sustainability

MDPI

Article

An Ensemble of a Prediction and Learning Mechanism for Improving Accuracy of Anomaly Detection in Network Intrusion Environments

Imran , Faisal Jamil and Dohyeun Kim *

Computer Engineering Department, Jeju National University, Jeju-si 63243, Korea ; imranjofficial@jejunu.ac.kr (I.); faisal@jejunu.ac.kr (F.J.)
* Correspondence: kimdh@jejunu.ac.kr

Abstract: The connectivity of our surrounding objects to the internet plays a tremendous role in our daily lives. Many network applications have been developed in every domain of life, including business, healthcare, smart homes, and smart cities, to name a few. As these network applications provide a wide range of services for large user groups, the network intruders are prone to developing intrusion skills for attack and malicious compliance. Therefore, safeguarding network applications and things connected to the internet has always been a point of interest for researchers. Many studies propose solutions for intrusion detection systems and intrusion prevention systems. Network communities have produced benchmark datasets available for researchers to improve the accuracy of intrusion detection systems. The scientific community has presented data mining and machine learning-based mechanisms to detect intrusion with high classification accuracy. This paper presents an intrusion detection system based on the ensemble of prediction and learning mechanisms to improve anomaly detection accuracy in a network intrusion environment. The learning mechanism is based on automated machine learning, and the prediction model is based on the Kalman filter. Performance analysis of the proposed intrusion detection system is evaluated using publicly available intrusion datasets UNSW-NB15 and CICIDS2017. The proposed model-based intrusion detection accuracy for the UNSW-NB15 dataset is 98.801 percent, and the CICIDS2017 dataset is 97.02 percent. The performance comparison results show that the proposed ensemble model-based intrusion detection significantly improves the intrusion detection accuracy.

Keywords: intrusion detection; intrusion accuracy; automated machine learning; CICIDS2017; UNSW-NB15

Citation: Imran; Jamil, F.; Kim, D. An Ensemble of Prediction and Learning Mechanism for Improving Accuracy of Anomaly Detection in Network Intrusion Environments. *Sustainability* **2021**, *13*, 10057. https://doi.org/10.3390/su131810057

Academic Editor: Manuel Fernandez-Veiga

Received: 15 July 2021
Accepted: 1 September 2021
Published: 8 September 2021

1. Introduction

An expeditious rise in the development of network and communication technologies leads to an immense amount of network data generated from a wide range of services. For instance, pervasive computing networks such as the Internet of Things (IoT) generate enormous data [1–3]. A wide range of network applications is developed in every domain of life, including business, healthcare, smart homes, and smart cities, to name a few [4–7]. The plethora of high-dimensional data increases the need for analysis tools based on advanced data mining and statistical methods [8,9]. There is a dire need to tune the contemporary data mining and statistical methods to address the challenges of the growing internet applications, such as bandwidth handling, network intrusion detection, and scalability. Network applications and resources' security using intrusion detection systems, intrusion prevention systems, and hybrid systems are becoming more challenging due to the enormous number of diverse networking applications. However, the rule-based approach for the analysis of enormous data has many limitations. The existing state-of-the-art intrusion detection-based systems focus on increasing the reliability aspect of these applications [10]. An efficient intrusion detection system can strengthen the defense system of such applications against anomalies and network intrusion attacks. The intrusion detection system also

provides real-time analysis of the collected critical reconnaissance data during defensive attacks. Intrusion detection systems based on artificial intelligence(AI) hold a significant potential to enhance the performance of detection mechanisms by learning from historical data and real-time data patterns.

Scientific community has presented various machine learning-based intrusion detection systems such as support vector machine (SVM) [11], Naive Bayes (NBs) [12], clustering [13], artificial neural network (ANN), and deep learning network (DNN) [14]. Conventional machine learning algorithms can better classify small and low dimension datasets. However, the classification accuracy of these algorithms deteriorates when it comes to addressing problems involving high dimensionality and nonlinearity. Hence, the need for intrusion detection models to address the classification accuracy problem increases as AI advances. For example, a convolutional neural network (CNN) [15] and long short-term memory (LSTM) [16] have been applied in natural language processing (NLP) and computer vision applications. The problem with deep learning techniques such as CNN and LSTM is adaptability to nonlinear and high-dimensional data. The issue of nonlinearity has been addressed in CNN and LSTM for modeling nonlinear systems [17–22]. In literature, the issues of high dimensional data are handled in CNN, and LSTM using a deep learning paradigm [23–26]. Automated machine learning (autoML) is a newly emerged subfield of machine learning and data science. The feasible adaptability of autoML makes it equally useful for trainees of machine learning, data scientists, and machine learning engineers. Research articles demonstrate that autoML can revolutionize constructing machine learning models without machine learning expertise and knowing technical specifications. AutoML architectures produce a code pipeline by suggesting and selecting a model from a list of machine learning model-based input datasets [27]. The selection is performed based on the accuracy of these machine learning models. AutoML results in coding the pipeline of the best performing model, which will be very difficult to find using manual configurations of the models' parameters.

This paper presents an intrusion detection system based on the ensemble of prediction and learning mechanisms to improve intrusion detection accuracy. The conceptual design of the proposed ensemble intrusion detection model for improving the performance accuracy of anomaly detection is envisioned in Figure 1. The proposed learning mechanism is based on autoML [28], and the prediction mechanism is based on Kalman filter [29]. First, the automated neural architecture search paradigm of autoML improves the accuracy of the learning model using parameters optimization. Then, the optimal DNN (o-DNN) model pipeline is created from an autoML based learning mechanism. Next, an optimal Kalman filter-based intrusion detection system is produced using measuring and updating errors. Finally, the o-DNN and Kalman filter are utilized to develop the ensemble intrusion detection model based on the weighted voting mechanism.

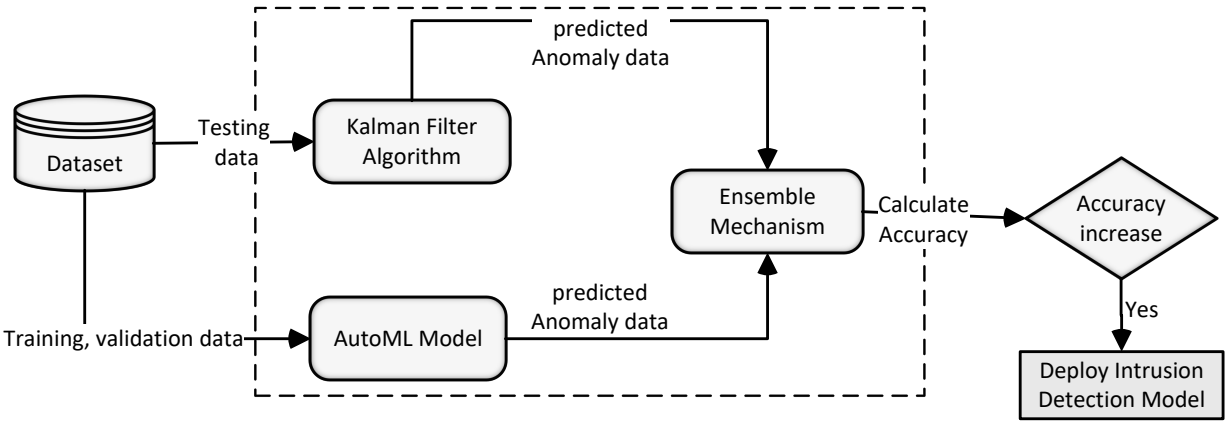


Figure 1. Conceptual diagram of ensemble of learning and prediction mechanism for anomaly detection.

The proposed ensemble-based intrusion detection model is selected based on accuracy comparison with DNN and Kalman filter-based intrusion detection models. If the accuracy of intrusion detection is improved using the proposed ensemble approach, it is deployed as an intrusion detection model in a network intrusion environment.

The main contributions of this paper are as follows:

- Design of ensemble mechanism based on learning and prediction models.
- Implementation of ensemble model based intrusion detection system for improving the detection accuracy.
- Case studies based on benchmarks' intrusion detection datasets.

The case studies based on intrusion detection datasets are used to assess the proposed ensemble mechanism for intrusion detection environments. The performance of the proposed model is compared with some contemporary models, including DNN, autoML, and other algorithms from the literature on these benchmark datasets. The case studies are evaluated using benchmark datasets UNSW-NB15 and CICIDS2017.

The rest of the paper is organized as follows: a brief literature review is presented in Section 2. Section 3 presents the methodology for the proposed intrusion detection system. Experimental results are discussed in Section 4. Section 5 presents performance analysis and significance of the study. Finally, the conclusion and future work directions is presented in Section 6.

2. Related Work

Artificial intelligence is taking over the current era and is changing the current era into a revolutionary practical world. Data analysis, predictive analytics and optimization models are used for many real-life applications [30–32]. Anomaly detection is a type of data analysis used to identify irregular and abnormal data from a given data set. Anomaly detection is the approach used in data mining applications for discovering and finding patterns inside the data [33]. It is also used as a standalone module in many studies related to machine learning and statistics applications. Deviation detection, outlier detection, and exception mining are related terms used for anomaly detection [34]. Narayana et al. defined anomaly as a mechanism generated from the deviation of several observations [35]. Anomaly detection is used in several scientific domains such as healthcare, intrusion detection, sensor network, and fraud detection, to name a few. Detecting irregularities in the network, identifying anomalies in financial transactions, detecting fraudulent activities, and detecting anomalies in medical images are some anomaly detection applications [36]. In networks, anomaly patterns can be identified based on the classification of packet data containing abnormal patterns.

Xie et al. published a survey study related to intrusion detection in wireless sensor networks [37]. According to most of the studies, intrusion detection depends on the communication medium; for example, wired connection-based techniques cannot be applied to the wireless communication medium. The survey emphasizes the need for standard anomaly detection techniques for all types of networks. One challenge for detecting anomalies in the network is the lack of a comprehensive dataset. Most of the current anomaly detection systems are based on supervised approaches that use labeled data knowledge. During the past few years, research has been conducted in network intrusion detection segregated into audit source, network behavior, detection method, location, frequency of usage, and detection method. In [38], Debar et al. presented a standard technique based on the extension of transaction-based detection paradigm. Axelsson et al. [39] proposed a study based on detection principle and focus on operational aspects. Furnell et al. [40] proposed an intrusion matrix based on the data scale and output type. Estevez-Tapiador et al. presented a wired-based network intrusion detection based on anomaly detection [41]. Boukerche et al. presented an outlier-based classified detection approach using the unsupervised and supervised models [42]. Under the supervised category, a proximity-based technique has been used recently [43].

