ランダムフォレストを用いたファイアーウォールの規則の生成

工学部　電気電子情報工学科　4年　松倉雄也

研究背景

ネットワークの現状とセキュリティシステム

- サイバー犯罪の増加により、対策が必要な状況にある。
- その対策としてファイアウォールがある。
- ファイアウォールは，あらかじめ設定された条件にしたがってパケットの通過と破棄を行い、不正な通信の遮断を実現する「アクセス制御」の一種。

従来のファイアーウォールの問題点

- 一般にファイアウォールのルールはセキュリティポリシーに沿って設定する. これは手動で設定される.
- 日々多様な攻撃にさらされるインターネット環境において手動では限界がある.

研究動向

- 丸藤信哉, “ランダムフォレストを用いたファイアーウォールの構築とファイアーウォールの仮想環境化”, 神奈川大学2018年度卒業論文

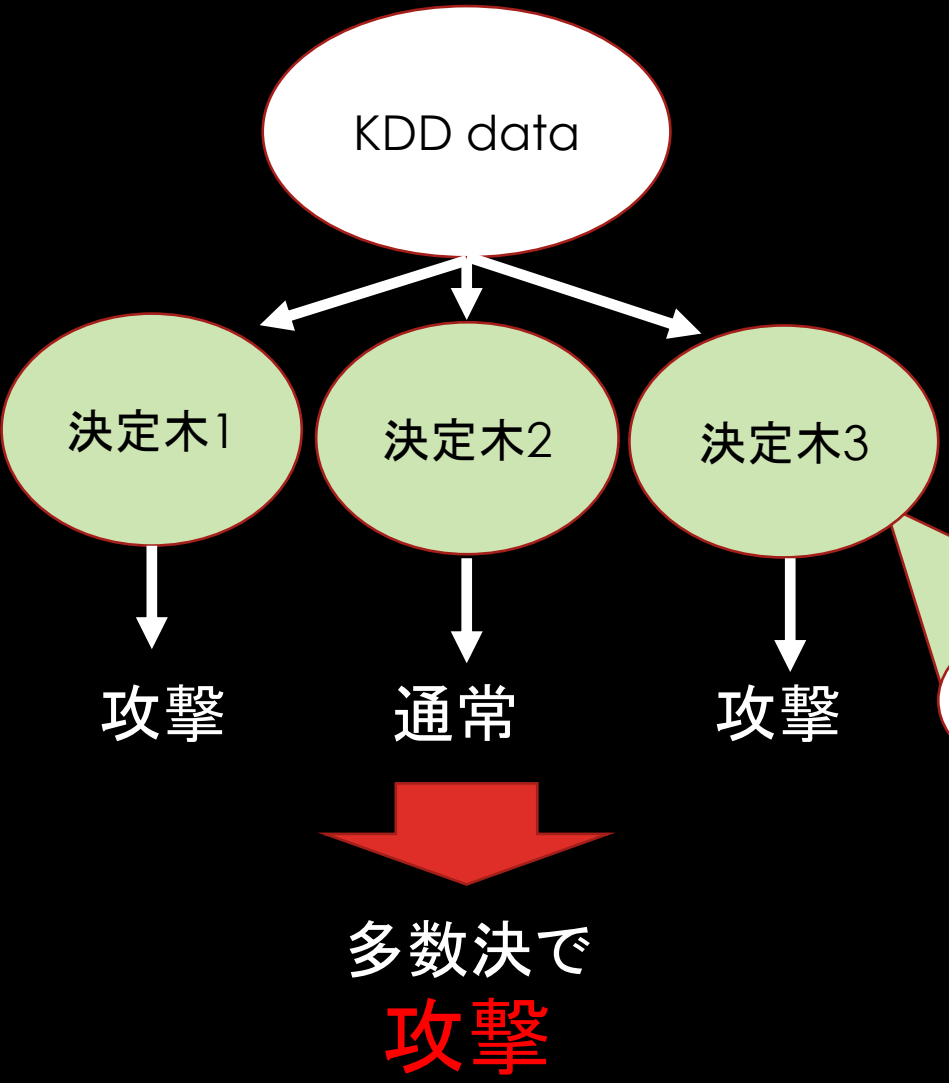
  → ランダムフォレストを用いて、ファイアウォールのフィルタリング設定、仮想環境化の提案

