IoMT is the use of machine learning algorithms. Machine learning allows for the automated analysis of large amounts of data, enabling the identification of patterns and anomalies that may indicate an intrusion. In this paper, we propose a machine learning-based intrusion detection for the IoMT and evaluate the performance using a realistic public dataset.

The remainder of this paper is arranged as follows: Section II presents the related work on anomaly detection for IoMT. Section III describes the methods used in the study such as the dataset and data pre-processing. Our experimental results are discussed and analyzed in Section IV Finally, Section V draws the conclusion.

II. Related Work

Anomaly detection (often referred to as intrusion detection) is considered one effective method to detect cyber-attacks in IoMT networks. In [2], a novel approach to detect malicious network traffic using artificial neural networks (ANNs) is presented for use in deep packet inspection-based IDS. Under repeated 10-fold cross-validation, the authors obtained an average accuracy of 98% and an average false positive rate of less than 2% using a range of real-world shell-code exploits and benign network traffic.

Thamilarasu et al. developed a Machine Learning-based IDS for IoMT networks using autonomous mobile agents [3]. The study used its own generated dataset. The study showed that the proposed IDS is able to detect attacks with high accuracy with minimal energy consumption overhead. However, the dataset used and produced in this study does not reflect the complete range of cyber-attacks at IoMT networks.

Zachos et al. proposed an anomaly-based intrusion detection system (AIDS) for IoMT networks [4]. The AIDS has a set of distributed monitoring and data acquisition (MDA) components running on each IoMT device, and a central detection (CD) component running on the gateway. The proposed study used six machine learning (ML) algorithms: Decision Tree, Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine, and K-Nearest Neighbor and considered computational costs in detecting abnormal data traffic and identifying malicious traffic in the IoMT network. The paper used the TON_IoT dataset for training and evaluation [5]. The study showed that the Naive Bayes algorithm produces better results compared to other ML algorithms.

A method to detect attack traffic using a deep neural network in the IoMT-Blockchain environment is proposed in [6]. The study used a multi-model autoencoder (MMAE) to effectively learn the fusion of low-dimensional feature representations between different features from the original data. The paper used two self-made datasets (TADA and TADB) collected in the IoMT-Blockchain network. TADA has DoS, Probe, R2L, PortScan, SSH, and U2R. TADB has Backdoor, DoS, Exploit, Analysis, Fuzers, and Worms. The paper claimed that the anomaly detection performance obtained by their method is relatively good.

The Duo-Secure IoMT framework using multi-modal sensory signal data to differentiate attack patterns and routing IoMT devices' data is proposed in [7]. The study used a combination of methods such as dynamic Fuzzy C-Means clustering with Bi-LSTM. The study used the WUSTL-EHMS dataset and their performance evaluation showed that the proposed method achieved 92.95% accuracy in identifying network malware.

III. Methodology

In this section, the details methodology used in this study is discussed. This includes the dataset used, pre-processing methods, and performance evaluation metrics.

A. Dataset

The use of IoT-related datasets that reflect real-world IoT applications plays an essential role in evaluating the accuracy as well as the efficiency of the intrusion detection models. However, there is a lack of availability of real-world datasets among the research community as most of the companies that deal with IoT devices are reluctant to share their log details due to privacy concerns. This creates an obstacle in the creation of intrusion detection models tailored to IoT, IoMT, or Industrial IoT (IIoT) applications.

One of the few publicly available datasets for research purposes is called the TON_IoT network dataset [5] (**T**elemetry data, **O**perating systems' data, and **N**etwork data) and is used in this research to develop the supervised machine learning models. The TON_IoT dataset contains major real-world threat vectors in IoT and IoMT networks.