Chandola et al. also presented another detailed survey study on anomaly detection [44]. Their study presents different techniques related to intrusion detections. Some studies proposed several anomaly detection techniques based on supervised, unsupervised, and clustering methods [45–49]. The lack of discussion and research problems in the available datasets are one of the research gaps that need to be addressed. The most used datasets for network anomaly detection are the DARPA/KDD, which developed in 2013. Various variants of datasets are developed based on this dataset to address the causes of data errors and inconsistency. As network anomaly detection based on the aforementioned dataset has no significant performance improvements; therefore, more anomaly detection datasets have been introduced recently to improve intrusion detection system efficiency. Some research surveys focused on these dataset issues and challenges to develop an efficient intrusion detection system [50]. The network attack profile feature relies on classification-based techniques and the size of the data [51]. The intrusion detection system process is based on the signature of the attack and the capability of intrusion detection system to detect the attack from data patterns [52]. The intrusion detection engine can also enhance the defense system using intelligent mechanisms for various attacks' variants. This process is quite expensive for creating a new attack in case of loss or replacement [53]. Furthermore, the regular traffic does not contain the knowledge base attack, and it will be raising the wrong alarms.

In summary, anomaly detection mechanisms are costly in terms of time and are relying on the existing network traffic dataset. Furthermore, keeping the standard profile up-to-date is very difficult in today's network. The network traffic analysis dataset does not have easy access due to privacy limitations. Examples of benchmark datasets for intrusion detection are DARPA/KDD, UNSW-NB15, CICIDS2017, and CSE-CIC-IDS2018 [54]. The main challenge that needs to be addressed is improving intrusion detection systems' accuracy on these benchmarks' datasets. Table 1 presents a summary of existing intrusion detection and prevention systems organized as applications, datasets, models, and relative demerits.

Table 1. Summary of existing intrusion detection and prevention systems.

Application	Datasets	Model	Relative Demerits
Anomaly Detection [55]	InSDN	TRW-CB algorithm	Standardized programmability and can predict anomalies in SOHO Network
DoS attacks detection [56]	KDD-99	Self-organizing maps, ANN	Lightweight DDoS Flooding Attack but do not have any flow rules installed.
Anomaly Detection [57]	NSL-KDD	DNN approach	Does not scale well for commercial product but is a good alternative solution for signature-based intrusion detection system
DDoS Detection System [58]	Simulated data	Stack auto-encoder and DNN	Detect all DDoS attack, but has a Controller bottleneck in a wide networks.
Intrusion Detection [59]	Simulated data	Self organizing map and learning vector quantization	Detect U2R attacks but limited to deep packet inspection technique.
Monitor traffic flow [60]	Simulated data	Flow analysis tool	Improve computation time of flow but difficult to handle due to batch processing. Flow analysis tools are not compatible with the MapReduce interface.
P2P botnet detection [61]	CAIDA,simulated data	Random forest	Process high bandwidth and efficiently analyze malicious traffic data. However, the high drop rate of packets and delay in detection make it inefficient for new complex threats.
Intrusion detection [62]	NSL-KDD 99	NB tree, random forest	Improved performance accuracy reduces false-positive rate for hybrid approaches, but the false-positive rate is high for non-hybrid approaches.
Phishing-based attack detection [63]	Simulated data	Collaborative mechanism	Practical method for generalization to any attacks but no validation with real datasets.
Intrusion detection [64]	KDD 99, CMDC 2012	OneR algorithm, KNN, SVM	Faster but feature reduction and training mechanism is real overhead.
Malware detection [65]	Simulated data	Choi–Williams distribution	Effective for Kelihos injection but not tested with real datasets.
Intrusion detection system [66]	Simulated data	RSFSA, fuzzy logic based SVM	Faster mechanism for decision attributes and log data reduction though not tested with real datasets.
Network traffic monitoring [67]	CAIDA	IP Trace Analysis System	Useful for passive analysis but does not provide a fine-grained analysis.

3. Materials and Methods

This section presents the design of the proposed ensemble mechanism-based intrusion detection methodology. The datasets and methods used for data preparation are briefly

discussed. The intrusion detection system is based on the ensemble of learning and prediction models. The main aim of the proposed ensemble mechanism-based intrusion detection system is to improve the accuracy of intrusion detection. The proposed intrusion detection system has two main phases. The first phase is training, and the second phase is deployment in an intrusion detection environment. The first phase of the methodology for the ensemble intrusion detection model based on comprehensive datasets is shown in Figure 2. Two comprehensive benchmark datasets are prepared using feature engineering techniques to assess the performance of the proposed ensemble model. The data are prepared and split into training, validation, and testing datasets for the training, testing, and validation of the proposed intrusion detection model.

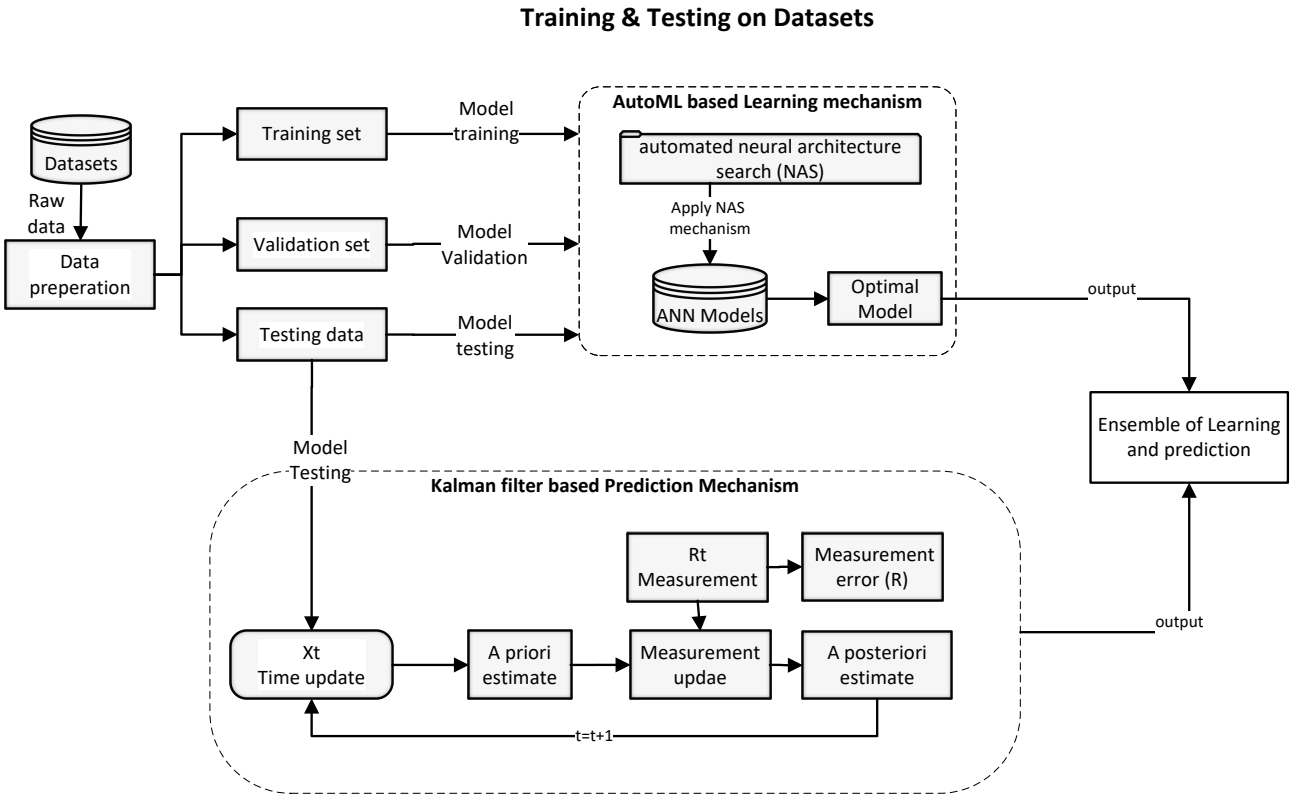


Figure 2. Ensemble model based on learning and prediction mechanisms.

The autoML based learning mechanism utilizes automated neural architecture search to obtain O-DNN using hyperparameter optimization, and accuracy metric. We used the Bayesian optimization (BayesOpt) method for hyperparameter optimization. Auto-kreas uses BayesOpt as an optimization method for neural network architecture search(NAS). BayesOpt has a sequential procedure for global optimization of black-box functions. The BayesOpt requires a dataset prepared for machine learning models and defined search space. The search space is based on a large set of neural network architectures. The search space consists of architectural parameters such as the number of layers, activation functions, and density. The next step involves defining an objective function that aims to search for architectural parameters to improve the training and testing accuracy. Edit distance function is used to find the distance between architectures. Let f(accuracy) be the objective function for maximizing the accuracy. The goal of the BayesOpt mechanism is to find a deep learning architecture D $\in$ set of architectures list, which maximizes f(accuracy). The Kalman filter model is used for intrusion detection, and its parameters are tuned to improve the accuracy. For instance, the final R parameter value of the Kalman filter is selected based on the average of R values which performs better for both comprehensive datasets.

The ensemble model is based on the O-DNN model and the Kalman filter. The ensemble model is based on the weighted voting mechanism. The predictions of both learning and prediction mechanisms are weighted based on their average prediction accuracy.

The second phase of the methodology is testing the proposed ensemble model for the intrusion detection system. Figure 3 presents the testing of the proposed intrusion detection system based on the ensemble model in the intrusion detection environment.

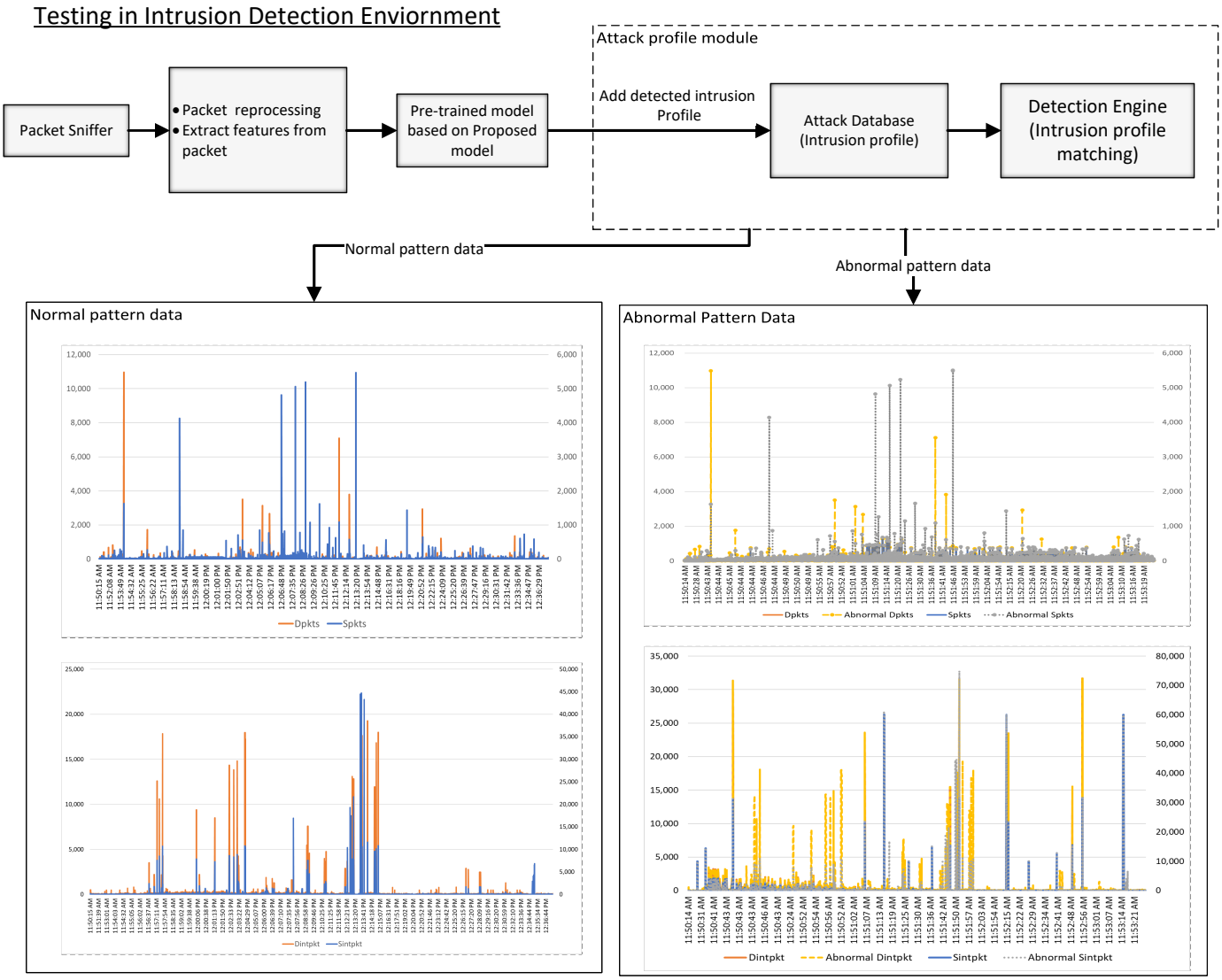


Figure 3. Intrusion detection system based on the proposed ensemble model.

