symmetry

MDPI

Article

Hierarchical Intrusion Detection Using Machine Learning and Knowledge Model

Martin Sarnovsky * and Jan Paralic

Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letna 9, 040 01 Košice, Slovakia; jan.paralic@tuke.sk

* Correspondence: martin.sarnovsky@tuke.sk

Received: 31 December 2019; Accepted: 25 January 2020; Published: 1 February 2020

Abstract: Intrusion detection systems (IDS) present a critical component of network infrastructures. Machine learning models are widely used in the IDS to learn the patterns in the network data and to detect the possible attacks in the network traffic. Ensemble models combining a variety of different machine learning models proved to be efficient in this domain. On the other hand, knowledge models have been explicitly designed for the description of the attacks and used in ontology-based IDS. In this paper, we propose a hierarchical IDS based on the original symmetrical combination of machine learning approach with knowledge-based approach to support detection of existing types and severity of new types of network attacks. Multi-stage hierarchical prediction consists of the predictive models able to distinguish the normal connections from the attacks and then to predict the attack classes and concrete attack types. The knowledge model enables to navigate through the attack taxonomy and to select the appropriate model to perform a prediction on the selected level. Designed IDS was evaluated on a widely used KDD 99 dataset and compared to similar approaches.

Keywords: intrusion detection; machine learning; classification; knowledge modelling

1. Introduction

With the increasing extent, diversity and added value of information and communication systems (ICT) usage, especially in today's networked world, the number and types of security attacks increase also (see, e.g., [1]). Companies, as well as public institutions, invest constantly growing parts of their ICT budgets on their network and computer security. In order to efficiently cope with this situation, research in the ICT security-related domain is very much needed.

We can observe two main approaches to IDS in the research literature. One focuses more on machine learning approaches with the aim to achieve the best possible prediction of attacks, or even specific attack types. The other one (knowledge-based approach) is more user-centric and tries to model the necessary background knowledge in the network security domain. This article presents a particular and specific contribution to the mentioned challenges in terms of knowledge-based IDS design, implementation and experimental evaluation. The main motivation is to get the best out of both worlds, i.e., extremely good performance in terms of prediction accuracy and false alarm rates, on one hand, and proper coverage of the relevant context as well as the provision of proper additional information to the network operators, on the other. We build on the current state-of-the-art in the two prominent areas in IDS, contributing to ICT security-threat prevention, i.e., machine learning and knowledge-based approaches. We are looking for symmetry in using both of these principal approaches, which means that both of them are equally important and mutually contribute to addressing intrusion detection and prevention challenges.

The rest of this paper is organized as follows. Related work is presented in the following Section 2. Main building blocks of this symmetry-following strategy are outlined in Section 3.1, followed by

the detailed description of the two main building blocks, namely the knowledge model (described in Section 3.2) and machine learning approaches (Section 3.3). The mutual cooperation of the two main building blocks of our solution is presented in Section 3.4.

Section 4 presents the experimental evaluation of the proposed symmetry-following system for intrusion detection. We start with the introduction of performance metrics used (Section 4.1) in the subsequent experiments. Experimental evaluation is divided into three parts. We first present the performance of the two main prediction models of our hierarchical IDS in Section 4.2.1, continue with the explanation of the interplay between our knowledge model and prediction models in Section 4.2.2 and performance of a specific prediction model targeted for seldom new attack types in Section 4.2.3. Finally, Section 5 summarizes the main findings from our experiments and highlights the advantages of the proposed solution, which brings the symmetry-following strategy.

2. Related Work

In recent years, the use of machine learning methods in network intrusion detection domain became a widely studied area of research. With several publicly available datasets, numerous different approaches emerged to tackle the problem of detection of network attacks. Currently, popular methods used in the intrusion detection domain are neural networks models. As they usually gain better performance than traditional machine learning models, they became popular methods in this area, although their explanatory capacity is often very limited. Numerous different approaches are used, including both shallow and deep learning methods [2], or a combination of both [3].

Standalone data analytics and machine learning methods are often combined into more advanced hybrid approaches, combining various methods including the use of hierarchical and layered models [4–7] and anomaly detection models [8,9] or enhancing the machine learning models with knowledge-based approaches [10,11]. The overall motivation of those approaches is usually aimed to provide the capability of detection of rare attacks without sacrificing the detection accuracy of the frequent ones and, on the other hand, minimizing the false alarm rate in such attacks. Ahmin et al. [12] proposed a hierarchical model combining tree and rule-based concepts and used three models (REP Tree, JRip, Forest PA), where the outputs of the two models are used as the inputs to the third one, creating a hierarchical model consisting of three classifiers, predicting either normal or attack connections and attack types. The approach was evaluated on the CICIDS 2017 dataset and achieved an accuracy of 96.65%, while keeping detection rate at 94,47%. Compared to other models on the same dataset, this approach gained superior accuracy performance, while still keeping reasonable training and evaluation time. In Reference [13], a layered-approach with multi-classifiers is presented. To improve the classification of minority attacks, authors used a combination of Naive Bayes and NBTree models on the NSL-KDD dataset. The IDS uses four layers, the top two for detection of major attacks (e.g., DoS, Probe) and the other two layers for minority attack classes (U2R, R2L). The model uses different reduced feature sets for each attack prediction; authors kept only a subset of relevant features for each attack class using domain knowledge and continuous experiments with the models. Evaluation proved an improvement in classification in minor classes compared to standard approaches, gaining overall accuracy of 99.05%. In Reference [14], Gain Ratio was used to select the features for each layer of the layered model consisting of trees, Naive Bayes and Perceptron models. In Reference [15], a similar goal was achieved using Conditional Random Fields approach. Recently, Zhou et al. [16] presented the ensemble model consisting of C4.5, Random Forest and PA Forest, where the decisions are based on the average of probabilities rule (voting based on average of probabilities). The ensemble model uses feature selection by applying a hybrid approach of combination of correlation-based feature selection and biologically inspired bat algorithm. The model was tested on KDD 99, CICIDS 2017 and NSL-KDD datasets, achieving superior performance compared to other similar models (accuracy of 99,9% on KDD, 96,8% on CICIDS 2017 and 99,1% on NSL-KDD).