The dataset files available in TON_IoT repository were generated by simulating nine varieties of attack scenarios (scanning, DoS, DDoS, ransomware, backdoor, data injection, cross-site scripting (XSS), password cracking attacks, and Man-In-The-Middle (MITM) attacks) against different IoT and IIoT devices to collect the data. The dataset includes heterogeneous data sources collected from Telemetry datasets of IoT and IIoT sensors, Windows 7/10 and Ubuntu 14/18 LTS operating system datasets, and IoT network traffic datasets. The ToN_IoT dataset combines four different data types: packet capture, Bro logs, sensor data, and OS logs.

In this work, we are using the training and testing split of the TON_IoT dataset (as used by the authors of [5] for evaluating the accuracy and efficiency of various machine learning algorithms). This processed dataset has 45 features and is divided into four components:

(i) Network_dataset: contains the traffic data that passed through the entire testbed.
(ii) IoT_dataset: contains the data related to various IoT/IIoT sensors simulated in the testbed.
(iii) Linux_dataset: contains data connected to the Ubuntu systems.
(iv) Windows_dataset: contains data connected to the Windows systems.

B. Pre-processing

The dataset available in CSV format was imported to the MATLAB environment using the import tool. Since the standard features in 'Train_Test_datasets' were found as a mix of both numerical and categorical values, all the features were imported as categorical variables to MATLAB in .mat file format and then each feature was separately converted into numerical values using the 'unique' function to facilitate their use in ML algorithms. For example, consider the feature 'proto' containing the categories of "tcp", "udp" and "icmp". While converting this categorical feature, the categories "tcp", "udp" and "icmp" is converted into numerical values "0", "1", and "2" respectively.

Furthermore, the dataset feature size was reduced from 45 to 44 by omitting the 'ts' feature, as this feature might lead some ML algorithms to overfit to the training data. Since the ML models take a significant amount of time to train and test the entire dataset, only 50% of the imported dataset was used in this study. The "normalize" function with normalization method "range" and "scale" were used to create two datasets: normalized dataset and standardized dataset respectively.

Each dataset was then sent to the classification learner app, which separates the data for training, validation, and testing purposes. In this study, 20% of the preprocessed data was assigned for testing, the remaining 80% was used for training the model. Validation prevents overfitting by estimating model performance on new data compared to training data and assisting in the selection of the optimal model. To validate the machine learning model, 25% of the training data was separated using the holdout approach. Before beginning the session, the necessary predictors and response attributes were chosen.

C. Evaluation Metrics

The machine learning models are evaluated using standard performance metrics: accuracy, precision, recall, and F1-score (see Table I) [8]. In Table I, true positive (TP) means anomalous traffic correctly identified, true negative (TN) means normal traffic correctly identified, false positive (FP) means normal traffic incorrectly identified as anomalous, and false negative (FN) means anomalous traffic incorrectly identified as normal.

TABLE I
PERFORMANCE METRICS

Performance Metric	Definition
Accuracy	$\frac{(TP+TN)}{(TP+TN+FP+FN)}$
Precision	$\frac{TP}{(TP+FP)}$
Recall	$\frac{TP}{(TP+FN)}$
F1-Score	$\frac{(2\times Precision\times Recall)}{(Precision+Recall)}$

IV. RESULTS

This section evaluates the test results of different algorithms with the standardized and normalized datasets and analyses their performance. The algorithms used in this study are: Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN). A robust set of default hyperparameters from the classification learner app in MATLAB are used in this work. The binary and multi-categorical classification in features 'label' and 'type' will be discussed separately.

A. Binary Classification

The response attribute set for this ML model will be the 'label' attribute which tells us if a network intrusion is detected or not.

1) Normalized dataset: The accuracy, precision, recall, and F1-score test results of different ML models trained with the normalized train-test network dataset are shown in Table II. From the given data, we can see that the LR provides the worst performance, with an accuracy of 82.2% and an F1-score of 76.5%. The best results were obtained from the DT algorithm with an accuracy of 99.96% and an F1-score of 99.94%.