The pre-trained model based on the ensemble mechanism is deployed in a networking environment to predict anomalies in the network data. For network capturing and pre-processing, a Raspberry Pi based capturing probe is used. The extracted data profile is given as an input to the ensemble model to analyze the data pattern for intrusions. If the input packet pattern is abnormal and matches the signatures in the intrusion detection engine, the packet is dropped. Suppose that the abnormal pattern signature is not present in the intrusion detection engine. In that case, the database is updated with the new signature, and the packet is dropped. In summary, the intrusion detection system based on the proposed ensemble model analyzes the packet for intrusion data patterns.

Datasets

An intrusion detection system is evaluated based on their performance analysis of comprehensive labeled data of normal and abnormal behavior to identify types of network attacks [68]. UNSWNB15 and CICIDS2017 pp. datasets previously used for network intrusion detection [69]. Older data sets such as KDD CUP 99 [70] and NSL KDD [71] have been widely used for evaluating performance of the intrusion detection system. Studies [72–74], evaluating intrusion detection system using these data sets do not reflect realistic output performance. The KDD CUP 99 data set contains an enormous amount of redundant data; therefore, it has been improved to obtain the NSL KDD dataset. As the NSL KDD dataset is not comprehensive, the comprehensive dataset of UNSWNB15 is prepared in the cyber range lab of the Australian center for cybersecurity. UNSW-NB15 represents nine types of attacks obtained through the IXIA PerfectStorm tool. The dataset contains 49 features developed using Bro-IDS and Argus tools. These datasets contain a limited number of network attacks and outdated information about packets. The training dataset contains 175,341 data instances, whereas the testing dataset contains 82,332 normal and attack data instances. Table 2 presents data distribution from the UNSW-NB15 dataset based on the types of data. The issues of the UNSW-NB15 dataset are addressed in the CICIDS2017 dataset.

Table 2. UNSW-NB15 dataset data distribution.

Type of Data	Description	No of Records
Normal	Normal network data	2,218,761
Analysis	Contains attacks such as spam, port scan and HTML web pages penetration	2677
Dos	Denial of service attack	16,353
Fuzzers	Causing a program suspension using randomly generated data feeding	24,246
Backdoors	Technique for bypassing system security	2329
Exploits	Exploiting the known security problems	44,525
Shellcode	Piece of code for exploiting the vulnerability of software	1511
Generic	Technique which targets all block ciphers	215,481
Worms	Worms replicate themselves to spread to other computers	174
Reconnaissance	Strikes which can simulate attacks to gather information	13,987

The CICIDS2017 dataset, on the other hand, is one of the updated intrusion detection systems' datasets. It contains benign, and seven general network flows' attacks. The dataset has been used to evaluate the performance of machine learning models on a set of networking traffic data features for anomaly detection. In the paper, the analysis of the CICIDS2017 shows that random forest outperforms other algorithms [75]. Table 3 presents a summary of the features of CICIDS2017.

Table 3. CICIDS2017 dataset data distribution.

Type of Data	Description	No of Records
Normal	Normal network data	2,358,036
DoS hulk	Denial of service attack data generated using hulk tool	231,073
DDos	Attack using multiple machines	41,835
Port scan	Attack using a port scan mechanism	158,930
Dos GoldenEye	Attack data using a GoldenEye tool	10,293
FTP patator	Brute force attack attempt for guessing FTP passwords	7938
SSH patator	Brute force attack attempt for guessing SSH passwords	5897
Botnet	Attacks using trojans and utilization of the victim system in the Botnet network	1966
DoS slow loris	Denial of service attacks using a Slow Loris tool	5796
DoS slow HTTP Test	Excess of HTTP get request to prevent HTTP usage	5499
Web attack	Brute force attack for finding personal identification numbers from web pages	1507
XSS Web attack	Malicious scripts injection in trusted websites	625
HeartBleed	Injection of malicious information into openSSL memory using openSSL exploitation	11
SQL injection Web attack	Attack using the famous attack method called SQL injection	21
Infiltration	Unauthorized access to system using infiltration methods and tools	36

As part of data pre-processing, dataset features are transformed and normalized. Categorical features such as attack types are converted into numerical values using one-hot-encoding. Data features with large values are normalized. The intrusion detection datasets contain many redundant data records, which causes biasness toward the many data records [76]. Furthermore, missing data records are also reasons for changing the characteristics of the data [77]. Hence, datasets are improved to solve data issues such as unbalancing between normal and abnormal data records and the missing values [78]. Therefore, data features with irrelevant and redundant data are eliminated, and necessary features are selected. For feature selection, the univariate feature selection technique with analysis of a variance (ANOVA) F-test is used [79,80]. An ANOVA f-test is used to analyze to determine x individual performance of all the features and strength of the relationship between the data feature and label features of the dataset. Sklearn library-based Select Percentile is used to select features based on the highest scores percentile. After selecting the subset of the features, the recursive feature elimination technique is applied to eliminate the irrelevant features. After the dataset is prepared, a model based on the ensemble of learning and prediction mechanism is trained and evaluated using the training and validation dataset. The pre-trained model of the ensemble of learning and prediction mechanism is used as an intrusion detection system model in the intrusion detection environment for improving the security of the environment against any malicious traffic. In recent network applications, such as the IoT paradigm-based cloud system, the system transmits data between the cloud and end-user through the IoT gateway. This gateway is an important location for deploying the intrusion detection system based on the proposed ensemble model. Like traditional IDSs, the proposed anomaly detection system uses a set of principal components: a sniffing unit that sniffs packets from the traffic and analyzes them and an intrusion database engine that stores rules and attack profiles. Response units drop the packet with anomalous attack patterns or maintain the traffic if regular. The analyzer unit is a critical component where the data processing techniques are applied on the packet, and detection models are deployed. An enormous amount of network data can be extracted from packets, so pre-processing the data is critical and requires more processing power and resources.

As explained earlier in the datasets, basic features are extracted for the data flow. Furthermore, the extracted data features are standardized for the pre-trained intrusion detection system model to reduce the data dimensionality problems. Therefore, the analyzer unit helps prevent biases and confusion for the intrusion detection system. The pre-processed data are sent to the intrusion detection model for making the final decision on each data batch. The intrusion detection model detects known and unknown attacks by learning from real-time data and updating the intrusion database engine. If the input data profile does not match the normal network profile stored in the intrusion database, it is classified as an attack. The response unit alerts the system's administrator with abnormal behavior based on the predicted attack profile. The response unit also updates the new signature of the attack types into the intrusion database engine.

4. Results and Discussion

In this section, the implementation results are discussed. Firstly, we will discuss the implementation environment and the datasets. The implementation environment and software used in the experimentation are illustrated in Table 4. Four sets of Raspberry Pi devices are utilized for the experimentation. However, two Raspberry Pi devices are enough to evaluate intrusion detection systems in the local area network. The other two Raspberry Pi devices are used for checking the performance of the intrusion detection system in an online environment such as a wide area network. One Raspberry Pi device is used for IoT server configurations that communicate with a PC-based server. The second Raspberry device is used to test the proposed ensemble model in a local area network-based intrusion environment. The third Raspberry Pi is connected to a network switch, where three virtual area networks are configured to make a wide area network. The third

Raspberry Pi based intrusion detection testing is implemented to evaluate the performance of the proposed ensemble model for larger networks like workplaces. Finally, the fourth Raspberry Pi is used to create a snort software-based intrusion prevention system for real-time traffic and packet analysis. Normal and abnormal traffic is generated on the PC server and passed to the networking server.

Table 4. Implementation environment.

Component	Description
Operating System	Windows 10
Hardware	4 sets of Raspberry PI devices
Memory	24 GB
Server	PC Server and networking server
Programming language	Python and Flask Framework
Simulation Softwares	Cooja Simulator, Wireshark,Keras
AutoML Framework	AutoKeras

As part of the descriptive analysis, the comprehensive dataset is analyzed using pre-processing techniques. As a result, the prepared datasets are easily interpretable for the proposed intrusion detection model. In addition, descriptive analytics are presented on the intrusion detection datasets to identify attacks and normal traffic data patterns. Finally, the datasets used for the case studies are split into training, test, and validation sets for training, testing, and validation of the proposed intrusion detection system model. To implement NAS, AutoKeras is used. AutoKeras is an autoML framework for deep learning using the Keras library's APIs. NAS is the process of neural network architecture searching to solve a problem using neural networks efficiently. For instance, an AutoKeras based anomaly detection engine is used to discover optimal classification models on classification datasets. Table 5 presents some parameters that resulted from the NAS in AutoKeras.

Table 5. AutoML constant hyperparameters resulted from NAS.

Batch Size	Learning Rate	Epochs
260	0.001	250

Table 6 presents Kalman filter configurations and prediction accuracy with various sets of R parameter values. For instance, for the UNSW-NB15 dataset, the Kalman filter with an R-value of 15 achieves high accuracy, whereas, for the CICIDS2017 dataset, the Kalman filter with an R-value of 10 achieves high accuracy.

Table 6. Kalman filter configurations and accuracy.

R	2	5	10	15	20
Accuracy of UNSW-NB15 dataset	95.08	95.78	97.12	98.801	96.95
Accuracy of CICIDS2017 dataset	96.28	96.78	97.02	95.801	95.25

The training, test, and validation split ratio used is 70, 20, and 10 percent, respectively, for both datasets. The ten-fold cross-validation mechanism is used with stratified splits, keeping 20 percent for testing and 80 percent for cross-validation. The averages of the ten folds are used to pick the best model. Statistics of normal and attack data instances in the training, test, and validation sets of the UNSW-NB15 dataset are given in Table 7. In the training set, there are 1,014,221 normal training data records and 157,748 attack data records. In the testing set, there are 289,777 normal testing data records and 45,071 attack data records. In the validation set, there are 144,889 normal validation data records and 22,535 attack data records.

