Journal of Network Intelligence ©2025 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **10**, Number **4**, November **2025**

Intrusion Detection Method for Intelligent Measurement Systems Based on Machine Learning

Liang Zhao

Joint Laboratory of Digital Technology for New Power System
Digital Grid Research Institute
China Southern Power Grid, Guangzhou 510630, China
hust123hust@126.com

Zhi-Ming Wang*

Joint Laboratory of Digital Technology for New Power System
Digital Grid Research Institute
China Southern Power Grid, Guangzhou 510630, China
jamewzm@163.com

Mei-Shan Zhong

Guangdong Provincial Key Laboratory of Digital Grid Technology
Digital Grid Research Institute
China Southern Power Grid, Guangzhou 510630, China
zhongmeishan123@126.com

Ze-Jie Tan

Guangdong Provincial Key Laboratory of Digital Grid Technology
Digital Grid Research Institute
China Southern Power Grid, Guangzhou, 510630, China
tanzj@csg.cn

Zhao-Hui Hu

Southern Power Grid Digital Platform Technology (Guangdong) Co., Ltd, Guangzhou 510630, China
Huzh@csg.cn

Ting-Wen Yu

Southern Power Grid Digital Platform Technology (Guangdong) Co., Ltd, Guangzhou 510630, China
ytw@csg.cn

*Corresponding author: Zhi-Ming Wang

Received May 13, 2024, revised November 21, 2024, accepted May 27, 2025.

Abstract. *The use of intrusion detection algorithms can effectively detect network threats in intelligent measurement system, which is of great significance in avoiding related losses caused by network security issues. Most intrusion detection algorithms have limitations in detecting attack types and only have good detection results for attack types within a single dataset. At the same time, the detected attacks cannot be classified comprehensively and in detail. Based on this issue, a intelligent measurement system attack dataset was constructed, and a feature selection method and a intelligent measurement system attack classification model were proposed using the sample size and time attribute characteristics of the sample data. This can effectively solve the problem of excessive feature attributes in the dataset leading to a decrease in classification accuracy in the later stage. At the same time, it expands the scope of intrusion detection in the intelligent measurement system and improves its accuracy. Compared with other algorithms, this method significantly improves the detection and classification accuracy of intelligent measurement system attacks, and can effectively detect 61 types of attacks such as fuzzy attacks and DOS attacks.*

Keywords: Intelligent measurement system, intrusion detection, classification, machine learning

1. **Introduction.** With the construction of smart grids, the level of informatization in the power system is getting higher and higher, and the threats and risks it faces are also becoming more severe. Security issues such as malicious attacks, data tampering, and virus planting are emerging one after another, causing damage to the integrity, availability, and confidentiality of assets in the intelligent measurement system, seriously threatening the safe operation of the intelligent measurement system [1]. After the emergence of the concept of network intrusion detection, the related technologies of network intrusion detection are constantly developing, and machine learning based network intrusion detection models have been widely applied [2]. The machine learning based intrusion detection model analogizes the problem of network intrusion detection to the classification problem of network traffic in a system, and uses machine learning methods to train intrusion detection models for classification and prediction of network attacks. At present, various machine learning algorithms, such as K-nearest neighbors, support vector machines, deep learning, artificial neural networks, etc., have been used for data preprocessing of network traffic and classification prediction, and have achieved good experimental results.

For example, Li et al. reviewed existing network intrusion detection methods and tools [3]. Kim et al. applied deep learning to intrusion detection and validated its effectiveness [4, 5]. Zhao et al. optimized the Grey Wolf algorithm by using deep confidence networks to repeatedly learn and train feature datasets. Finally, the effectiveness of their method was experimentally verified [6]. Hassan et al. simulated intrusion detection systems based on deep learning and proposed a deep learning approach, establishing a recurrent neural network architecture for detecting intrusion threats [7]. Hsu et al. proposed a hybrid intrusion detection model that combines convolutional neural networks and recurrent neural networks to maintain long-term dependencies between extraction functions and prevent overfitting on frequent connections [8]. Wu et al. proposed an enhanced protocol based on their protocol and used formal/informal security analysis to demonstrate the security of the improved protocol [9]. Ren proposed an efficient network intrusion detection model based on LightGBM [10] to solve the problems of gradually opening up terminal networks, device dispersion, and difficulty in security monitoring on the power client side. Yang has conducted in-depth research on big data and machine learning technologies, which can be used to provide pre identification of threats in advanced metrology facilities [11]. Ding et al. combined big data analysis technology and equipment evaluation technology to propose a time series based data anomaly detection method, in order to further improve the utilization rate of equipment information in equipment anomaly detection methods [12]. Wu et al. propose an enhanced authentication and key agreement protocol. The security of protocol is rigorously analyzed using the Real-Or-Random model, informal security analysis, and the AVISPA tool [13]. Afroz et al. proposed a two-layer intrusion detection model based on RF and SVM, which reduces the dimensionality of data and uses random forest and support vector machine as classifiers. The results indicate that the model performs well in detecting various attacks and achieves the expected quality of a functional IDS model [14]. Fu et al. designed an IL-SVM-KNN classifier that combines SVM and K-nearest neighbor algorithm. This classifier has the ability to distinguish abnormal traffic from normal traffic and determine the type of abnormal traffic. Compared with KNN algorithm and SVM algorithm, the accuracy of the proposed classifier is significantly improved [15]. Pozi et al. proposed a method to improve attack detection rate using support vector machines and genetic algorithms, mainly using a classifier method called GPSVM [16]. Kalubi et al. proposed an intrusion detection model that combines LSTM and GRU in deep learning methods [17]. Erfani et al. proposed a model that combines deep belief networks and support vector machines, which is suitable for large-scale and high-dimensional data detection [18]. Wu et al. show the detailed attack steps and propesed an enhanced scheme based on PAuth. Formal and informal security analyses are provided to demonstrate the improved security of the proposed scheme [19].

Traditional intelligent measurement systems are mainly isolated from external networks through tools such as firewalls. However, with the application of new technologies such as cloud computing and the Internet of Things, the degree of interconnection between networks continues to deepen, and the integration scale is getting larger and larger [20]. The data size of the traffic dataset of intelligent measurement systems is gradually increasing, and the degree of data redundancy is also increasing [21]. The difficulty of security protection is greatly increased, and it is crucial to effectively detect network intrusion behavior. Therefore, the paper proposes a feature selection method and an intelligent measurement system attack classification model based on the characteristics of sample size and time attributes of sample data. This model combines deep learning algorithms and neural network algorithms to detect abnormal behavior in the network, effectively solving the problem of excessive feature attributes in the dataset leading to a decrease in classification accuracy in the later stage, and alleviating the serious harm caused by such problems to the system. At the same time, it provides reference for relevant personnel to handle the subsequent intrusion detection of intelligent measurement systems, which is of great significance for ensuring the security of intelligent measurement systems.

