Received: November 24, 2023. Revised: January 3, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Analysis of Weight-Based Voting Classifier for Intrusion Detection System

Miftahul Hasanah[1] **Rizqy Ahsana Putri**[1] **Muhammad Aidiel Rachman Putra**[1] **Tohari Ahmad**[1]*

[1]*Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
* Corresponding author's Email: tohari@if.its.ac.id

Abstract: The evolution of technology and the internet has accelerated the pace of communication and information exchange. Despite technological advancements, the significant weakness lies in the persistent threat of cybercrime, manifesting in various forms like malware, phishing, and ransomware. To solve the cybercrime problems, this research aims to create an intrusion detection system model using a novel framework. In general, the proposed method consists of 3 stages: Data preprocessing, feature selection using ANOVA F-value combined with cross validation, and classification using weight-based voting classifier. Some machine learning methods used in the weight-based voting classifier are random forest, K-nearest neighbour, and logistic regression. The experiment results show that weight order and weight combination affect the detection performance. The proposed method produces an excellent precision value of 98.66%, higher than the single voting classifier.

Keywords: IDS, Information security, National security, Network security, UNSW-NB15, Weight-based voting classifier.

1. Introduction

The internet has become a basic need and an inseparable part of an individual. Advances in computers and the internet enable information to propagate at astonishing speeds, often within mere fractions of a second. Despite the advances in computers and internet technologies, the vulnerabilities still occur. One of the most concerning vulnerabilities of the internet lies in the security and confidentiality of online data exchanges [1]. Currently, numerous malicious actors and irresponsible parties are actively circulating, deliberately seizing, and misusing personal data for illegitimate purposes [2]. To prevent more individuals from falling into cybercrime attacks, it is crucial to develop an intrusion detection system (IDS) capable of detecting anomalies and protecting the network [3].

An intrusion detection system is a security technology designed to monitor network or system activity and detect unauthorized or suspicious behavior [4]. The main objective is to identify and address potential security threats either in real time or within a very short timeframe. This means that when suspicious activity is detected, the system can generate alerts and respond quickly to mitigate and prevent any further damage [5]. This becomes particularly crucial to prevent unauthorized access or hacking attempts that can occur in a matter of seconds [6]. The use of IDS is a critical component of information security, enabling organizations to uphold data integrity and confidentiality while protecting systems and networks against numerous security threats [7]. IDS plays an important role in identifying potential attacks and responding to them quickly, thus keeping IT infrastructure secure.

Along with the development of research related to IDS, machine learning is also developing quite rapidly. Research conducted by [8] developed a machine learning based on a voting classifier, which was implemented to detect anomalies in the network. Basically, a voting classifier represents a machine learning method that aggregates predictions from various individual models to produce a final prediction [9]. The idea behind ensemble methods like voting classifiers is to leverage the collective

DOI: 10.22266/ijies2024.0430.17

wisdom of different models to improve predictive accuracy and robustness.

There are two main types of voting classifiers: (1) hard voting classifier and (2) soft voting classifier [10]. In the hard voting classifier, multiple machine learning models are trained on the same dataset, and each model makes its own prediction. The final prediction of the ensemble is determined by a majority vote among the individual models [11]. On the other hand, soft voting considers the probabilities that each model assigns to each class label [12]. The final prediction is obtained by aggregating these probabilities, usually by summing or averaging them. The selected class has the highest cumulative probability [13]. Leveraging probability-based information provided by models, Soft Voting tends to have greater adaptability and potential for higher accuracy [14]. Additionally, it allows the models to be assigned varying weights, giving the Ensemble Method more flexibility over how decisions are made [14, 15]. While soft voting is generally more effective when working with a wide range of models, Hard Voting is easier to execute [16]. Research done in [8] concluded that the performance of machine learning with a voting classifier was better than that of a single classifier model.

The evolution of machine learning continues into a weight-based voting classifier method, which is an improvement of the original voting classifier. This approach gives specific weight to the contribution of each model, allowing for better adjustments to the contributions of models with different expertise or judgment [17]. By considering the confidence level of each model [18], weight-based voting classifiers can provide more accurate and reliable results.

This research proposes a new framework that includes an innovative implementation of feature selection. The feature selection stage is undoubtedly crucial because not all features in the dataset are useful [19]. As a new approach, this research combines ANOVA F-value and cross-validation (CV) in the feature selection process. The implementation of CV combined with the feature selection method is still rarely developed. In this proposed research, ANOVA F-value was chosen as the basis for the feature selection method because ANOVA F-value can determine the extent of the average differences between groups, and this ability is significantly important in the context of IDS. These advantages can be helpful for identifying which features have significant differences between normal and intrusion activities. To optimize the feature selection process, CV is also applied to evaluate model performance at each iteration and to ensure that the feature selection process does not only depend on one particular subset of data.

Based on the previous explanation, this research proposes a new approach to improve machine learning performance in the IDS domain. This approach implements a weight-based voting classifier combined with the ANOVA F-value CV feature selection method. Besides, this paper also describes the analysis related to weight order and weight combination in the development of models.