In the area of knowledge models for intrusion detection systems, there have been ontologies explicitly designed for the description of the network attacks [10,17,18], more specifically targeted

to specific issues related to the domain, e.g., vulnerabilities [19] or, on the other hand, covering a broader range, e.g., entire cybersecurity [20]. Such ontologies were in the past used to develop the ontology-based IDS. In Reference [21], authors present the DAML+OIL attack ontology and corresponding detection system applied on the KDD 99 data. The ontology includes high-level domain concepts (e.g., Attack, Host, Component, etc.) and defines the basic relations between the protocols (Components) and attacks, organized in the taxonomy. Abdoli et al. [22] use the ontology to capture the semantic relations between the attacks in distributed IDS and improve the detection using such knowledge. Authors studied the attack scenarios and their effects and identified the attack relationships, which were incorporated into the ontology. Ontology was evaluated on KDD 99 dataset, focusing on the DoS attack classes, recognizing 99.9% of DoS attacks in the testing data. In distributed IDS, ontologies were also used to support cooperative detection [23]. In Reference [24], ontology-based, situation-aware intrusion detection system is presented. The approach is based on the integration of various heterogeneous data sources used to create the semantically rich knowledge base. Reasoner processes the streams from the sources, extracts the knowledge and uses the ontology to infer the possibility of an attack. More recently, OWL-S attack ontology was used in IDS to detect the protocol-specific attacks [25], which was evaluated on the KDD 99 dataset. In Reference [26], authors describe the usage of the ontology in semantic intrusion detection system for malware detection.

The approach presented in this paper combines both studied areas—hybrid hierarchical classifiers with knowledge-based approaches. Hierarchical classification is based on the taxonomy of the attack classes and uses various classification models to handle the prediction on a specific level of the target attribute. To predict the particular class of attack, we used an ensemble model with weighted voting for particular classes to achieve more robust classification even in minority classes. The entire process is guided by the ontology, which is used to navigate through the attack taxonomy and retrieve the corresponding model to perform the prediction and domain-specific knowledge related to the severity of the attacks, which could complement the predictions.

3. Hierarchical Intrusion Detection

3.1. The Overall Architecture of the Proposed System

The main objective of the proposed approach is to combine the multi-layered hierarchical model approach of using various predictive models to detect the intrusions on different levels of the attack type taxonomy and combine it with the knowledge obtained from the domain-specific knowledge contained in the ontology. Scheme depicted in Figure 1 outlines how the knowledge model is used in the prediction.

Hierarchical classification can be divided into two separate phases.

1. Normal/Attack separation—the first phase is a binary classification task. The classifier used in this phase is used to distinguish normal traffic and attacks. If a connection is labelled as a normal one, then an alarm is not raised. Otherwise, the suspicious connection is processed by a set of models to determine the class of attack during the phase 2.
2. Attack class and type prediction—this phase is guided by the taxonomy of the attacks from the knowledge model. The system hierarchically processes the taxonomy and selects the appropriate model to classify the instance on a particular level of a class hierarchy.
3. When a class of attack is predicted, ontology is queried for all relevant sub-types of the attack type and to retrieve the suitable model to predict the particular sub-type. Knowledge model can also be used to extract specific domain-related information as a new attribute, which could be used either to improve the classifier's performance or to provide context, domain-specific information which could complement the predictive model.

More in-detail description of the predictive models and their evaluation will be provided in the following sub-sections.

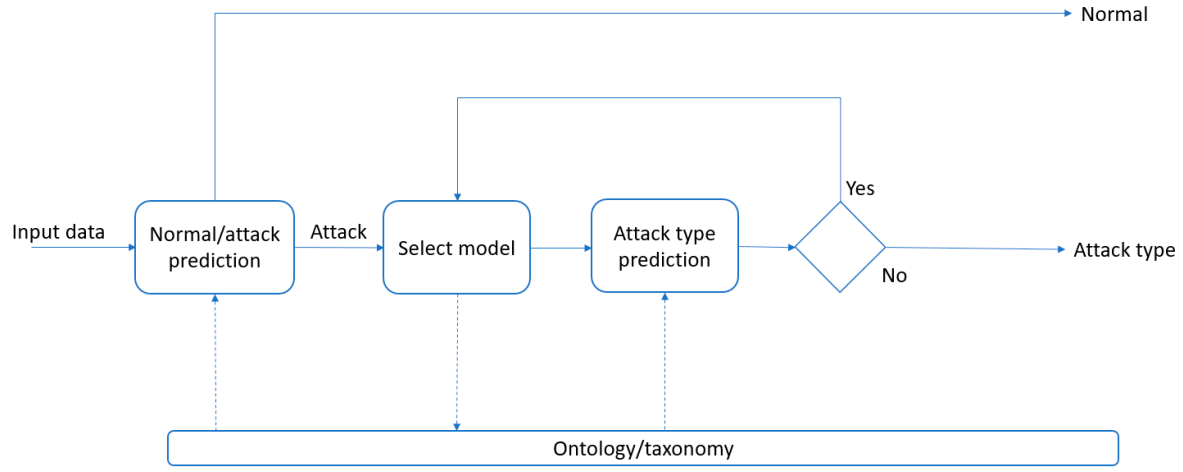

Figure 1. The overall architecture of the proposed intrusion detection system.