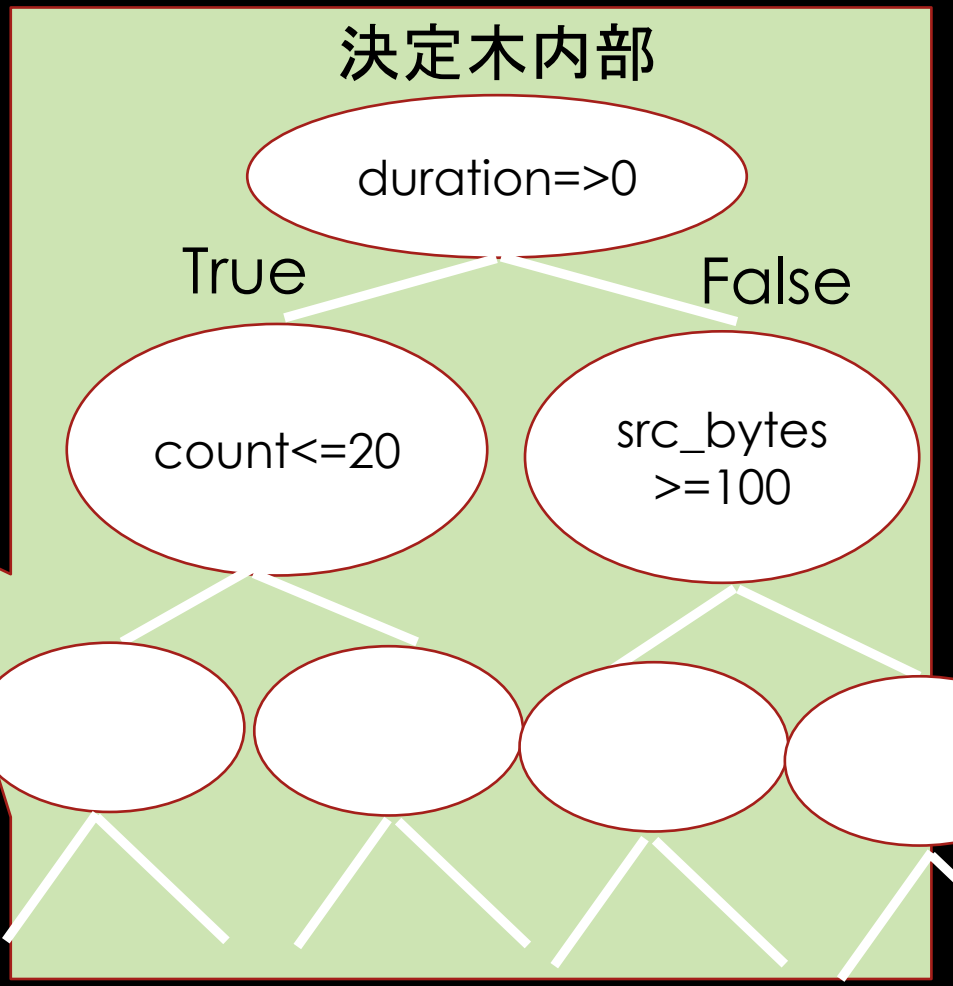
- 野末大貴,”ランダムフォレストを用いたフィルタリングルール”,神奈川大学2018年度卒業論文