This paper has several sections. The relevant works are discussed in section 2, the suggested approach is explained in section 3, and the experimental findings are given in section 4. Section 5 ends with a conclusion.

2. Related works

Research on intrusion detection systems has been widely discussed in recent times. A frequently discussed topic is how the system can improve the detection performance. Several approaches implement the common machine learning methods, such as random forest, support vector machine, and decision tree [20, 21]. Besides, the other research focuses on reducing the dimensionality of features in the dataset using several feature selection techniques such as particle swarm optimization, Chi-Square, and grasshopper optimization [1, 22–24]. Another research implemented a hybrid algorithm to improve the accuracy and efficiency of IDS [25].

Shanthi and Maruthi [20] used the isolation forest model and support vector machine to detect an anomaly in network traffic. The research proposed an effective anomaly detection model to handle a complex and large number of datasets. The best results obtained reached an accuracy of 99.00% using the isolation forest model and 95.00% using the support vector machine. The other research implementing a support vector machine for IDS was conducted by Hamzah and Othman [21]. The research analyses each kernel function of SVM in nonlinear data classification for wireless sensor network intrusion detection system (WSN-IDS). The best performance was achieved by the RBF kernel with an accuracy of 91.00%. Although both studies achieved a high accuracy score, theirf sensitivity (recall), precision, and F-score are still relatively low. At the same time, model sensitivity is more crucial in IDS for detecting anomalies.

Zhang et al. [26] applied an optimized IPSO-SVM algorithm to detect network intrusion. The improved particle swarm optimization (IPSO) was implemented to find and select the optimal

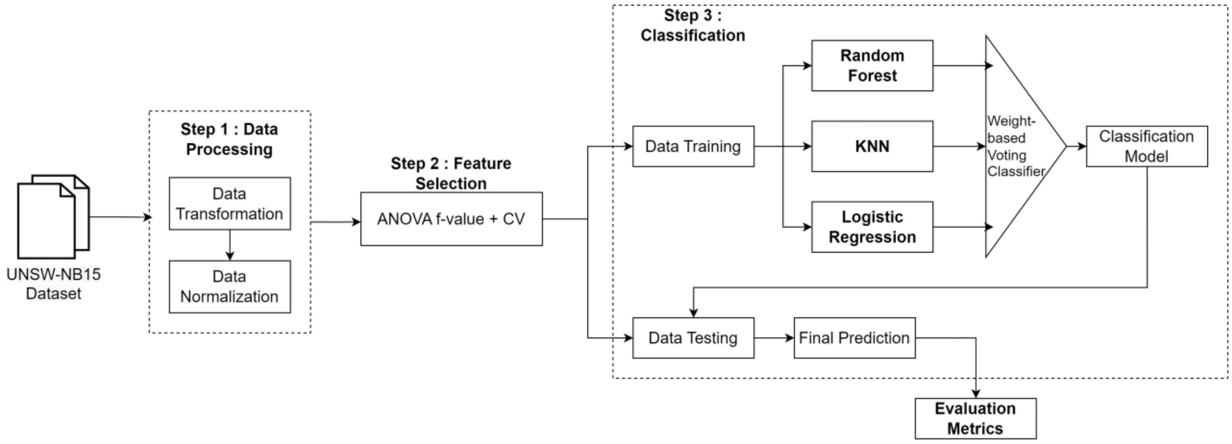

Figure 1. Research methodology

parameters of SVM as the basis classifier. The proposed method successfully recognized the intrusion signal with more than 82% accuracy. Unfortunately, this method still faces problems in the form of intrusion misrecognition. An improved algorithm combining the random forest, decision tree, and multilayer perceptron (MLP) algorithm was developed by Zhour et al. [25]. The research introduced a hybrid method to detect network intrusion in the NSL KDD, UNSW-NB15, and CIC-IDS-2017 datasets. Although the model achieves a high accuracy of 99.70% in the NSL-KDD dataset, it only reaches 77.99% in the UNSW-NB15 dataset. This indicates that the proposed method lacks sufficient capability to effectively detect attacks within the UNSW-NB15 dataset.

Ajdani and Ghaffary [22] implement the Particle Swarm Optimization algorithm as the feature selection technique. PSO was implemented to select the most compelling features of the dataset, and Random Forest was used as the classifier. The proposed method effectively identified abnormal activities with a 93.00% accuracy rate. Additionally, it demonstrates enhanced learning speed when processing a large volume of data. Similar research using the feature selection technique was conducted by Chen et al. [23]. The research applied CNN to extract the essential features of the dataset, and PSO was used to optimize the SVM parameters as the classifiers. The combination of these algorithms achieves an accuracy of 94.50%. Unfortunately, the proposed method only performs better in small sample data.

Several other approaches are used to improve the performance of traditional IDS. Gao et al. [27] proposed an innovative network intrusion detection framework based on extreme learning machine (ELM) and multi-voting technology (MVT). The proposed method successfully reduced the time consumption of the detection process with an 88.93% accuracy rate. The model's accuracy could be much higher due to the proposed method utilizing only 1%, 5%, and 10% of the dataset, which might cause a substantial loss of information. The voting classifier method was also implemented by Puri et al. [28]. The research utilized the SMOTE algorithm to normalize the dataset, which resulted in more extensive data. It also utilized the SHAP method to identify the significant features to understand their influence on the model's output. The proposed method using the voting classifier model reaches an accuracy of 93.91%. Unfortunately, implementing SHAP requires much computational time because it needs to run across all possible combinations of parameters.