3.2. Network Intrusion Knowledge Model

The central part of the knowledge model is the attack taxonomy. Figure 2 illustrates the overall taxonomy of the network attack included KDD 99 dataset, which is one of the (still) most frequently used data collections in the selected domain [27,28]. Attacks are divided into four groups (DOS, R2L, U2R, Probe), each of them including specific attack types.

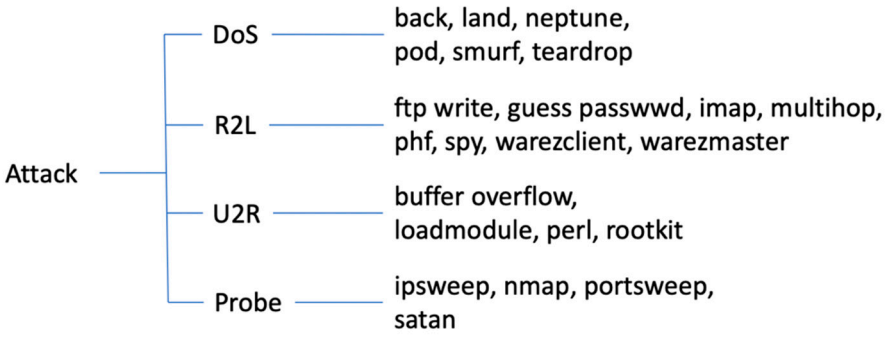

Figure 2. Taxonomy of the target attribute in KDD 99 dataset.

Besides the taxonomy, the knowledge model also covers the domain-specific knowledge from the network intrusion field. Basic concepts and relations were extracted from the already existing security domain models and standards [29]. As the objective of the knowledge model was to use it in the data analytical tasks, the concepts and properties had to be able to be mapped to the data used in the process. Moreover, concepts related to the classification models had to be included, to create the relation between the particular classifier and its usability on the specific level of target attribute hierarchy. Figure 3 visualizes the top-level concepts and relations between them.

- Connection class represents particular connections, whether normal ones or attacks. The class forms a class hierarchy when a sub-class Attacks represents the attack. Attack sub-classes (TypeOfAttack) represent the classes of the attacks (e.g., DoS, r2l, etc.); concrete attacks types are modelled as a sub-class of the ConcreteAttack type classes (e.g., back, land, etc.).
- Effect class covers all possible effects that an attack affects (e.g., slowing down of the server response, gaining root access for the user, service outage, etc.)
- Mechanism class and its sub-classes describe all possible mechanisms of particular attacks (e.g., poor environment maintenance, incorrect configuration of the components, etc.)

- Flag characterizes the normal or error states of the specific connections (e.g., service not responding, denied the connection, etc.)
- Protocol represents the protocols used in the connection (e.g., TCP, UDP, etc.)
- Service concept describes service types related to the connection (e.g., http, telnet, etc.)
- Severity describes how severe the possible attack type effects could be (low, medium and high).
- Targets define the possible targets of the particular attack type (e.g., user, network, data, etc.).
- Models concept covers the classification models used to predict the given target attribute

Ontology was implemented in OWL (Web Ontology Language). Particular ontology instances represent the particular connections (e.g., connection records from the data). Trained and serialized classification models are instantiated as the instances of the *Model* class. The models could be accessed using their *URI* property, which contains the URI of the stored serialized model in the system.

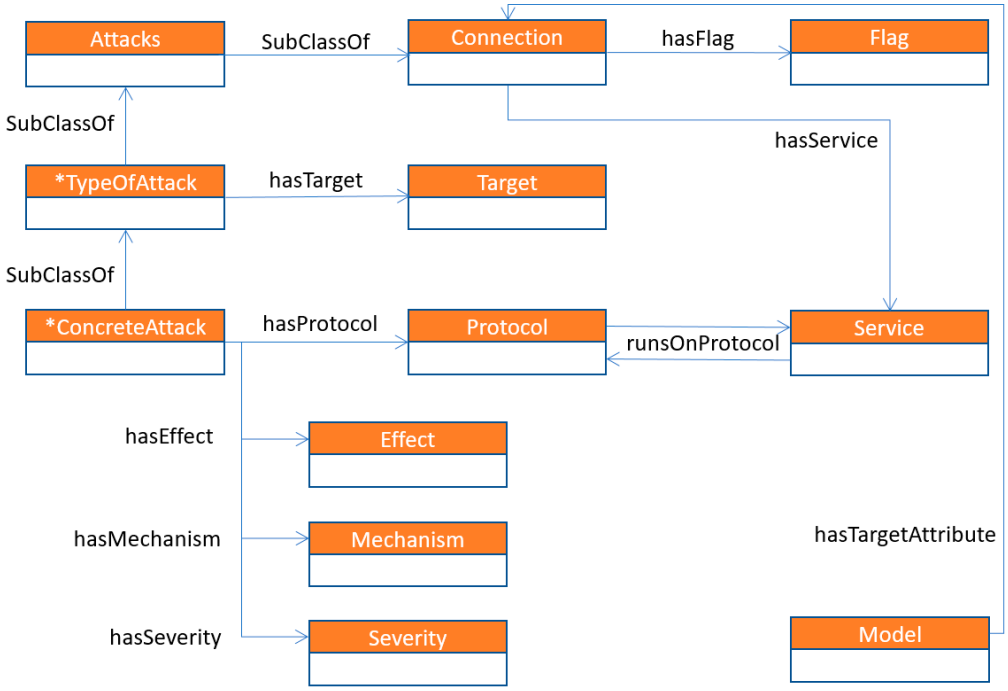

Figure 3. Knowledge model for the network intrusion detection domain.