TABLE II
TEST RESULTS FOR BINARY CLASSIFICATION USING THE NORMALIZED DATASET

	DT	RF	LR	SVM	KNN
TP	16055	16058	13336	15720	15972
TN	30031	29658	24571	29769	29912
FP	12	385	5472	274	131
FN	6	3	2725	341	89
Accuracy	99.96	99.16	82.22	98.67	99.52
Precision	99.93	97.66	70.91	98.29	99.19
Recall	99.96	99.98	83.03	97.88	99.45
F1-Score	99.94	98.81	76.49	98.08	99.32

Figure 1 compares the accuracy and F1-score for different ML models. While all other ML algorithms except LR show excellent accuracy (between 98% and 100%), LR shows comparatively worse performance (<85% accuracy and <75% F1-score).

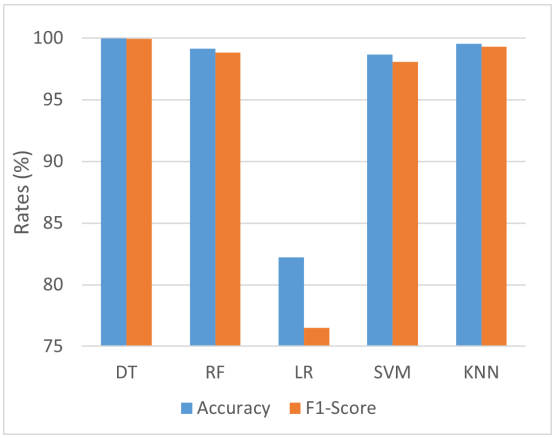

Fig. 1. Accuracy and F1-score for different algorithms while testing on the normalized dataset

2) Standardized dataset: Table III shows the test results obtained by different ML algorithms trained with the standardized dataset. Like the normalized dataset results, LR again gives the worst performance with an accuracy of 76.6% and an F1-score of 72.8%. Here, the SVM model performs best, with an accuracy of 99.6% and an F1-score of 99.4%. Both DT and RF obtain very similar results which could be due to the tree-base structure of their design.

TABLE III
TEST RESULTS FOR BINARY CLASSIFICATION USING THE STANDARDIZED DATASET

	DT	RF	LR	SVM	KNN
TP	16059	16059	14419	15984	15747
TN	29660	29660	20899	29920	29776
FP	383	383	9144	123	267
FN	2	2	1642	77	314
Accuracy	99.16	99.16	76.61	99.57	98.74
Precision	97.67	97.67	61.19	99.24	98.33
Recall	99.99	99.99	89.78	99.52	98.04
F1-Score	98.82	98.82	72.78	99.38	98.19

The accuracy and F1-score results are plotted in Figure 2 to increase readability. While the F1-score is lower compared to accuracy, the model trained with SVM gives a fairly equal response. Again, all the ML models except the LR algorithm gives out a good result between 98% and 100%, whereas LR shows a comparatively worse performance (between 61.19% and 89.78%).

The models that give the best results while trained with both the normalized and standardized datasets are compared in Figure 3. From the chart, we can firmly understood that the ML model trained with DT algorithm and the normalized dataset performs better compared to the other algorithms.

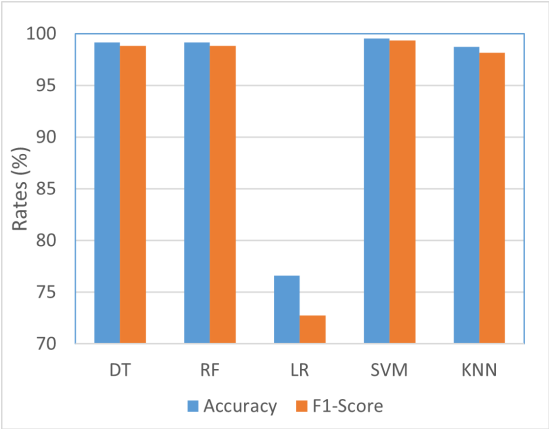

Fig. 2. Accuracy and F1-score for different algorithms while testing on the standardized dataset