2. **Intelligent measurement system network attack data processing.** The paper uses four types of publicly available network datasets to construct an intrusion detection dataset for intelligent energy measurement systems, in order to facilitate subsequent network attack detection operations.

2.1. **Network attack dataset.** The paper constructs intrusion detection datasets for intelligent energy measurement systems using four types of publicly available network datasets, including CIC-IDS-2018, CIC-DDOS (2019), NSL-KDD, and UNSW-NB15 datasets.

2.1.1. *CIC-IDS-2018 Dataset.* The CIC-IDS-2018 dataset consists of 70 columns of attributes, including 69 feature attribute columns and 1 category label column. The specific attribute records are shown in Table 1.

The category label (column 70) marks network traffic as normal traffic or one of the six types of attacks. The six types of attacks are:

Table 1. CIC-IDS-2018 Dataset Attribute Feature Table

CIC-IDS-2018 Dataset Attribute Features			
DstPort	Protocol	Time stamp	Flow Duration
Fwd Pkt Len Std	Bwd Pkt Len Max	Bwd Pkt Len Min	Bwd Pkt Len Mean
Fwd IAT Tot	Fwd IAT Mean	FwdIAT Std	Bwd IAT Min
Bwd Pkts/s	Pkt Len Min	Pkt Len Max	Pkt Len Mean
Bwd Seg Size Avg	Fwd Byts/b Avg	Fwd Pkts/b Avg	Fwd Blk Rate Avg
Init Fwd Win Byts	Init Bwd Win Byts	Fwd Act Data Pkts	Fwd Seg Size Min
Idle Min	Idle Max	Fwd Pkt Len Mean	Flow IAT Min
TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	TotLen Fwd Pkts
Flow IAT Mean	Flow IAT Std	Flow IAT Max	Flow Pkts/s
Bwd URG Flags	Fwd Header Len	Bwd Header Len	Fwd URG Flags
SYN Flag Cnt	RST Flag Cnt	PSH Flag Cnt	FIN Flag Cnt
Subflow Fwd Pkts	Subflow Fwd Byts	Subflow Bwd Pkts	Bwd Blk Rate Avg
Active Min	Idle Mean	Idle Std	Active Max
Label	Subflow Bwd Byts	ACK Flag Cnt	Active Std
Bwd Pkts/b Avg	Pkt Len Var	Bwd PSH Flags	Flow Byts/s
Tot Bwd Pkts	Fwd Pkts/s	Active Mean	Bwd Byts/b Avg
Pkt Len Std	Fwd PSH Flags	Bwd Pkt Len Std	Tot Fwd Pkts

Table 2. CIC-IDS-2018 Testing and Training Dataset Type Distribution

Category labels	CIC-IDS-2018-traing	CIC-IDS-2018-testing
Brute Force	116817	38939
Botnet	84507	28169
Infiltration	48417	16193
Benign	251253	83751
Total	502506	167502

Brute Force attack: Attackers calculate passwords one by one until they find the correct password.

Botnet attack: Attackers spread ”zombie programs” to form a ”zombie network”, achieving the goal of controlling computers and issuing remote commands.

Denial of Service (DoS) attack: Attackers occupy a large amount of memory resources, preventing legitimate users from accessing the computer or processing normal requests.

Distributed Denial of Service (DDOS) attack: Attackers target websites and servers, interrupt network services, exhaust application resources, and prevent legitimate users from accessing computers or processing normal requests.

Web attack: Attackers use website application vulnerability scanners to attack vulnerable websites.

Infiltration of the Network from Inside: Attackers send malicious files to victims via email while exploiting application vulnerabilities to execute backdoors on the victim's computer. Among them, due to the fact that DOS and DDOS attacks in other datasets have the same characteristics and a large number as this dataset, for the CIC-IDS-2018 dataset, the DOS attack and DDOS attack data are removed, and abnormal data is selected with a probability of 30% and 10% and placed in the training and testing sets respectively. At the same time, normal data equal to the total number of abnormal data is placed, so that the distribution of normal and abnormal data in the dataset is uniform.

The training and testing sets for CIC-IDS-2018 are shown in Table 2.

Table 3. CIC-DDOS(2019) Testing and Training Dataset Type Distribution

CIC-DDOS(2019)Dataset Attribute Features			
DstPort	Protocol	Time stamp	Flow Duration
Fwd Pkt Len Std	Bwd Pkt Len Max	Bwd Pkt Len Min	Bwd Pkt Len Mean
Fwd IAT Tot	Fwd IAT Mean	FwdIAT Std	Bwd IAT Min
Bwd Pkts/s	Pkt Len Min	Pkt Len Max	Pkt Len Mean
Bwd Seg Size Avg	Fwd Byts/b Avg	Fwd Pkts/b Avg	Fwd Blk Rate Avg
Init Fwd Win Byts	Init Bwd Win Byts	Fwd Act Data Pkts	Fwd Seg Size Min
Flow ID	Source IP	Source Port	Destination IP
Idle Min	Idle Max	Fwd Pkt Len Mean	Label
Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts
Bwd Pkt Len Std	Flow Byts/s	Flow Pkts/s	Flow IAT Mean
Fwd PSH Flags	Bwd PSH Flags	Fwd URG Flags	Bwd URG Flags
Pkt Len Std	Pkt Len Var	FIN Flag Cnt	SYN Flag Cnt
Bwd Byts/b Avg	Bwd Pkts/b Avg	Bwd Blk Rate Avg	Subflow Fwd Pkts
Active Mean	Active Std	Active Max	Active Min
SimillarHTTP	Inbound	Fwd Pkts/s	ACK Flag Cnt
Fwd Pkt Len Max	Fwd Header Len	Subflow Fwd Byts	Fwd Pkt Len Min
Flow IAT Std	RST Flag Cnt	Idle Mean	Flow IAT Max
Bwd Header Len	Subflow Bwd Pkts	Subflow Bwd Byts	Flow IAT Min
PSH Flag Cnt	Idle Std		

Table 4. The Number of Different Attack Types in the CIC-DDOS(2019) Dataset

Category labels	CIC-DDOS(2019)_1	CIC-DDOS(2019)_2
PortMap	1048576	1048576
NetBIOS	1048576	1048576
LDAP	1048576	1048576
MSSQL	1048576	1048576
UDP	1048576	1048576
UDP-Lag	370606	725166
SYN	1048576	1048576
NTP	1048576	—
DNS	1048576	—
SNMP	1048576	—
TFTP	1048576	—
Total	10856366	7016622

2.1.2. *CIC-DDOS(2019) Dataset.* The CIC-DDOS(2019) dataset consists of 74 attribute columns, including 73 feature columns and 1 category label column. The specific attribute records are shown in Table 3.