3.3. Machine learning Models for Detection of the Network Attacks on KDD 99 Dataset

To evaluate the proposed approach, we used the KDD Cup 1999 competition dataset 1999 [30], a commonly used network intrusion detection data collection. Dataset consists of the device logs in a LAN network collected over nine weeks. We used the 10% sample subset of the dataset, which consisted of 494,021 records. Each record represents a connection and is labelled as a normal one, or as an attack (exactly one attack type assigned). The target attribute (attack type) can be organized in a concept hierarchy, which is specified in the knowledge model (see Figure 2). There are 24 different attack types present in the dataset, and each attack type belongs to one of the four attack classes. The target attribute is heavily imbalanced, Table 1. summarizes the number of records of attack types and attack classes.

Table 1. Attack types and classes.

Attack	Attack Class	Number of Samples
back	DoS	2203
land		21
neptune		107,201
pod		264
smurf		280,790
teardrop		979
satan	Probe	1589
ipsweep		1247
nmap		231
portsweep		1040
guess_passwd	R2L	53
ftp_write		8
imap		12
phf		4
multihop		7
warezmaster		20
warezclient		1020
spy		2
buffer_overflow	U2R	30
loadmodule		9
perl		3
rootkit		10
normal	Normal	97,227

Each connection is described using a set of features: basic features of the connection, content features and traffic features (overall 32 features). The first group describes the protocol type, duration of the connection, service on the destination of the attack and other features relevant to the standard TCP connection description. Content features are attributes that could be linked to the domain-specific knowledge. Traffic features describe the two-second time window attributes, e.g., a number of connections to the same host in such window. During the pre-processing, we mostly focused on feature selection. According to the work of Zhou and Cheng in [16], we selected only the most relevant attributes: service, src_bytes, dst_bytes, logged_in, num_file_creations, srv_count, serror_rate, rerror_rate, srv_diff_host_rate, dst_host_count, dst_host_diff_srv_rate, dst_host_srv_diff_host_rate.

To illustrate the proposed approach on the KDD 99 network intrusion dataset, Figure 4 depicts the structure of the prediction models used on different levels of the target attribute class hierarchy. During the first phase, an *attack detection model* is used for prediction on the top-level of the class hierarchy, e.g., to distinguish between the attack connections from the normal ones. The classifier was trained on all instances of the dataset and target attribute was transformed to a binary one reflecting the class hierarchy. The main goal of the classifier is to reliably predict the normal connections and separate them from the attack ones.

If the model detects an attack connection, the *ensemble model* is used to predict one of the four types of the attack. In this case, we used an ensemble of classifiers with a weighted voting scheme, trained on all attack instances in the dataset. The main reason to use such an approach was the effect of class imbalance on this level. Standard machine learning models were able to gain good accuracy, achieved mostly by good performance on the dominant class (in this case DoS attacks). However, the models struggled to predict minor classes such as U2R.

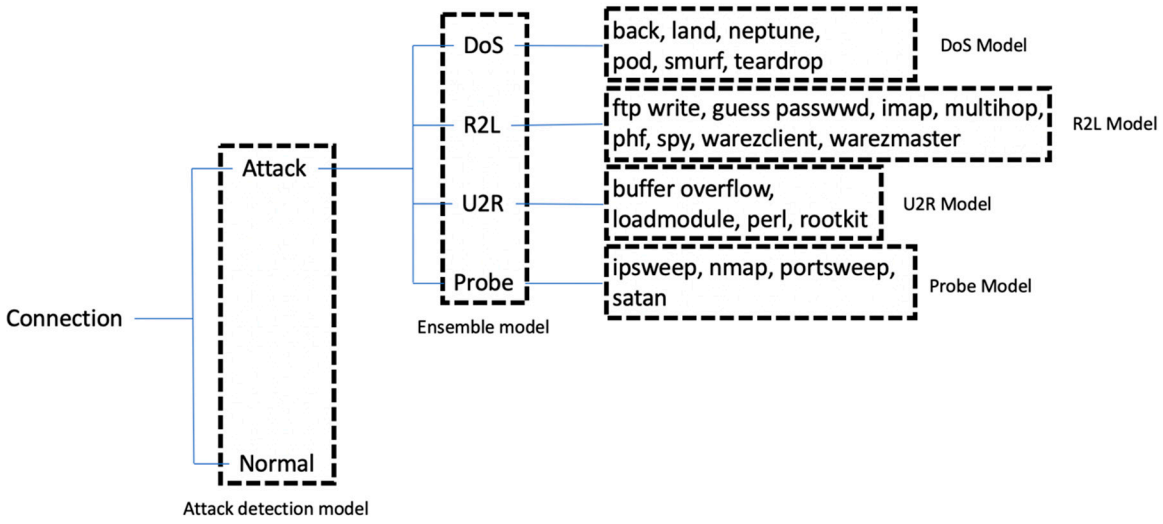

Figure 4. Different models used for predictions on different levels of target attribute class hierarchy.