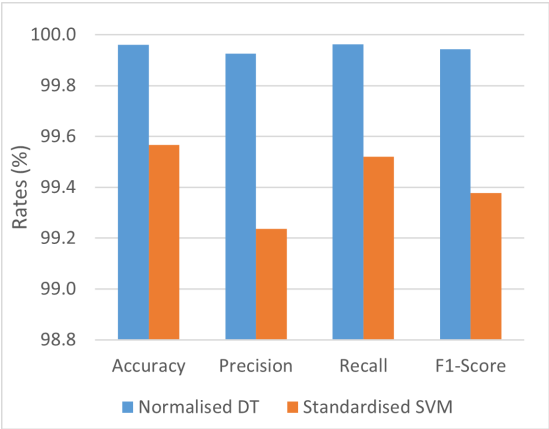

Fig. 3. Chart comparing best models from the normalized and standardized datasets for binary classification

B. Categorical Classification

Here, the 'type' attribute from "Train_Test_datasets" is used as the response attribute to train the network. There will be a total of 10 categories including 9 attack types and a normal type data which needs to be distinguished by the ML models.

1) Normalized dataset: Table IV shows the full results for the three ML algorithms used for categorical classification using the normalized dataset. The RF classifier produces TP and FP values of zero for some of the attack types which means precision and F1-Score are therefore undefined. The RF algorithm performs well in classifying normal, backdoor, and DoS data with an F1-score of 99.45%, 98.13%, and 96.57% respectively; however, it performs poorly for the other attack types. The SVM algorithm shows good performance for classifying most of the attack types - giving an F1-score of above 94% for most attacks. At the same time, the algorithm struggles in predicting the MITM network attack - giving an F1-score of 43.17%. Overall, the SVM model gives an average F1-score of 91.89%. Similarly, the KNN algorithm also gives a good categorical classification, classifying 6 of the categories with an F1-score of above 95%. As we have seen with the SVM model, KNN also performs worst in classifying MITM network attacks, but gives a better performance than SVM with an F1-score of 45.5% compared to 43.2% for the SVM model. The KNN model also gives an average F1-score of 91.13%.

Table V gives an overall comparison of the performance of different algorithms to classify different network category attacks using the normalized dataset. From the table we can see that the SVM model gives a better accuracy of 99.65% and F1-score of 91.89% as compared to the KNN model with accuracy and F1-score of 99.64% and 91.13% respectively.

2) Standardized dataset: Now, let us look at the performance of different algorithms to the standardized dataset in classifying different types of network anomalies.

Table VI shows the full results for the three ML algorithms used for categorical classification using the standardized dataset. Again, the RF classifier produces TP and FP values of zero for some of the attack types which means precision and F1-Score are therefore undefined. Even though the algorithm gives a good F1-score of 99.45% for predicting normal network traffic, it gives the worst performance for the ransomware category with F1-score of only 1.18%. The SVM algorithm shows better performance, with 8 out of 10 categories giving an F1-score of above 95%. The algorithm performs very well in classifying the backdoor network attack, whereas it again struggles in classifying the MITM attack. The SVM algorithm also gives an average F1-score of 91.83%. The KNN algorithm also performs well in predicting the categories using the standardized dataset, with 6 out of 10 categories having an F1-score above 95%. As we have seen before, the MITM attack was classified with the least performance, achieving an F1-score of 46.3%. The KNN model also gives an average accuracy and F1 score of 99.52% and 90.07% respectively.

The comparison of the overall performance of the different algorithms we tested in classifying different network category attacks using the standardized dataset is shown in Table VII. From the table, we can see that the SVM model gives better accuracy of 99.66% and F1-score of 91.83% as compared to the KNN model with accuracy and F1-score of 99.52% and 90.07% respectively. This may be partly explained by SVMs being particularly good at dealing with high dimensional datasets with limited numbers of samples [9].