The category label (column 74) marks traffic as normal traffic or one of the eleven types of attacks. Eleven types of attacks are classified based on the different transport protocols used by attackers, and can be divided into TCP based attacks (MSSQL, SYN), UDP based attacks (NTP, UDP, UDP Lag), and attacks that can use TCP or UDP protocols (DNS, SNMP, BIOS, LDAP, PortMapT, TFTP). The training set of the CIC-DDOS(2019) dataset includes 10856366 records, while the testing set includes 7016622 records. The number of labels for each category is shown in Table 4.

Table 5. CIC-DDOS(2019) Training and Testing Set Different Types of Number

Category labels	CIC-DDOS(2019)training	CIC-DDOS(2019)testing
PortMap	10485	10485
NetBIOS	10485	10485
LDAP	10485	10485
MSSQL	10485	10485
UDP	10485	10485
UDP-Lag	3706	7251
SYN	10485	10485
NTP	10485	—
DNS	10485	—
SNMP	10485	—
TFTP	10485	—
Total	108556	70161

Among them, random sampling was carried out on the CIC-DDOS(2019) dataset, and abnormal data was selected with a probability of 3% and 1% and placed in the training and testing sets respectively. At the same time, normal data that was close to the total number of abnormal data was placed to evenly distribute the number of normal and abnormal data in the dataset. The training and testing sets for CIC-DDOS(2019) are shown in Table 5.

2.1.3. *NSL-KDD Dataset.* The NSL-KDD dataset consists of 42 columns of attributes, including 41 features and 1 category label. The first 41 columns of features include 9 TCP basic connection features, 13 content connection features, 9 time-based network traffic statistics features, and 10 host based network traffic statistics features. The specific attribute records are shown in Table 6.

The category label (column 42) marks traffic as normal traffic or one of the four types of attacks. The four types of attacks are:

Denial of Service (DoS) attacks; Unauthorized Remote Access Attack (R2L): Attackers who do not have the computer's account can send packets to the computer and gain local access to the computer; Unauthorized Access to Local Users (U2R): Obtain root privileges by first accessing unauthorized or low privileged user accounts to identify system vulnerabilities; Probe attack: Obtaining computer network information for attack. The number of labels in the NSL-KDD dataset is shown in Table 7.

2.1.4. *UNSW-NB15 Dataset.* The UNSW-NB15 dataset consists of 49 attribute columns, including 48 feature columns and 1 category label column. The specific attribute records are shown in Table 8.

The category label (column 49) marks traffic as normal traffic or one of the nine types of attacks. The nine types of attacks are:

Fuzzy attack: The attacker inputs a large amount of random data into the application, causing it to crash, and uses fuzzy testing tools to discover application vulnerabilities.

Analyze attacks: scanning different attack ports, infiltrating spam and HTML files.

Universal attack: applicable to all block ciphers (with given block and key sizes).

Reconnaissance attack: All attacks that simulate collecting information.

Shellcode attack: Code used as a payload when exploiting software vulnerabilities.

Worm attack: Attackers self replicate in order to spread to other computers.

Table 6. NSL-KDD Dataset Attribute Characteristics

Basic Connection Features Columns			
1∼2	flag		src_bytes
3∼5	duration	protocol_type	service
6∼7	dst_bytes	wrong_fragment	
8∼9	land	urgent	
Content Connection Features Columns			
10∼11	hot	num_failed_logins	
12∼14	logged_in	compromised	num_compromised
15∼16	num_root	num_file_creations	
17∼19	root_shell	su_attempted	num_shells
20∼22	is_hot_login	num_outbound_cmds	is_guest_login
Time-based Network Traffic Statistics Features Columns			
23∼25	count	serror_rate	srv_count
26∼27	srv_serror_rate		rerror_rate
28∼29	same_srv_rate	srv_rerror_rate	
30∼31	diff_srv_rate	srv_diff_host_rate	
Host Based Network Traffic Statistics Features Columns			
32∼33	dst_host_count	dst_host_same_srv_rate	
34∼35	dst_host_srv_count	dst_host_diff_srv_rate	
36∼37	dst_host_srv_diff_host_rate		dst_host_serror_rate
38∼39	dst_host_srv_rerror_rate		dst_host_same_src_port_rate
40∼42	dst_host_srv_serror_rate	dst_host_rerror_rate	Label

Table 7. The Number of Different Attack Types in the NSL-KDD Dataset

Category labels	KDDTrain+	KDDTest+
Normal	67343	9711
DoS	45927	7458
Probe	11656	2421
R2L	995	2754
U2R	52	200
Total	125973	22544

The training set of the UNSW-NB15 dataset includes 125973 records, and the testing set includes 22544 records. The number of labels for each category is shown in Table 9.

2.2. **Data preprocessing.**

The paper constructs an intrusion detection dataset for intelligent energy measurement systems using four types of publicly available network datasets, compares the relationships between feature attributes, and performs data processing operations including data integration, missing data checking, column name checking, data transformation, feature selection, etc.

2.2.1. *Data Integration.* Merge four different source datasets to form an intelligent measurement system intrusion detection dataset with diverse attack types and wide coverage. The specific types and quantities of intrusion detection datasets are shown in Table 10.

2.2.2. *Missing data check.* Check if there are any empty values in each dimension of each data record and count the number of empty values. If there are empty values and the

Table 8. UNSW-NB15 Dataset Attribute Characteristics

UNSW-NB15 Dataset Attribute Characteristics			
srcip	sport	dstip	dsport
dttl(num_shell)	sloss	dloss	service
stcpb	dtcpb	smeansz	dmeansz
Sintpkt	Dintpkt	tcprtt	synack
ct_srv_src	ct_srv_dst	ct_dst_ltm	ct_src_ ltm
protocol-type	state	duration	srcbyte
Sload	Dload	Spkts	Dpkts
trans_depth	res_bdy_len	Sjit	Djit
ackdat	is_sm_ips_ports	land	ct_flw_http_mthd
ct_src_dport_ltm	ct_dst_sport_ltm	ct_dst_src_ltm	attack_cat
dstbyte	Stime	sttl	Ltime
dwin	is_ftp_login	swin	ct_ftp_cmd
Label			

Table 9. The Number of Different Attack Types in the UNSW-NB15 Dataset

Category labels	UNSW-NB15 -training	UNSW-NB15 -testing
Fuzzy attack	995	9711
Analyze attacks	45927	7458
Backdoor attack	11656	2421
Vulnerability exploitation	52	200
Universal attack	2390	1009
Reconnaissance attack	7865	230
Shell code attack	2852	149
Worm attack	23	232
Total	125973	22544

number of empty values is much smaller than the overall data sample size, filter the samples with empty values and use samples without NAN values.