Some standard classifiers performed very well when predicting the specific classes but failed (or produced significant errors) on the other ones. For example, when training a decision tree model on the attack connections, the model performed very well when classifying the DoS and R2L classes, on the other hand, missed an almost significant amount of the Probing attacks and was not able to detect the U2R attacks at all. That leads us to the idea of complementing such classifier with a set of other models, able to reliably predict the other classes. To perform a final prediction, weighted voting could be implemented, based on the classification performance in prediction of a particular class. The weighting scheme is depicted in Table 2.

Table 2. Weighting scheme used in the ensemble model.

Weighting Scheme	Class 1	Class 2	Class 3	Class 4
model 1	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$w_{1,4}$
model 2	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,4}$
model 3	$w_{3,1}$	$w_{3,2}$	$w_{3,3}$	$w_{3,4}$
...	...	...	...	...

After the prediction of the attack class, particular models were employed to predict the concrete type of the specific attack class. Four different models were trained, each for prediction of the particular attack types for each attack class. The models were trained using only instances of particular attack class records from the dataset. This proved to be difficult on minority class (U2R), as the dataset contains very few records of that type.

Then, when an attack class is predicted, a set of models are dedicated to predicting the concrete attack type (e.g., DoS model to predict the concrete type of DoS attack, when such attack is detected by the IDS). There are four different models trained, each dedicated to prediction of each attack class' types.

All models were trained in the Python environment using standard packages such as scikit-learn in Python. Predictive models are then serialized using native or supporting tools (e.g., Pickle in Python). Serialized models URI (Uniform Resource Identifier) was then passed to the knowledge model.

3.4. Use of Knowledge Model in Multi-Stage Intrusion Detection

Implemented semantic model is used in the hierarchical classification mostly to navigate through the target class taxonomy and to select the appropriate model to perform a prediction on the chosen level. The detection system is implemented in Python and communicates with the ontology using RDFlib package, which enables to specify the SPARQL queries and retrieve the results as a Python

object. When predicting the unknown connection, the system checks the ontology taxonomy and starts from the top level. Using SPARQL query, it checks the top level of the class hierarchy and retrieves the corresponding model for given prediction type using the hasTargetAttribute property (e.g., attack detection). As the classification models are serialized and stored, they can be identified and retrieved using their URIs. Once a classifier is retrieved, it is deserialized in the system and used to predict the given unknown example. Once the prediction is performed, the system checks the prediction and queries the ontology to see if there is a classifier able to process the record further (e.g., to predict the type of the attack) and retrieves the corresponding classifier.

As the prediction of concrete sub-type of the attacks could be in some cases tricky (minority classes), we considered a retrieval of the information from the ontology, which could be beneficial without predicting a correct attack type. Moreover, in the network intrusion domain, new attacks (attacks, that were not present in the training data) can appear when using the models. In such a case, the system is able to predict the attack class and use domain-specific information from the knowledge model to enhance the prediction with domain-specific knowledge, e.g., severity of the detected attack. Besides using the class hierarchy, we experimented by using the domain-specific knowledge from the ontology to improve the detection system. We decided we can leverage the expert information about the possible effects of the given attacks and their severity. If the model is not reliable enough to predict the concrete attack sub-type, the system can be used to provide at least information on how severe the attack would be. Using the Effect class and hasSeverity property, we were able to retrieve the severity values of the particular attacks and enhance the dataset with this newly acquired attribute. We experimented then with the training of the classifiers able to predict the severity of the attacks, which could serve as a supporting source of information, complementing the attack type prediction or providing the severity information.

4. Experimental Evaluation

4.1. Performance Metrics

To evaluate the performance of the designed approach, we used metrics commonly employed in classification tasks such as precision and recall. To evaluate the prediction of the particular classes, we used the confusion matrix. The matrix is essential, as it precisely depicts both, correctly and incorrectly classified records. Especially in case of not balanced target classes, we could estimate the classification errors. When predicting the network attacks on the first level of target taxonomy, we used the confusion matrix for binary classification and measured the standard evaluation metrics:

Precision:

$$P = \frac{TP}{TP + FP}, \tag{1}$$

Recall:

$$R = \frac{TP}{TP + FN}, \tag{2}$$

where:

- TP (True Positive): when predicted network attack is in fact an attack,
- TN (True Negative): when predicted normal record is in fact normal record,
- FN (False Negative): when predicted normal record is in fact an attack,
- FP (False Positive): when predicted network attack is in fact a normal record.

From the perspective of the performance evaluation of the entire network intrusion detection, we decided to evaluate the models also from the perspective of missed attacks and raised false alarms. To evaluate such aspects, F-measure combining both, precision and recall metrics can be used, as well as FAR (False Alarm Rate), which is the total number of incorrectly identified attacks (FP) divided by the total number of normal records (TN + FP).

Precision and recall metrics could be directly used to evaluate the first level of hierarchical classification, as the target attribute was a binary one. In the following stages, we computed the precision and recall metrics for each target attribute value and to measure the overall precision and recall; we used macro-averaging. This was more important, as it was crucial to precisely see the number of false alarms and missed attacks to correctly estimate the model quality. Then, we used a multi-class confusion matrix to estimate the overall classifier performance.

4.2. Performance Evaluation

4.2.1. Model Training and Evaluation

For the attack detection model, we used a classifier based on decision trees. Dataset consisted of all dataset records; target attribute was transformed to a binary one, labelling the normal and attack connections. We split the dataset in a 70/30 training/testing ratio. The classifier was trained without limitation of maximum depth of the tree and used gini as a splitting criterion. The testing data were not used in the training of any of the models (ensemble and attack type models) and were also used to evaluate the whole system. Attack detection model was evaluated on the testing set, and the model gained accuracy of 0.9997; confusion matrix is depicted in Table 3.