  → ランダムフォレストを用いて、フィルタリングルールを設定木の数に着目

ランダムフォレスト

攻撃 or 通常？

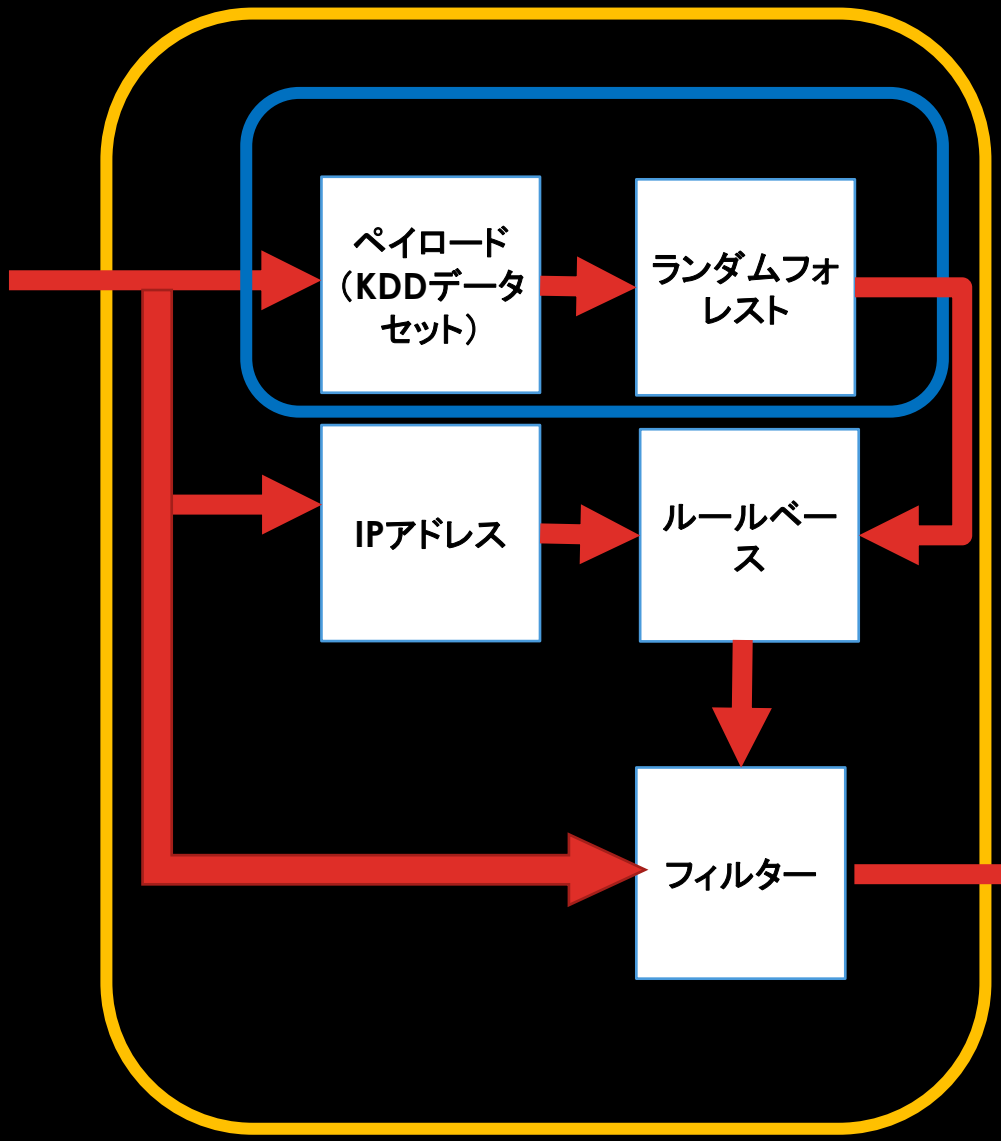




研究目的

- 教師あり機械学習の1つであるランダムフォレストを利用して、条件を満たすファイアーウォールのフィルタリングルールを生成する。
- 本研究では使用するデータをKDDデータセットに限定する.
- 先行研究より様々な条件のデータをランダムフォレストで学習させ，より実用的なルールを作成するシステムを提案する.