2.2.3. *List check.* Preliminary screening of column names (attribute names) in the dataset was conducted, and redundant operations were performed on columns (attributes) with the same name. After data integration, a total of 259 feature attributes were found in the intrusion detection dataset. Specifically, two or more columns with the same name were removed, including Protocol, Timestamp, Label, Dst Port, Source Port, and Destination IP. The remaining 105 feature attributes were all named differently, totaling 154.

2.2.4. *Data conversion.* The range of differences between the maximum and minimum values of certain features in the data is very large, such as the value range of duration being [0,58329], the value range of src_bytes being [0,1.3 * 109], and the value range of dst_bytes being [0, 1.3 * 109]. Firstly, the logarithmic scaling method is used to numerically scale the features with a larger range, resulting in a range of values for duration of [0,4.77], src_bytes of [0,9.11], and dst_bytes of [0,9.11]. Secondly, each feature is linearly mapped to the range of [0,1] according to Formula (1), where Max represents the maximum value of each feature and Min represents the minimum value of each feature.

$$x_i = \frac{x_i - Min}{Max - Min} \tag{1}$$

Table 10. Measurement System Intrusion Detection Dataset

Category labels	training	testing
Brute Force	116817	38939
Heartbleed	8526	9757
Botnet	84507	28169
DoS	45927	7458
DDOS	108556	70161
Web	5837	5794
Infiltration	48417	16193
Probe	11656	2421
R2L	995	2754
U2R	52	200
Fuzzy attack	995	9711
Analyze attacks	45927	7458
Backdoor attack	11656	2421
Vulnerability exploitation	52	200
Universal attack	2390	1009
Reconnaissance attack	7865	230
Shell code attack	2852	149
Worm attack	23	232
Benign	318596	93462
Total	820123	296718

The dataset contains character features (protocol_type, service, flag, and attack_type), as shown in Table 11.

However, due to the fact that most of the data in deep learning is in the form of a numerical matrix, a unique hot encoding method is adopted to convert non numerical features into numerical features. An N-bit state register is used to encode N states. For example, the protocol type feature in the feature column has three feature values, namely TCP, UDP, and ICMP, which are respectively represented as 001, 010, and 100 after unique hot encoding. There are 70 characteristic values for the same service type, and 11 characteristic values for the flag type. There are a total of 19 categories and 61 subcategories of label types in the dataset. Due to the large number of label types in subcategories, the paper uses large category labels, as shown in Table 12.

After the above data preprocessing, a dataset of intelligent measurement system network attacks is formed.

3. **Network attack feature selection method for intelligent measurement systems based on mRMR.** The mRMR method is a feature selection algorithm that maximizes correlation and minimizes redundancy. This algorithm scores each feature based on correlation, sets a threshold, or filters the number of features to be selected. This method differs from other feature selection algorithms in that it is not only easy to operate and has low implementation complexity, but also utilizes two indicators, maximum correlation and minimum redundancy, to achieve the conditions of ”best” classification results and ”less” selection of attribute features. Therefore, based on the maximum correlation and minimum redundancy methods, this paper proposes a feature selection method based on multi distance weighting and Grammer algorithm, establishes a feature selection model, selects features of network traffic data packets, and excludes highly similar features.

Protocol(2)	**Service(3)**		
udp	other	urh_j	time
icmp	link	ssh	hostnames
tcp	netbios_ssn	http_8001	name
Flag(4)	smtp	iso_tsap	ecr_i
S3	netstat	aol	bgp
OTH	ctf	sql_net	telnet
S1	ntp_u	shell	domain
S2	harvest	supdup	ftp_data
RSTO	efs	auth	nnsp
RSTRs	klogin	whois	courier
RSTRS0	systat	discard	finger
SF	exec	sunrpc	uucp_path
SH	nntp	urp_i	X11
REJ	pop_3	rje	Imap4
S0	printer	ftp	mtp
	vmnet	daytime	login
	netbios_ns	domain_u	tftp_u
	pop_2	pm_dump	kshell
	gopher	IRC	netbios_dgm
	csnet_ns	http_443	uucp
	private	Red_i	eco_i
	http_2784	z39_50	remote_job
	echo	tim_i	idap
	http		

Table 11. The Category Label of the Dataset After One-Hot Encoding

Category labels	Attack type	Category labels	Attack type
0	Brute Force	10	Fuzzy attack
1	Heartbleed	11	Analyze attacks
2	Botnet	12	Backdoor attack
3	DoS	13	Vulnerability exploitation
4	DDOS	14	Universal attack
5	Web	15	Reconnaissance attack
6	Infiltration	16	Shell code attack
7	Probe	17	Worm attack
8	R2L	18	Benign
9	U2R		

The paper uses the Grammer algorithm to construct the first layer of a feature selection model, which is used to calculate the maximum correlation between feature attributes. Firstly, for the correlation between multivariate classification attributes, chi square statistics are used to measure them. The chi square statistics are calculated based on the observed frequency and expected frequency of the attributes, and the formula is as follows:

$$\chi^2 = \sum_i \sum_j \frac{(n_{ij} - \pi_{ij})^2}{\pi_{ij}} \qquad (2)$$

Among them, n_{ij} is the actual frequency of the jth feature attribute of the ith data,π_{ij} is the expected frequency of the jth feature attribute of the ith data, and the actual frequency=the actual quantity of the attribute/the total number of samples, and the theoretical frequency (expected frequency)=the expected quantity of the attribute/the total number of samples.

For the intrusion detection dataset in the paper, assuming that the contingency table formed by the dataset is an $R \times C$ pattern, which is a two-dimensional list of R rows and C columns, the V consistency coefficient of the Kramer method is:

$$V = \sqrt{\frac{\chi^2}{nmin\left[(R-1),(C-1)\right]}} \tag{3}$$

Moreover, the Kramer algorithm is a measure of the correlation between multiple categorical variables, with a range of values for the V consistency coefficient of [0,1], where 0 indicates that the two variables are independent or independent of each other, and 1 indicates that the two variables are completely correlated.