A weight-based voting classifier was implemented in another research domain [29, 30]. Kumar et al. [29] applied a weight-based voting classifier for classifying breast cancer. The proposed model demonstrated the highest performance compared to the single, hard, and soft voting classifier, particularly in metrics of accuracy and sensitivity. Similar research using a weight-based voting classifier was also conducted by Aziz and Dimililer [30] to analyze Twitter sentiment. This study found that the suggested weighted voting classifier approach boosts sentiment classification performance beyond the single classifiers and a primary majority voting classifier. Thus, the advantage of a weight-based voting classifier is needed to improve the IDS performance.

Based on the previous studies, this research implemented a weight-based voting classifier for the Intrusion Detection System. The proposed method also utilized ANOVA F-value combined with cross-validation as the feature selection technique. The combination of these methods could improve the model’s performance. This research also analyzes every classifier's weight combination and weight order to see whether this impacts performance results.

3. The Proposed method

The process in this research consists of several stages. The first stage is data preprocessing, which consists of transformation and normalization. The second stage performs feature selection with ANOVA F-value-based cross-validation. After that, the data was classified by implementing a Weight-based voting classifier and evaluated with accuracy, precision, recall, and F1-score metrics. Fig. 1 illustrates the research flowchart, which will be further explained in the following sections for a comprehensive understanding.

3.1 Data preprocessing

The preprocessing stage in this research consists of 2 parts, namely (1) data transformation, and (2) data normalization. Basically, data transformation is a series of techniques used to change raw data into a form or format that is more suitable or useful for machine learning models. Data transformation was implemented in this research because several categorical features in the dataset need to be converted to numerical data [31]. This is an essential step because categorical data cannot be processed with some machine learning methods. In implementing data transformation, this research utilized one of the Python libraries, namely label encoder.

The next step after successfully carrying out data transformation is normalizing the data. The method used for data normalization in this study is Z-score. Z-score, better known as standard scaler, is a normalization method that is very popular because of its good ability to normalize data. Eq. (1) is the mathematical formula of the Z-score method [32].

$$Z = \frac{(x-\mu)}{\sigma} \tag{1}$$

where:
Z: Z-score value
x: observed value
μ: mean
σ: standard deviation

The Z-score data normalization is a standard method used in statistical analysis and machine learning to standardize data. It aims to transform data with a mean (average) of zero and a standard deviation of one [33]. By using Z-score normalization, the distribution and range of the dataset are centralized, which can assist machine learning algorithms in scaling and producing better results. Here are the steps for normalizing data with Z-Score:

1) Calculate the mean. For each feature (column) in the data, calculate the mean value of the entire dataset. This mean is used as the center of the data distribution.
2) Calculate the standard deviation. After computing the mean, calculate the standard deviation of the entire dataset for each feature. The standard deviation measures the data's spread and quantifies how far individual data points are from the mean.
3) Normalize Z-score. For each data point in each feature, compute the Z-score using Eq. (1).

3.2 Feature selection

As mentioned in the previous section, this research also implements feature selection to eliminate unimportant features in the dataset. There are many feature selection methods, but this research uses ANOVA F-value as the feature selection method. ANOVA F-value is a concept used to measure the significant influence or difference between a particular feature and the target variable or class in a classification or regression problem [34]. It is used to understand whether a feature is important or relevant in making predictions. Here is how ANOVA F-value works:

1) Data is divided into various groups or classes based on the target variable or category to predict. For example, in a classification problem, each class has its own group of data.
2) F-value Calculation: For each feature, ANOVA F-value calculations are performed to measure how much the means of feature values differ among the various groups or classes. This involves comparing the variation between groups to the variation within groups.
3) The results of the F-value calculations are used to rank features based on their significance. Features with high F-values indicate that it significantly impacts distinguishing among groups or classes.
4) Feature Selection: Based on the F-value ranking, the best-performing features are selected to include in the machine learning model. The number of features included in the model was decided depending on the objectives and computational constraints.

The feature selection method using ANOVA F-value involves a hyperparameter known as the threshold, designed to control the number of features

Table 1. Performance of single classifier

Alg.	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
RF	**95.51**	97.38	**96.09**	**96.73**
KNN	94.00	96.32	94.96	95.64
LR	92.20	**98.61**	90.77	94.52

Table 2. Classifier order combination based on the highest to lowest weight

Id	Model's Order		
	1	2	3
O1	RF	KNN	LR
O2	RF	LR	KNN
O3	KNN	LR	RF
O4	KNN	RF	LR
O5	LR	KNN	RF
O6	LR	RFF	KNN

retained in the model. This threshold determines the extent to which features are deemed relevant for model construction. In the pursuit of an optimal threshold value, a cross-validation approach is employed, wherein experimentation is conducted with varying numbers of features in each model testing iteration. For instance, in a dataset comprising 40 features, exploration is undertaken with thresholds ranging from 1 to 40, and the model's performance is assessed in each trial. The threshold value that yields the highest accuracy or meets predetermined evaluation criteria is selected as the optimal setting. This approach facilitates adaptive feature selection by systematically exploring diverse combinations of feature quantities to cater to specific model requirements [1].