Figure 4 plots the results of the best-performing algorithms while using the normalized and standardized datasets. Even though both the models exhibit a very similar response, the models trained with normalized dataset provide a slightly better F1-score of 91.89% as compared to the 91.83% provided by the models trained with standardized dataset. On the other hand, the models trained with standardized dataset provides a slightly better accuracy of 99.66% as compared to the 99.65% provided by the models trained with normalized dataset.

Considering the score difference between the accuracy of the models as much less compared to F1-score, and preferring F1-score as a better metric to evaluate the model performance, this study infers that the TON_IoT dataset pre-processed with

TABLE IV
TESTING RESULTS FOR CATEGORICAL CLASSIFICATION USING THE NORMALIZED DATASET

ML algorithm	Category	TP	TN	FP	FN	Accuracy	Precision	Recall	F1-Score	Support
RF	Backdoor	1968	44061	51	24	99.84	97.47	98.80	98.13	1992
	DDoS	0	44111	0	1993	95.68	–	0.00	–	1993
	DoS	1858	44114	0	132	99.71	100.00	93.37	96.57	1990
	Injection	0	44128	0	1976	95.71	–	0.00	–	1976
	MITM	0	45998	0	106	99.77	–	0.00	–	106
	Normal	30026	15744	318	16	99.28	98.95	99.95	99.45	30042
	Password	2007	37623	6474	0	85.96	23.66	100.00	38.27	2007
	Ransomware	8	44080	10	2006	95.63	44.44	0.40	0.79	2014
	Scanning	1756	42469	1628	251	95.92	51.89	87.49	65.15	2007
	XSS	0	44127	0	1977	95.71	–	0.00	–	1977
	Total / Average	**37623**	**406455**	**8481**	**8481**	**96.32**	**–**	**48.00**	**–**	**46104**
SVM	Backdoor	1968	44111	1	6	99.98	99.95	99.70	99.82	1992
	DDoS	1917	44083	27	77	99.77	98.61	96.14	97.36	1994
	DoS	1838	44109	6	151	99.66	99.67	92.41	95.90	1989
	Injection	1811	44071	57	165	99.52	96.95	91.65	94.22	1976
	MITM	30	45995	3	76	99.83	90.91	28.30	43.17	106
	Normal	29984	15485	576	59	98.62	98.12	99.80	98.95	30043
	Password	1966	44083	13	42	99.88	99.34	97.91	98.62	2008
	Ransomware	1899	44022	68	115	99.60	96.54	94.29	95.40	2014
	Scanning	1993	44045	53	13	99.86	96.41	99.35	98.37	2006
	XSS	1870	44122	6	106	99.76	99.68	94.64	97.09	1976
	Total / Average	**45294**	**414126**	**810**	**810**	**99.65**	**97.72**	**89.42**	**91.13**	**46104**
KNN	Backdoor	1988	44111	1	4	99.99	99.95	99.80	99.87	1992
	DDoS	1895	44055	56	98	99.67	97.13	95.08	96.10	1993
	DoS	1926	44039	75	64	99.70	96.25	96.78	96.52	1990
	Injection	1837	43879	249	139	99.16	88.06	92.97	90.45	1976
	MITM	43	45958	40	63	99.78	51.81	40.57	45.50	106
	Normal	29945	15970	92	97	99.59	99.69	99.68	99.69	30042
	Password	1830	44028	69	177	99.47	96.37	91.18	93.70	2007
	Ransomware	2002	44043	47	12	99.87	97.71	99.40	98.55	2014
	Scanning	1969	44026	71	38	99.76	96.52	98.11	97.31	2007
	XSS	1847	44005	122	130	99.45	93.80	93.42	93.61	1977
	Total / Average	**45282**	**414114**	**822**	**822**	**99.64**	**91.73**	**90.70**	**91.13**	**46104**