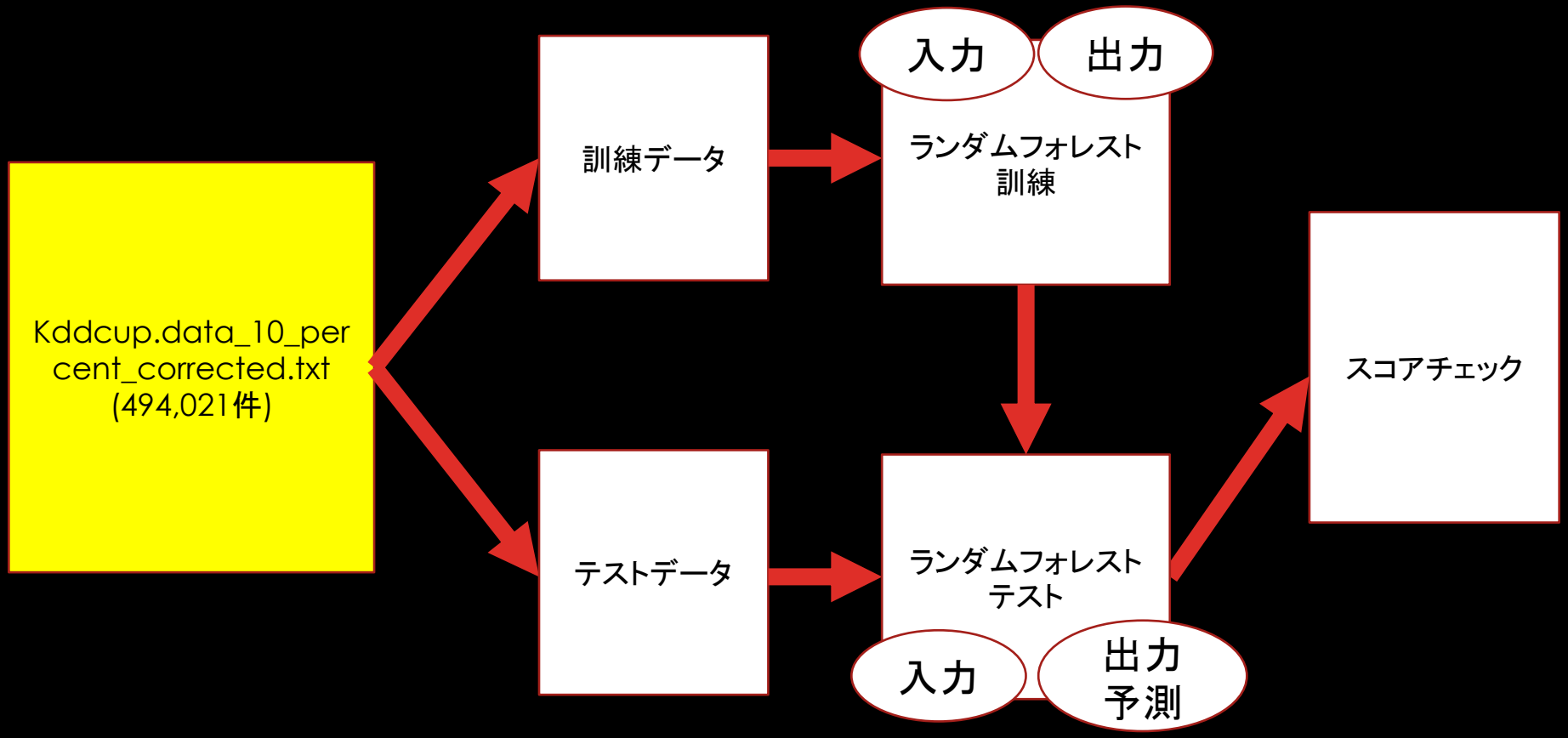
提案手法について



1. 入力をKDDデータセットに限定したときのランダムフォレストの予測精度
2. KDDデータセットの中でも，特に重要度の高いデータに絞った時のランダムフォレストの予測精度
3. 学習していないデータのランダムフォレストの予測精度

提案手法1(基本システム)

- 正規分布によって合成しない現実の観測データを学習入力データとする。
- KDDデータセット内のKddcup.data_10_percent_corrected.txtをそのままランダムフォレストの入力とする。
- 訓練データの割合を90%，10%，1%と変化させたときのランダムフォレストによる予測を行う.



特徴値一覧

特徴値		
duration	num file creations	diff_srv_rate
src_bytes	num_shells	srv_diff_host_rate
dst_bytes	num_access_files	dst_host_count
land	num outbound_cmds	dst_host_srv_count
wrong_fragment	is_host_login	dst_host_same srv_rate
urgent	is_guest_login	dst_host_diff_srv_rate
hot	count	dst_host_same_src_port_rate
num_failed_logins	srv_count	dst_host_srv_diff_host_rate
logged_in	serror_rate	dst_host_serror_rate
num_compromised	srv_serror_rate	dst_host_srv_serror_rate
root_shell	rerror_rate	dst_host_rerror_rate
su_attempted	srv_rerror_rate	dst_host_srv_rerror_rate
num_root	same_srv_rate	