3.3 Weight-based voting classifier

In detecting anomalies, this research implemented a method called a weight-based voting classifier, which is a classification-based approach that combines several supervised learning methods. What is significantly different between a regular voting classifier and a weight-based voting classifier is the weight given to each model that participates in the classification process. The basic concept of the weight-based voting classifier is to give different weights to each model in the ensemble method so that some models have a more significant influence than others in making the final decision [35]. The tuning of weights for each model is elaborated more in the next section.

Some classification methods used to participate in the weight-based voting classifier are logistic regression, random forest, and K-nearest neighbour. These methods were chosen because they were proven to produce the best accuracy compared to other classification methods. In this research, the entire implementation uses the Python programming language and leverages several available libraries. For all Python methods and libraries, default hyperparameters are employed for each function. Consequently, the research does not conduct fine-tuning or optimal hyperparameters analysis.

3.4 Weight tuning for each model

According to the previous explanation, the process of tuning the weights for each model are explained in this section. The total weights experimented in this study is composed as Eq. (2). The research provides several combinations of each classifier weight to see if it affects the performance results. The first combination is denoted by Eq. (3), such as $w_1 = 0.5,\ w_2 = 0.3$, and $w_3 = 0.2$. It means the first classifier with the highest weight will not win the vote if the second and third classifiers have opposite predictions. The second combination is formulated as Eq. (4), such as $w_1 = 0.6,\ w_2 = 0.3,\ w_3 = 0.1$, and $w_1 = 0.7,\ w_2 = 0.2, w_3 = 0.1$. This combination allows a classifier with the highest weight to win the voting classifier, whether the second or third classifier is the opposite. The last combination uses almost equal distribution for each weight, such as $w_1 = 0.4,\ w_2 = 0.3, w_3 = 0.3$, and $w_1 = 0.4,\ \ w_2 = 0.4, w_3 = 0.2$. This combination option does not provide any tendency for one classifier to win the vote.

$$\sum_{i=1}^{3} w_i = 1 \tag{2}$$

$$w_1 = w_2 + w_3 \tag{3}$$

$$w_1 > w_2 + w_3 \tag{4}$$

where:
w_i: the weight of i

4. Results and discussion

This section discusses the experimental results of the distinct weight analysis of the voting classifier. The program is built using Python, and the experiments are carried out on Google collaboratory.

4.1 Dataset

Several relevant datasets of intrusion detection systems, such as KDD98, KDDCUP99, and NSLKDD, do not represent the current network threat environment and modern attacks because the datasets were created decades ago. Therefore, the dataset used in this study is the UNSW-NB15 dataset, which tends to be a more recent dataset for IDS [36]. The dataset consists of 49 features with two labels, attack and normal. It also classifies the attacks into nine categories of attacks: Fuzzers, Analysis, backdoors, DoS exploits, generic, reconnaissance, shellcode, and worms. This study uses the training set of the dataset, which consists of 175,341 records, and only classifies whether network traffic contains normal or attack activities.

4.2 Evaluation metrics

Several indicators to measure the performance of the proposed method are accuracy, precision, recall, and F1-score. Some of these performance indicators can be calculated by Eq. (5), Eq. (6), Eq. (7), and Eq. (8), where TP represents the amount of data that correctly predicts the positive class, TN is the amount of data that correctly predicts the negative class, FP is the amount of data that is predicted to be positive but the actual data is negative, and FN is the amount of data that is predicted to be negative but the actual data is positive.

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{5}$$

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

The precision, recall, and F1 scores are often used to evaluate the IDS model. Precision is important to measure how well the model can avoid false alarms. The higher the precision, the fewer false alarms the system generates. In addition, recall or sensitivity is also an essential metric in IDS, as the objective is to detect as many intrusions as possible. Recall measures the system performance to detect all actual intrusions. A high recall value indicates that the system can detect more intrusions. Meanwhile, the F1-score is a metric measuring the balance between precision and recall. F1-score helps obtain an optimal balance between avoiding false alarms and detecting intrusions.

4.3 Preprocessing and feature selection results

This section explains the result of preprocessing and feature selection. The first preprocessing step is data transformation using label encoder. The UNSW-NB15 dataset has three categorical features: proto, service, and state, each of which contains 133, 13, and 9 categories, respectively. The next step is the normalization of the data using the Z-score method. After going through this stage, the dataset range and distribution have become similar, improving the performance of each single classifier model. The last step is feature selection using ANOVA-CV. From the total of 43 features in the UNSW-NB15, 19 features were selected: state, rate, sttl, sload, dload, swin, stcpb, dtcpb, dwin, dmean, ct_srv_src, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_src_ltm, ct_srv_dst, is_sm_ips_ports.