TABLE V
COMPARISON OF DIFFERENT ML ALGORITHMS FOR CATEGORICAL CLASSIFICATION USING NORMALIZED DATASET

	Accuracy	Precision	Recall	F1-Score
RF	96.32	–	48.00	–
SVM	99.65	97.72	89.42	91.89
KNN	99.64	91.73	90.70	91.13

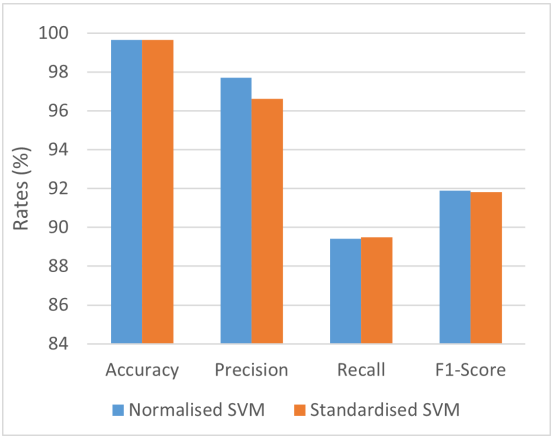

Fig. 4. Chart comparing best models from the normalized and standardized datasets for categorical classification

the normalization technique and classified with the SVM algorithm shows the best performance.

This study was conducted with only 50% of the original train test split of the network dataset due to computational constraints - with the large dataset size increasing the training time beyond what was manageable with limited resources. It was noted that the feature selection option can considerably reduce the training time as compared to the model trained without feature selection. During the study, one of the models that trained without feature selection took five minutes to complete training whereas the same model with feature selection took only three minutes. This suggests that 40% of the training time can be saved by using the feature selection method at a cost of a very small reduction in model performance.

V. CONCLUSION

In this paper, we propose a machine learning-based network anomaly detection system for the IoMT and evaluate the performance using a realistic public dataset. We implement various machine learning algorithms: Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) with the TON_IoT dataset. Two types of classifications are implemented: binary and categorical. In the categorical classification, evaluation for nine attack scenarios (scanning, DoS,

TABLE VI
TESTING RESULTS FOR CATEGORICAL CLASSIFICATION USING THE STANDARDIZED DATASET

ML algorithm	Category	TP	TN	FP	FN	Accuracy	Precision	Recall	F1-Score	Support
RF	Backdoor	1967	44055	57	25	99.82	97.18	98.74	97.96	1992
	DDoS	0	44110	0	1994	95.67	–	0.00	–	1994
	DoS	1829	44115	0	160	99.65	100.00	91.96	95.81	1989
	Injection	0	44128	0	1976	95.71	–	0.00	–	1976
	MITM	0	45998	0	106	99.77	–	0.00	–	106
	Normal	30017	15755	306	26	99.28	98.99	99.91	99.45	30043
	Password	2008	37567	6529	0	85.84	23.52	100.00	38.08	2008
	Ransomware	12	44083	7	2002	95.64	63.16	0.60	1.18	2014
	Scanning	1743	42469	1629	263	95.90	51.69	86.89	64.82	2006
	XSS	0	44128	0	1976	95.71	–	0.00	–	1976
	Total / Average	**37576**	**406408**	**8528**	**8528**	**96.32**	**–**	**47.81**	**–**	**46104**
SVM	Backdoor	1990	44111	0	3	99.99	100.0	99.85	99.92	1993
	DDoS	1939	44078	32	55	99.81	98.38	97.24	97.81	1994
	DoS	1858	44099	15	132	99.68	99.20	93.37	96.19	1990
	Injection	1808	44084	44	168	99.54	97.62	91.50	94.46	1976
	MITM	30	45990	8	76	99.82	78.95	28.30	41.67	106
	Normal	29987	15491	571	55	98.64	98.13	99.82	98.97	30042
	Password	1963	44086	11	44	99.88	99.44	97.81	98.62	2007
	Ransomware	1888	44027	63	265	99.59	96.77	93.74	95.23	2014
	Scanning	1973	44055	43	33	99.84	97.87	98.35	98.11	2006
	XSS	1876	44123	5	100	99.77	99.73	94.94	97.28	1976
	Total / Average	**45312**	**414144**	**792**	**792**	**99.66**	**96.61**	**89.49**	**91.83**	**46104**
KNN	Backdoor	1988	44104	8	4	99.97	99.60	99.80	99.70	1992
	DDoS	1867	44058	52	127	99.61	97.29	93.63	95.43	1994
	DoS	1895	44041	74	94	99.64	96.24	95.27	95.76	1989
	Injection	1830	43753	375	146	98.87	82.99	92.61	87.54	1976
	MITM	41	45968	30	65	99.79	57.75	38.68	46.33	106
	Normal	29871	15934	127	172	99.35	99.58	99.43	99.50	30043
	Password	1734	44000	96	274	99.20	94.75	86.35	90.36	2008
	Ransomware	2002	43999	91	12	99.78	95.65	99.40	97.49	2014
	Scanning	1976	44021	77	30	99.76	96.25	98.50	97.36	2006
	XSS	1800	43958	170	176	99.25	91.37	91.09	91.23	1976
	Total / Average	**45004**	**413836**	**1100**	**1100**	**99.52**	**91.15**	**89.48**	**90.07**	**46104**