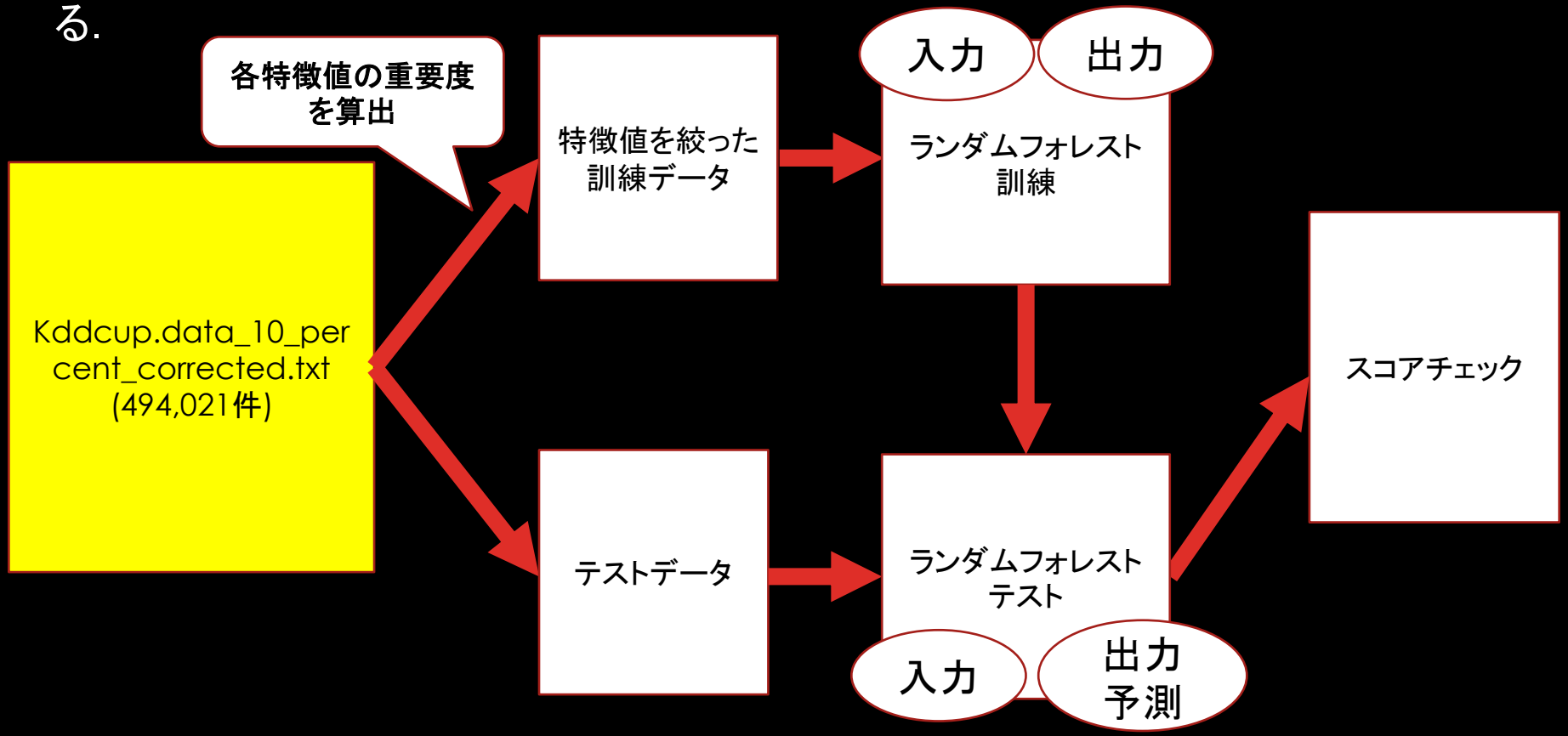
KDDCUP.DATA_10_PERCENT_CORRECTED.TXTのラベル一覧

ラベル一覧

normal	teardrop	rootkit
smurf	pod	loadmodule
neptune	nmap	ftp_write
back	guess_passwd	multihop
satan	buffer_overflow	phf
ipsweep	land	perl
portsweep	warezmaster	spy
warezclient	imap	