Table 3. Results of the decision tree classifier for attack prediction (attack detection model).

Attack Detection Model	Normal	Attack	Precision	Recall
Normal	29,095	11	0.999	0.999
Attack	35	119,066		

An *ensemble classification model* was trained using the attack records from the dataset (97277 normal records were removed from training). In the ensemble model, we used Naive Bayes and Decision Tree classifiers as the base models. We chose the different tree and Naive Bayes models in order to have models that correctly cover all attack classes in the dataset. When training tree-based models, experiments proved, that such models were performing reasonably on Probe, DoS and R2L attacks. On the other hand, U2R (minor class) class produced either many false alarms, or the models were not able to detect the U2R attacks at all. Therefore, separate one-vs-all models were trained to detect just the U2R attacks. Ensemble model was then able to leverage both classifier types—tree-based model to detect the R2L, Probe and DoS attacks and one-vs-all model to detect the U2R attacks. The training set consisted only of attack connections (396743 records) and the models were trained and evaluated on such data using the 70/30 training/testing split and evaluated their performance on the particular classes. Class accuracies were then used to set the weights of the ensemble model created of these base classifiers. Then, we evaluated the overall ensemble model on the testing set of the KDD 99 dataset (we did not use the records used in training). Table 4 depicts the confusion matrix of the ensemble model performance when classifying the attack records.

Table 4. Confusion matrix of the ensemble model for attack type prediction (ensemble classification model).

Ensemble Model	Probe	U2R	DoS	R2L	Precision	Recall
Probe	1279	0	1	0	0.992	0.992
U2R	0	15	0	0	1	0.882
DoS	6	0	117,385	0	0.999	0.999
R2L	4	2	0	331	0.982	1

On the lowest level of the target attribute hierarchy, four classifiers were employed, each for selection of particular subsets of the attacks. We used tree-based classifiers, and each model was trained

on the dataset consisting of records of particular attack class. Table 5 summarizes the performance of the models.

Table 5. Performance metrics of particular models used to predict the particular kind of the specific attack types.

	Probe	U2R	DoS	R2L
Overall accuracy	0.991	0.937	0.999	0.989
Precision	0.989	0.927	0.999	0.879
Recall	0.989	0.875	0.999	0.833

4.2.2. Overall Approach Performance

When all particular models in the presented hierarchical detection system were trained, the overall approach was evaluated on the testing subset of KDD 99 dataset. The testing set consisted of records, previously not used during the training of the models, sampled from the KDD 99 dataset. Serialized models were stored and instantiated in the ontology, which was used to navigate in the target attribute hierarchy. For each test record, attack detection model was applied to decide, if the record is an attack or not. To retrieve the attack detection model instance from the ontology, SPARQL query was used. If a record was labelled as an attack, ontology was used to retrieve class taxonomy. Target attribute was specified according to the class hierarchy, and the particular model able to predict the attack types on that level was retrieved and used. Examples of used SPARQL queries are summarized in Table 6.

Table 6. Results of the decision tree classifier for attack prediction.

SPARQL	Action
SELECT ?lname WHERE { ?inst a onto:Connections. ?inst onto:hasModel ?lname	Retrieve the classifier able to predict the attack at the Connection level (decide if the connection is an attack or not)
SELECT ?lname WHERE { ?inst a onto:Attacks. ?inst onto:hasModel ?lname	Select the model for prediction of the attack type

The detection system was evaluated using the same metrics as used in the particular model's evaluation. Table 7. summarizes the system performance—we used the system to predict the type of the attack—and compares the results with other similar approaches [16]. Table 8 presents the confusion matrix of the system and summarizes the precision and recall of particular classes.

Table 7. Performance of the detection system to classify the attack classes.

Classifier	Accuracy	Precision	F-measure	FAR
C4.5	0.969	0.947	0.970	0.005
Random Forests	0.964	0.998	0.986	0.025
ForestPA	0.975	0.998	0.998	0.002
Ensemble model	0.976	0.998	0.998	0.001
Our approach	0.998	0.998	0.998	0.001

Table 8. Performance of the detection system to classify the attack classes.

Ensemble Model	Probe	U2R	DoS	R2L	Normal	Precision	Recall
Probe	1176	0	5	0	7	0.998	0.999
U2R	0	15	0	0	5	0.750	0.937
DoS	4	0	117,547	0	1	0.999	0.999
R2L	3	1	0	346	7	0.969	0.997
Normal	1	0	3	1	48,454	0.999	0.999

4.2.3. Attack Severity Prediction

To demonstrate the usefulness of the ontology, supplementary information related to the attack detection can be provided. For example, in case that the prediction of the concrete attacks is not accurate enough, class prediction can be enhanced by providing the additional aspects of the attack, such as attack severity. Ontology can be used to predict the attack class and the possible severity of the attack. Severity information can be instantiated in the ontology in Severity concept and extracted to data records as a newly created feature. Therefore, we can train a classification model to predict the severity value of the predicted attack. Table 9 describes the severity values of the attack types.

Table 9. The severity of the attacks.