4.4 Analysis of the classifier weighting order on performance results

The experiment starts with testing the performance of each single classifier provided in Table 1. The results show that the random forest algorithm obtains the highest accuracy, recall, and F-1 score values. In contrast, logistic regression has the lowest value of accuracy, recall, and F-1 score and achieves the highest value in precision. This research aims to identify the influence of weighting orders and combinations of voting classifier performance.

This section analyses the weight order of each classifier based on the weight combination explained before, as shown in Table 2. The experiment uses all six possible sequences of the three existing algorithms. The first analysis identifies the order's influence on Eq. (3). The results provided in Table 3 show that the classifier order does not affect the performance results on all metric evaluations. It is because when the best single classifier has the highest weight, it will win the voting classifier. However, when the highest weight is given to the worst single classifier, the final prediction still depends on two other classifiers. So, when the first-order classifier has a different prediction from the other classifiers, the final prediction will be based on the majority voting.

The second analysis identifies the combination of weights based on Eq. (4). This combination means the final prediction only depends on the classifier with the highest weight and ignores the other two classifiers. Therefore, the classifier orders matter to this weight combination. Tables 4 and 5 show better

Table 3. Performance evaluation of weight combination = 0.5, 0.3, 0.2

Metric	Order Combination as Table 2					
	O1(%)	O2(%)	O3(%)	O4(%)	O5(%)	O6(%)
Accuracy	95.05	95.05	95.05	95.05	95.05	95.05
Precision	98.66	98.66	98.66	98.66	98.66	98.66
Recall	94.34	94.34	94.34	94.34	94.34	94.34
F1-Score	96.45	96.45	96.45	96.45	96.45	96.45

Table 4. Performance evaluation of weight combination = 0.6, 0.3, 0.1

Metric	Order Combination as Table 2					
	O1(%)	O2(%)	O3(%)	O4(%)	O5(%)	O6(%)
Accuracy	95.51	95.51	94.00	94.00	92.20	92.20
Precision	97.38	97.38	96.32	96.32	98.61	98.61
Recall	96.09	96.09	94.96	94.96	90.77	90.77
F1-Score	96.73	96.73	95.64	95.64	94.52	94.52

Table 5. Performance evaluation of weight combination = 0.7, 0.2, 0.1

Metric	Order Combination as Table 2					
	O1(%)	O2(%)	O3(%)	O4(%)	O5(%)	O6(%)
Accuracy	95.51	95.51	94.00	94.00	92.20	92.20
Precision	97.38	97.38	96.32	96.32	98.61	98.61
Recall	96.09	96.09	94.96	94.96	90.77	90.77
F1-Score	96.73	96.73	95.64	95.64	94.52	94.52

Table 6. Performance evaluation of weight combination = 0.4, 0.3, 0.3

Metric	Order Combination as Table 2					
	O1(%)	O2(%)	O3(%)	O4(%)	O5(%)	O6(%)
Accuracy	95.05	95.05	95.05	95.05	95.05	95.05
Precision	98.66	98.66	98.66	98.66	98.66	98.66
Recall	94.34	94.34	94.34	94.34	94.34	94.34
F1-Score	96.45	96.45	96.45	96.45	96.45	96.45

Table 7. Performance evaluation of weight combination = 0.4, 0.4, 0.2

Metric	Order Combination as Table 2					
	O1(%)	O2(%)	O3(%)	O4(%)	O5(%)	O6(%)
Accuracy	95.05	95.05	95.05	95.05	95.05	95.05
Precision	98.66	98.66	98.66	98.66	98.66	98.66
Recall	94.34	94.34	94.34	94.34	94.34	94.34
F1-Score	96.45	96.45	96.45	96.45	96.45	96.45

results if the highest weight is given to the classifier with the best performance. On the contrary, if the worst classifier is given the highest weight, the results will worsen. The last analysis evaluates the influence of weight order with almost equal distribution. The experimental results in Tables 6 and 7 show that the classifier order does not affect the performance result. It produces the exact value of each metric evaluation for all possible classifier orders.

4.5 Analysis of the weight combination on each metric performance

This section analyses the influence of weight combinations on the performance of each metric evaluation results. The first weight combination formulated in Eq. (3) performs a lower value than Random Forest, the best classifier algorithm for

Table 8. Performance comparison between the proposed method and other studies

Method	Feature Selection	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Nururrahmah and Ahmad [1]	CHI2CV	NB	68.11	71.21	69.15	71.09
		DT	92.73	92.13	90.11	91.12
		SVM	96.71	95.15	95.15	95.56
		KNN	80.08	82.89	79.51	79.55
Dickson and Thomas [37]	-	NB	73.00	65.70	71.90	68.60
	-	SVM	65.00	59.00	59.00	66.60
	-	J48	98.00	66.90	66.90	75.90
Kasongo and Sun [38]	-	FFDNN	87.48	N/A	N/A	N/A
	WFEU	FFDNN	85.48	N/A	N/A	N/A
Ajdani and Ghaffary [22]	PSO	RF	93.00	N/A	N/A	N/A
Zhang et al. [39]	MSCNN	LSTM	89.80	N/A	N/A	N/A
Zhour et al. [25]	-	Voting classifier	77.99	N/A	N/A	N/A
Gao et al. [27]	-	Voting classifier	89.28	N/A	N/A	N/A
Proposed Method	**ANOVA F-value CV**	**Weight-based voting classifier**	**95.51**	**98.66**	**96.09**	**96.73**

accuracy, recall, and F1-score. Based on the result in Table 3, the voting classifier with this weight combination does not improve the accuracy, recall, and F1-score performance of the highest single classifier performance. On the contrary, the voting classifier performance is better for the precision metric using this weight combination. The voting classifier achieves a precision score of 98.66%, and the highest precision performance of a single classifier, Logistic Regression, only reaches 98.61%.