The paper uses a multi distance weighted function to construct the second layer of a feature selection model, measuring the independence of each feature. The farther the distance, the higher its independence and the lower its redundancy. In multidimensional data space structures, Euclidean distance is a method of measuring the distance between two vectors in a multidimensional space. It effectively captures the overall difference between two vectors in space, but does not take into account the similarity between individual feature attributes. There is a problem with directly applying Euclidean distance between feature vectors, which is that differences in feature metrics may affect the accuracy of redundancy. To address this issue, the paper introduces the concept of weighted Euclidean distance to more accurately measure the redundancy between feature vectors. The calculation expression is:

$$NED\left(X,Y\right) = \sqrt{\sum_{K=1}^{N} \frac{w_i}{W}\left(X_k - y_k\right)^2} \tag{4}$$

Among them, $\frac{w_i}{W}$represents the normalization factor, $w_i = e^{-\frac{|x_i - y_i|}{\sigma}}$,$W = \sum w_i$;$\sigma$ is the regulatory factor, and in the paper, $\sigma = 1$. After obtaining the maximum correlation coefficient and maximum Euclidean distance for each feature value, the relationship between each feature attribute is obtained according to Formula (5), with a range of values between (0, 1). The larger the value, the greater the correlation between the two types of features and the smaller the similarity of information. They are considered as retained features.

$$S = \frac{NED\left(X,Y\right) + V}{2} \tag{5}$$

4. **A Classification Method for Network Attacks in Intelligent Measurement Systems.** A intrusion detection model based on GRU_KNN was proposed. The model combines GRU network and classifiers such as KNN and SVM, with each part cascading with each other. The entire model is divided into three main parts: feature learning processing model, KNN model, and SVM model.

(1) Feature processing model: Using recurrent neural networks to extract features from the processed intrusion detection dataset of the power monitoring system.

(2) SVM model: Classify based on the features extracted by the feature processing model, use the one-to-many method of SVM classifier, and output the predicted classification results.

(3) KNN model: uses the SVM model to output other class sample data for classification, calculates the distance between samples using weighted Euclidean distance, and outputs the predicted classification results.

4.1. **Design of GRU Feature Learning Model.** Gated Recurrent Unit (GRU) is a recurrent neural network structure similar to Long Short Term Memory Network (LSTM), but with a more simplified design. Unlike the input gate, forget gate, and output gate of LSTM, GRU controls information flow by using reset gates and update gates. In most cases, GRU and LSTM perform similarly, but they may differ when processing datasets of different sizes. GRU is usually more suitable for smaller datasets because it has fewer parameters and a simpler structure, allowing for more efficient processing of smaller scale data. The dataset size used in the experiment is still relatively small, so the GRU method is used for feature processing of the dataset.

Using GRU as the first layer of the intrusion detection model, utilizing the capabilities of recurrent neural networks to process time series data. Recurrent neural networks process input sequences in time steps, utilize hidden states (also known as memory units) to store information from previous time steps, and update internal states based on current input information and hidden states. Compared to other neural network algorithms, the gated recurrent unit method is concise and easy to train. It includes two key gate controllers: an update gate and a reset gate, which can update internal memory units or combine new input information with previous memory. This method can effectively preserve the time series characteristics of the data, making the model more suitable for processing temporal data. Due to its simple structure, the training efficiency has also been significantly improved. The GRU structure is shown in Figure 1.

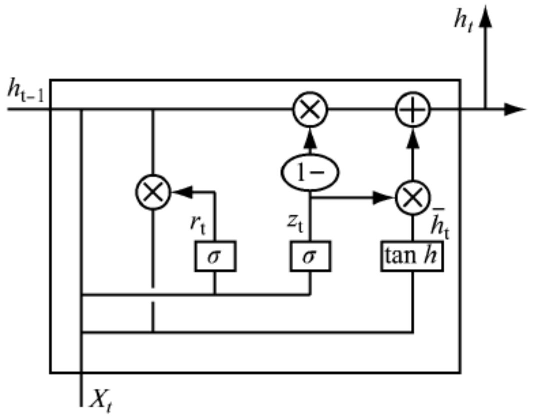

Figure 1. GRU Structure

In the structure of GRU cell units, x_t is the current input state, h_t is the current output state, h_{t-1} is the hidden state of the previous time step, that is, the output state of the previous GRU unit, z_t represents the update gate, r_t represents the reset gate, and $\widetilde{h_t}$ represents candidate state information. The update gate z_t determines whether to update the hidden state h_{t-1} to the current state h_t, specifically represented as:

$$z_t = \sigma\left(W_s\left[h_{t-1}, x_t\right]\right) \tag{6}$$

Among them, σ is the sigmoid function, and W_s is the weight parameter. The reset gate r_t determines whether to reset the hidden state h_{t-1}, represented by:

$$r_t = \sigma(W_r \times [h_{t-1}, x_t]) \tag{7}$$

Among them, W_{r} is the weight parameter. The historical information of data recorded by $\tilde{h}_t$ and h_t is represented as:

$$\tilde{h}_t = \tan h(W_S \times [r_t \times h_{t-1}, x_t]) \tag{8}$$

$$h_t = (1 - z_t)\, h_{t-1} + z_t h_t \tag{9}$$

This paper establishes a network model for feature learning, as shown in Figure 2.

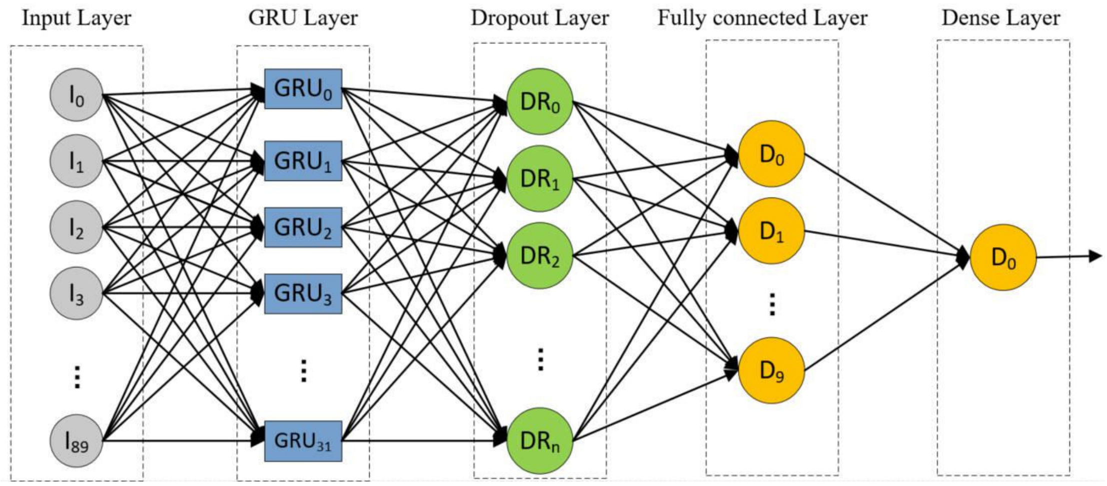

Figure 2. Feature Learning Netwwork Model

4.2. **Design of SVM_KNN classification model.** Using SVM and KNN algorithms for intrusion detection model classifier, using different methods to classify different types of samples based on their relative numbers in the dataset.