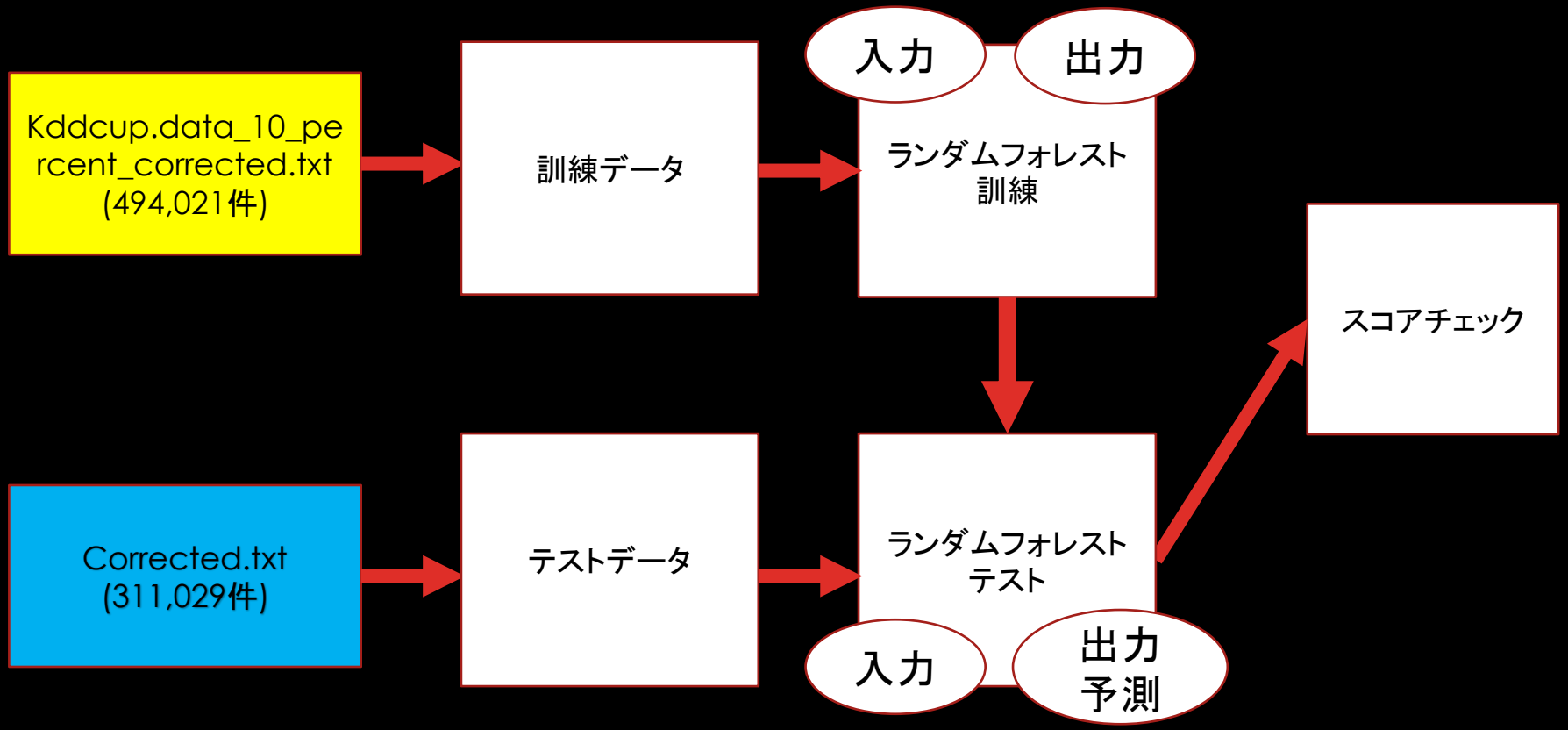
提案手法2(特徴のある訓練データを学習)

- 特徴値を絞ったときのランダムフォレストによる精度を調べる.
- 提案手法1で各特徴値の重要度を算出
- 重要度の高かったデータのみを抽出，訓練データとし，ランダムフォレストによる予測を行う. これにより，どのデータがあれば精度が保証されるのか調べることができる.



提案手法3(未知のデータの予測)

- 未知のデータに対するランダムフォレストの精度を調べる.
- 訓練データには, Kddcup.data_10_percent_corrected.txt, テストデータにはKDDデータセット内の攻撃ラベルの種類を15種類追加したCorrected.txtを使用.
- 学習したデータ22種類　＋　学習していないデータ15種類の合計37種類についてランダムフォレストの予測を確認する.



CORRECTEDのラベル一覧

ラベル一覧					
normal	teardrop	imap	mscan	snmgetattaack	saint
smurf	pod	rootkit	ps	httptunnel	xsnoop
neptune	nmap	loadmodule	xlock	worm	snmpguess
back	guess_passwd	ftp_write	apache2	mailbomb	
satan	buffer_overflow	multihop	sendmail	sglattack	
ipsweep	land	phf	sterm	processtable	
portsweep	warezmaster	perl	udpstorm	named	

提案手法1の結果

表1　訓練時間，テスト時間，正答率

教師データの割合(%)	訓練時間(sec)	テスト時間(sec)	正答率
90	12.682	0.352	0.9997
10	0.921	2.686	0.9994
1	0.115	2.223	0.9971

- 結果は表1のようになる.
- 1列目は教師データの割合，2列目はランダムフォレストの訓練にかかった時間，3列目はランダムフォレストの予測にかかった時間，4列目は予測した結果の正答率.
- 表1より教師データの割合が少なくとも高い精度だということがわかる.