The following analysis evaluates the weight combination composed in Eq. (4). The results in Tables 4 and 5 show that the performance of the voting classifier has the same value as the single classifier value with the highest weight. For example, when Random Forest is given the highest weight, the voting classifier performance in all evaluation metrics is the same as random forest performance shown in Table 1. Therefore, the highest accuracy, recall, and F1-score values of this weight combination are achieved when random forest is given the highest weight, and the precision value is the highest when logistic regression has the highest weight. These experimental results prove that this weight combination of the voting classifier does not improve the performance of a single classifier in all evaluation metrics.

The last weight combination results in Tables 6 and 7 show no difference for all possible orders in each evaluation metric. The accuracy, recall, and F1-score are lower than that of the random forest. On the other hand, this combination improves the precision value of logistic regression as the best single classifier in precision. Based on the experimental results on both the single and voting classifiers, the performance of the voting classifier might be improved when the performance of every single classifier did not show a significant difference. Hence, the accuracy, recall, and F1-score value of the voting classifier is not higher than the highest value of single classifier performance. At the same time, it increased in precision because the that of each single classifier is not much different.

4.6 Comparison with previous research

The proposed method is compared with several studies related to UNSW-NB15 identification. Evaluation metrics used to compare the performance results are accuracy, precision, recall, and F1-score, the results of which can be seen in Table 8. The proposed method has achieved the highest value in terms of precision, recall, and F1-score among all other studies. The high performance of the proposed method comes from optimal data preprocessing and effective feature selection techniques. Despite Dickson and Thomas [37] and Nururrahmah and Ahmad [1] obtaining better accuracy, their precision, recall, and F1-score are remarkably lower than the proposed method's. It indicates an unequal distribution of samples across predicted classes or class imbalance, resulting in high accuracy but low precision and recall for the minority class. Also, models generating many false positives or false negatives could yield high accuracy but lower precision and recall due to frequent misclassifications. On the other hand, the Accuracy score of the proposed method and Dickson and Thomas [37] and Nururrahmah and Ahmad [1] do not indicate a significant difference. The comparison results show that the proposed method demonstrates more stable performance toward all evaluation metrics.

5. Conclusion

This research proposed an approach to detect network intrusions using weight-based voting classifiers. This research consists of three main processes: (1) Data preprocessing, (2) Feature selection, and (3) Classification using a weight-based voting classifier. The experiments used the open-source dataset from UNSW-NB15.

After conducting experiments, the findings revealed the successful detection capability of the weight-based voting classifier model in discerning the presence or absence of anomalies within a network. The accuracy and precision obtained were also relatively high, reaching 95.51% and 98.66%, respectively. Diverging slightly from previous studies, this research effectively detects anomalies and examines the impact of weight order and weight combinations in the weight-based voting classifier.

It is depicted that the order of weights does not affect the model performance results when the weight distributions are not much different. Using this combination of weights, the accuracy, recall, and F1-score values' performance is lower than a single classifier's performance. However, it produces higher precision than the single classifier. Then, the order of the weights will have an effect if one of the weights is given a larger portion than the sum of the other two weights. This combination of weights causes the performance value for each metric to be the same as the value of the single classifier that occupies the most significant weight.

In this research, weight tuning is still done manually. Future research can develop new approach methods that can find weight combinations automatically. Automating the weight-tuning process could lead to more efficient and accurate model configurations, enhancing the overall performance of the weight-based voting classifier.

Conflicts of interest

The authors have no conflicts of interest to disclose.

Author contributions

Conceptualization, MH, RAP, MARP, and TA; methodology, MH, RAP, MARP, and TA; software; MH and RAP; validation, MH and RAP; formal analysis, MH, RAP, MARP, and TA; investigation, MH, RAP, MARP, and TA; resources, MH and RAP; data curation, MH and RAP; writing—original draft preparation, MH and RAP; writing—review and editing, TA and MARP; visualization, MH and RAP; supervision, TA and MARP; project administration, TA; funding acquisition, TA.

Acknowledgments

This research was supported by Pendidikan Magister menuju Doktor untuk Sarjana Unggul (PMDSU) grant from The Ministry of Education, Culture, Research and Technology, The Republic of Indonesia, and Institut Teknologi Sepuluh Nopember, Indonesia.

References

[1] A. T. Nururrahmah and T. Ahmad, "CHI2CV : Feature Selection using Chi-Square with Cross-Validation for Intrusion Detection System", In: *Proc. of 2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–6, 2023, doi: 10.1109/ISDFS58141.2023.10131731.