Table 7. Statistics of normal vs. attack data records in the UNSW-NB15 dataset.

Label	Training Dataset	Testing Dataset	Validation Dataset
Normal data	1,014,221	289,777	144,889
Attack data	157,748	45,071	22,535

Statistics of normal and attack data instances in the training, test, and validation sets of the CICIDS2017 dataset are given in Table 8. There are 318,014 normal training data records 7800 attack data records. There are 90,861 normal testing data records and 2229 test set-based attack data records. There are 45,431 normal validation data records and 1114 validation set-based attack data records.

Table 8. Statistics of normal vs. attack data records count in the CICIDS2017 dataset.

Label	Training Dataset	Testing Dataset	Validation Dataset
Normal data	318,014	90,861	45,431
Attack data	7800	2229	1114

The attack data of the UNSW-NB15 dataset comprises nine types: analysis, fuzzers, shellcode, reconnaissance, generic, backdoor, doS, and exploits. Statistics of these attack data in training, test, and validation datasets are given in Table 9.

Table 9. Statistics of attack data count in training, testing, and validation of UNSW-NB15.

Label	Training Dataset	Testing Dataset	Validation Dataset
Analysis	1249	357	178
Fuzzers	12,273	3507	1753
Shellcode	898	257	128
Reconnaissance	6946	1985	992
Generic	102,930	29,409	14,704
Backdoor	1139	325	163
DoS	9905	2830	1415
Exploits	22,172	6335	3167
Worms	85	24	12

CICIDS2017 dataset's data imbalance issue is addressed by introducing new labeling attack types. The newly labeled attack data of the CICIDS2017 dataset is comprised of six types: bot, brute force, infiltration, DoS/DDoS, web attack, and port scan. Statistics of these attack data in training, testing, and validation sets are given in Table 10.

Table 10. Statistics of attack data count in training, testing, and validation of the CICIDS2017 dataset.

Label	Training Dataset	Testing Dataset	Validation Dataset
Bot	257	73	37
Brute Force	1902	543	278
Infiltration	4	1	1
Port scan	22,317	6376	3188
Web attack	298	85	43
DoS/Ddos	390,072	111,449	55,724

Before performing correlation analysis, feature engineering techniques are applied to the dataset, including analysis of feature types such as finding continuous, categorical, date, and features with text. The dataset is transformed by encoding categorical features using one-hot-encoding (one-of-K) and converting text features to numeric. The mean values of the feature data are used for handling missing values. The performance of machine

learning models is good if the data are good. Hence, appropriate pre-processing and cleaning of the data are a significant step. Those features that contribute most to a machine learning model's performance should be used for training the intrusion detection system model to improve the accuracy of intrusion detection.

Feature selection is a technique used to eliminate features from the dataset that do not solve or contribute to the problem's solution. For the selection of the features, feature importance and correlation analysis methods are used. Data correlation analysis is used to understand the relationship between the dataset's features. The association of multiple features with other dataset features is analyzed during correlation analysis. Correlation analysis is used in the literature as an essential technique for feature selection and reduction. Figure 4 visualizes a heatmap of correlation analysis of the dataset.

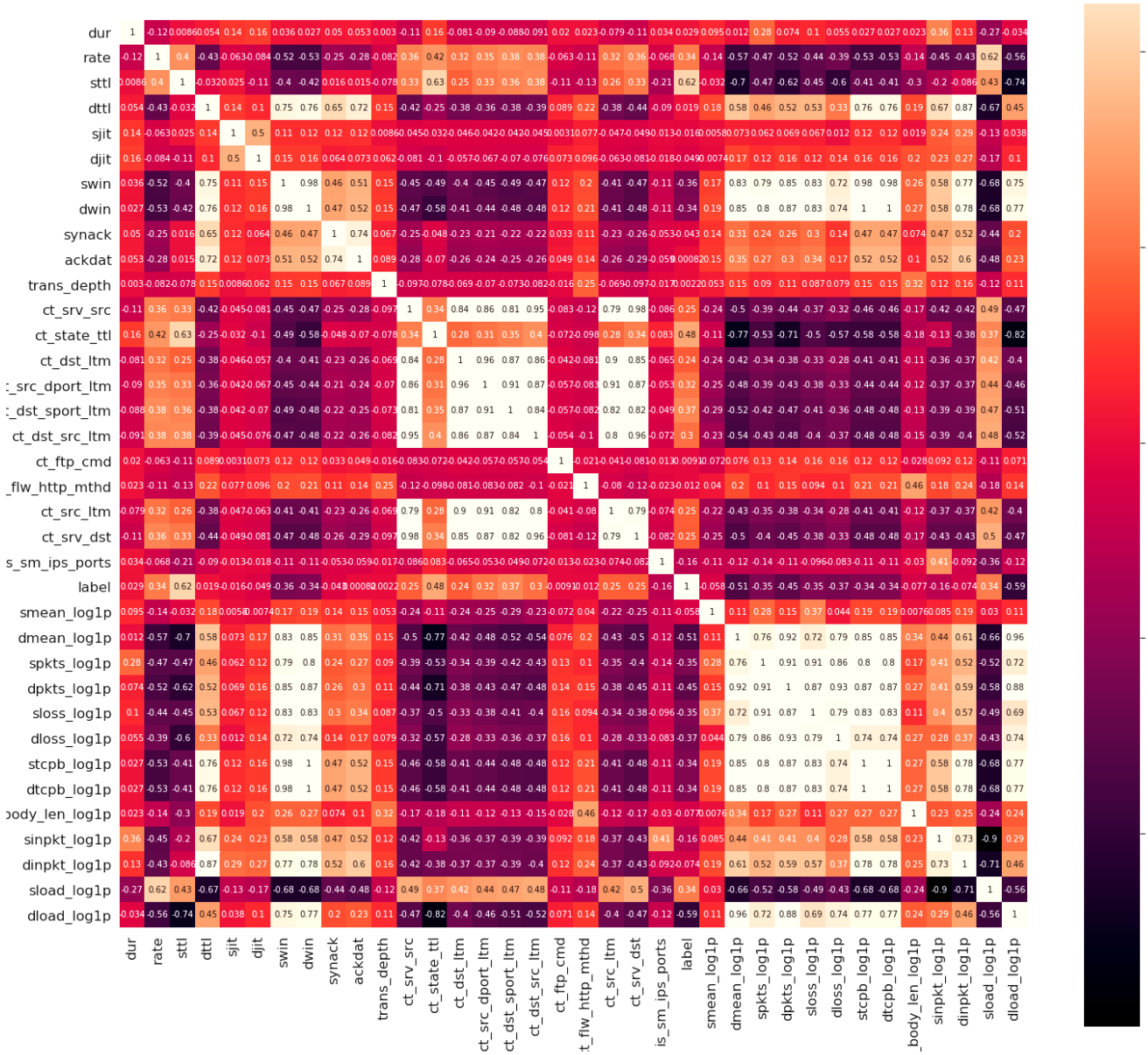


Figure 4. Correlation analysis of the dataset.

The highly correlated features are listed below:

- Source to destination packet count (Spkts), source to destination bytes (sbytes), and source packets re-transmitted or dropped (sloss).

- Destination to source packet count (Dpkts), destination to source bytes (dbytes), and destination packets re-transmitted or dropped (dloss).
- Source inter-packet arrival time (sinpkt), IP address, and port number based on source (is_sm_ips_ports).
- Source TCP window advertisement (swin), and destination TCP window advertisement (dwin).
- The sum of TCP acknowledgment data features (Tcprtt) and time between SYN and SYN_ACK packet of the TCP (synack).
- Connection count with same service and source address (Ct_srv_src), connection count with same service and destination address (ct_srv_dst), connection count of same source and destination address (ct_dst_src_ltm), connection count of the same source address and destination port number (ct_src_dport_ltm), and connection count of the same destination address and source port number (ct_dst_sport_ltm).
- Access to ftp session (Is_ftp_login), and (Flows count in ftp session consists of commands(ct_ftp_cmd).

Features such as sloss, dloss, sbytes, and dbytes are dropped because they have a high correlation with spkts and dpkts features. Stcpb and dtcpb are dropped because the range of the TCP base sequence is high (0 to 5×10^9). However, connections containing anomalies are close to 0. Tcprtt is the round trip time of connection setup and the sum of synack and ackdat features. Tcprtt is dropped because it does not add any extra information to the model. Synack, sload, dload, sjit, djit, and ackdat are dropped because they have not played a role in improving the accuracy of the model.

As explained earlier, the data correlation is used to analyze the relationship between data features. For instance, correlation analysis in machine learning is used to understand the relationship between input data features and output data features. If the machine learning model is trained with a set of data features with little correlation, the results are more likely to be inaccurate [81–83]. For example, data features' Ct_srv_src' and 'ct_srv_dst' are highly correlated but have poor correlation with the rest of the data features and output feature. Consequently, both 'Ct_srv_src' and 'ct_srv_dst' are dropped. 'Dur' feature is the total duration recorded, 'Dur' feature is dropped due to no correlation with the label data feature. The 'rate' feature is dropped because it has a value range of up to 1M and anomalous connections are mostly around zero. 'Sinpkt' and 'dinpkt' are highly correlated, so we dropped 'dinpkt' as 'sinpkt' has a better correlation with the rest of the data features.

Figure 5 presents the normal and abnormal mean of packet flow size. The mean packet flow size is an essential network measure. Likewise, the flow byte size is a fundamental metric for network measurement. X-axis presents the packet arrival time from the source to destination and vice versa. Y-axis presents comparison of the normal and abnormal mean packet size.

The mean of packet flow size is an essential measure as low-rate distributed denial-of-service is a challenge to network security. In this attack, many attack packets flow similar to regular packet flow is sent to throttle legitimate flows. Figure 6 presents a normal and anomalous packet count. The mean packet flow size is an essential network measure. 'SPKTS' stands for source to destination packet count, whereas 'dpkts' stands for the destination to source packet count. 'Spkts' and 'dpkts' contain integer values of packet count. The primary axis presents source to destination packet count comparison in terms of normal packet count. The secondary axis presents destination to source packet count comparison in terms of anomalous packet count.