Attack Type	Severity Level
ftp_write	low
guess_passwd	low
spy	low
warezclient	low
warezmaster	low
buffer_overflow	medium
loadmodule	medium
perl	medium
rootkit	medium
phf	medium
imap	medium
multihop	medium
ipsweep	medium
portsweep	medium
nmap	medium
satan	high
back	high
land	high
neptune	high
pod	high
smurf	high
teardrop	high

We used the Severity attribute as the target attribute and trained the Random Forest model to predict the severity level of predicted attack type (predicted by the ensemble model). To train the severity predictor, we used the 10% KDD 99 dataset, split in 70/30 train/test ratio. Table 10. summarizes the prediction of severity types for particular attack classes and evaluation of the accuracy of severity prediction. We ran the severity prediction model as a complementary model to ensemble classifier predicting the attack classes. For each attack class, the severity of the attack was predicted. Confusion table summarizes the severity attribute predictions for particular attack types from the testing set and precision and recall metrics of the severity model. Both, precision and recall are in this case very high. This is the result of a newly created target attribute (Severity level), which has three different values, and particular attacks are well separated within the categories specified by those values. Therefore,

standard classifiers perform very well in prediction of such attribute, and the number of missed predictions is minimal.

Table 10. Confusion matrix of the ensemble model for attack severity prediction.

	High	Low	Medium	Precision	Recall
DoS	117,695	0	0	0.999	0.999
Probe	443	0	779		
R2L	0	346	6		
U2R	0	0	20		

5. Discussion and Future Work

The presented approach described the IDS based on a combination of hierarchical ensemble model and knowledge model in the form of an ontology. Both hierarchical and ensemble approaches are relatively popular models used to tackle with the network intrusion detection; the IDS presented in this paper combines both approaches. Hierarchical aspect is considered when the system performs the prediction on a different level of the target attribute generalization. An ensemble model is then employed to detect the attack type to maintain the performance of prediction in major and also minor classes. The ontology then serves to automatically navigate through the target attribute and to select the proper model on the specific level of target generalization. The results proved that the achieved results are comparable to the current state-of-the-art approaches when it comes to performance evaluation using standard metrics. Integration with the knowledge model could be beneficial, as it enables to automate the navigation through the target attribute taxonomy and selection of the appropriate predictive model.

On the other hand, the complexity of the hierarchical and ensemble approach is higher when it comes to training. As each of the models are trained separately, the requirements on training resources and training time are significantly higher than in other compared models. This limitation could be significant when such an approach is expected to be deployed in a real-world environment on dynamic, ever-changing data streams. From this perspective, a very interesting aspect in case of network intrusion detection is the ability to detect the new attacks that are not present in the training set. In real-world scenarios, this corresponds to a specific type of the concept drift (when a new class value appears). To remove this limitation of the standard approaches, either concept drift detection methods or periodic re-training of the models should be employed. Especially in case of ensemble models, early detection of such phenomena is crucial to update the predictive models as soon as possible, so as to not miss any newly appearing attacks. In case of complex ensemble models, the re-training of the partial models could be non-trivial. Moreover, in case of domain knowledge, model is involved (as in the presented IDS model), the update of the model structure is necessary (e.g., addition of the new attack type into the taxonomy).

What could be an also interesting area of research in this field is the deeper integration of the domain-specific knowledge with prediction. The main issue of standard predictive models used in intrusion detection is that the predictions depend on the context of particular events and commonly, such context considers only previous events and their properties. In real-world applications (such as IDS), this context should be expanded with the other, domain knowledge. Predictions then should be the result of information describing the particular event, information about the previous events and information obtained from the domain knowledge model. In this case, the expanded context could be represented by the new, derived features or by the specific expert rules. Such domain expert rules could be used to cover the detection of very rare events (e.g., rare attacks), which could be difficult to extract from the historical data due to a significantly low number of examples. To support such context representation, the knowledge model has to be expanded with more domain concepts and properties, expanding the range of knowledge covered by the ontology.

6. Conclusions

In this paper, we proposed an original symmetrical combination of the knowledge-based approach and machine learning techniques to build a hierarchical intrusion detection system able to support detection of existing types and severity of new types of network attacks. Implemented knowledge model is used in the hierarchical classification mostly to navigate through the target class taxonomy and to select the appropriate model to perform a prediction on the selected level. In case of rare attack types, where the number of available training data is too low, severity prediction model is applied, and the network operator is provided, with information available in the knowledge model about the most probable class of attacks.

The performance of the proposed knowledge-based hierarchical IDS is 0,998 in terms of precision as well as recall and 0,001 in terms of FAR, which outperforms other state-of-the-art approaches presented in Section 2. Moreover, the proposed system is also able to partially cover emerging types of attacks in terms of their predicted severity and additional information stored in the knowledge model. As a result, our knowledge-based approach leads to efficient and information-rich decision support for network operators. In the future work, we plan to extend our approach to make it more dynamic in terms of learning new types of attacks and deciding on the right moments when the available prediction models need to be retrained.

Author Contributions: Conceptualization, J.P.; methodology, M.S. and J.P.; resources, M.S.; software, M.S.; supervision, J.P.; validation, M.S.; writing—original draft, M.S. and J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Slovak Research and Development Agency under the contract No. APVV-16-0213 and by the VEGA project under grant No. 1/0493/16.