[2] R. Sardar and T. Anees, "Web of Things: Security Challenges and Mechanisms", *IEEE Access*, Vol. 9, pp. 31695–31711, 2021, doi: 10.1109/ACCESS.2021.3057655.

[3] B. Dash, M. F. Ansari, P. Sharma, and A. Ali, "Threats and Opportunities with AI-based Cyber Security Intrusion Detection: A Review", *International Journal of Software Engineering & Applications*, Vol. 13, No. 5, pp. 13–21, 2022, doi: 10.5121/ijsea.2022.13502.

[4] E. Sandhya and A. Kumarappan, "Enhancing the Performance of an Intrusion Detection System Using Spider Monkey Optimization in IoT", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 6, pp. 30–39, 2021, doi: 10.22266/ijies2021.1231.04.

[5] P. Maniriho, L. Mahoro, E. Niyigaba, Z. Bizimana, and T. Ahmad, "Detecting Intrusions in Computer Network Traffic with Machine Learning Approaches", *International Journal of Intelligent Engineering and Systems*, Vol. 13,

No. 3, pp. 433–445, 2020, doi: 10.22266/ijies2020.0630.39.

[6] M. Riyadh, B. Ali, and D. Alshibani, "IDS-MIU: An Intrusion Detection System Based on Machine Learning Techniques for Mixed type, Incomplete, and Uncertain Data Set", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 493–502, 2021, doi: 10.22266/ijies2021.0630.41.

[7] L. Ashiku and C. Dagli, "Network Intrusion Detection System using Deep Learning", *Procedia Computer Science*, Vol. 185, pp. 239–247, 2021, doi: 10.1016/j.procs.2021.05.025.

[8] S. Sridevi, R. Prabha, K. N. Reddy, K. M. Monica, G. A. Senthil, and M. Razmah, "Network Intrusion Detection System using Supervised Learning based Voting Classifier", In: *Proc. of 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, pp. 1–6, 2022, doi: 10.1109/IC3IOT53935.2022.9767903.

[9] G. Pradipta, R. Wardoyo, A. Musdholifah, and I. Sanjaya, "Improving Classifiaction Performance of Fetal Umbilical Cord Using Combination of SMOTE Method and Multiclassifier Voting in Imbalanced Data and Small Dataset", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 441–454, 2020, doi: 10.22266/ijies2020.1031.39.

[10] H. Mohamed, A. Hamza, and H. Hefny, "An Efficient Intrusion Detection Approach Using Ensemble Deep Learning models for IoT", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 350–363, 2023, doi: 10.22266/ijies2023.0228.31.

[11] S. T. Vimala and J. P. M. Dhas, "SDN based DDoS attack detection system by exploiting ensemble classification for cloud computing", *International Journal of Intelligent Engineering and Systems*, Vol. 11, No. 6, pp. 282–291, 2018, doi: 10.22266/IJIES2018.1231.28.

[12] N. Rai, N. Kaushik, D. Kumar, C. Raj, and A. Ali, "Mortality prediction of COVID-19 patients using soft voting classifier", *International Journal of Cognitive Computing in Engineering*, Vol. 3, pp. 172–179, 2022, doi: 10.1016/j.ijcce.2022.09.001.

[13] W. H. Chu, Y. J. Li, J. C. Chang, and Y. C. F. Wang, "Spot and Learn: A Maximum-Entropy Patch Sampler for Few-Shot Image Classification", In: *Proc. of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6244–6253, 2019, doi: 10.1109/CVPR.2019.00641.

[14] E. Tasci, "Voting combinations-based ensemble of fine-tuned convolutional neural networks for food image recognition", *Multimed Tools Appl*, Vol. 79, Nos. 41–42, pp. 30397–30418, 2020, doi: 10.1007/s11042-020-09486-1.

[15] G. Gao, H. Wang, and P. Gao, "Establishing a Credit Risk Evaluation System for SMEs Using the Soft Voting Fusion Model", *Risks*, Vol. 9, No. 11, p. 202, 2021, doi: 10.3390/risks9110202.

[16] A. Mahabub, M. I. Mahmud, and F. Hossain, "A robust system for message filtering using an ensemble machine learning supervised approach", *ICIC Express Letters, Part B: Applications*, Vol. 10, No. 9, pp. 805–811, 2019, doi: 10.24507/icicelb.10.09.805.

[17] V. C. Osamor and A. F. Okezie, "Enhancing the weighted voting ensemble algorithm for tuberculosis predictive diagnosis", *Sci Rep*, Vol. 11, No. 1, p. 14806, 2021, doi: 10.1038/s41598-021-94347-6.

[18] N. Tóth and B. Pataki, "Classification confidence weighted majority voting using decision tree classifiers", *International Journal of Intelligent Computing and Cybernetics*, Vol. 1, No. 2, pp. 169–192, 2008, doi: 10.1108/17563780810874708.

[19] A. Navot, R. G. Bachrach, Y. Navot, and N. Tishby, "Is Feature Selection Still Necessary?", *International Statistical and Optimization Perspectives Workshop "Subspace, Latent Structure and Feature Selection*", pp. 127–138, 2006, doi: 10.1007/11752790_8.