提案手法2の結果

表2　各特徴値の重要度

特徴値名	重要度
duration	0.0016210157
src_bytes	0.1263076275
dst_bytes	0.0352500850
land	0.0000104756
wrong_fragment	0.0029302551
urgent	0
hot	0.0036142893
num_failed_logins	0.0000308608
logged_in	0.0230412084
num_compromised	0.0034286638
root_shell	0.0000786936
su_attempted	0.0000148023
num_root	0.0000386694
num file creations	0.0000442285
num_shells	0.0000220786
num_access_files	0.0000269207
num outbound_cmds	0
is_host_login	0
is_guest_login	0.0005598335
count	0.1629632356
srv_count	0.1455674649
serror_rate	0.0150678392
srv_serror_rate	0.0135818495
rerror_rate	0.0035466521
srv_rerror_rate	0.0026508874
same_srv_rate	0.0742159337
diff_srv_rate	0.0663862049
srv_diff_host_rate	0.0031534383
dst_host_count	0.0145519402
dst_host_srv_count	0.0489110855
dst_host_same srv_rate	0.0485380440
dst_host_diff_srv_rate	0.0389474895
dst_host_same_src_port_rate	0.1073409357
dst_host_srv_diff_host_rate	0.0128981430
dst_host_serror_rate	0.0214626321
dst_host_srv_serror_rate	0.0124951073
dst_host_rerror_rate	0.0055668407
dst_host_srv_rerror_rate	0.0051345688

- 各特徴値の重要度を算出した結果は表2のようになる.
- この中から重要度の高かった特徴値上位4個(src_bytes, count, srv_count, dst_host_same_src_port_rate)を抜き出し，ランダムフォレストによる訓練，予測を行う.

提案手法2の結果

表3　訓練時間，テスト時間，正答率

教師データの割合(%)	訓練時間(sec)	テスト時間(sec)	結果
90	10.653	0.339	0.9969
10	0.698	2.793	0.9959
1	0.115	2.262	0.9933

- 結果は表3のようになる.
- 表3よりKDDデータセットの場合は，重要度の高い特徴値が数個あれば高い精度が保証される.

提案手法3の結果

表4　訓練時間，テスト時間，正答率

訓練時間(sec)	テスト時間(sec)	正答率
9.171	2.169	0.9188

表5　予測結果の誤判定部分(一部抜粋)

No.	ラベル名	予測ラベル
5	snmpgetattack	normal
813	normal	guess_passwd
8078	multihop	mscan
8136	xsnoop	mscan
81796	mscan	normal
91214	ps	normal
93759	xlock	normal
142711	apache2	normal
213747	sendmail	normal

- ランダムフォレストによる正答率は表4のようになる.
- 表5はランダムフォレストが誤判定したデータを一部抜粋したものである.
- 表5の1列目は列番号，2列目は正しいラベルの名称，3列目はランダムフォレストによる予測ラベルである.
- 色のついたラベルは訓練データにはなかった未知のラベルである.
- 正答率は約92%となり，訓練していない未知のデータはほとんど誤判定が出力されていた.

考察

- 提案手法1，2から訓練データの割合が90%，10%，1%ともに分類精度は99%以上であった．この結果は訓練データが少なくても，ほぼすべてのデータにおいて，攻撃であるか，通常の通信であるかを正確に分類できていることを示している．
- しかし，ファイアウォールにブロックリストとして登録するにはIPアドレスも必要となる．そのため，今後は分類したデータをIPアドレスに紐付け，ファイアーウォールに自動登録するという作業が必要になるであろう．

考察

- 提案手法3の学習していない未知の攻撃が混ざったデータを使用した結果は，精度91.87%であった．また，誤判定部分をまとめた結果より，未知の攻撃のほとんどはnormalと予測していた．
- これでは本来攻撃である通信が通常の通信として処理されてしまい，攻撃の通信を遮断できなくなる．このことからランダムフォレストは学習していない未知の攻撃に対しては効果を発揮できない．
- 対策としてはランダムフォレストを並列化し，アンサンブル学習させ，大量の未知の通信を学習させる必要がある．