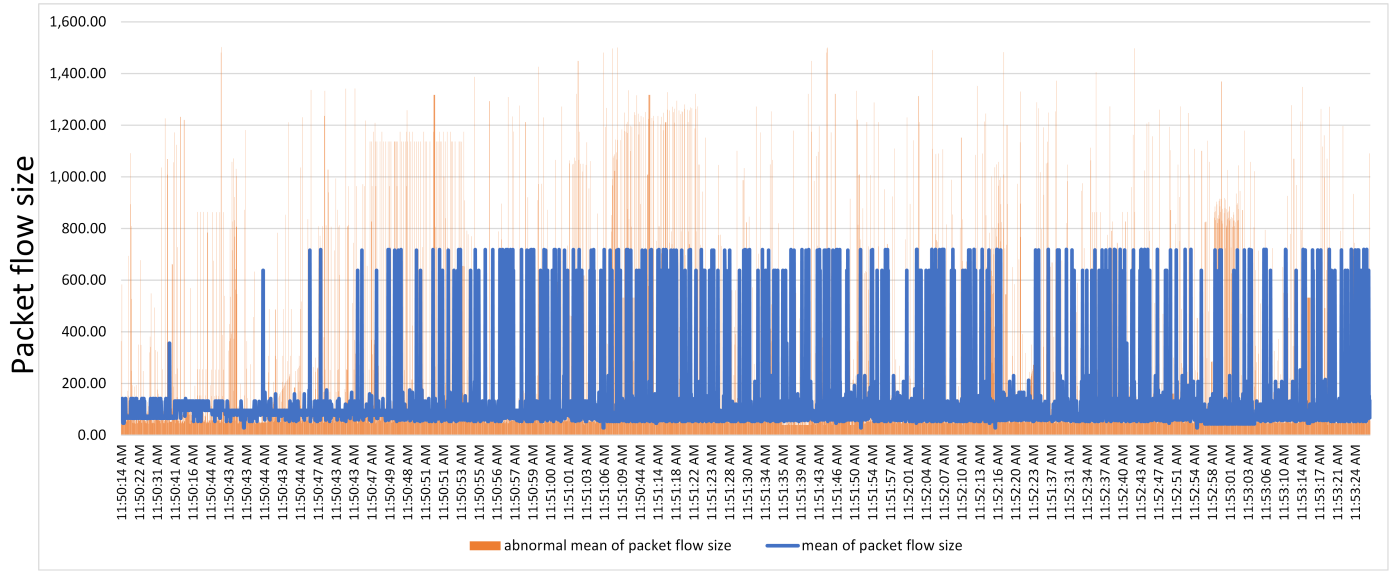


Figure 5. Mean of packet flow size by source and destination.

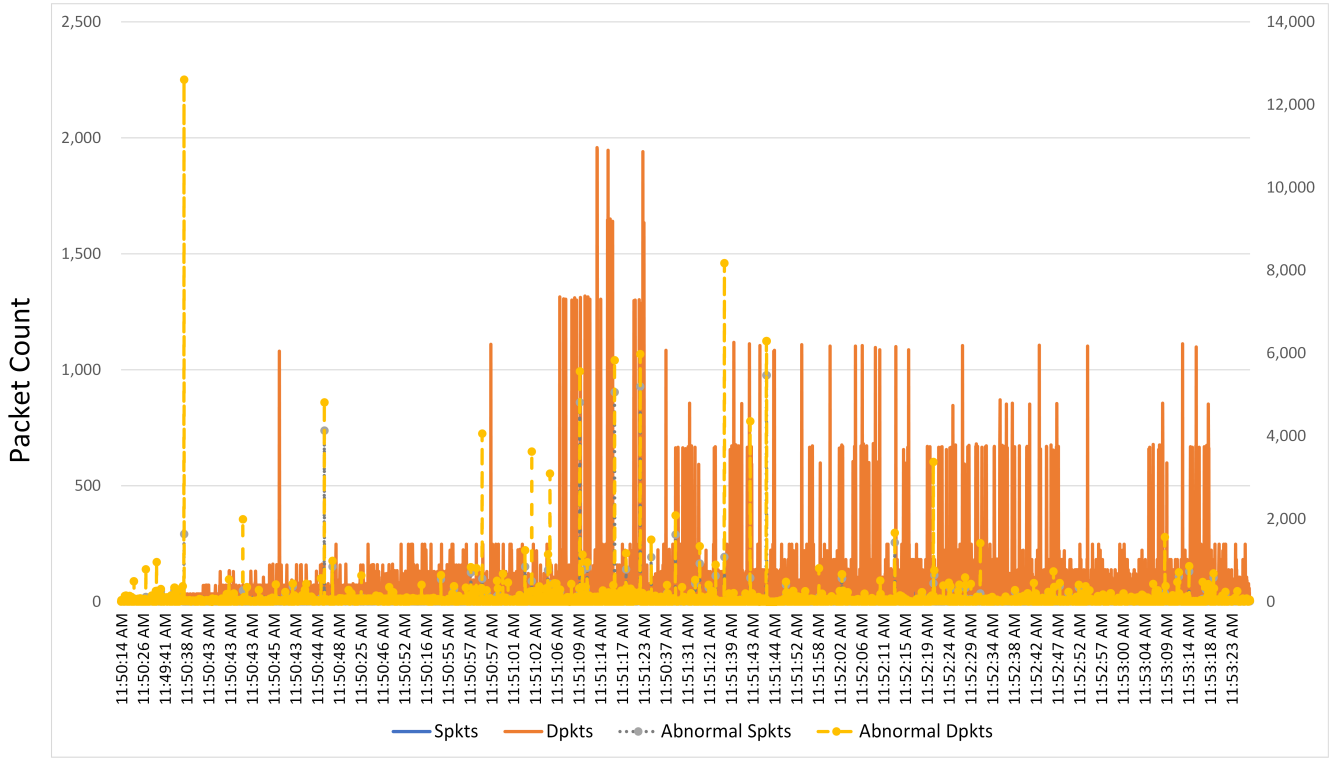
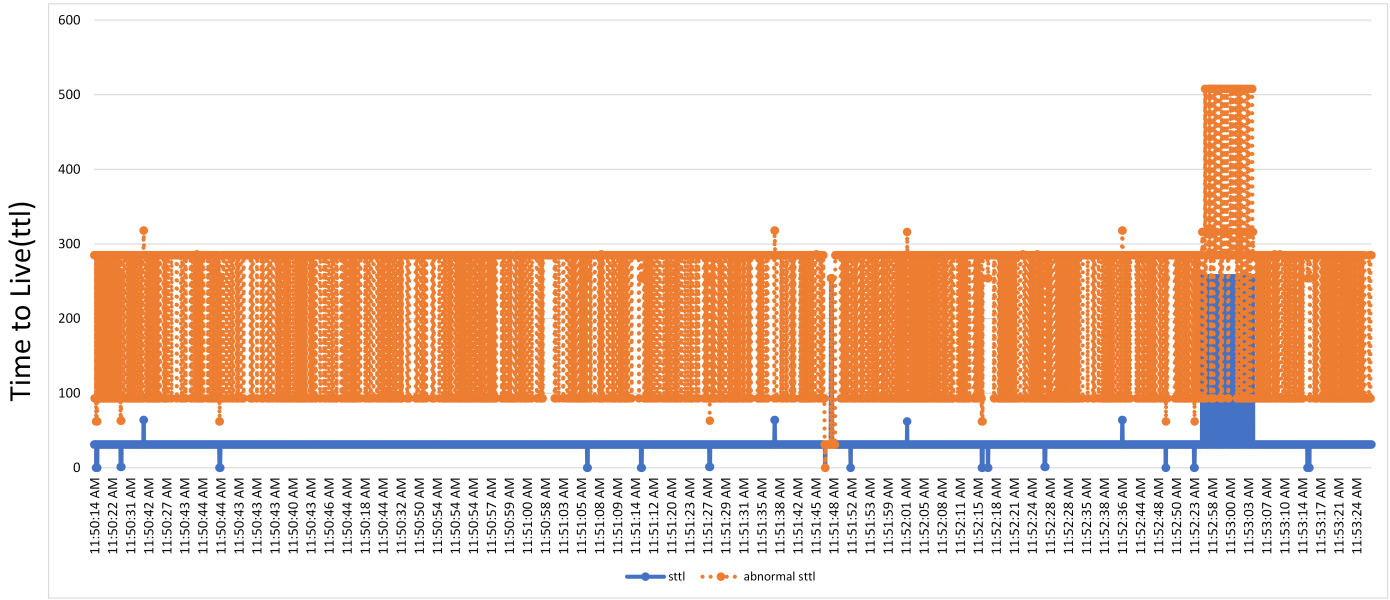
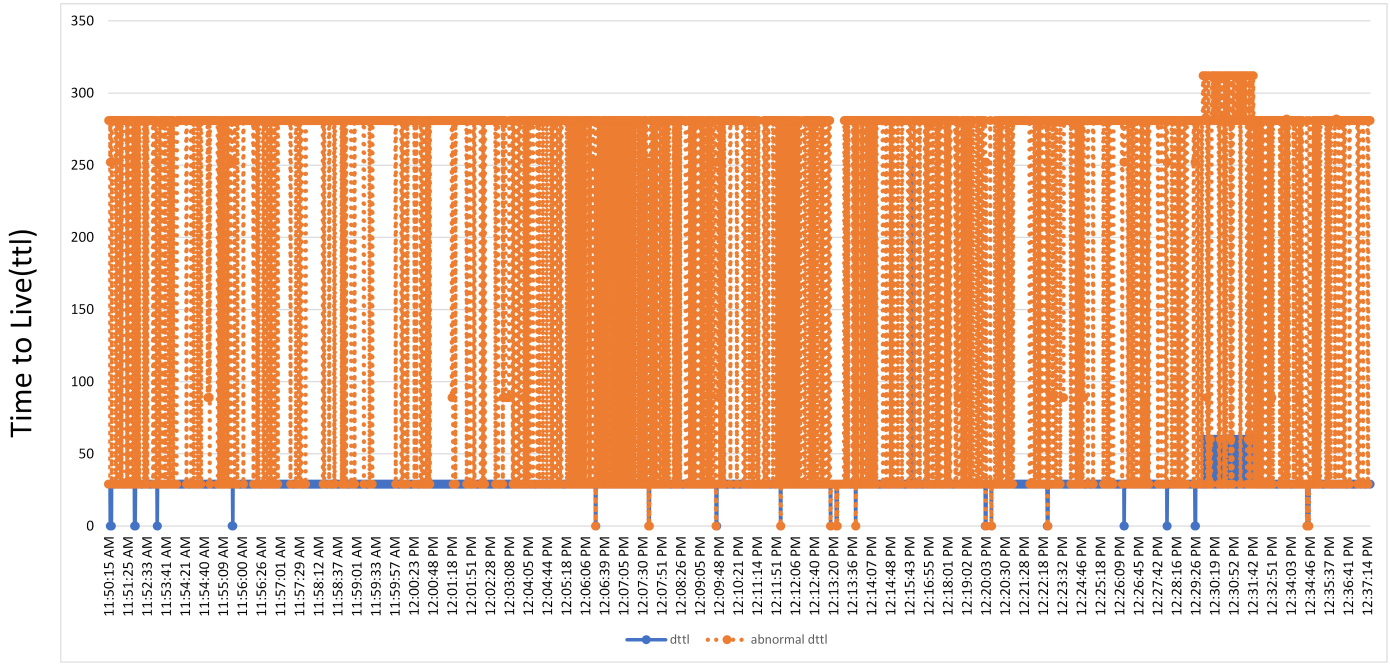


Figure 6. Packet count based analysis.