The SVM classifier classifies the dataset based on the partition hyperplane. In most cases, SVM is used for binary classification problems and can only divide data into two different categories: 0 or 1. However, in practical scenarios, most problems are multi classification problems. Therefore, in SVM classifiers, it is necessary to choose appropriate methods to solve the classification problem of attacking data. Due to the fact that the one against-one method can effectively reduce the amount of test data and improve data processing speed in the construction of SVM multi classifiers, the feature attribute values output by the previous layer of recurrent neural network and corresponding data records are used as inputs to the SVM classifier. The one-on-one method is used to construct the SVM classifier, which is used to classify five types of R2L, U2R, fuzzy attack, vulnerability exploitation, and worm attack that account for less than 0.01% of the dataset. This method can be divided into six categories: R2L, U2R, fuzzy attack, vulnerability exploitation, worm attack, and other categories. Other categories include R2L, U2R, vulnerability exploitation, worm attack, and other categories. All data categories (including normal traffic and attack traffic) except for the five types of 2R, fuzz attack, vulnerability exploitation, and worm attack.

The core idea of one-on-one method is to design an SVM binary classifier between any two attack categories when constructing an intrusion detection multi classification model. For example, normal traffic data and attack data are classified as binary, and the first type of attack is classified as binary with other types of attacks. Therefore, an SVM binary classifier should be designed for the 19 categories of data in the intrusion detection dataset. Perform binary classification on the preprocessed dataset, count the votes of each data record in the corresponding category, and the attack type with the most votes is the category corresponding to that data record. The specific process of using SVM method to achieve multi classification is as follows:

Input: Intrusion detection data samples after feature processing.

Output: Category of intrusion detection data samples.

Firstly, initialize the categories of the data samples, including R2L: 0, U2R: 1, fuzz attack: 2, vulnerability exploitation: 3, worm attack: 4, other categories: 5.

Secondly, an SVM binary classifier is used to distinguish between categories 0 and 1, adding one vote to the recognized category;

Classify categories 0 and 1, adding one vote to the identified category;

Classify categories 1 and 2, adding one vote to the identified category;

And so on, for other pairs of categories;

Classify categories 3 and 5, adding one vote to the identified category;

Classify categories 4 and 5, adding one vote to the identified category;

Finally, calculate the number of votes received for the intrusion detection dataset in five categories and classify the sample as the category with the highest number of votes.

The selection of kernel functions in SVM needs to consider the non-linear and large number of data samples. Due to the inability of linear kernel functions to handle such nonlinear data well, and the high computational complexity of polynomial kernel functions and sigmoid kernel functions, we chose to use Gaussian kernel functions to construct an SVM classifier for multi classification of data, mapping the data from the original space to an infinite dimensional space. This can be more convenient for model design and optimization. The Gaussian kernel functions are as follows:

$$\kappa\left(X_1, X_2\right) = exp\left(-\frac{\left\|X_1 - X_2\right\|^2}{2\sigma^2}\right) \tag{10}$$

Among them,$\left\|X_1 - X_2\right\|^2$can be seen as the squared Euclidean distance between two feature vectors, σ It is a user-defined parameter used to determine the speed at which the function value drops to 0.

In SVM classifiers, the initialization setting of kernel function parameters has a significant impact on the performance of the classifier. Kernel function parameters reflect the degree to which a single sample affects the optimal hyperplane when mapping data samples to high-dimensional space. When the kernel function parameters are set to a larger value, it means that a single data sample has a closer impact on the optimal hyperplane, resulting in fewer support vector selections and higher complexity of the model; When the parameters of the kernel function are set to a small value, it means that the influence of the data sample on the optimal hyperplane is far, resulting in more support vector choices and lower complexity of the model.

In addition, the penalty factor C is used to balance the training error and complexity of the model, and to penalize misclassified data samples. When the C value is large, the punishment for intrusion detection error is greater, resulting in an increase in model complexity and fewer support vectors; When the C value is small, the punishment for intrusion detection error is relatively small, the model is relatively simple, and it is easy to ignore outlier intrusion sample data, resulting in more support vectors. When designing an SVM classifier, it is necessary to carefully select the values of kernel function parameters and penalty factors to achieve an appropriate balance between model complexity and performance. By adjusting these parameters reasonably, the accuracy and efficiency of intrusion detection systems can be improved, thereby better protecting the security of intelligent measurement systems.

By combining the feature attribute values output by the previous recurrent neural network with the corresponding data records as inputs to the KNN classifier. Using the KNN principle, classify the remaining large sample data into categories including brute force cracking attacks, zombie networks, DOS, DDOS, Web, Infiltration, Probe, analysis attacks, backdoor attacks, general attacks, reconnaissance attacks, shell code attacks, and normal data traffic.

The classic K-nearest neighbor (KNN) algorithm is a simple and intuitive supervised learning algorithm, but it may have some limitations in practice. One of the main issues is that it assumes that each attribute of the sample points has the same impact on distance, so fixed weights are used to calculate the distance between sample points. However, in real-world data, the contribution of each attribute may be different, and if these differences are not taken into account when calculating distances, it may lead to serious deviations between actual and predicted results. To solve this problem, the weighted KNN algorithm can be used. This method adjusts the contribution of attributes to distance by assigning different weights to each attribute, thereby better reflecting the importance of attributes. For example, after determining the important attributes of a dataset through feature selection methods, assign higher weights to these attributes. In addition, when determining the K value, the influence of the number of neighboring samples in different categories on classification accuracy is considered to ensure that the classification accuracy in different categories reaches the optimal value, thereby avoiding the problem of misclassification caused by improper initial value setting of algorithm parameters, which may lead to a decrease in algorithm efficiency.