[20] K. Shanthi and R. Maruthi, "Machine Learning Approach for Anomaly-Based Intrusion Detection Systems Using Isolation Forest Model and Support Vector Machine", In: *Proc. of 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 136–139, 2023, doi: 10.1109/ICIRCA57980.2023.10220620.

[21] M. A. Hamzah and S. H. Othman, "A Review of Support Vector Machine-based Intrusion Detection System for Wireless Sensor Network with Different Kernel Functions", *International Journal of Innovative Computing*, Vol. 11, No. 1, pp. 59–67, 2021, doi: 10.11113/ijic.v11n1.303.

[22] M. Ajdani and H. Ghaffary, "Introduced a new method for enhancement of intrusion detection with random forest and PSO algorithm", *Security and Privacy*, Vol. 4, No. 2, 2021, doi: 10.1002/spy2.147.

[23] C. Chen, X. Xu, G. Wang, and L. Yang, "Network intrusion detection model based on neural network feature extraction and PSO-

SVM", In: *Proc. of 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pp. 1462–1465, 2022, doi: 10.1109/ICSP54964.2022.9778404.

[24] S. R. Chikkalwar and Y. Garapati, "Autoencoder – Support Vector Machine – Grasshopper Optimization for Intrusion Detection System", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 406–414, 2022, doi: 10.22266/ijies2022.0831.36.

[25] R. Zhour, C. Khalid, and K. Abdellatif, "Hybrid intrusion detection system based on Random forest, decision tree and Multilayer Perceptron (MLP) algorithms", In: *Proc. of 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1–5, 2023, doi: 10.1109/WINCOM59760.2023.10322983.

[26] R. Zhang, Y. Song, and X. Wang, "Network Intrusion Detection Scheme Based on IPSO-SVM Algorithm", In: *Proc. of 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, pp. 1011–1014, 2022, doi: 10.1109/IPEC54454.2022.9777568.

[27] J. Gao, S. Chai, C. Zhang, B. Zhang, and L. Cui, "A Novel Intrusion Detection System based on Extreme Machine Learning and Multi-Voting Technology", In: *Proc. of 2019 Chinese Control Conference (CCC)*, pp. 8909–8914, 2019, doi: 10.23919/ChiCC.2019.8865258.

[28] N. Puri, P. Saggar, A. Kaur, and P. Garg, "Application of ensemble Machine Learning models for phishing detection on web networks", In: *Proc. of 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, pp. 296–303, 2022, doi: 10.1109/CCiCT56684.2022.00062.

[29] A. Kumar, R. Sushil, and A. K. Tiwari, "Classification of Breast Cancer using User-Defined Weighted Ensemble Voting Scheme", In: *Proc. of TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON)*, pp. 134–139, 2021, doi: 10.1109/TENCON54134.2021.9707374.

[30] R. H. H. Aziz and N. Dimililer, "Twitter Sentiment Analysis using an Ensemble Weighted Majority Vote Classifier", In: *Proc. of 2020 International Conference on Advanced Science and Engineering (ICOASE)*, pp. 103–109, 2020, doi: 10.1109/ICOASE51841.2020.9436590.

[31] M. A. R. Putra, T. Ahmad, and D. P. Hostiadi, "Analysis of Botnet Attack Communication Pattern Behavior on Computer Networks", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 533–544, 2022, doi: 10.22266/ijies2022.0831.48.

[32] D. Shankar, G. V. S. George, and N. Kanya, "OptiBiNet_GRU: Robust Network Intrusion Detection System Using Optimum Bi-Directional Gated Recurrent Unit", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 75–91, 2023, doi: 10.22266/ijies2023.0630.06.

[33] C. Andrade, "Z Scores, Standard Scores, and Composite Test Scores Explained", *Indian J Psychol Med*, Vol. 43, No. 6, pp. 555–557, 2021, doi: 10.1177/02537176211046525.

[34] L. W. Mdakane and W. Kleynhans, "Feature Selection and Classification of Oil Spill From Vessels Using Sentinel-1 Wide–Swath Synthetic Aperture Radar Data", *IEEE Geoscience and Remote Sensing Letters*, Vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2020.3025641.

[35] A. Dogan and D. Birant, "A Weighted Majority Voting Ensemble Approach for Classification", In: *Proc. of 2019 4th International Conference on Computer Science and Engineering (UBMK)*, pp. 1–6, Sep. 2019, doi: 10.1109/UBMK.2019.8907028.

[36] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)", In: *Proc. of 2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, Nov. 2015, doi: 10.1109/MilCIS.2015.7348942.

[37] A. Dickson and C. Thomas, "Analysis of UNSW-NB15 Dataset Using Machine Learning Classifiers", *Machine Learning and Metaheuristics Algorithms, and Applications*, pp. 198–207, 2021.

[38] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system", *Comput Secur*, Vol. 92, p. 101752, 2020, doi: 10.1016/j.cose.2020.101752.

[39] J. Zhang, Y. Ling, X. Fu, X. Yang, G. Xiong, and R. Zhang, "Model of the intrusion detection system based on the integration of spatial-temporal features", *Comput Secur*, Vol. 89, p. 101681, 2020, doi: 10.1016/j.cose.2019.101681.