Figure 7 presents normal and anomalous time to live (TTL). TTL, also called hop limit, is a procedure for limiting the lifetime of the data packet in a computer network. TTL is usually implemented as a timestamp or counter mechanism embedded in the data packets. Figure 7a presents a source to destination time live comparison with anomalous packet time to live. Figure 7b presents a destination to source time to live comparison with anomalous time to live.



(**a**) Source to destination time to live



(**b**) Destination to source time to live

Figure 7. Normal and anomalous time to live.

Figure 8 presents normal and anomalous inter-packet arrival time (inpkt). Inpkt is the time taken by a packet to arrive on the host node over a period. It is also known as a delay. 'Sintpkt' is the source interpacket arrival time in milliseconds, whereas 'dintpkt' is the destination interpacket arrival time in milliseconds. 'Sintpkt' and 'dintpkt' features contain float values of inter-packet arrival. The primary axis presents source and destination normal inter-packet arrival. The secondary axis presents abnormal source and destination inter-packet arrival.

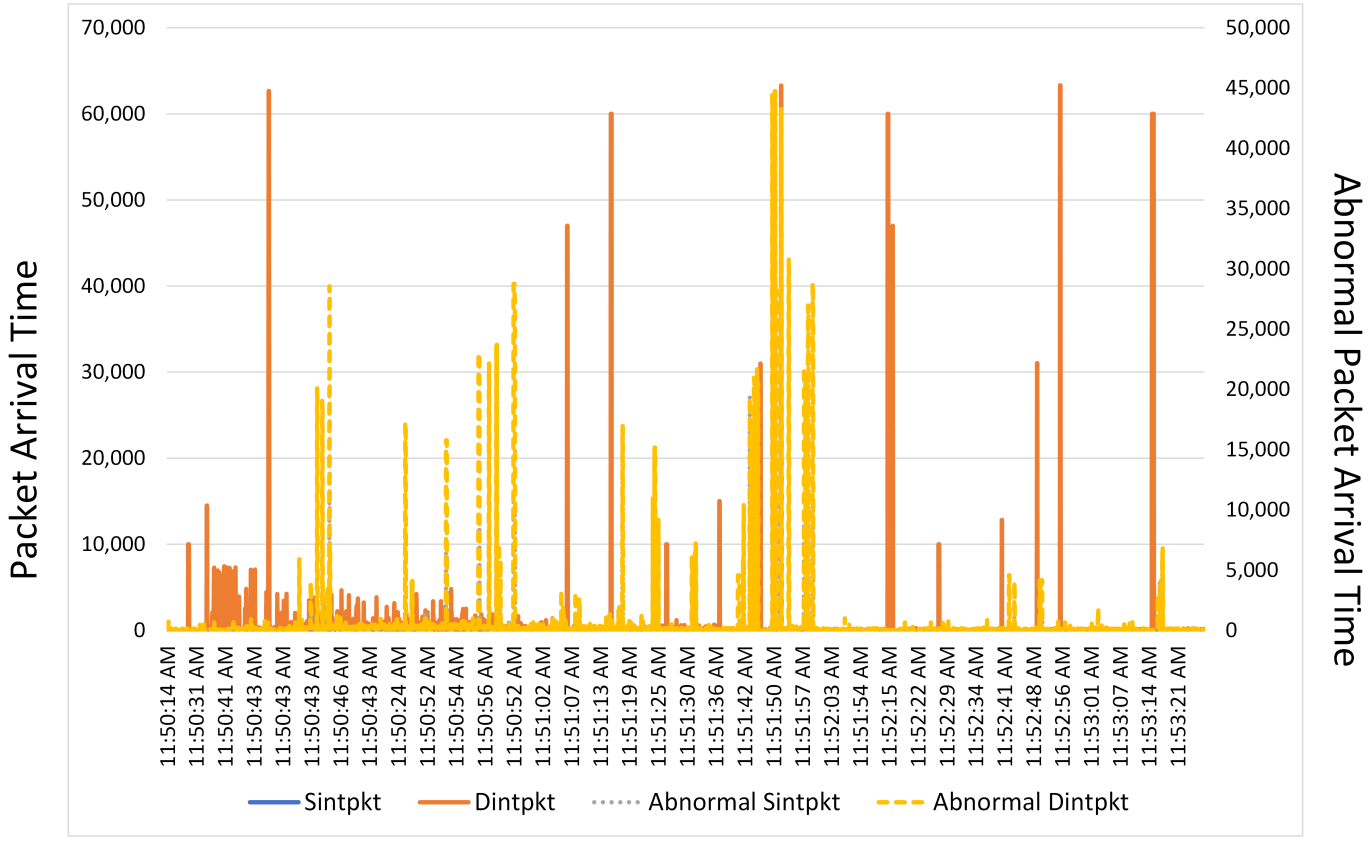


Figure 8. Inter-packet arrival time based analysis.

5. Performance Analysis and Discussion

This section presents the comparative performance analysis of the proposed ensemble model-based intrusion detection system and state-of-the-art intrusion detection models. Classification metrics such as accuracy, detection rate, false alarm rate, and F1 score are used for the evaluation of the models. First, we discuss the classification metrics used for evaluation and then present the comparison of the performance analysis. The comparison of the performance analysis is made in two stages. The first stage of the comparison is based on the state-of-the-art machine learning models implemented during this study. The second stage is the comparison of the performance analysis with state-of-the-art intrusion detection models from the literature. Now, we discuss the classification metrics along with their mathematical representation.

Accuracy is the percentage of correctly classified anomalies among sets of network data samples. Detection rate is the ratio of positively identified intrusion samples detected correctly [84]. An anomalies detection rate is calculated using Equation (1):

$$Detection\ Rate = \frac{Number\ of\ True\ Positive}{Number\ of\ True\ Positive + Number\ of\ False\ Positive} \quad (1)$$

False alarm rate is the ratio of negative identified intrusion samples, which are identified as positive ones [85]. The false alarm rate of anomalies is calculated using Equation (2):

$$False\ Alarm\ Rate = \frac{False\ Positive}{False\ Positive + True\ Negative} \quad (2)$$

F1 score is used to calculate the average of precision and recall. The confusion metric is the mathematical matrix used for all these performance measures. Equation (3) presents the formula for the calculation of the accuracy of classification models [86]:

$$Accuracy = \frac{Number\ of\ True\ Positive + Number\ of\ True\ Negative}{Number\ of\ (True\ Positive + True\ Negative + False\ Positive + False\ Negative)} \quad (3)$$

F1 score is defined as the weighted average or harmonic mean of precision and recall metrics values [87]. The F1 score value is between zero and one. The F1 score presents the precision of a classification model. High precision and lower recall lead to high accuracy but are inefficient for large data instances. Therefore, classification models with more excellent F1 scores are better and vice versa. Equation (4) presents the F1 score:

$$F1\ score = 2 * \frac{1}{\frac{1}{Precision\ of\ Intrusiondetectionsystem} + \frac{1}{recall\ of\ Intrusiondetectionsystem}} \quad (4)$$

The precision of a detection system based on supervised machine learning is given in Equation (5). Thus, the intrusion detection model's precision is the number of correct anomaly prediction results divided by the total number of anomaly prediction results:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (5)$$

A recall is the number of correct predicted anomalies divided by the number of total samples of an intrusion detection identified anomalies. The recall is given in Equation (6):

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (6)$$

Performance analysis of the proposed intrusion detection system model is compared with state-of-the-art machine learning-based intrusion detection systems from the literature and models implemented during the experimentation. Most of the classification models consider accuracy as the main measuring metric. Despite this, we also compared the proposed intrusion detection model performance with literature on the same datasets and implemented state-of-the-art models using detection rate and F1 score. Table 11 presents a performance comparison of the proposed intrusion detection system model with DNN, and AutoML for the UNSW-NB15 dataset. Table 12 presents a performance comparison of the proposed intrusion detection system model on the CICIDS2017 dataset. The proposed intrusion detection system model has identified anomalies with an accuracy of 98.801 percent and F1 score of 98.76 percent on the UNSW-NB15 dataset. The intrusion detection system model based on DNN with one layer performs better than the five variants of DNN models implemented in this study.

Table 11. Comparison of the proposed model with state-of-the-art models on UNSW-NB15.

Intrusion Detection Model	Accuracy	Detection Rate	F1 Score
DNN with 1 layer	78.4%	94.42	82.0
DNN with 2 layers	75.1%	97.09	78.05
DNN with 3 layers	76.3%	96.03	79.5
DNN with 4 layers	76.5%	94.6	80.1
DNN with 5 layers	78.90%	95.1	79.6
AutoML	96.40%	95.0	95.09
Proposed intrusion detection model	98.801%	97.92	98.76

The DNN model-based intrusion detection with one layer performs better than the other versions of DNN in terms of classification accuracy and F1 score. The DNN with two layers performs better in terms of detection rate as compared to other variants of the

DNN. The proposed intrusion detection model classified anomalies with 97.02 percent accuracy and the F1 score of 96 percent on the CICIDS2017 dataset. In summary, an intrusion detection system based on one layer performs better than the five variants of the implemented DNN.

Table 12. Comparison of the proposed model with state-of-the-art models on the CICIDS2017 dataset.

Intrusion Detection Model	Accuracy	Detection Rate	F1 Score
DNN with 1 layer	96.3%	90.8	93.9
DNN with 2 layers	95.1%	89.07	91.09
DNN with 3 layers	94.4%	85.04	91.2
DNN with 4 layers	93.6%	83.6	90.1
DNN with 5 layers	93.1%	82.7	89.4
AutoML	96.0%	90.0	94.09
Proposed intrusion detection model	97.02%	92.80	96.0

Significance and Comparison

This section presents the significance and comparison of the proposed ensemble model. The comparison of the proposed ensemble mechanism-based intrusion detection system with the existing state-of-the-art models shows the significance of the proposed intrusion detection model. The performance results of the proposed intrusion detection model with existing models are compared in terms of accuracy, detection rate, false alarm rate, and F1 score. The performance results presented in Table 13 show that the proposed ensemble model for the intrusion detection model significantly improves the intrusion detection accuracy, detection rate, and F1 score. The accuracy of the proposed ensemble model-based intrusion detection system is 98.801, the detection rate is 97.92 percent, and the F1 score is 98.76. SGM-CNN, a CNN model with synthetic minority over-sampling technique(SMOTE) and Gaussian mixture, is used to develop intrusion detection which performs better than the rest of the intrusion detection models. The accuracy, detection rate, and F1 score of the SGM-CNN are 96.54, 96.54, and 97.26, respectively. The SCM3 + RF model is an ensemble approach for intrusion detection based on the subset combination method (SCM). SCM3 is used to produce the final subset by selecting data features from two subsets SCM1 and SCM2. The intrusion detection is based on the random forest (RF) model. The ensemble model of SCM3 and RF achieved an accuracy of 95.87, the detection rate of 97.80, and a false alarm rate of 7.70. Table 13 presents significant improvement in the accuracy, detection rate, and F1 score of the intrusion detection model compared to existing intrusion detection models.

Table 13. Comparison of the proposed intrusion detection system model with the state of the art.