Therefore, the weighted KNN algorithm was introduced to optimize the weights of samples with different distances, and the weighted Euclidean distance between the test sample point and each other sample point was calculated. The formula is as follows:

$$NED\left(X,Y\right)=\sqrt{\sum_{K=1}^{N}\frac{w_i}{W}\left(X_K-Y_K\right)^2} \tag{11}$$

Among them, N is the number of attributes each record has, $N = 154$. X and Y represent any two records (vectors) in the dataset, X_K represents the K-th attribute in X , and Y_K represents the K-th attribute in Y , $\frac{w_i}{W}$ represents the normalization factor, $w_i = e^{-\frac{|x_i-y_i|}{\sigma}}$, $W = \sum w_i$, σ is the regulatory factor, and in the paper, $\sigma = 1$.

The steps to classify large sample data using the weighted KNN algorithm are as follows:

Firstly, for the output data of the SVM classifier, calculate the weighted Euclidean distance between the test set samples and the training set samples of the dataset;

Secondly, initialize the neighboring value K and sort the calculated weighted Euclidean distance, selecting the K sample points with the smallest distance based on the sorting results;

Again, count the number of occurrences of each attack category in the K nearest neighbor samples;

Finally, compare the categories to which K points belong, and classify the sample points into the category with the highest proportion based on the principle of minority obeying majority.

5. **Experiment and Result Analysis.**

The paper uses the four publicly available network intrusion detection datasets mentioned above as benchmarks to verify the proposed method's effectiveness. The intrusion detection model is shown in Figure 3.

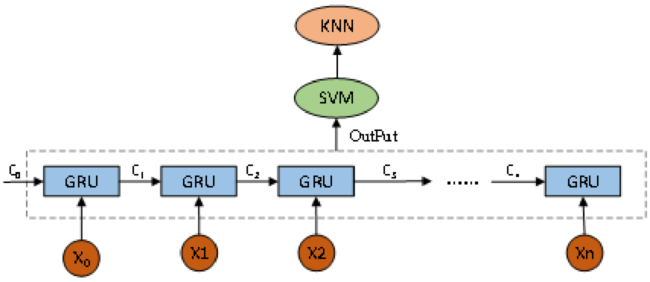

Figure 3. Intrusion Detection Model

Table 12. Simulation Parametric Design

Parameter Name	Allocation
Optimizer	Adam
loss function	Binary_crossentropy
Batch size	256
Epochs	10
dropout	0.5

5.1. **Evaluation indicators.** In this experiment, accuracy (Acc) was mainly used as the evaluation indicator.

The accuracy represents the proportion of correctly predicted sample N_{R} to the total sample N_{S}, calculated as Formula (12).

$$Acc = \frac{N_R}{N_S} \tag{12}$$

5.2. **Result Analysis.** In the various experimental parameters of the GRU-SKNN network architecture, the batch size and epochs parameters have an impact on the training accuracy and required training time of the model. In order to improve training efficiency, repeated tests have found that a batch size of 256 can reach the optimal state, which can obtain more accurate descent direction and reduce training oscillation amplitude during the training process. At the same time, using dropout to solve the overfitting problem during neural network training. Dropout refers to the process of discarding network neurons with a certain probability during neural network training, in order to reduce the degree of interaction between neurons, simplify the neural network structure, and effectively avoid overfitting. Finally, this experiment determined through testing that the GS-KNN network architecture consists of a 32 unit GRU layer, a Dropout layer with a drop rate of 0.5, a fully connected network layer with 10 neurons, and an S-KNN layer. The specific parameter settings are shown in Table 13.

The preprocessed data is integrated and divided into training and dataset sets. The full feature classification model and the KNN, LSTM, GRU, GRU_SVM after G-mRMR feature selection algorithm, as well as the proposed GRU_KNN model are used for classification. Figures 4-6 show the training results of each training model, and Table 14 shows the accuracy of each training model.

According to the classification training results and accuracy in the table above, it can be seen that the intrusion detection model based on feature selection and GRU+KNN algorithm has improved accuracy by 13.3% compared to traditional GRU models, 6.5% compared to GRU+SVM models, 12.5% compared to LSTM models, and 12% compared

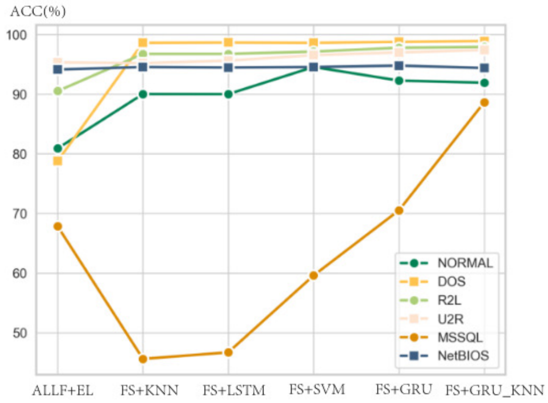

Figure 4. LDAP/SSDP/UDPLag/DNS Classification Model

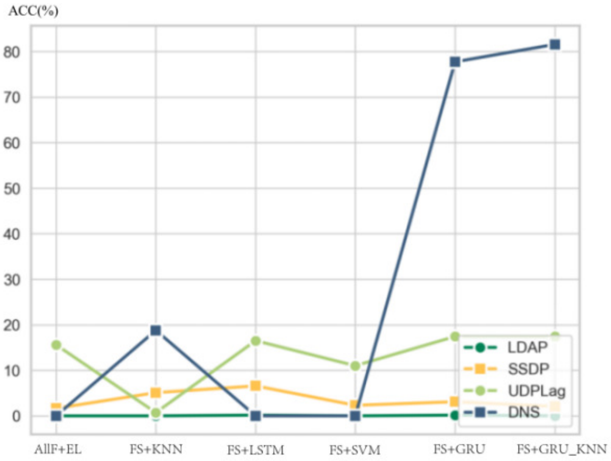

Figure 5. NORMAL/DOS/R2L/U2R/MSSQL/NetBIOS Classification Model

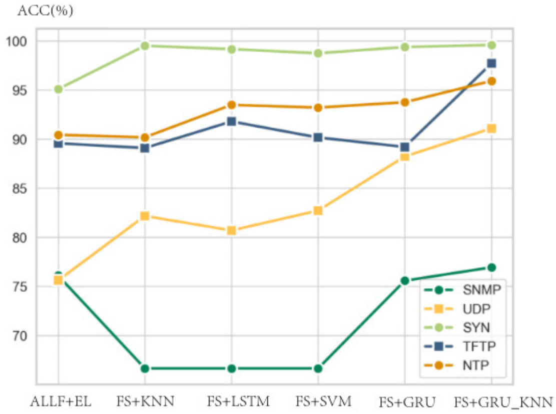

Figure 6. SNMP/UDP/SYN/TFTP/NTP Classification Model

to KNN models. It can be seen that the proposed intrusion detection model based on GRU+KNN algorithm performs the best in terms of accuracy among all models.