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

References

1. Park, J. Advances in Future Internet and the Industrial Internet of Things. *Symmetry* **2019**, *11*, 244. [CrossRef]
2. Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. A Deep Learning Approach for Network Intrusion Detection System. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), New York, NY, USA, 3–5 December 2016.
3. Khan, M.A.; Karim, M.d.R.; Kim, Y. A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network. *Symmetry* **2019**, *11*, 583. [CrossRef]
4. Ahmim, A.; Ghoualmi Zine, N. A new hierarchical intrusion detection system based on a binary tree of classifiers. *Inf. Comput. Secur.* **2015**, *23*, 31–57. [CrossRef]
5. Ahmim, A.; Ghoualmi-Zine, N. A New Fast and High Performance Intrusion Detection System. *Int. J. Secur. Appl.* **2013**, *7*, 67–80. [CrossRef]
6. Kevric, J.; Jukic, S.; Subasi, A. An effective combining classifier approach using tree algorithms for network intrusion detection. *Neural Comput. Appl.* **2017**, *28*, 1051–1058. [CrossRef]
7. Srivastav, N.; Challa, R.K. Novel intrusion detection system integrating layered framework with neural network. In Proceedings of the 2013 3rd IEEE International Advance Computing Conference (IACC), Ghaziabad, India, 22–23 February 2013; pp. 682–689.
8. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
9. Samrin, R.; Vasumathi, D. Review on anomaly based network intrusion detection system. In Proceedings of the 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, India, 15–16 December 2017; pp. 141–147.
10. Arunadevi, M.; Perumal, S.K. Ontology based approach for network security. In Proceedings of the 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), Ramanathapuram, India, 25–27 May 2016; pp. 573–578.
11. Salahi, A.; Ansarinia, M. Predicting network attacks using ontology-driven inference. *arXiv* **2013**, arXiv:13040913.

12. Ahmim, A.; Maglaras, L.; Ferrag, M.A.; Derdour, M.; Janicke, H. A novel hierarchical intrusion detection system based on decision tree and rules-based models. *arXiv* **2018**, arXiv:181209059.
13. Sharma, N.; Mukherjee, S. A Novel Multi-Classifier Layered Approach to Improve Minority Attack Detection in IDS. *Procedia Technol.* **2012**, *6*, 913–921. [CrossRef]
14. Ibrahim, H.E.; Badr, S.M.; Shaheen, M.A. Adaptive layered approach using machine learning techniques with gain ratio for intrusion detection systems. *arXiv* **2012**, arXiv:12107650.
15. Gupta, K.K.; Nath, B.; Kotagiri, R. Layered Approach Using Conditional Random Fields for Intrusion Detection. *IEEE Trans. Dependable Secur. Comput.* **2010**, *7*, 35–49. [CrossRef]
16. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. An efficient intrusion detection system based on feature selection and ensemble classifier. *arXiv* **2019**, arXiv:190401352.
17. Abdoli, F.; Meibody, N.; Bazoubandi, R. An Attacks Ontology for computer and networks attack. In *Innovations and Advances in Computer Sciences and Engineering*; Sobh, T., Ed.; Springer: Dordrecht, The Netherlands, 2010; pp. 473–476. ISBN 978-90-481-3657-5.
18. Razzaq, A.; Anwar, Z.; Ahmad, H.F.; Latif, K.; Munir, F. Ontology for attack detection: An intelligent approach to web application security. *Comput. Secur.* **2014**, *45*, 124–146. [CrossRef]
19. Zhu, L.; Zhang, Z.; Xia, G.; Jiang, C. Research on Vulnerability Ontology Model. In Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 24–26 May 2019; pp. 657–661.
20. Syed, Z.; Padia, A.; Finin, T.; Matthews, L.; Anupam, J. UCO: Unified Cybersecurity Ontology. In Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security, Phoenix, Arizona, 12–13 February 2016.
21. Hung, S.-S.; Liu, D.S.-M. A User-centric Intrusion Detection System by Using Ontology Approach. In Proceedings of the 9th Joint Conference on Information Sciences (JCIS), Kaohsiung, Taiwan, 8–9 October 2006; Atlantis Press: Kaohsiung, Taiwan.
22. Abdoli, F.; Kahani, M. Ontology-based distributed intrusion detection system. In Proceedings of the 2009 14th International CSI Computer Conference, Tehran, Iran, 20–21 October 2009; pp. 65–70.
23. Abdoli, F.; Kahani, M. Using Attacks Ontology in Distributed Intrusion Detection System. In *Advances in Computer and Information Sciences and Engineering*; Sobh, T., Ed.; Springer: Dordrecht, The Netherlands, 2008; pp. 153–158. ISBN 978-1-4020-8740-0.
24. More, S.; Matthews, M.; Joshi, A.; Finin, T. A Knowledge-Based Approach to Intrusion Detection Modeling. In Proceedings of the 2012 IEEE Symposium on Security and Privacy Workshops, San Francisco, CA, USA, 24–25 May 2012; pp. 75–81.
25. Karande, H.A.; Gupta, S.S. Ontology based intrusion detection system for web application security. In Proceedings of the 2015 International Conference on Communication Networks (ICCN), Gwalior, India, 19–21 November 2015; pp. 228–232.
26. Can, Ö.; Ünallır, M.O.; Sezer, E.; Bursa, O.; Erdoğdu, B. A semantic web enabled host intrusion detection system. *Int. J. Metadata Semant. Ontol.* **2018**, *13*, 68. [CrossRef]
27. Divekar, A.; Parekh, M.; Savla, V.; Mishra, R.; Shirole, M. Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives. In Proceedings of the 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, Nepal, 25–27 October 2018; pp. 1–8.
28. Özgür, A.; Erdem, H. A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. *PeerJ Preprints* **2016**, *4*, e1954v1.
29. Mavroeidis, V.; Bromander, S. Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, Greece, 11–13 September 2017; pp. 91–98.
30. 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.