TABLE VII
COMPARISON OF DIFFERENT ML ALGORITHMS FOR CATEGORICAL CLASSIFICATION USING NORMALIZED DATASET

	Accuracy	Precision	Recall	F1-Score
RF	96.32	–	48.00	–
SVM	99.65	97.72	89.42	91.89
KNN	99.64	91.73	90.70	91.13

DDoS, ransomware, backdoor, data injection, cross-site scripting (XSS), password cracking attack, and Man-in-The-Middle (MITM)) are performed. Our experiments show that in binary classification for the normalized dataset, DT provides the best results with an accuracy of 99.96% and F1-score of 99.94%. In binary classification for the standardized dataset, SVM gives the highest performance with an accuracy of 99.6% and F1-score of 99.4%. In categorical classification using the normalized dataset, the SVM model gives the best accuracy of 99.65% and F1-score of 91.89%. In categorical classification for the standardized dataset, again the SVM model gives the highest accuracy of 99.66% and F1-score of 91.83%.

REFERENCES

[1] B. Pranggono and A. Arabo, "Covid-19 pandemic cybersecurity issues," *Internet Technology Letters*, vol. 4, no. 2, p. e247, 2021.

[2] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *ICT Express*, vol. 4, no. 2, pp. 95–99, 2018.

[3] G. Thamilarasu, A. Odesile, and A. Hoang, "An intrusion detection system for internet of medical things," *IEEE Access*, vol. 8, pp. 181 560–181 576, 2020.

[4] G. Zachos, I. Essop, G. Mantas, K. Porfyrakis, J. Ribeiro, and J. Rodriguez, "An anomaly-based intrusion detection system for internet of medical things networks," *Electronics*, vol. 10, no. 21, p. 2562, 2021.

[5] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165 130–165 150, 2020.

[6] J. Wang, H. Jin, J. Chen, J. Tan, and K. Zhong, "Anomaly detection in internet of medical things with blockchain from the perspective of deep neural network," *Information Sciences*, vol. 617, pp. 133–149, 2022.

[7] S. Wagan, J. Koo, I. Siddiqui, N. Qureshi, M. Attique, and D. Shin, *A Fuzzy-Based Duo-Secure Multi-Modal Framework for IoMT Anomaly Detection*. Journal of King Saud University - Computer and Information Sciences, 2023, vol. 35, no. 1.

[8] P. Mishra, V. Varadharajan, U. Tupakula, and E. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2018.

[9] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.