Table 13. Accuracy of each training model(%)

Attack type	ALLF+EL	FS+GRU	FS+GS-KNN
NORMAL	80.90	92.27	91.91
DOS	78.79	98.76	98.90
R2L	90.51	97.77	97.92
U2R	95.32	96.99	97.41
PROBE	90.76	97.29	97.49
LDAP	0.00	0.16	0.00
MSSQL	67.83	70.51	88.63
NetBIOS	94.13	94.78	94.38
SNMP	76.10	75.58	79.94
SSDP	1.66	3.10	6.07
UDP	75.61	88.23	91.09
SYN	95.09	99.37	99.58
TFTP	89.57	89.20	97.70
NTP	90.43	93.75	95.92
UDPLag	15.60	17.43	17.43
DNS	0.00	77.74	85.57

6. **Conclusions.** In order to more effectively detect abnormal behavior in the intelligent measurement system network in a large amount of complex network data, this paper proposes a feature selection method and an intelligent measurement system network attack classification method based on the characteristics of sample size and time attribute of sample data. This method combines neural network algorithms and basic classification methods to detect abnormal behavior in the network, effectively solving the problem of excessive feature attributes in the dataset leading to a decrease in classification accuracy in the later stage. The paper evaluates the algorithm based on the network public intrusion detection datasets CIC-IDS-2018, CIC-DDOS (2019), NSL-KDD, and UNSW-NB15 as benchmark datasets, demonstrating that the algorithm has good detection performance. Compared with other algorithms, the accuracy of the paper's algorithm detection is significantly better than other algorithms.

Acknowledgment. This work is partially supported by the National Key R&D Program Projects (No. 2022YFB3206800). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] N. Chaabouni, M. Mosbah and A. Zemmari. "Network intrusion detection for IoT security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671-2701, 2019.

[2] S. Hansman, R. Hunt. "A taxonomy of network and computer attacks," *Computers and Security*, vol. 24, no. 1, pp.:31–43, 2005.

[3] Y. Li, Y. Xu and Z. Liu. "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," *Measurement*, vol. 154, pp. 107450, 2020.

[4] M. Al-Hawawreh, N. Moustafa and E. Sitnikova. "Identification of malicious activities in industrial Internet of Things based on deep learning models," *Journal of Information Security and Applications*, vol. 21, pp. 1-11, 2018.

[5] A. Kim, M. Park and D. H. Lee. "AI-IDS: Application of deep learning to real-Time web intrusion detection," *IEEE Access*, vol. 8, pp.70245-70261, 2020.

[6] Y.-J. Zhao, L. Shi and X. Qi. "Transformer fault detection based on enhanced gray wolf optimization VMD-DBN," *Electrical Measurement & Instrumentation*, vol. 61, no. 2, pp. 157-163, 2024.

[7] M.-M. Hassan, A. Gumaei and A. Alsanad. "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386-396, 2020.

[8] C.-M. Hsu, M.-Z. Azhari and H.-Y. Hsieh. "Robust network intrusion detection scheme using long-short term memory based convolutional neural networks," *Mobile Networks and Applications*, vol. 26, no. 3, pp.1137-1144, 2021.

[9] T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, Y. Tian and N.-A. Al-Nabhan. "An enhanced pairing-based authentication scheme for smart grid communications," *Journal of Ambient Intelligence and Humanized Computing*, 2021. [Online]. Available: https://doi.org/10.1007/s12652-020-02740-2.

[10] Z.-H. Ren."An efficient intrusion detection model for power client side terminal network," *Electrical Measurement & Instrumentation*, vol. 59, no. 5, pp. 149-157, 2022.

[11] Z.-Y. Yang. "Overall architecture analysis system of abnormal electricity behavior based on big data and machine learning," *Electrical Measurement & Instrumentation*, vol. 60, no. 6, pp. 167-173, 2023.

[12] J.-Q. Ding, Y. Wen and Q.-S. Lv. "Abnormal state detection method of power equipment based on time series and neural network," *Electrical Measurement & Instrumentation*, vol. 61, no. 2, pp. 185-190, 2024.

[13] T.-Y. Wu, L.-Y. Wang and C.-M. Chen. "Enhancing the security: a lightweight authentication and key agreement protocol for smart medical services in the IoHT," *Mathematics*, vol. 11(17), pp. 3701, 2023.

[14] S. Afroz, S.-A. Islam and S.-N. Rafa, "A two layer machine learning system for intrusion detection based on random forest and support vector machine," in *IEEE International Women in Engineering Conference on Electrical and Computer Engineering.* IEEE, 2020, pp. 300-303.

[15] Z.-X. Fu, Y. Xu and Z.-D. Wu. "Intrusion detection method for SVM-KNN network based on incremental learning," *Computer Engineering*, vol. 46, no. 04, pp. 115-122, 2020.

[16] M. Pozi, M.-N. Sulaiman and N. Mustapha. "Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming," *Neural Processing Letters*, vol. 44, no. 2, pp. 1-12, 2015.

[17] R. Bhadoria, N. Bhoj and H. Zaini. "Artificial intelligence for creating low latency and predictive intrusion detection with security enhancement in power systems," *Applied Sciences*, vol. 11, no. 24, pp. 11988, 2021.

[18] M. Pozi, N.-M. Sulaiman and N. Mustapha. "Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming," *Neural Processing Letters*, vol. 44, no. 2, pp. 1-12, 2015.

[19] T.-Y. Wu, F.-F. Kong, Q. Meng, S. Kumari and C.-M. Chen. "Rotating behind security: an enhanced authentication protocol for IoT-enabled devices in distributed cloud computing architecture," *EURASIP Journal on Wireless Communications and Networking*, vol. 2023, pp. 36, 2023.

[20] S.-M. Erfani, S. Rajasegarar and S. Karunasekera. "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, pp. 58, 2016.

[21] N. Zhu, C. Zhu and L. Zhou. "Optimization of the random forest hyperparameters for power industrial control systems intrusion detection using an improved grid search algorithm," *Applied Sciences*, vol. 12, no. 20, pp. 1045, 2022.