Intrusion Detection Model	Accuracy	Detection Rate	False Alarm Rate	F1 Score
CSCADE-ANN [88]	86.40	86.74	13.10	–
SGM-CNN [89]	96.54	96.54	–	97.26
SCM3 + RF [90]	95.87	97.80	7.70	–
RCNF [91]	95.98	4.02		–
ICVAE-DNN [92]	89.08	95.68	19.01	90.61
Intrusion detection system for IICS [93]	92.4	93	8.2	–
Bi-directional LSTM [94]	85	85	–	86
Proposed intrusion detection system	98.801	97.92	–	98.76

In short, the main contribution of this study is to improve the accuracy of the intrusion detection model. The improvement in detection accuracy is evident from the comparative analysis of the proposed intrusion detection system with state-of-the-art models developed

during the case study and with existing intrusion detection models from the literature such as SMOTE and Gaussian mixture.

6. Conclusions

An expeditious rise in the development of network applications leads to an immense amount of network data generated from a wide range of services for large user groups. Safeguarding network applications and things connected to the internet has always been a point of interest for researchers. Many studies propose solutions for intrusion detection systems and intrusion prevention systems. Nevertheless, there is a dire need to tune the contemporary data mining and statistical methods to address the challenges of the growing internet applications, such as bandwidth handling, network intrusion detection, and scalability. This paper presents an intrusion detection system based on the ensemble of prediction and learning mechanisms to improve anomaly detection accuracy in a network intrusion environment. Case studies of intrusion detection are implemented using publicly available benchmark intrusion detection datasets UNSW-NB15 and CICIDS2017. The performance of the proposed model is compared with some contemporary models, including DNN, autoML, and other algorithms from the literature on these benchmark datasets. The performance evaluation is compared in terms of accuracy, precision, recall, and F1 score. The proposed model accuracy for the UNSW-NB15 dataset is 98.801 percent, and the CICIDS2017 dataset is 97.02 percent. The performance comparison analysis shows significant improvements in the intrusion accuracy, detection rate, and F1 score. As part of future work, the proposed intrusion detection model will be leveraged for IoT-cloud applications for detecting anomalies in the sensing data.

Author Contributions: I. conceived the idea for this paper, designed the experiments, and wrote the paper. F.J. assisted in the experimental design, review and editing. D.K. supervised and proof-read the study of an ensemble of a prediction and learning mechanism for improving accuracy of anomaly detection in network intrusion environments. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Energy Cloud R&D Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (2019M3F2A1073387), and this research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2018R1D1A1A09082919)), Any correspondence related to this paper should be addressed to DoHyeun Kim.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ghaffar, Z.; Alshahrani, A.; Fayaz, M.; Alghamdi, A.M.; Gwak, J. A Topical Review on Machine Learning, Software Defined Networking, Internet of Things Applications: Research Limitations and Challenges. *Electronics* **2021**, *10*, 880. [CrossRef]
2. Ahmad, S.; Kim, D. Design and Implementation of Thermal Comfort System based on Tasks Allocation Mechanism in Smart Homes. *Sustainability* **2019**, *11*, 5849. [CrossRef]
3. Ahmad, S.; Kim, D.H. Quantum GIS Based Descriptive and Predictive Data Analysis for Effective Planning of Waste Management. *IEEE Access* **2020**, *8*, 46193–46205. [CrossRef]
4. Iqbal, N.; Ahmad, S.; Kim, D.H. Health Monitoring System for Elderly Patients Using Intelligent Task Mapping Mechanism in Closed Loop Healthcare Environment. *Symmetry* **2021**, *13*, 357. [CrossRef]
5. Imran; Ahmad, S.; Kim, D.H. A Task Orchestration Approach for Efficient Mountain Fire Detection Based on Microservice and Predictive Analysis in IoT Environment. *J. Intell. Fuzzy Syst.* **2021**, *40*, 5681–5696. [CrossRef]
6. Iqbal, N.; Ahmad, S.; Kim, D.H. Towards Mountain Fire Safety Using Fire Spread Predictive Analytics and Mountain Fire Containment in IoT Environment. *Sustainability* **2021**, *13*, 2461. [CrossRef]
7. Iqba, N.; Ahmad , S.; Ahmad, R.; Kim, D.-H. A Scheduling Mechanism Based on Optimization Using IoT-Tasks Orchestration for Efficient Patient Health Monitoring. *Sensors* **2021**, *21*, 5430. [CrossRef] [PubMed]
8. Camastra, F. Data dimensionality estimation methods: A survey. *Pattern Recognit.* **2003**, *36*, 2945–2954. [CrossRef]
9. Di Mauro, M.; Galatro, G.; Fortino, G.; Liotta, A. Supervised feature selection techniques in network intrusion detection: A critical review. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104216. [CrossRef]
10. Liao, H.J.; Lin, C.H.R.; Lin, Y.C.; Tung, K.Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [CrossRef]

11. Bhati, B.S.; Rai, C. Analysis of Support Vector Machine-based Intrusion Detection Techniques. *Arab. J. Sci. Eng.* **2020**, *45*, 2371–2383. [CrossRef]
12. Kanth, A.R. Gaussian Naıve Bayes Based Intrusion Detection System. In Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019), Hyderabad, India, 13–15 December 2020; Springer Nature: Berlin/Heidelberg, Germany, 2020; Volume 1182, p. 150.
13. Markiewicz, R.P.; Sgandurra, D. Clust-IT: Clustering-based intrusion detection in IoT environments. In Proceedings of the 15th International Conference on Availability, Reliability and Security, Virtual, 25–28 August 2020; pp. 1–9.
14. Sarker, I.H.; Abushark, Y.B.; Alsolami, F.; Khan, A.I. IntruDTree: A Machine Learning Based Cyber Security Intrusion Detection Model. *Symmetry* **2020**, *12*, 754. [CrossRef]
15. Zarándy, Á.; Rekeczky, C.; Szolgay, P.; Chua, L.O. Overview of CNN research: 25 years history and the current trends. In Proceedings of the 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, Portugal, 24–27 May 2015; pp. 401–404.
16. Irie, K.; Tüske, Z.; Alkhouli, T.; Schlüter, R.; Ney, H. LSTM, GRU, Highway and a Bit of Attention: An Empirical Overview for Language Modeling in Speech Recognition. In Proceedings of the Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, 8–12 September 2016, pp. 3519–3523.
17. Jiang, Y.; Yang, F.; Zhu, H.; Zhou, D.; Zeng, X. Nonlinear CNN: Improving CNNs with quadratic convolutions. *Neural Comput. Appl.* **2019**, *32*, 8507–8516. [CrossRef]
18. Gonzalez, J.; Yu, W. Nonlinear system modeling using LSTM neural networks. *IFAC-PapersOnLine* **2018**, *51*, 485–489. [CrossRef]
19. Tan, Y.; Hu, C.; Zhang, K.; Zheng, K.; Davis, E.A.; Park, J.S. LSTM-Based Anomaly Detection for Non-Linear Dynamical System. *IEEE Access* **2020**, *8*, 103301–103308. [CrossRef]
20. Marchi, E.; Vesperini, F.; Weninger, F.; Eyben, F.; Squartini, S.; Schuller, B. Nonlinear prediction with LSTM recurrent neural networks for acoustic novelty detection. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–7.
21. Zoumpourlis, G.; Doumanoglou, A.; Vretos, N.; Daras, P. Nonlinear convolution filters for CNN-based learning. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4761–4769.
22. Corinto, F.; Biey, M.; Gilli, M. Nonlinear coupled CNN models for multiscale image analysis. *Int. J. Circ. Theory Appl.* **2006**, *34*, 77–88. [CrossRef]
23. Shamsolmoali, P.; Jain, D.K.; Zareapoor, M.; Yang, J.; Alam, M.A. High-dimensional multimedia classification using deep CNN and extended residual units. *Multimed. Tools Appl.* **2019**, *78*, 23867–23882. [CrossRef]
24. Cheikhrouhou, O.; Mahmud, R.; Zouari, R.; Ibrahim, M.; Zaguia, A.; Gia, T. N. One-Dimensional CNN Approach for ECG Arrhythmia Analysis in Fog-Cloud Environments. *IEEE Access* **2021**, *9*, 103513–103523. [CrossRef]
25. Praanna, K.; Sruthi, S.; Kalyani, K.; Tejaswi, A.S. A CNN-LSTM Model for Intrusion Detection System from High Dimensional Data. *J. Inf. Comput. Sci.* **2020**, *10*, 1362–1370.
26. Malaiya, R.K.; Kwon, D.; Kim, J.; Suh, S.C.; Kim, H.; Kim, I. An empirical evaluation of deep learning for network anomaly detection. In Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 5–8 March 2018; pp. 893–898.
27. Yao, Q.; Wang, M.; Chen, Y.; Dai, W.; Yi-Qi, H.; Yu-Feng, L.; Wei-Wei, T.; Qiang, Y.; Yang, Y. Taking human out of learning applications: A survey on automated machine learning. *arXiv* **2018**, arXiv:1810.13306.
28. Gijsbers, P.; LeDell, E.; Thomas, J.; Poirier, S.; Bischl, B.; Vanschoren, J. An open source AutoML benchmark. *arXiv* **2019**, arXiv:1907.00909.
29. Haught, J.; Hopkinson, K.; Stuckey, N.; Dop, M.; Stirling, A. A Kalman filter-based prediction system for better network context-awareness. In Proceedings of the 2010 Winter Simulation Conference, Baltimore, MD, USA, 5–8 December 2010; pp. 2927–2934.
30. Wahid, F.; Fayaz, M.; Aljarbouh, A.; Mir, M.; Aamir, M.; Imran. Energy Consumption Optimization and User Comfort Maximization in Smart Buildings Using a Hybrid of the Firefly and Genetic Algorithms. *Energies* **2020**, *13*, 4363. [CrossRef]
31. Rizwan, A.; Iqbal, N.; Ahmad, R.; Kim, D.-H. WR-SVM Model Based on the Margin Radius Approach for Solving the Minimum Enclosing Ball Problem in Support Vector Machine Classification. *Appl. Sci.* **2021**, *11*, 4657. [CrossRef]
32. Khan, A.-N.; Iqbal, N.; Rizwan, A.; Ahmad, R.; Kim, D.-H. An Ensemble Energy Consumption Forecasting Model Based on Spatial-Temporal Clustering Analysis in Residential Buildings. *Energies* **2021**, *14*, 3020. [CrossRef]
33. Agrawal, S.; Agrawal, J.Survey on anomaly detection using data mining techniques. *Procedia Comput. Sci.* **2015**, *60*, 708–713. [CrossRef]
34. Pathan, A.S.K. *The State of the Art in Intrusion Prevention and Detection*; CRC Press: Boca Raton, FL, USA, 2014.
35. Narayana, V.L.; Gopi, A.P.; Khadherbhi, S.R.; Pavani, V. Accurate identification and detection of outliers in networks using group random forest methodoly. *J. Crit. Rev.* **2020**, *7*, 381–384.
36. Demestichas, K.; Peppes, N.; Alexakis, T.; Adamopoulou, E. An Advanced Abnormal Behavior Detection Engine Embedding Autoencoders for the Investigation of Financial Transactions. *Information* **2021**, *12*, 34. [CrossRef]
37. Xie, M.; Han, S.; Tian, B.; Parvin, S. Anomaly detection in wireless sensor networks: A survey. *J. Netw. Comput. Appl.* **2011**, *34*, 1302–1325. [CrossRef]
38. Debar, H.; Dacier, M.; Wespi, A. A revised taxonomy for intrusion-detection systems. In *Annales Des Télécommunications*; Springer: Berlin/Heidelberg, Germany, 2000; Volume 55, pp. 361–378.

39. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowl.-Based Syst.* **2020**, *189*, 105124. [CrossRef]
40. Tucker, C.J.; Furnell, S.M.; Ghita, B.V.; Brooke, P.J. A new taxonomy for comparing intrusion detection systems. *Internet Res.* **2007**, *17*, 1. [CrossRef]
41. Estevez-Tapiador, J.M.; Garcia-Teodoro, P.; Diaz-Verdejo, J.E. Anomaly detection methods in wired networks: A survey and taxonomy. *Comput. Commun.* **2004**, *27*, 1569–1584. [CrossRef]
42. Boukerche, A.; Zheng, L.; Alfandi, O. Outlier detection: Methods, models, and classification. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–37. [CrossRef]
43. Gogoi, P.; Bhattacharyya, D.; Borah, B.; Kalita, J.K. A survey of outlier detection methods in network anomaly identification. *Comput. J.* **2011**, *54*, 570–588. [CrossRef]
44. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–58. [CrossRef]
45. Patcha, A.; Park, J.M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Comput. Netw.* **2007**, *51*, 3448–3470. [CrossRef]
46. Hodge, V.; Austin, J. A survey of outlier detection methodologies. *Artif. Intell. Rev.* **2004**, *22*, 85–126. [CrossRef]
47. Kiani, R.; Keshavarzi, A.; Bohlouli, M. Detection of thin boundaries between different types of anomalies in outlier detection using enhanced neural networks. *Appl. Artif. Intell.* **2020**, *34*, 345–377. [CrossRef]
48. Safaei, M.; Asadi, S.; Driss, M.; Boulila, W.; Alsaeedi, A.; Chizari, H.; Abdullah, R.; Safaei, M. A systematic literature review on outlier detection in wireless sensor networks. *Symmetry* **2020**, *12*, 328. [CrossRef]
49. Markou, M.; Singh, S. Novelty detection: A review—Part 2: Neural network based approaches. *Signal Process.* **2003**, *83*, 2499–2521. [CrossRef]
50. Ahmed, M.; Mahmood, A.N.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]
51. Treinen, J.J. System, Method and Program Product for Identifying Network-Attack Profiles and Blocking Network Intrusions. U.S. Patent 8,056,115, 8 November 2011.
52. Mhatre, A.J.; Kiggins, A.J.; Diggins, M.F. Attack Traffic Signature Generation Using Statistical Pattern Recognition. U.S. Patent 8,997,227, 31 March 2015.
53. Peng, Y. Research of network intrusion detection system based on snort and NTOP. In Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery, Chongqing, China, 29–31 May 2012; pp. 2764–2768.
54. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
55. Mehdi, S.A.; Khalid, J.; Khayam, S.A. Revisiting traffic anomaly detection using software defined networking. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Menlo Park, CA, USA, 20–21 September 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 161–180.
56. Braga, R.; Mota, E.; Passito, A. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In Proceedings of the IEEE Local Computer Network Conference, Denver, CO, USA, 10–14 October 2021; pp. 408–415.
57. Tang, T.A.; Mhamdi, L.; McLernon, D.; Zaidi, S.A.R.; Ghogho, M. Deep learning approach for network intrusion detection in software defined networking. In Proceedings of the 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, 26–29 October 2016; pp. 258–263.
58. Niyaz, Q.; Sun, W.; Javaid, A.Y. A deep learning based DDoS detection system in software-defined networking (SDN). *arXiv* **2016**, arXiv:1611.07400.
59. Jankowski, D.; Amanowicz, M. On efficiency of selected machine learning algorithms for intrusion detection in software defined networks. *Int. J. Electron. Telecommun.* **2016**, *62*, 247–252. [CrossRef]
60. Lee, Y.; Kang, W.; Son, H. An internet traffic analysis method with mapreduce. In Proceedings of the 2010 IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps), Osaka, Japan, 19–23 April 2010; pp. 357–361. [CrossRef]
61. Singh, K.; Guntuku, S.C.; Thakur, A.; Hota, C. Big data analytics framework for peer-to-peer botnet detection using random forests. *Inform. Sci.* **2014**, *278*, 488–497. [CrossRef]
62. Bhat, A.H.; Patra, S.; Jena, D. Machine learning approach for intrusion detection on cloud virtual machines. *Int. J. Appl. Innov. Eng. Manag.* **2013**, *2*, 56–66
63. Chen, Z.; Han, F.; Cao, J.; Jiang, X.; Chen, S. Cloud computing-based forensic analysis for collaborative network security management system. *Tsinghua Sci. Technol.* **2013**, *18*, 40–50. [CrossRef]
64. Chen, T.; Zhang, X.; Jin, S.; Kim, O. Efficient classification using parallel and scalable compressed model and its application on intrusion detection. *Expert Syst. Appl.* **2014**, *41*, 5972–5983 [CrossRef]
65. Marnerides, A.; Watson, M.R.; Shirazi, N.; Mauthe, A.; Hutchison, D. Malware analysis in cloud computing: Network and system characteristics. In Proceedings of the 2013 IEEE Globecom Workshops (GC Wkshps), Atlanta, GA, USA, 9–13 December 2013; pp. 482–487. [CrossRef]
66. Muthurajkumar, S.; Kulothungan, K.; Vijayalakshmi, M.; Jaisankar, N.; Kannan, A. A rough set based feature selection algorithm for effective intrusion detection in cloud model. In Proceedings of the International Conference on Advances in Communication, Network, and Computing, Beijing, China, 23–24 May 2013; pp. 8–13

67. Wang, H.; Ding, W.; Xia, Z. A cloud-pattern based network traffic analysis platform for passive measurement. In Proceedings of the 2012 International Conference on, Cloud and Service Computing (CSC), Shanghai, China, 22–24 November 2012; pp. 1–7. [CrossRef]
68. Gogoi, P.; Bhuyan, M.H.; Bhattacharyya, D.; Kalita, J.K. Packet and flow based network intrusion dataset. In Proceedings of the International Conference on Contemporary Computing, Noida, India, 6–8 August 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 322–334.
69. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6.
70. Cup, K. 2007. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 07 september 2021).
71. Su, T.; Sun, H.; Zhu, J.; Wang, S.; Li, Y. BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. *IEEE Access* **2020**, *8*, 29575–29585. [CrossRef]
72. McHugh, J. Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **2000**, *3*, 262–294. [CrossRef]
73. Mahoney, M.V.; Chan, P.K. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, 8–10 September 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 220–237.
74. Vasudevan, A.; Harshini, E.; Selvakumar, S. SSENet-2011: A network intrusion detection system dataset and its comparison with KDD CUP 99 dataset. In Proceedings of the 2011 Second Asian Himalayas International Conference on Internet (AH-ICI), Kathmundu, Nepal, 4–6 November 2011; pp. 1–5.
75. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. A detailed analysis of the cicids2017 data set. In Proceedings of the International Conference on Information Systems Security and Privacy, Funchal, Portugal, 22–24 January 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 172–188.
76. Sahu, S.K.; Sarangi, S.; Jena, S.K. A detail analysis on intrusion detection datasets. In Proceedings of the 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, India, 21–22 February 2014; pp. 1348–1353.
77. Groenwold, R.H.; Dekkers, O.M. Missing data: The impact of what is not there. *Eur. J. Endocrinol.* **2020**, *183*, E7–E9. [CrossRef] [PubMed]
78. Dal, P.A.; Caelen, O.; Bontempi, G. When is undersampling effective in unbalanced classification tasks? In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Porto, Portugal, 7–11 September 2015; Springer: Cham, Switzerlan, 2015; pp. 200–215.
79. Biney, G.; Okyere, G.A.; Alhassan, A. Adaptive scheme for ANOVA models. *J. Adv. Math. Comput. Sci.* **2020**, 12–23. [CrossRef]
80. Feir-Walsh, B.J.; Toothaker, L.E. An empirical comparison of the ANOVA F-test, normal scores test and Kruskal–Wallis test under violation of assumptions. *Educ. Psychol. Meas.* **1974**, *34*, 789–799. [CrossRef]
81. Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. In Proceedings of the 2014 Science and Information Conference, London, UK, 27–29 August 2014; pp. 372–378.
82. Ahmad, S.; Iqbal, N.; Jamil, F.; Kim, D. Optimal Policy-Making for Municipal Waste Management Based on Predictive Model Optimization. *IEEE Access* **2020**, *8*, 218458–218469. [CrossRef]
83. Imran, I.; Zaman, U.; Waqar, M.; Zaman, A. Using Machine Learning Algorithms for Housing Price Prediction: The Case of Islamabad Housing Data. *Soft Comput. Mach. Intell.* **2021**, *1*, 11–23.
84. Muda, Z.; Yassin, W.; Sulaiman, M.N.; Udzir, N.I. Intrusion detection based on k-means clustering and OneR classification. In Proceedings of the 2011 7th International Conference on Information Assurance and Security (IAS), Melacca, Malaysia, 5–8 December 2011; pp. 192–197.
85. Om, H.; Kundu, A. A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. In Proceedings of the 2012 1st International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, 15–17 March 2012; pp. 131–136. [CrossRef]
86. Novaković, J.D.; Veljović, A.; Ilić, S.S.; Papić, Ž.; Milica, T. Evaluation of classification models in machine learning. *Theory Appl. Math. Comput. Sci.* **2017**, *7*, 39–46.
87. Goutte, C.; Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In Proceedings of the European Conference on Information Retrieval, Santiago de Compostela, Spain, 21–23 March 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 345–359.
88. Baig, M.M.; Awais, M.M.; El-Alfy, E.S.M. A multiclass cascade of artificial neural network for network intrusion detection. *J. Intell. Fuzzy Syst.* **2017**, *32*, 2875–2883. [CrossRef]
89. Zhang, H.; Huang, L.; Wu, C.Q.; Li, Z. An Effective Convolutional Neural Network Based on SMOTE and Gaussian Mixture Model for Intrusion Detection in Imbalanced Dataset. *Comput. Netw.* **2020**, *177*, 107315. [CrossRef]
90. Binbusayyis, A.; Vaiyapuri, T. Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach. *IEEE Access* **2019**, *7*, 106495–106513. [CrossRef]
91. Moustafa, N.; Slay, J. RCNF: Real-time collaborative network forensic scheme for evidence analysis. *arXiv* **2017**, arXiv:1711.02824.
92. Yang, Y.; Zheng, K.; Wu, C.; Yang, Y. Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. *Sensors* **2019**, *19*, 2528. [CrossRef]

93. Muna, A.H.; Moustafa, N.; Sitnikova, E. Identification of malicious activities in industrial internet of things based on deep learning models. *J. Inf. Secur. Appl.* **2018**, *41*. [CrossRef]
94. Yang, S. Research on network behavior anomaly analysis based on bidirectional LSTM. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019; pp. 798–802.